



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

**“DISEÑO, IMPLEMENTACIÓN Y PROGRAMACIÓN DE UN ROBOT
MÓVIL PARA INTEGRAR A UN SISTEMA MPS PARA
TRANSPORTE DE MATERIALES”**

TESIS DE GRADO

PREVIO LA OBTENCIÓN DEL TÍTULO DE:

**INGENIERO EN ELECTRÓNICA, CONTROL Y REDES
INDUSTRIALES.**

AUTORES: JORGE ANTONIO LÓPEZ PAGUAY
GALO EDUARDO SANTILLÁN MENESES

TUTOR: ING. MARCO VITERI

RIOBAMBA – ECUADOR

2015

DEDICATORIA

A Dios por su protección y a mis padres por sus cuidados, a mis hermanas, a mis apreciados profesores, a mis amigos y seres amados, quienes me brindaron su apoyo para finalizar mi tesis.

Galo Santillán

DEDICATORIA

A Dios que me ha dado luz en mi camino, a mis padres y hermanos que son mi fuente de inspiración, a mis familiares y a mis amigos, y en especial a MICHELLE quien nunca dejo de creer en un mañana. Por Uds., mis amores mi EVLA y MIMICHU, por una vida juntos.

Jorge López

AGRADECIMIENTO

Agradezco a la Escuela Superior
Politécnica De Chimborazo. A
Dios, a mis padres, familiares, al
Ing. Marco Viteri por su apoyo y en
especial a todos mis amigos.

Galo Santillán

AGRADECIMIENTO

Agradezco a Dios, a la Escuela Superior Politécnica de Chimborazo por abrirme las puertas, a mis padres, hermanos, mi Abuelita, a mis tíos, primos y al Ing. Marco Viteri por haber sido más que un Tutor un amigo en el que podemos confiar.

Jorge López

FIRMAS DE RESPONSABILIDAD

| NOMBRE | FIRMA | FECHA |
|---|--------------|--------------|
| Ing. Gonzalo Samaniego, PhD DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA | | |
| Ing. Alberto Arellano DIRECTOR DE ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES | | |
| Ing. Marco Viteri DIRECTOR DE TESIS | | |
| Ing. Paul Romero MIEMBRO DEL TRIBUNAL | | |
| COORDINADOR SISBIB-ESPOCH | | |

NOTA DE LA TESIS

.....

DERECHOS DE AUTOR

“Nosotros, Galo Eduardo Santillán Meneses y Jorge Antonio López Paguay, somos responsables de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado de título “DISEÑO, IMPLEMENTACIÓN Y PROGRAMACIÓN DE UN ROBOT MÓVIL PARA INTEGRAR A UN SISTEMA MPS PARA TRANSPORTE DE MATERIALES” , y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.”

Galo Eduardo Santillán Meneses

Jorge Antonio López Paguay

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

FIRMAS DE RESPONSABILIDAD

RESPONSABILIDAD DEL AUTOR

ÍNDICE GENERAL

ÍNDICE DE ABREVIATURAS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

INTRODUCCIÓN

CAPÍTULO I

| | |
|--|----|
| 1. MARCO REFERENCIAL | 22 |
| 1.1 ANTECEDENTES..... | 22 |
| 1.2 JUSTIFICACIÓN DEL PROYECTO DE TESIS..... | 23 |
| 1.3 OBJETIVOS | 23 |
| 1.3.1 OBJETIVO GENERAL | 23 |
| 1.3.2 OBJETIVOS ESPECÍFICOS..... | 24 |
| 1.4. HIPOTESIS..... | 24 |

CAPÍTULO II

SISTEMA ROBÓTICO MÓVIL

| | |
|---|----|
| 2. FUNDAMENTOS BÁSICOS..... | 25 |
| 2.1. INTRODUCCIÓN A LA ROBÓTICA..... | 25 |
| 2.2. DEFINICION DE ROBOT INDUSTRIAL | 26 |
| 2.3. MORFOLOGIA DE UN ROBOT | 28 |

| | |
|--|----|
| 2.3.1. ESTRUCTURA MECANICA | 28 |
| 2.3.2.1. TRANSMISIÓN POR BANDA Y CADENA | 29 |
| 2.3.3. SISTEMA DE ACCIONAMIENTO O ACTUADORES | 31 |
| 2.3.3.1. ACTUADORES ELÉCTRICOS | 31 |
| 2.3.4. SISTEMA SENSORIAL..... | 32 |
| 2.3.5. SISTEMA DE CONTROL | 32 |
| 2.3.6. ELEMENTOS TERMINALES | 34 |
| 2.4. CLASIFICACION DE LOS ROBOTS..... | 35 |
| 2.4.1. Clasificación de los Robots según su Nivel de Inteligencia..... | 35 |
| 2.4.2. Clasificación de los Robots según el Nivel de Control Ejecutan | 35 |
| 2.4.3. Clasificación de los Robots según el Nivel del Lenguaje de Programación ... | 36 |
| 2.4.4. Clasificación de los Robots según su Aplicación..... | 36 |
| 2.4.4. Clasificación de los Robots según su Arquitectura..... | 36 |
| 2.5. ROBOTICA MOVIL | 40 |
| 2.5.1. ROBOTS MÓVILES | 40 |
| 2.5.1.1. ROBOTS CON RUEDAS..... | 41 |
| 2.5.1.2. ROBOTS CON PATAS | 41 |
| 2.6. NAVIGACIÓN DE ROBOTS MÓVILES | 42 |
| 2.7. ODOMETRÍA..... | 44 |
| 2.8. CONFIGURACIÓN OMNIDIRECCIONAL | 45 |
| 2.9. MODELO CINEMATICO DEL ROBOTINO | 47 |
| 2.10. ESTIMACIÓN DE LA POSICIÓN DE UN ROBOT MÓVIL | 53 |
| 2.11. ESTIMACIÓN EXPLÍCITA DE LA POSICIÓN. | 55 |
| 2.12. SISTEMAS ODOMÉTRICOS..... | 55 |
| 2.13. SISTEMAS DE NAVIGACIÓN INERCIAL (INS)..... | 56 |
| 2.14. ESTIMACIÓN MEDIANTE PERCEPCIÓN DEL ENTORNO | 57 |
| 2.15. PLATAFORMA CONTROLADA POR PLC | 58 |
| 2.15.1. CONTROLADOR LOGICO PROGRAMABLE | 58 |
| 2.15.2. ESTRUCTURA DEL PLC | 59 |

| | |
|---|----|
| 2.15.2.1. ESTRUCTURA COMPACTA | 61 |
| 2.15.2.2. ESTRUCTURA SEMIMODULAR..... | 61 |
| 2.15.2.3. ESTRUCTURA MODULAR | 62 |
| 2.15.3. CICLO DE FUNCIONAMIENTO DE UN PLC..... | 63 |
| 2.15.4. GRAFCET | 64 |
| 2.15.4.1. ELEMENTOS GRAFICOS | 65 |
| 2.15.4.2. REGLAS DE EVOLUCION..... | 67 |
| 2.15.4.3. ESTRUCTURAS DEL GRAFCET | 68 |
| 2.15.5. DESCRIPCIÓN GENERAL DE LAS COMUNICACIONES | 70 |
| 2.15.5.1 BUS DE COMUNICACIÓN MODBUS | 72 |
| 2.15.5.2. BUS DE COMUNICACIÓN ETHERNET..... | 74 |
| 2.15.6. ACTUADORES NEUMÁTICOS..... | 75 |
| 2.15.7. SENSORES MAGNÉTICOS..... | 77 |
| 2.16. MÉTODOS DE NAVEGACIÓN | 78 |
| 2.16.1. CARTOGRAFÍA. | 78 |
| 2.17. DISEÑO Y FUNCIÓN DEL ROBOT | 80 |
| 2.17.1. UNIDAD DE ACCIONAMIENTO | 82 |
| 2.17.2. LA UNIDAD DE CONTROL | 83 |
| 2.17.3. FUENTE DE ALIMENTACIÓN..... | 83 |
| 2.17.4. SENSORES..... | 84 |
| 2.17.5. PUNTO DE ACCESO INALÁMBRICO BLUETOOTH..... | 85 |
| 2.18. PLANIFICACIÓN DE LA TRAYECTORIA | 86 |
| 2.19. GENERACIÓN DE CAMINOS | 87 |
| CAPÍTULO III | |
| IMPLEMENTACIÓN DEL ROBOT MÓVIL PALETIZADOR | |
| 3.1. SENSORES UTILIZADOS | 88 |
| 3.2. MOVIMIENTOS LINEALES SISTEMA ROBÓTICO | 92 |
| 3.3. COMPONENTES DEL ROBOT MOVIL PALETIZADOR | 96 |
| 3.3.1. GEARMOTOR 131:1 METAL 37DX57L MM 64 CPR ENCODER..... | 96 |

| | |
|---|-----|
| 3.3.2 ENCODER..... | 98 |
| 3.3.3 ARDUINO | 99 |
| 3.3.3.1 ARDUINO MEGA 2560..... | 100 |
| 3.3.4. GIROSCOPIO GY-521 MPU6050 MODULO 6DOF..... | 102 |
| 3.3.5. INFRARED DISTANCE SHARP GP2Y0A41SK (4-30cm) | 103 |
| 3.3.6. PUENTE H DOBLE L298N MODULO | 106 |
| 3.3.7. BLUETOOTH HC-06 MODULE..... | 108 |
| 3.3.8. FOTSENSOR RETRORREFLECTANTE IBEST M18..... | 108 |
| 3.3.9. CAMARA WEB OMEGA 8 MEGAPIXELES-3636K..... | 110 |
| 3.3.10. ACTUADOR LINEAL | 111 |
| 3.4. SOFTWARE UTILIZADO..... | 112 |
| 3.4.1. ARDUINO ENVIROMENT 1.0.5..... | 112 |
| 3.4.1.1. DISEÑO DEL PROGRAMA EN ARDUINO..... | 114 |
| 3.4.2. TWIDO SUITE 2.31 | 116 |
| 3.4.2.1. DISEÑO EN DIAGRAMA DE CONTACTOS (LADDER)..... | 117 |
| 3.4.2.2. CONFIGURACIÓN DEL PLC Y RECOLECCIÓN DE DATOS DESDE EL MPS..... | 118 |
| 3.11.2.3 EDICION DEL PROGRAMA | 121 |
| 3.4.3. LABVIEW 2013 | 125 |
| 3.4.3.1. DISEÑO DEL PROGRAMA EN LABVIEW | 128 |
| 3.4.3.2. ADQUISICION DE IMÁGENES..... | 131 |
| 3.4.3.3. TRATAMIENTO DE LAS IMÁGENES | 135 |
| 3.4.3.5. IMPLEMENTACIÓN DEL ALGORITMO A* | 138 |
| 3.4.3.6. ALGORITMO A* DINAMICO | 141 |
| 3.4.3.7. ADQUISICION DE LA TRAYECTORIA..... | 145 |
| 3.4.3.8. INTEGRACIÓN DEL MODELO CINEMÁTICO..... | 146 |
| 3.4.3.8. TRANSMISION Y RECEPCION DE INFORMACIÓN..... | 147 |
| 3.4.3.9. INDICACIÓN DE PUNTOS FINALES DE TRAYECTORIA AL ROBOT PALETIZADOR MEDIANTE PLATAFORMA CONTROLADA POR PLC.. | 149 |

CAPÍTULO IV

PRUEBAS Y RESULTADOS

| | |
|---|-----|
| 4.1. INTRODUCCIÓN | 150 |
| 4.2. ANALISIS DEL ROBOT | 150 |
| 4.3. PRUEBAS MECÁNICAS | 152 |
| 4.4. PRUEBAS DE ADQUISICIÓN DE IMÁGENES | 153 |
| 4.5. PRUEBA Y CALIBRACIÓN DE SENSORES..... | 155 |
| 4.6. PRUEBAS DE CONSUMO DE CORRIENTE..... | 156 |
| 4.7. PRUEBAS DE SOFTWARE DE CONTROL..... | 158 |
| 4.9. PORCENTAJE DE AUTOMATIZACIÓN | 160 |
| 4.10. PRUEBA DE HIPOTESIS..... | 161 |

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMARY

ANEXOS

BIBLIOGRAFÍA

ÍNDICE DE ABREVIATURAS

| | |
|-----------------|--|
| AC.- | Corriente alterna |
| AFNOR.- | Asociación Francesa de Normalización |
| BMP.- | Imagen de mapa de bits. |
| CPR.- | Ciclos por revolución. |
| DC.- | Corriente directa |
| DOF.- | En inglés Depth of Field; en español profundidad de campo. |
| GDL.- | Grado de libertad |
| HMI.- | Human Machine Interface. |
| I/O.- | Entrada/Salida de datos. |
| I2C.- | Es un bus de comunicaciones en serie. |
| INS.- | Sistemas de navegación inercial |
| IP.- | Dirección del dispositivo en una red pública o privada. |
| ISO.- | Organización Internacional de Estándares |
| MEMS.- | Sistemas micro electromecánicos. |
| MODBUS.- | Protocolo de comunicación serial basado en el modelo maestro/esclavo. |
| MPS.- | Sistema de Producción Modular. |
| MPU.- | Unidad de procesamiento de movimiento. |
| PLC.- | Programable Logic Controller. |
| PSD.- | Posición detector sensible. |
| PWM.- | Acrónimo en inglés pulse-width modulation, modulación por ancho de pulso |

- RIA.-** Asociación de Industrias Robóticas
- RGB.-** En inglés Red, Green, Blue; es la composición del color en términos de la intensidad de los colores.
- Rx.-** Receptor.
- SCADA.-** Supervisory Control and Data Acquisition.
- Tx.-** Transmisor.
- UARTs.-** Acrónimo en inglés Universal Asynchronous Receiver-Transmitter.
- Wc.-** La velocidad angular de giro.

ÍNDICE DE FIGURAS

| | | |
|---------------|---|----|
| Figura II 1. | Robot móvil articulado..... | 26 |
| Figura II 2. | Robot industrial Mitsubishi PA-10 | 27 |
| Figura II 3. | Transmisión por banda y cadena..... | 29 |
| Figura II 4. | Reductor de velocidad..... | 30 |
| Figura II 5. | Tipos de sensores | 32 |
| Figura II 6. | Sistema en lazo abierto | 33 |
| Figura II 7. | Sistema en lazo cerrado..... | 33 |
| Figura II 8. | Carretilla elevadora | 34 |
| Figura II 9. | Robot Poli-articulado | 37 |
| Figura II 10. | Robotino..... | 38 |
| Figura II 11. | Robot Androide..... | 38 |
| Figura II 12. | Robot zoomórfico | 39 |
| Figura II 13. | Robot hibrido | 40 |
| Figura II 14. | Robot con ruedas..... | 41 |
| Figura II 15. | Robot cuadrúpedo | 42 |
| Figura II 16. | Robot oruga..... | 42 |
| Figura II 17. | Navegación reactiva | 44 |
| Figura II 18. | Robot omnidireccional | 45 |
| Figura II 19. | Modelo de la configuración omnidireccional..... | 46 |
| Figura II 20. | Rueda en contacto con el suelo | 47 |
| Figura II 21. | Esquema cinemático robot omnidireccional | 48 |
| Figura II 22. | Sistema sensorial..... | 57 |
| Figura II 23. | Estructura de PLC | 59 |
| Figura II 24. | PLC Compacto..... | 61 |
| Figura II 25. | Estructura Semimodular..... | 62 |
| Figura II 26. | Estructura PLC modular..... | 62 |
| Figura II 27. | Ciclo de funcionamiento de un PLC | 64 |
| Figura II 28. | Ejemplo de estructura GRAFCET | 65 |
| Figura II 29. | Representación de Etapas..... | 66 |
| Figura II 30. | Líneas de Evolución..... | 66 |
| Figura II 31. | Transiciones | 67 |
| Figura II 32. | Secuencia Lineal | 68 |

| | | |
|----------------|--|-----|
| Figura II 33. | Divergencia y Convergencia en "o" | 69 |
| Figura II 34. | Divergencia y Convergencia en "Y" | 70 |
| Figura II 35. | Esquema conexión remota | 71 |
| Figura II 36. | Modulos de comunicación | 72 |
| Figura II 37. | Esquema de ejemplos de comunicación Modbus..... | 73 |
| Figura II 38. | Ejemplo de Arquitectura Ethernet..... | 74 |
| Figura II 39. | Cilindro de simple efecto | 75 |
| Figura II 40. | Cilindro de doble efecto..... | 76 |
| Figura II 41. | Electroválvula 5/2 | 77 |
| Figura II 42. | Sensor Magnético..... | 77 |
| Figura II 43. | Panel de aluminio compuesto..... | 80 |
| Figura II 44. | Estructura aluminio compuesto..... | 81 |
| Figura II 45. | Estructura chasis | 81 |
| Figura II 46. | Unidad de accionamiento..... | 82 |
| Figura II 47. | Baterías 12 voltios..... | 83 |
| Figura II 48. | Ubicación de los sensores infrarrojos | 85 |
| Figura II 49. | Sistema de coordenadas global, local asociado al robot | 87 |
| Figura III 50. | Sensor Inductivo | 89 |
| Figura III 51. | Giroscopio electrónico..... | 89 |
| Figura III 52. | Distancia real recorrida del robot..... | 92 |
| Figura III 53. | Rodillo omnidireccional | 94 |
| Figura III 54. | Gearmotor 37Dx57L..... | 96 |
| Figura III 55. | Dimensiones motoreductores (mm) de metal 37D | 97 |
| Figura III 56. | CPR encoder 64 | 98 |
| Figura III 57. | Arduino Mega 2560 R3 | 100 |
| Figura III 58. | GY-521 | 102 |
| Figura III 59. | Orientación de ejes de sensibilidad y polaridad de rotación..... | 103 |
| Figura III 60. | Sensor (GP2Y0A41SK0F)..... | 103 |
| Figura III 61. | Relación voltaje-distancia sensor GP2Y0A41SK0F | 105 |
| Figura III 62. | Módulo L298N | 106 |
| Figura III 63. | Partes del módulo L298N..... | 107 |
| Figura III 64. | Bluetooth hc-06 module | 108 |
| Figura III 65. | Sensor PESI-R18POC3MD | 108 |
| Figura III 66. | Diagramas de conexiones de sensores polarizados..... | 109 |
| Figura III 67. | Contactos en un detector inductivo..... | 110 |

| | | |
|-----------------|---|-----|
| Figura III 68. | Camara web de 8 Mpx..... | 110 |
| Figura III 69. | Actuador lineal..... | 111 |
| Figura III 70. | Entorno Arduino 1.0.5..... | 113 |
| Figura III 71. | Pantalla inicio Twido 2.31..... | 116 |
| Figura III 72. | Algoritmo Robot Paletizador..... | 117 |
| Figura III 73. | Creación de nuevo proyecto Twido Soft..... | 118 |
| Figura III 74. | Descripción del Hardware..... | 119 |
| Figura III 75. | Configuración de red PLC..... | 119 |
| Figura III 76. | Asignación de Alias..... | 120 |
| Figura III 77. | Configuración de memorias del HMI..... | 120 |
| Figura III 78. | Edición del Programa en Twido..... | 122 |
| Figura III 79. | Transmisión de programa a PLC TWIDO..... | 125 |
| Figura III 80. | Pantalla de Inicio Labview 2013..... | 126 |
| Figura III 81. | Vista del panel frontal..... | 127 |
| Figura III 82. | Vista del diagrama de bloques..... | 128 |
| Figura III 83. | Creación de nuevo Proyecto en Labview..... | 129 |
| Figura III 84. | Creación de servidor I/O..... | 130 |
| Figura III 85. | Configuración de servidor Modbus I/O..... | 130 |
| Figura III 86. | Creación de variables compartidas..... | 131 |
| Figura III 87. | Consola de sistema distribuido..... | 131 |
| Figura III 88. | Adquisición de imagen..... | 132 |
| Figura III 89. | IMAQ CREATE..... | 132 |
| Figura III 90. | Proceso de inicialización cámara..... | 133 |
| Figura III 91. | Selección y toma de fotografía (Panel Frontal)..... | 133 |
| Figura III 92. | IMAQdx Snap..... | 134 |
| Figura III 93. | IMAQ write..... | 134 |
| Figura III 94. | IMAQdx Close..... | 134 |
| Figura III 95. | Etapas de tratamiento de imágenes..... | 135 |
| Figura III 96. | IMAQ ReadFile..... | 136 |
| Figura III 97. | Ruta de almacenamiento de la imagen..... | 136 |
| Figura III 98. | IMAQ Resample..... | 136 |
| Figura III 99. | Filtro e inversión de color..... | 137 |
| Figura III 100. | IMAQ ImageToArray..... | 138 |
| Figura III 101. | Implementación de algoritmo A*..... | 138 |
| Figura III 102. | Selección de puntos..... | 139 |

| | | |
|-----------------|---|-----|
| Figura III 103. | Selección de punto inicial..... | 140 |
| Figura III 104. | Puntos de referencia (Panel Frontal)..... | 140 |
| Figura III 105. | Create Occupancy Grid Map | 141 |
| Figura III 106. | Get Cell Reference..... | 141 |
| Figura III 107. | Algoritmo AD* | 142 |
| Figura III 108. | Initialize AD* | 143 |
| Figura III 109. | Bloque AD* | 143 |
| Figura III 110. | Get Cells in Path | 144 |
| Figura III 111. | Get All Occupancy Grid Cells | 144 |
| Figura III 112. | Close Occupancy Grid Map..... | 144 |
| Figura III 113. | Adquisición de la trayectoria | 145 |
| Figura III 114. | Integración del modelo cinemático..... | 146 |
| Figura III 115. | Etapa de transmisión y recepción de información | 148 |
| Figura III 116. | Sensores Plataforma controlada por PLC | 149 |
| Figura IV 117. | Configuración Omnidireccional | 152 |
| Figura IV 118. | Captura de Fotografía de la Plataforma | 154 |
| Figura IV 119. | Configuración Seguidor de Línea..... | 156 |
| Figura IV 120. | Parámetros de configuración del serial..... | 159 |
| Figura IV 121. | Control de Trayectoria..... | 159 |

ÍNDICE DE TABLAS

| | | |
|-----------------------|---|-----|
| Tabla II I. | Parámetros de configuración cinemática | 49 |
| Tabla II II. | Características técnicas sensor inductivo IME12..... | 84 |
| Tabla II III. | Características físicas rodillo omnidireccional..... | 93 |
| Tabla III IV. | Dimensiones rodillo omnidireccional | 94 |
| Tabla III V. | Conexión de cables..... | 99 |
| Tabla III VI. | Datos Técnicos..... | 100 |
| Tabla III VII. | Tabla de entradas y salidas de PLC TWIDO CAE 40 DRF | 121 |
| Tabla IV VIII. | Comparación de Superficies | 153 |
| Tabla IV IX. | Censo de carga | 157 |
| Tabla IV X. | Procesos Automatizados | 160 |
| Tabla IV XI. | Tabla Chi Cuadrado | 162 |

ÍNDICE DE ANEXOS

| | | |
|-----------------|---|-----|
| Anexo 1 | Manual técnico robot móvil paletizador..... | 163 |
| Anexo 2 | Diseño de piezas en Solid Works..... | 176 |
| Anexo 3 | Datasheet Arduino Mega 2560..... | 187 |
| Anexo 4 | Datasheet Gear Motor Pololu 37D..... | 190 |
| Anexo 5 | Datasheet Puente H L298N..... | 194 |
| Anexo 6 | Datasheet Sensor Sick IME 08..... | 197 |
| Anexo 7 | Datasheet PLC TWIDO CAE40DRF..... | 200 |
| Anexo 8 | Datasheet CS1-J Series..... | 203 |
| Anexo 9 | Datasheet GP2Y0A41SK0F SHARP (4-30cm)..... | 205 |
| Anexo 10 | Código de programa Arduino..... | 208 |
| Anexo 11 | Código de programa Labview..... | 213 |

INTRODUCCIÓN

En el mundo existen muchas empresas dedicadas a la creación de robots industriales móviles, los cuales reciben información a través de sus sensores y se emplean sobre todo para el transporte de mercancías en cadenas de producción y almacenes.

También se utilizan estos robots en lugares de difícil acceso, peligrosos o muy distantes, como es el caso de la inspección de centrales nucleares o la exploración espacial. Para lo cual se usan una gran variedad de técnicas y componentes para lograr un objetivo.

En el Ecuador la robótica no está desarrollada, limitándose a empresas pequeñas creadas por jóvenes que terminan sus estudios y tratan de establecerse con un negocio propio. La maquinaria y los equipos utilizados por estos pequeños negocios son mayormente importados de otros países, sin embargo existen varios proyectos en universidades a nivel nacional y local destinadas a procesos didácticos y demostrativos de las posibles aplicaciones que se puede dar a la robótica.

Los robots móviles son un foco importante de la investigación actual y casi toda universidad importante tiene uno o más laboratorios que se centran en la investigación de robots móviles. Los robots móviles se encuentran también en la industria y los servicios.

Además la robótica autónoma es el área de la robótica que desarrolla robots capaces de desplazarse y actuar sin intervención humana. Para ello el robot debe percibir su entorno y actuar de forma adecuada, además de llevar a cabo su tarea.

La robótica ha tenido grandes avances en entornos estructurados, en los que el controlador del robot puede tener un mapa detallado de su entorno. Conforme decrece el grado de estructuración del entorno las tareas se tornan más complejas. Esto ocurre cuando el robot es móvil y debe tener información de su posición en el mapa interno. Los mecanismos pueden ser absolutos o relativos, por ejemplo, usando GPS y odometría, respectivamente.

En el ámbito local en la ESPOCH existen varias investigaciones e implementaciones, que han sido enfocados al ámbito académico y a la participación en concursos de robótica nacionales e internacionales.

CAPÍTULO I

1. MARCO REFERENCIAL

1.1 ANTECEDENTES

El mundo de la robótica es muy amplio y abarca estudios de varias áreas como la física, la mecánica, las matemáticas entre otras, además existen diversas clasificaciones dependiendo de la función, aplicación, forma, etc. Entonces cuando se inicia el estudio de un robot hay que identificar cual es el objetivo del mismo o el trabajo que va a cumplir y la forma del mismo.

Las principales aplicaciones de un robot móvil son las de exploración y transporte; la creación de un robot móvil autónomo es de gran complejidad debido a que se requiere que el robot conozca el entorno o espacio de trabajo y la trayectoria a seguir de un punto inicial a un punto final.

En el caso de un robot móvil el entorno que lo rodea muy pocas veces será constante; y usualmente tendra a encontrar obstáculos en su trayectoria, para que el robot sea capaz de moverse libremente debe estar dotado de sensores que le permitan interactuar con el medio ambiente que lo rodea dando le opción de tomar decisiones en base a la información obtenida por estos y procesar la información para determinar la acción a desarrollar por los actuadores que en el caso de un robot móvil son los motores que le proveen de movimiento.

1.2 JUSTIFICACIÓN DEL PROYECTO DE TESIS

El área de Automatización Industrial de toda empresa se necesita tener equipos con tecnología actual: mecánica, electrónica, neumática, hidráulica, sistemas informáticos, hardware y software, incluyendo: sensores, actuadores, autómatas programables, robots industriales, sistemas automáticos de inspección y control de calidad, sistemas de visión artificial, etc.

El presente trabajo se justifica plenamente porque se desea aportar con un robot y desarrollo de software que pueda ser usado en procesos industriales; esta tesis completará proyectos previos existentes de un sistema de producción modular; permitiendo transportar elementos desde un punto inicial hacia una posición final en un área específica que para este caso es una mesa de trabajo de 2.4x2.4 metros de manera autónoma, simulando procesos industriales reales y satisfaciendo la necesidad de llevar o transportar piezas de un área de producción a otra.

Para lo cual se hace uso de un conjunto de algoritmos de navegación, para asegurar la precisión y disminuir los márgenes de error en la ubicación de rutas y un robot móvil omnidireccional que es capaz de moverse en cualquier dirección de manera instantánea eliminando el problema de movilidad; lo cual permitirá obtener un ahorro de costos al optimizar los tiempos de producción y trabajar en ambientes poco favorables o peligrosos, evitando así un desperdicio de tiempo y recursos; además de ser una tarea repetitiva y cansada para un ser humano.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

- Diseñar, implementar y programar un Robot móvil para integrar a un sistema de producción modular y paletizado en una línea de montaje; para transporte de materiales.

1.3.2 OBJETIVOS ESPECÍFICOS

- Implementar un robot con fines académicos para incorporarse en el sistema de producción modular de paletizado.
- Desarrollar un software dedicado al tratamiento de imágenes mediante visión artificial para la localización y ubicación de posiciones.
- Seleccionar e implementar los controladores que permitirán la automatización del Sistema de acuerdo a las condiciones de funcionamiento de un robot móvil autónomo.
- Implementar los dispositivos que van a permitir la alimentación, detección, transporte, y control de pallets.
- Combinar y acoplar diferentes algoritmos de navegación para obtener la mejor precisión en el seguimiento y cálculo de trayectorias.
- Implementar el robot móvil en el sistema de paletizado.

1.4. HIPÓTESIS

La incorporación del robot móvil paletizador al sistema MPS, mejorará el índice de automatización del sistema.

CAPÍTULO II

SISTEMA ROBÓTICO MÓVIL

2. FUNDAMENTOS BÁSICOS

2.1. INTRODUCCIÓN A LA ROBÓTICA

La palabra robot fue usada por primera vez en el año 1921. Su origen es la palabra eslava robota, que se refiere al trabajo realizado de manera forzada. Pero fue el escritor Issac Asimov el máximo impulsor de la palabra robot. En octubre de 1945 publicó en la revista Galaxy Science Fiction una historia en la que por primera vez enunció sus tres leyes de la robótica.

1. Un robot no puede hacer daño a un ser humano o, por inacción, permitir que un ser humano sufra daño.
2. Un robot debe obedecer las órdenes dadas por los seres humanos, excepto si estas órdenes entrasen en conflicto con la 1ª Ley.
3. Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la 1ª o la 2ª Ley. [1]

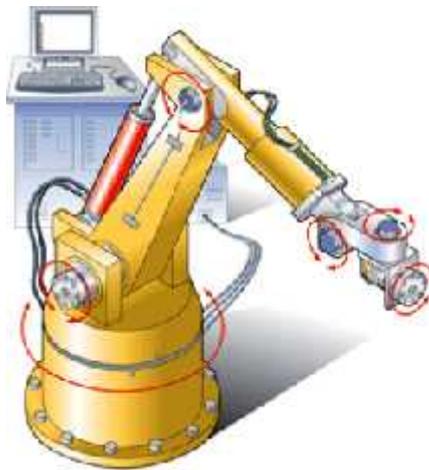
La evolución de los robots desde sus primeros balbuceos ha sido vertiginosa. En poco más de 30 años las investigaciones y desarrollos sobre la robótica han permitido que los robots tomen posición en casi todas las áreas productivas y tipos de industria. En

pequeñas o grandes fábricas los robots pueden sustituir al hombre en aquellas tareas repetitivas y hostiles.

2.2. DEFINICIÓN DE ROBOT INDUSTRIAL

Existen algunas dificultades al momento de establecer una definición formal de lo que es un robot industrial. La primera de estas dificultades surge de la diferencia conceptual entre el mercado japonés y el euro-americano de lo que es un robot y de lo que es un manipulador. Así, mientras que para los japoneses un robot industrial es cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación, el mercado occidental es más restrictivo, exigiendo un mayor complejidad, sobre todo en lo relativo al control. [2]

Figura II 1. ROBOT MÓVIL ARTICULADO



Fuente: <http://tecnowebstudio.com/definicion-sobre-lo-que-es-un-robot-industrial>

La definición más aceptada es la de la Asociación de Industrias Robóticas (RIA), según la cual expresa lo siguiente:

Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas"

Esta definición, ligeramente modificada, ha sido adoptada por la Organización Internacional de Estándares (ISO) que define al robot industrial como:

"Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas" [2].

Figura II 2. Robot industrial Mitsubishi PA-10



Fuente: <http://ssti.ua.es/es/infraestructuras-de-apoyo/laboratorio-de-robotica-y-sensorizacion.html>

Una definición más completa es la establecida por la Asociación Francesa de Normalización (AFNOR), que define al robot industrial como:

Robot: manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectoria variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material. [3]

2.3. MORFOLOGÍA DE UN ROBOT

Un robot está constituido por 6 elementos componentes, los cuales se mencionan a continuación:

- Estructura Mecánica
- Transmisiones y reducciones
- Sistema de Accionamiento
- Sistema Sensorial
- Sistema de Control
- Elementos Terminales

Cada uno de estos elementos será examinado a continuación:

2.3.1. ESTRUCTURA MECÁNICA

Mecánicamente un robot está constituido por una serie de elementos o eslabones, y actuadores en ciertos casos, los que se encuentran unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos.

El movimiento de cada articulación puede ser de desplazamiento, de giro, o de una combinación de ambos.

Cada uno de los movimientos independientes que puede realizar cada articulación con respecto a la anterior, se denomina grado de libertad (GDL).

2.3.2. TRANSMISIONES Y REDUCTORES

Las transmisiones son los elementos cuya función es la de transmitir el movimiento desde los actuadores hacia las articulaciones. Las transmisiones usualmente vienen junto a los reductores, que son los encargados de adaptar el par y la velocidad de la salida del actuador a los valores adecuados para el movimiento de los elementos del robot.

Actualmente existen robots industriales que utilizan acoplamiento directo entre accionamiento y articulación, los cuales son ventajosos, en numerosas ocasiones dependiendo del trabajo a desempeñar. Se trata de casos particulares dentro de la generalidad que en los robots industriales actuales supone la existencia de sistemas de transmisión junto con reductores para el acoplamiento entre actuadores y articulaciones.

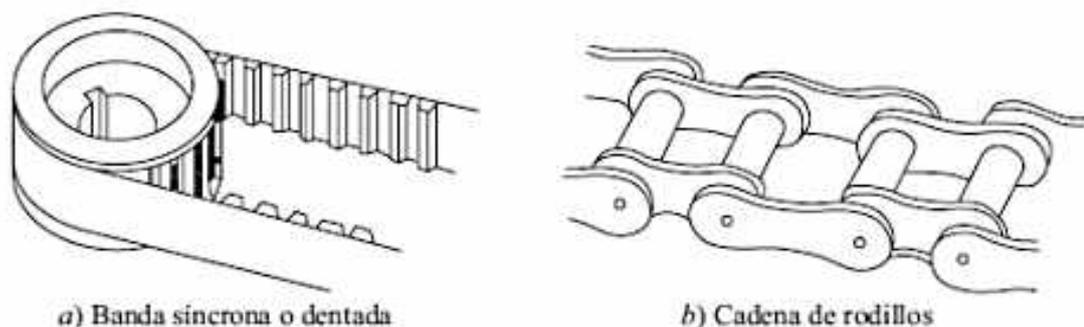
Para que exista un confiable sistema de transmisión este debe cumplir con una serie de características básicas: debe tener un tamaño y peso reducido, se ha de evitar holgaduras considerables y se deben buscar transmisiones con gran rendimiento; además deben estar hechas de materiales adecuados y resistentes.

Las más usadas son aquellas que cuentan con movimiento circular tanto a la entrada como a la salida. Como ejemplo se tienen los engranajes, las correas dentadas y las cadenas. Para el caso de nuestro robot se tiene una caja reductora acoplada directamente al motor con un eje de salida.

2.3.2.1. TRANSMISIÓN POR BANDA Y CADENA

Los accionamientos por banda se utilizan mucho en la robótica, especialmente la banda síncrona, como se muestra en la figura 2a). Sin embargo, su vida útil es breve, ya que dependen de la tensión de la banda para producir agarre a través de la polea. Las cadenas, por otro lado, como se muestra en la figura 2b), son por lo general más económicas. En la figura II 3 se ilustra los tipos de transmisión:

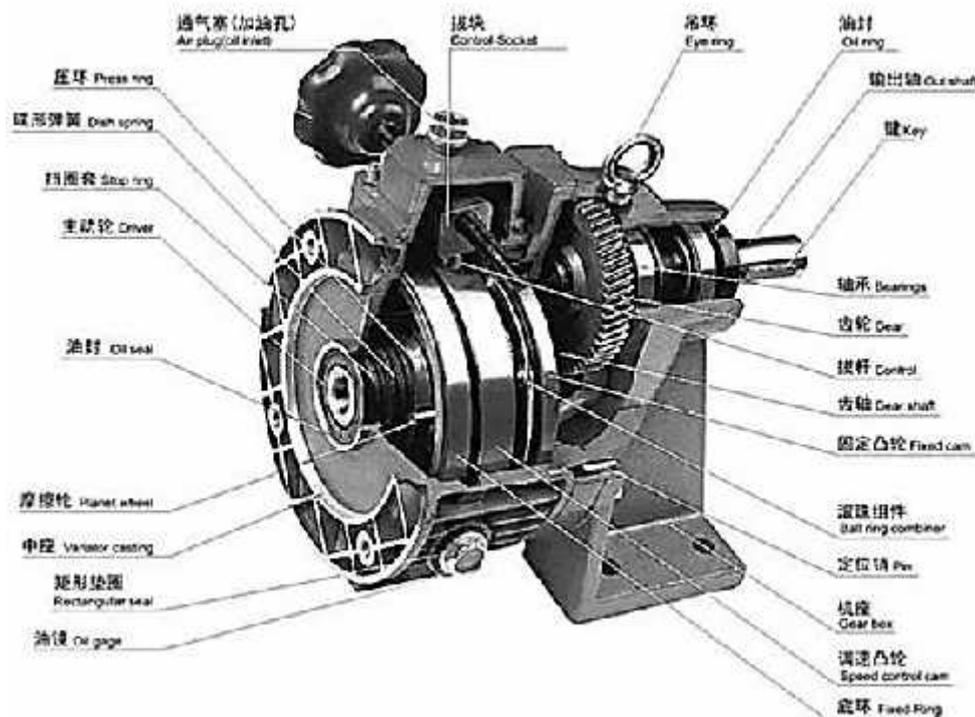
Figura II 3. Transmisión por banda y cadena



Fuente: <http://creandoelfuturo.net/es/morfologia-del-robot/transmisiones-reductores>

Con respecto a los reductores, al contrario que con las transmisiones, sí que existen determinados sistemas usados de manera preferente en los robots industriales. Esto se debe a que a los reductores utilizados en robótica se les exige unas condiciones de funcionamiento muy restrictivas. La exigencia de estas características viene motivada por las altas prestaciones que se le pide al robot en cuando a precisión y velocidad de posicionamiento.

Figura II 4. Reductor de velocidad



Fuente: <http://spanish.alibaba.com/product-gs/stepless-adjustable-reducer-gearbox-617545617.html>

Lo que se hace es buscar reductores de ligeros, de reducido tamaño, de bajo desgaste o rozamiento y que sean capaces de realizar una reducción elevada de velocidad en un único paso.

Puestos que los robots trabajan en ciclos cortos esto implican continuos arranques y paradas, por lo cual es de gran importancia que el reductor sea capaz de soportar pares elevados puntuales.

2.3.3. SISTEMA DE ACCIONAMIENTO O ACTUADORES

Los actuadores tienen por misión generar el movimiento de los elementos del robot según las órdenes dadas por la unidad de control. Los actuadores utilizados en robótica pueden emplear energía neumática, hidráulica o eléctrica. [4].

2.3.3.1. ACTUADORES ELÉCTRICOS

Las características de control, sencillez y precisión de los accionamientos eléctricos han hecho que sean los más usados en los robots actuales.

Dentro de los actuadores eléctricos actuales se pueden distinguir 3 tipos diferentes:

- Motores de corriente continua (DC)
- Motores de corriente alterna (AC)
- Motores pasó a paso.

Motores de corriente continua (DC):

Son los más usados en la actualidad debido a su facilidad de control. Los motores DC están formados por dos devanados internos, inductor e inducido, que se alimenta con corriente continua:

- El inductor, también denominado devanado de excitación, está situado en el estator y crea un campo magnético de dirección fija, denominado excitación.
- El inducido, situado en el rotor, hace girar al mismo debido a la fuerza de Lorentz que aparece como combinación de la corriente circulante por él y del campo magnético de excitación. Recibe la corriente del exterior a través del colector de delgas, en el que se apoyan unas escobillas de grafito.

Para que se pueda realizar la conversión de energía eléctrica en energía mecánica de forma continua es necesario que los campos magnéticos del estator y del rotor permanezcan estáticos entre sí.

Los motores eléctricos de corriente directa son los más usados en robótica son los motores de 12VCD - 24VCD con cajas reductoras, servomotores y motores paso a paso.

2.3.4. SISTEMA SENSORIAL

Para conseguir que un robot realice una tarea con la adecuada precisión, velocidad e inteligencia, será preciso que tenga conocimiento tanto de su propio estado como del estado de su entorno. La información relacionada con su estado (fundamentalmente la posición de sus articulaciones) la que consigue con los denominados sensores internos, mientras que la que se refiere al estado de su entorno, se adquiere con los sensores externos. [5].

La mayoría de los sensores cumplen con la función de informar acerca de la activación de un proceso, la presencia de un objeto sin importar el material o su composición, la distancia a la que se encuentra un determinado obstáculo, información de los factores ambientales en donde se encuentre instalado el sensor, entre otras aplicaciones.

Figura II 5. Tipos de sensores



Fuente: <http://1612182.blogspot.com/2013/05/21-sensores-y-transductores.html>

2.3.5. SISTEMA DE CONTROL

Se entiende por sistema de control a la combinación de componentes que actúan juntos para realizar el control de un proceso sin intervención del ser humano. Este control se puede hacer de forma continua, es decir en todo momento o de forma discreta, es decir cada cierto tiempo. El sistema de control se caracteriza por estar formado por elementos

de entrada, de control y de salida; los elementos de control son los que deciden las acciones que va a realizar el sistema suelen ser los denominados microprocesadores.

Cuando el sistema es continuo, el control se realiza con elementos continuos. Cuando el sistema es discreto, el control se realiza con elementos digitales como el ordenador, por lo que hay que digitalizar los valores antes de su procesamiento y volver a convertirlos tras el procesamiento. [6]

En cualquier caso existen dos tipos de sistemas, sistemas en lazo abierto y sistemas en lazo cerrado.

Sistemas de control en lazo abierto: son los que en su ejecución no influyen las salidas.

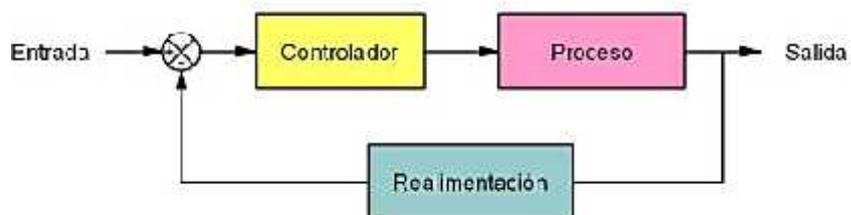
Figura II 6. Sistema en lazo abierto



Fuente: < López J., Santillán G.

Sistemas de control en lazo cerrado: son aquellos en los que influyen las salidas, lo que se denomina retroalimentación.

Figura II 7. Sistema en lazo cerrado



Fuente: López J., Santillán G.

2.3.6. ELEMENTOS TERMINALES

También llamados efectores finales; son los encargados de interactuar directamente con el entorno del robot. Pueden ser tanto elementos de aprensión como de herramientas. [4]

Figura II 8. Carretilla elevadora



Fuente: Festo-Robotino

Para las aplicaciones industriales, las capacidades de un robot básico deben aumentarse por medio de dispositivos adicionales. Podríamos denominar a estos dispositivos como los periféricos del robot. En robótica, el término de actuador final se utiliza para describir la mano o herramienta que está unida a la muñeca. El actuador final representa la herramienta especial que permite al robot de uso general realizar una aplicación particular, y debe diseñarse específicamente para dicha aplicación.

Los elementos terminales son los encargados de interactuar directamente con el entorno del robot. Se clasifican en sujeción, operación, manipulación, montaje, y especiales.

2.4. CLASIFICACIÓN DE LOS ROBOTS

2.4.1. CLASIFICACIÓN DE LOS ROBOTS SEGÚN SU NIVEL DE INTELIGENCIA

Los japoneses (JIRA) tienen su propia forma de clasificar a los robots dentro de seis clases basada en el Nivel de Inteligencia del Robot:

- 1) Dispositivos de manejo manual; controlados por una persona.
- 2) Robots de secuencia arreglada.
- 3) Robots de secuencia variable; donde un operador puede modificar la secuencia fácilmente.
- 4) Robots regeneradores; el operador conduce el robot a través de la tarea.
- 5) Robots de control numérico; donde el operador alimenta la programación del movimiento, hasta que se enseñe manualmente la tarea.
- 6) Robots inteligentes; pueden entender e interactuar con cambios en el medio ambiente.

2.4.2. CLASIFICACIÓN DE LOS ROBOTS SEGÚN EL NIVEL DE CONTROL EJECUTAN

- 1) Nivel de inteligencia artificial.-** donde el programa aceptará un comando como "levantar el producto" y descomponerlo dentro de una secuencia de comandos de bajo nivel basados en un modelo estratégico de las tareas.
- 2) Nivel de modo de control.-** donde los movimientos del sistema son modelados, para lo que se incluye la interacción dinámica entre los diferentes mecanismos, trayectorias planeadas, y los puntos de asignación seleccionados.
- 3) Niveles de servo-sistemas.-** donde los actuadores controlan los parámetros de los mecanismos con el uso de una retroalimentación interna de los datos obtenidos por los

sensores, y la ruta es modificada sobre la base de los datos que se obtienen de sensores externos. Todas las detecciones de fallas y mecanismos de corrección son implementados en este nivel.

2.4.3. CLASIFICACIÓN DE LOS ROBOTS SEGÚN EL NIVEL DEL LENGUAJE DE PROGRAMACIÓN

- 1) **Sistemas guiados.-** en el cual el usuario conduce el robot a través de los movimientos a ser realizados.
- 2) **Sistemas de programación de nivel-robot.-** en los cuales el usuario escribe un programa de computadora al especificar el movimiento y el sentido.
- 3) **Sistemas de programación de nivel-tarea.-** en el cual el usuario especifica la operación por sus acciones sobre los objetos que el robot manipula.

2.4.4. CLASIFICACIÓN DE LOS ROBOTS SEGÚN SU APLICACIÓN

- 1) **Robots Médicos.-** fundamentalmente, prótesis para disminuir los efectos físicos que se adaptan al cuerpo y están dotados de potentes sistemas de mando. Con ellos se logra igualar al cuerpo con precisión los movimientos y funciones de los órganos o extremidades que suplen.
- 2) **Exoesqueletos Robóticos.-** es el esqueleto externo continuo que recubre, protege y soporta el cuerpo de un animal, hongo protista.
- 3) **Robots Industriales.-** es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales.

2.4.4. CLASIFICACIÓN DE LOS ROBOTS SEGÚN SU ARQUITECTURA

La arquitectura, definida por el tipo de configuración general del robot, se clasifica en lo siguiente:

1) Poli-articulados.- Bajo este grupo están los robots de muy diversa forma y configuración cuya característica común es la de ser básicamente sedentarios -aunque excepcionalmente pueden ser guiados para efectuar desplazamientos limitados en un determinado espacio de trabajo según uno o más sistemas de coordenadas y con un número limitado de grados de libertad".

En este grupo se encuentran los manipuladores, los robots industriales, los robots cartesianos y algunos robots industriales y se emplean cuando es preciso abarcar una zona de trabajo relativamente amplia o alargada, actuar sobre objetos con un plano de simetría vertical o deducir el espacio ocupado en el suelo.

Figura II 9. Robot Poli-articulado



Fuente: mossa17.wordpress.com

2) Móviles.- Son robots que tienen la capacidad de moverse o desplazarse, basada en carros o plataformas. Siguen una trayectoria guiándose por la información obtenida de su entorno mediante sensores internos o externos. Estos robots son utilizados para el transporte de piezas de un punto a otro de una cadena de fabricación o en un entorno de difícil acceso.

Estos robots se guían mediante pistas o trayectorias pre establecidas; a través de la radiación electromagnética de circuitos empotrados en el suelo, o a través de bandas detectadas fotoeléctricamente, pueden esquivar obstáculos en su camino y estar dotados de un nivel relativamente elevado de inteligencia que les permita tomar decisiones de corrección de camino o selección de diferentes caminos dependiendo del costo, tiempo,

distancia entre otras variables. Lo que permitirá seleccionar una trayectoria óptima dependiendo de la aplicación que se le quiera dar a un robot móvil.

Figura II 10. Robotino



Fuente: Festo-Robotino

3) Androides.- Son robots que intentan reproducir total o parcialmente la forma y el comportamiento cinemático del ser humano. Actualmente los androides son todavía dispositivos muy poco evolucionados y sin utilidad práctica, y destinados, fundamentalmente, al estudio y experimentación.

Figura II 11. Robot Androide



Fuente: www.taringa.net

El problema o reto que presentan estos robots es la de la locomoción bípeda es decir sobre sus extremidades inferiores imitando a una persona. En este caso, el principal

problema es controlar dinámicamente y coordinadamente en el tiempo real el proceso y mantener simultáneamente el equilibrio del robot.

4) Zoomórficos.- Los robots zoomórficos, constituyen una clase caracterizada principalmente por sus sistemas de locomoción que imitan a los diversos seres vivos. A pesar de la disparidad morfológica de sus posibles sistemas de locomoción es conveniente agrupar a los robots zoomórficos en dos categorías principales: caminadores y no caminadores. El grupo de los robots zoomórficos no caminadores está muy poco evolucionado. Por el contrario, los robots zoomórficos caminadores multípedos son numerosos y están siendo desarrollados como verdaderos vehículos terrenos, piloteando o autónomos, capaces de evolucionar en superficies muy irregulares.

Figura II 12. Robot zoomórfico



Fuente: robots10.galeon.com

Las aplicaciones de estos robots están enfocadas al campo de la exploración espacial y en el estudio de los volcanes.

5) Híbridos.- Estos robots son de difícil clasificación ya que su estructura se sitúa entre una combinación con alguna de las anteriores expuestas, bien sea por conjunción o por yuxtaposición. Por ejemplo, un dispositivo segmentado articulado y con ruedas, al mismo tiempo uno de los atributos de los robots móviles y de los robots zoomórficos. De igual forma pueden considerarse híbridos algunos robots formados por la

yuxtaposición de un cuerpo formado por un carro móvil y de un brazo semejante al de los robots industriales.

En similar situación se localizan algunos robots antropomorfos y que no pueden clasificarse ni como móviles ni como andróides, tal es el caso de los robots personales. [7]

Figura II 13. Robot híbrido



Fuente: www.taringa.net

2.5. ROBÓTICA MÓVIL

En los últimos años, el área de los robots móviles ha tenido una importante actividad y desarrollo, ya que ha sido el sistema preferido por investigadores en las áreas de Inteligencia Artificial, Control Inteligente e Instrumentación.

Es el área de la robótica que desarrolla robots capaces de desplazarse y actuar sin intervención humana. Debido a su movilidad y manipulación, eventualmente podrían reemplazar a los humanos en sistemas de manufactura y servicios industriales.

2.5.1. ROBOTS MÓVILES

Un robot móvil es una máquina automática que es capaz de movimiento en cualquier ambiente dado. Los robots móviles tienen la capacidad de moverse en su entorno y no se fijan a una ubicación física. Este tipo de robots tienen un gran rango de aplicaciones, se pueden clasificar según el tipo de medio que usan para movilizarse así:

- Ruedas

- Oruga
- Patas

2.5.1.1. ROBOTS CON RUEDAS

Los vehículos de ruedas son, con mucho, los más populares por varias razones prácticas, son más sencillos y más fáciles de construir, la carga que pueden transportar es mayor que la que puede soportar otros tipos de robots móviles

A esto se puede añadir el que se pueden utilizar vehículos de ruedas de radio control para usarlos como bases de robots.

Figura II 14. Robot con ruedas



Fuente: www.limoncellodigital.com

La principal desventaja de las ruedas es su empleo en terreno irregular, en el que se comportan bastante mal. Normalmente un vehículo de ruedas podrá sobrepasar un obstáculo que tenga una altura no superior al radio de sus ruedas.

2.5.1.2. ROBOTS CON PATAS

Potencialmente los robots con patas pueden superar, con mayor facilidad que los otros, los problemas de los terrenos irregulares. Según el número de patas del robot, se pueden denominar cuadrúpedos, hexápodos, octópodos.

Figura II 15. Robot cuadrúpedo



Fuente: desocupe312.blogspot.com

2.5.1.3. ROBOT ORUGA

Las orugas son unos sistemas de tracción muy peculiares que destacan por su gran potencia y son aptos para todo tipo de terreno la tracción se reparte de manera uniforme a lo largo de cada articulación es de sencillo manejo y gracias a su gran tracción en las llantas y al diseño mismo de la oruga se lo utiliza en terrenos inestables o irregulares de difícil acceso.

Figura II 16. Robot oruga



Fuente: www.xatakaciencia.com

2.6. NAVEGACIÓN DE ROBOTS MÓVILES

La navegación es la técnica de conducir un robot móvil mientras atraviesa un entorno para alcanzar un destino o meta sin chocar con ningún obstáculo. Cuando se desconoce el entorno, el robot debe poseer la capacidad de reaccionar ante situaciones inesperadas,

esto se logra a través de la percepción del entorno mediante el uso de sensores. Mientras que si se trata de un entorno conocido, el uso de los sensores se vuelve secundario y las tareas a seguir serían: planificar una óptima trayectoria libre de obstáculos dependiendo de los puntos de partida y llegada y obviamente que el robot pueda seguir y cumplir físicamente esta trayectoria.

La navegación se puede expresar en las siguientes etapas:

Medición:

- Sensar el entorno
- Detectar objetos
- Ingresar comandos de usuario

Modelación:

- Mapa del entorno
- Modelo de objetos
- Mapa de caminos

Percepción:

- Buscar caminos
- Detectar situaciones de colisión
- Conocer el mapa

Planeación:

- Descomponer las tareas en subtareas
- Seleccionar caminos
- Escoger alternativas cuando el camino es bloqueado

Acción:

- Navegar
- Atravesar caminos evitando colisiones
- Control en base al modelo cinemático o dinámico del robot.

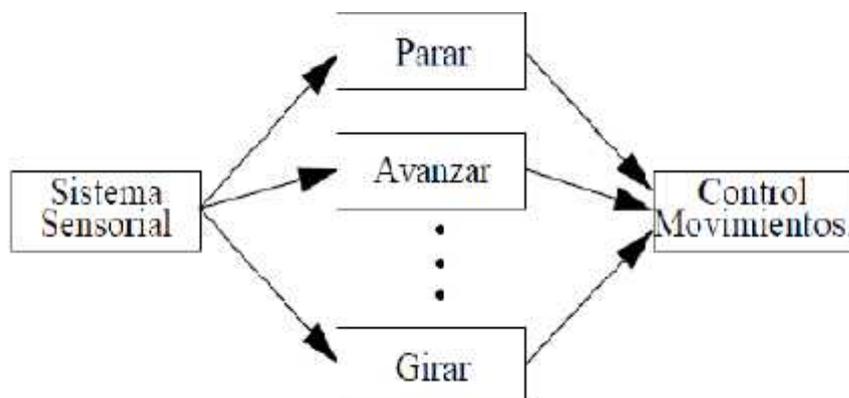
2.7. ODOMETRÍA

La odometría es el cálculo de la posición y orientación del robot móvil. Para corregir los problemas de los robots con caminos definidos, algunos robots siguen un camino pre-programado, utilizando marcas en el piso para corregir su posición, algunos utilizan rayos láser o colocan marcas a lo largo de su trayectoria. [8]

Mediante el uso de la odometría del vehículo se puede realizar esta acción, pero debido a la naturaleza del método y a las características de los sensores utilizados, la estimación efectuada se ve afectada por errores acumulativos (Watanabe y Yuta, 1.990). Cuando dichos errores alcanzan niveles indeseables se hace necesario eliminarlos mediante la utilización de algoritmos de estimación de la posición basados en referencias externas. La navegación estratégica tiene sus limitaciones en entornos dinámicos no conocidos, ya que requiere un completo conocimiento de la dinámica de los posibles obstáculos móviles, además de una adecuada actualización del mapa de entorno.

Por otra parte, se podrían utilizar sensores como: ultrasónicos, infrarrojos, táctiles, etc., para que reaccionen en el entorno dinámico, pero aquí estaría perdiendo importancia la planificación y seguimiento de caminos. En base a esto como se observa en la Figura II 17, se podría aplicar una arquitectura descompuesta en módulos cada uno especializado en una tarea específica, llamadas comportamientos.

Figura II 17. Navegación reactiva



Fuente: <http://bibdigital.epn.edu.ec/bitstream/15000/3772/1/CD-3550.pdf>

En este sistema de navegación, según la información proporcionada por los sensores, se activan uno o más comportamientos, siendo el comportamiento final la suma de cada

acción simple. Este tipo de navegación ha sido aplicada en múltiples aplicaciones para entornos desconocidos, inclusive dinámicos y sin colisionar con los obstáculos, pero con dificultad pueden seguir un plan establecido, lo que es necesario en misiones reales.

2.8. CONFIGURACIÓN OMNIDIRECCIONAL.

Este tipo de configuración esta provista de ruedas omnidireccionales, lo que hace que los cálculos de odometría sean más complicados; pero el robot podrá moverse en cualquier dirección.

Figura II 18. Robot omnidireccional



Fuente: http://metaltech.gronerth.com/triciclo_robot_con_ruedas_omnidireccionales-p36668.html

Esta configuración ostenta un alto grado de maniobrabilidad en un entorno de trabajo, puesto que se pueden desplazar en cualquier trayectoria del eje XY, sin la necesidad de reorientarse.

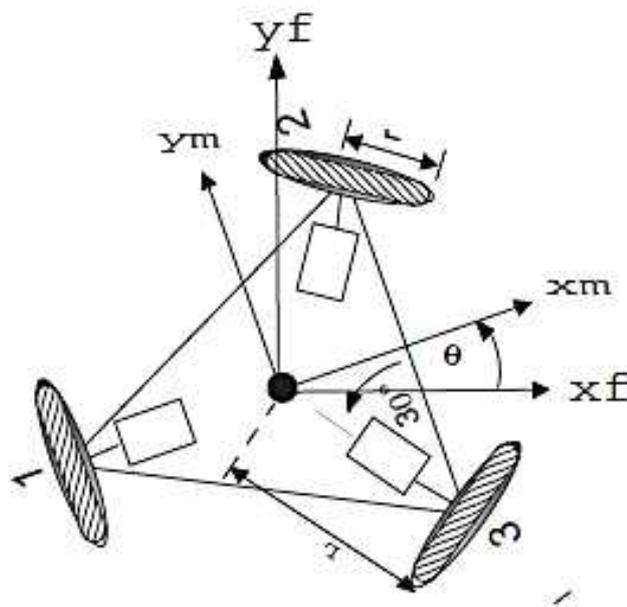
La rueda omnidireccional, se define como una rueda estándar, a la cual se le ha dotado de una corona de rodillos; cuyos ejes de giro de los rodillos resultan perpendiculares a la dirección normal de avance. De este modo, al aplicarle una fuerza lateral, los rodillos giran sobre sí mismo y permite que la componente de la velocidad en el eje x (V_x), no sea nulo, y por tanto, se elimina la restricción de no holomicidad. Las ruedas omnidireccionales tienen por lo general un costo elevado.

El modelo cinemático de esta configuración se muestra en la figura II 19.

La configuración cinemática de este robot, se define por una estructura triangular equilátera, en cuyos vértices se encuentran colocadas tres ruedas omnidireccionales.

La distancia desde el centro del triángulo equilátero a cualquiera de las ruedas es L . Todas las ruedas son no direccionales. Se representa como: ω_1 , ω_2 y ω_3 , a la velocidad angular de la rueda 1, rueda 2 y rueda 3.

Figura II 19. Modelo de la configuración omnidireccional



Fuente: <http://www.tamps.cinvestav.mx/~mgomez/Odometria/node9.html>

Finalmente, el modelo cinemático en el marco del robot, queda representado por medio de la siguiente matriz. Donde r es el radio de las ruedas. [9]

$$\begin{bmatrix} V_x \\ V_y \\ \theta^\bullet \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{\sqrt{3}}r & \frac{1}{\sqrt{3}}r \\ \frac{2}{3}r & \frac{-1}{3}r & \frac{-1}{3}r \\ \frac{r}{3L} & \frac{r}{3L} & \frac{r}{3L} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

Por lo tanto estas son las ecuaciones matemáticas del modelo cinemático de un robot omnidireccional.

2.9. MODELO CINEMATICO DEL ROBOTINO

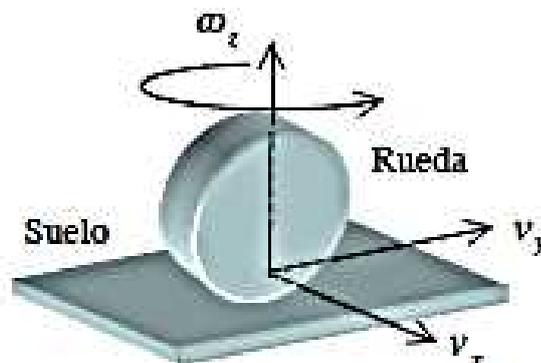
La cinemática, se centra en el estudio del movimiento del robot en función de su geometría. Entre las aplicaciones inmediatas se encuentran la posibilidad de utilizarlo como modelo matemático de partida para el diseño del controlador, la simulación del comportamiento cinemático del vehículo, o para establecer las ecuaciones de los cálculos odométricos.

Normalmente, se consideran las siguientes limitaciones para la construcción del modelo cinemático:

- El robot se mueve sobre una superficie plana.
- No existen elementos flexibles en la estructura del robot (incluidas las ruedas).
- Las ruedas poseen uno o ningún eje de direccionamiento, de manera que este último siempre es perpendicular al suelo.
- No se consideran ningún tipo de fricciones en elementos móviles del vehículo, o contra el suelo.

El comportamiento cinemático se establece en el principio de que las ruedas están en contacto con el suelo se comportan como una articulación planar de tres grados de libertad, tal y como aparece en la figura II 20.

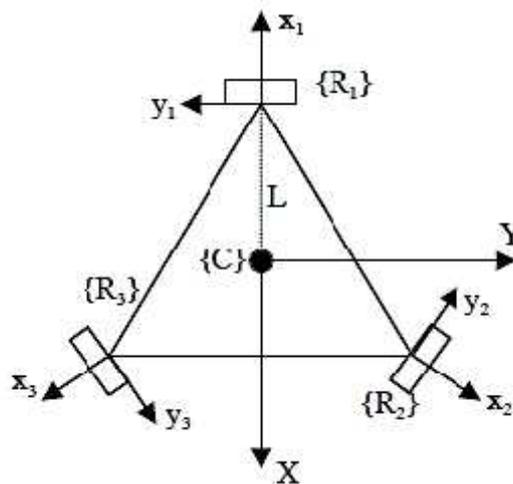
Figura II 20. Rueda en contacto con el suelo



Al suponerse la rueda como un elemento rígido, ésta entra en contacto con el suelo en un solo punto, que sirve de origen al sistema de referencias solidario dibujado en la figura II 21. Se utiliza para definir los tres grados de libertad antes mencionados.

La dirección v_y determina el sentido normal de avance de la rueda; el eje v_x indica los deslizamientos laterales, y w_z la velocidad rotacional que se produce cuando el vehículo realiza un giro. En el caso de una rueda convencional, la componente v_x , se supone siempre nula, sin embargo, existen ruedas omnidireccionales diseñadas para eliminar la mencionada restricción.

Figura II 21. Esquema cinemático robot omnidireccional



Fuente: <http://ingenieria1.udistrital.edu.co/digital/article/view/280/459>

La cinemática directa tiene como objetivo el cálculo de la velocidad lineal y angular del robot a partir de las correspondientes aportaciones de cada una de sus ruedas.

Como se aprecia en la mencionada figura II 21, la configuración cinemática del robot se define por una estructura triangular equilátera, en cuyos vértices se han dispuesto tres ruedas omnidireccionales

La distancia del origen del sistema $\{C\}$ (situado en el centro geométrico) a cualquiera de las ruedas viene dada por L . Todas las ruedas se definen como no direccionables, y por tanto, se produce la igualdad entre los siguientes sistemas coordenados $\{F_i\}=\{D_i\}=\{R_i\}$, es decir, para toda i , se cumple $b_i = 0^\circ$ y $g_i = 0^\circ$. La tabla II.I recoge los valores de los parámetros del modelo cinemático.

Tabla II I Parámetros de configuración cinemática

| | Rueda 1 | Rueda 2 | Rueda 3 |
|-------------|----------------|--|---|
| α_i | 180° | 60° | -60° |
| β_i | 0° | 0° | 0° |
| γ_i | 0° | 0° | 0° |
| δ_i | (0,0,0) | (0,0,0) | (0,0,0) |
| λ_i | (-L,0,0) | $\left(\frac{L}{2}, \frac{L\sqrt{3}}{2}, 0\right)$ | $\left(\frac{L}{2}, -\frac{L\sqrt{3}}{2}, 0\right)$ |

Fuente: MODELADO CINEMATICO DE UN ROBOT MÓVIL OMNI-DIRECCIONAL.pdf

Con el objeto de obtener el jacobiano de la rueda, se multiplica la matriz J_i , detallada en (1), por la matriz de conversión de la actuación para ruedas omnidireccionales presentada en la expresión (2).

$$(1) \quad \begin{pmatrix} v_{Cx} \\ v_{Cy} \\ \omega_C \end{pmatrix} = \begin{pmatrix} c_i & -s_i & p_{iy} & -\lambda_{iy} \\ s_i & c_i & -p_{ix} & \lambda_{ix} \\ 0 & 0 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} v_{ix} \\ v_{iy} \\ \omega_i \\ \beta_i \end{pmatrix} =$$

$$V_C = \hat{J}_i \cdot \dot{q}_i$$

Donde C_i representa al coseno y S_i al seno del ángulo i ; además V_{cx} y V_{cy} son las componentes de V_c , que es la velocidad del robot con respecto al centro de masa del mismo u el origen del sistema.

$$(2) \quad W_i \cdot \dot{q}_i = \begin{pmatrix} 0 & r & 0 \\ -R & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \omega_{ix} \\ \omega_{ir} \\ \omega_{iz} \end{pmatrix}$$

La matriz W_i modela, en este ejemplo una rueda de radio R , omnidireccional, con rodillos de radio r a noventa grados, tractora y no direccionable. Por otro lado, con respecto al vector q , W_{ix} es el grado de actuación del motor, W_{ir} la velocidad angular de giro de los rodillos y W_{iz} el deslizamiento rotacional en el eje vertical de la rueda.

De este modo, el jacobiano de la rueda i -ésima queda reflejado como sigue en (3):

$$(3) \quad J_i = \begin{pmatrix} R \cdot s_i & r \cdot c_i & \lambda_i \\ -R \cdot c_i & r \cdot s_i & -\lambda_i \\ 0 & 0 & 1 \end{pmatrix}$$

Se sustituyen los parámetros de la tabla II-I en la expresión (3), y se obtienen los jacobianos de cada una de las ruedas.

$$(4) \quad J_1 = \begin{pmatrix} 0 & -r & 0 \\ R & 0 & L \\ 0 & 0 & 1 \end{pmatrix}$$

$$(5) \quad J_2 = \begin{pmatrix} \frac{\sqrt{3} \cdot R}{2} & \frac{r}{2} & \frac{\sqrt{3} \cdot L}{2} \\ -\frac{R}{2} & -\frac{\sqrt{3} \cdot r}{2} & -\frac{L}{2} \\ 0 & 0 & 1 \end{pmatrix}$$

$$(6) \quad J_3 = \begin{pmatrix} -\frac{\sqrt{3} \cdot R}{2} & \frac{r}{2} & -\frac{\sqrt{3} \cdot L}{2} \\ -\frac{R}{2} & -\frac{\sqrt{3} \cdot r}{2} & -\frac{L}{2} \\ 0 & 0 & 1 \end{pmatrix}$$

Las matrices J_1 , J_2 y J_3 , se utilizan para resolver el jacobiano completo del vehículo como aparece detallado en (7) necesario para calcular la velocidad en el punto V_c .

$$(7) \quad V_c = \underbrace{(A^T \cdot A)^{-1} \cdot A^T \cdot B \cdot \dot{q}}_J$$

$$V_c = J \cdot \dot{q}$$

Donde la matriz J representa el jacobiano completo del vehículo.

$$(8) \quad J = \begin{pmatrix} 0 & -\frac{r}{3} & 0 & \frac{R}{a} & \frac{r}{6} & \frac{L}{a} & -\frac{R}{a} & \frac{r}{6} & -\frac{L}{a} \\ \frac{R}{3} & 0 & \frac{L}{3} & -\frac{R}{6} & \frac{r}{a} & -\frac{L}{6} & -\frac{R}{6} & -\frac{r}{a} & -\frac{L}{6} \\ 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} \end{pmatrix}$$

$$a = 2\sqrt{3}$$

Esta matriz jacobiana relaciona la velocidad del vehículo con las de giro que aparecen en las ruedas: actuado; deslizamiento en el eje vertical, y de los rodillos. Desde el punto de vista del control interesan exclusivamente los grados actuados. Para obtenerlo se imponen la condición de no deslizamiento lo que supone que el robot está actuando de manera adecuada para que no se deslice; la expresión es la siguiente:

$$(9) \quad \Omega(A) \cdot B \cdot \dot{q} = 0$$

$$\begin{pmatrix} 0 & \frac{2r}{3} & 0 & \frac{R}{a} & \frac{r}{6} & \frac{L}{a} & -\frac{R}{a} & \frac{r}{6} & -\frac{L}{a} \\ -\frac{2R}{3} & 0 & -\frac{2L}{3} & -\frac{R}{6} & \frac{r}{a} & -\frac{L}{6} & -\frac{R}{6} & -\frac{r}{a} & -\frac{L}{6} \\ 0 & 0 & -\frac{2}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & -\frac{r}{3} & 0 & -\frac{R}{\sqrt{3}} & -\frac{r}{3} & -\frac{L}{\sqrt{3}} & -\frac{R}{a} & \frac{r}{6} & -\frac{L}{a} \\ \frac{R}{3} & 0 & \frac{L}{3} & \frac{R}{3} & -\frac{r}{\sqrt{3}} & \frac{L}{3} & -\frac{R}{6} & -\frac{r}{a} & -\frac{L}{6} \\ 0 & 0 & \frac{1}{3} & 0 & 0 & -\frac{2}{3} & 0 & 0 & \frac{1}{3} \\ 0 & -\frac{r}{3} & 0 & \frac{R}{a} & \frac{r}{6} & \frac{L}{a} & \frac{R}{\sqrt{3}} & -\frac{r}{3} & \frac{L}{\sqrt{3}} \\ \frac{R}{3} & 0 & \frac{L}{3} & -\frac{R}{6} & \frac{r}{a} & -\frac{L}{6} & \frac{R}{3} & \frac{r}{\sqrt{3}} & \frac{L}{3} \\ 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & -\frac{2}{3} \end{pmatrix} \cdot \dot{q} = 0$$

El sistema anterior resulta indeterminado, ya que existen tres filas que son combinación lineal de las otras. Por ello, se despejan las variables no actuadas de las ruedas en función de las que sí lo son. Se obtiene como resultado las siguientes ecuaciones:

$$(10) \quad \begin{aligned} \omega_{1r} &= \frac{R(\omega_{3x} - \omega_{2x})}{r\sqrt{3}} & \omega_{1z} &= -\frac{R(\omega_{1x} + \omega_{2x} + \omega_{3x})}{3L} \\ \omega_{2r} &= \frac{R(\omega_{1x} - \omega_{3x})}{r\sqrt{3}} & \omega_{2z} &= \omega_{1z} \\ \omega_{3r} &= \frac{R(\omega_{2x} - \omega_{1x})}{r\sqrt{3}} & \omega_{3z} &= \omega_{1z} \end{aligned}$$

Al sustituir el resultado obtenido en el punto (10) en el jacobiano completo de la ecuación del punto (8), se obtienen las velocidades globales en función de las actuaciones.

$$(11) \quad \begin{aligned} v_{cx} &= \frac{R(\omega_{2x} - \omega_{3x})}{\sqrt{3}} \\ v_{cy} &= \frac{R(2\omega_{1x} - \omega_{2x} - \omega_{3x})}{3} \\ \omega_c &= -\frac{R(\omega_{1x} + \omega_{2x} + \omega_{3x})}{3L} \end{aligned}$$

La ecuación del punto (11) en forma matricial constituye el jacobiano actuado del robot móvil en estudio, el cual se representa en el punto (12);

Donde R es el radio de la llanta

Y L la longitud desde el centro de masa hasta el eje de la llanta

$$(12) \quad \begin{pmatrix} v_{cx} \\ v_{cy} \\ \omega_c \end{pmatrix} = \overbrace{\begin{pmatrix} 0 & \frac{R}{\sqrt{3}} & -\frac{R}{\sqrt{3}} \\ \frac{2R}{3} & \frac{R}{3} & -\frac{R}{3} \\ \frac{R}{3L} & \frac{R}{3L} & -\frac{R}{3L} \end{pmatrix}}^J \cdot \begin{pmatrix} \omega_{1x} \\ \omega_{2x} \\ \omega_{3x} \end{pmatrix}$$

La velocidad para cada una de las ruedas se puede expresar en una función lineal de la velocidad y la velocidad angular del robot como se muestra:

$$(13) \quad \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{3} - \theta) & -\cos(\frac{\pi}{3} - \theta) & L \\ \sin(\frac{\pi}{3} + \theta) & -\cos(\frac{\pi}{3} + \theta) & L \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$$

Finalmente partiendo de la ecuación (13) y sabiendo que la velocidad angular de una rueda es $\dot{\phi}_i = \frac{v_i}{r}$, para lograr un movimiento en coordenadas cartesianas a partir de $\{ \dot{x}, \dot{y}, \dot{\theta} \}$; se obtienen las velocidades angulares de las ruedas mediante la siguiente transformación lineal:

$$(14) \quad \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{pmatrix} = \frac{1}{r} \begin{pmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{3} - \theta) & -\cos(\frac{\pi}{3} - \theta) & L \\ \sin(\frac{\pi}{3} + \theta) & -\cos(\frac{\pi}{3} + \theta) & L \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$$

Donde r es el radio de las ruedas, L es la distancia desde el centro del robot a las ruedas, y θ es el ángulo entre el eje X y la primera rueda.

2.10. ESTIMACIÓN DE LA POSICIÓN DE UN ROBOT MÓVIL

Los robots móviles se caracterizan por su capacidad de desplazarse de un lugar a otro en forma autónoma (capacidad de percibir, modelar, planificar y actuar para alcanzar unos objetivos sin la intervención, o con poca intervención de personas), ya sea en un lugar conocido parcialmente o desconocido en su totalidad.

Un robot móvil es un sistema en el cual se encuentran inmersos diversos subsistemas de locomoción, control de movimientos, percepción y planificación que interactúan entre sí. El subsistema de percepción hace que el robot sea capaz de interactuar en entornos cambiantes, así como poder reaccionar ante eventos inesperados, lo que exige la existencia de un sistema sensorial que suministre información sobre el entorno. Esta información requerida debe permitir al robot realizar tres tareas fundamentales: estimar

su posición y orientación, mantener actualizado el mapa del entorno y detectar los posibles obstáculos.

El robot móvil rara vez va equipado con un único sensor para realizar todas estas tareas, sino que la práctica más habitual consiste en combinar dentro del sistema sensorial varios sensores que en mayor o menor medida se complementan.

Algunas veces se emplean varios sensores redundantes con el propósito de validar la información adquirida.

Para que un robot móvil pueda afrontar tareas como generar trayectorias, evitar obstáculos, etc. Se requiere que este sea capaz de determinar su localización (posición y orientación) con respecto a un sistema de referencia absoluto.

La estimación de la posición de un robot móvil, viene dado por: el tipo de entorno por el cual ha de navegar, el conocimiento que se tenga sobre el entorno, tipos de sensores que dispone, y de la tarea a realizar. [9]

La mayoría de robots móviles disponen de encoders, para detectar su movimiento, lo que les permite estimar en cada instante su posición, empleándose modelos de locomoción. Sin embargo, esta estimación no resulta muy conveniente para la mayoría de aplicaciones, ya que no es lo suficientemente precisa para ello.

El motivo es que los errores, por más pequeños que sean se van acumulando durante la navegación, por lo cual se suelen usar sistemas de posicionamiento externo.

Para la estimación de la posición y orientación de un robot móvil autónomo se distinguen los estimadores explícitos y los estimadores basados en la percepción del entorno. Los estimadores explícitos proporcionan directamente la posición y orientación del robot por medio de medidas, sin tener que realizar procesamiento de información para la interpretación del entorno.

Los estimadores basados en la percepción del entorno emplean sensores que suministran información del exterior, por medio de la cual se puede saber la localización del vehículo mediante comparación de esta información con otros datos o modelos conocidos (pueden ser mapas, marcas naturales, objetos, etc).

El problema de la estimación está en que las medidas se encuentran asociadas a una cierta cantidad de ruido. Tanto las propias medidas como las estimaciones realizadas a partir de estas, tendrán una naturaleza aleatoria. Si no existiese el ruido, la posición y orientación del robot móvil se obtendrían simplemente al resolver el modelo matemático.

2.11. ESTIMACIÓN EXPLÍCITA DE LA POSICIÓN.

Se considera como estimación explícita a todos aquellos sistemas capaces de estimar la posición del robot móvil sin que se realice una interpretación del entorno.

En los sistemas de estimación explícita se distingue la estimación basada en medidas internas y la estimación basada en estaciones de transmisión. [10]

La estimación basada en medidas internas trabaja solamente con los sensores internos del robot móvil. Los sensores empleados pueden ser: giroscopios, encoders, detectores de norte, acelerómetros, tacómetros, etc.

La estimación basada en estaciones de transmisión se basa en el empleo de dos unidades, la unidad montada sobre el robot móvil y la unidad o unidades externas. La unidad montada sobre el robot móvil actúa como sensor receptor, mientras que las externas actúan como emisores o transmisores de señales de referencia.

Estimadores basados en medidas internas.

La posición y orientación de un robot móvil puede obtenerse integrando la trayectoria recorrida por este a partir de una serie de medidas internas como pueden ser: las vueltas que dan las ruedas, la velocidad, aceleración, cambios de orientación, etc. Se pueden distinguir los sistemas odométricos y los sistemas de navegación inercial. [10].

2.12. SISTEMAS ODOMÉTRICOS

La odometría tiene por objeto determinar la posición y orientación del robot móvil a partir del número de pulsos obtenidos cuando giran las ruedas. Se utilizan codificadores ópticos de elevada precisión en al menos dos ruedas. La simplicidad y el bajo costo es

una gran ventaja que nos ofrece el sistema odométrico. Sin embargo, es necesaria una calibración constante, debido al desgaste y pérdida de presión de las ruedas, desajuste de los ejes, etc. Esta técnica es vulnerable a las imperfecciones en el suelo, al deslizamiento de las ruedas y a las variaciones en la carga transportada (aunque en este caso es posible diseñar un modelo para corregir las desviaciones introducidas).

La idea fundamental de la odometría es la integración de los incrementos del movimiento en el tiempo, lo que produce inevitablemente una acumulación de errores. La acumulación de errores en la orientación, causa grandes errores en la posición, los cuales se incrementan proporcionalmente con la distancia recorrida por el robot.

A pesar de estas limitaciones, la odometría es una parte importante en los sistemas de navegación del robot. Las tareas de navegación serán simplificadas, si se mejora la precisión de la odometría. [9]

2.13. SISTEMAS DE NAVEGACIÓN INERCIAL (INS)

Este sistema obtiene la posición y orientación del robot móvil por medio de las medidas de aceleraciones y ángulos de orientación. Para obtener la posición, se integra la aceleración obteniéndose la velocidad, la cual se integra para finalmente obtener la posición. Como se indicó, este sistema emplea la aceleración para la obtención de la posición, para lo cual hace uso de acelerómetros que suelen estar basados en sistemas pendulares.

La precisión del acelerómetro resulta crítica, ya que, los errores en la aceleración aunque sean pequeños afectan la obtención de la posición, debido a la doble integración de la aceleración. Para medir la orientación, se emplean giroscopios e inclinómetros. [9]

También es posible medir el ángulo de orientación mediante brújulas. Los sistemas de navegación inercial no son afectados por los problemas derivados de la interacción del vehículo con el suelo. En este sistema se pueden corregir los efectos de las ondulaciones e irregularidades del terreno, lo que hace que en la práctica, sean mucho más fiables y precisos que los sistemas basados en odometría. Estos sistemas son más frágiles y caros que los sistemas basados en odometría.

2.14. ESTIMACIÓN MEDIANTE PERCEPCIÓN DEL ENTORNO

Consiste en dotar al robot móvil de un sistema sensorial capaz de proporcionar suficiente información del entorno como para que este pueda de forma autónoma determinar su localización. El sistema sensorial puede operar en base a distintos tipos de sensores (cámaras, sonares, escáner láser, etc.) como se indica en la figura II 22; y seleccionando determinados tipos de datos u objetos a partir del conjunto de información adquirida (marcas naturales, puntos de interés, entorno completo percibido, etc.).

En cualquier caso, la localización del robot móvil se determina a través del emparejamiento de los datos extraídos del entorno por el sistema sensorial, con datos previamente conocidos del entorno.

Los sensores utilizados para la navegación de un robot móvil pueden situarse en dos grupos: activos y pasivos. [9]

Figura II 22. Sistema sensorial



Fuente: dspace.ups.edu.ec/bitstream/123456789/175/2/Capitulo%201.pdf

Los sensores activos son aquellos que emiten algún tipo de energía al medio (luz infrarroja, ultrasonidos, luz láser, ondas de radio, etc.).

Los sensores denominados pasivos se limitan a captar la energía ya existente en el medio. De este tipo son las cámaras de video, las cuales perciben el entorno a través de la cantidad de luz que les llegan procedente de fuentes luminosas o bien a través de reflexiones en los objetos del entorno.

2.15. PLATAFORMA CONTROLADA POR PLC

La plataforma controlada por PLC es una alternativa al controlador de robot tradicional, el cual mejora la integración en la línea de producción, mejora los tiempos de respuesta y en general productividad global del MPS.

2.15.1. CONTROLADOR LÓGICO PROGRAMABLE

PLC (Programmable Logic Controller – Controlador Lógico Programable), es un dispositivo digital electrónico de memoria programable, que guarda ordenes previamente programadas, permitiendo la ejecución de funciones específicas, como son lógicas, secuenciales, temporizadas, de conteo, matemáticas, con la tarea de controlar equipos y procesos.

También, se lo define como un dispositivo electrónico, que realiza unas tarea programada de forma secuencial, La ejecución del programa, puede ser detenida, para realizar operaciones lógicas que interrumpirán momentáneamente el ciclo de funcionamiento , para poder cumplir operaciones consideradas con más prioridad al momento de programar, pero garantizando la total ejecución del programa principal.

Estos controladores son muy usados en ambientes industriales, donde la velocidad de toma de decisiones es prioritario, para así responder en tiempo real a posibles problemas q se puedan suscitar.

También se puede definir como un equipo electrónico, el cual realiza la ejecución de un programa de forma cíclica. La ejecución del programa puede ser interrumpida momentáneamente para realizar otras tareas consideradas más prioritarias, pero el aspecto más importante es la garantía de ejecución completa del programa principal.

Estos controladores son utilizados en ambientes industriales donde la decisión y la acción deben ser tomadas en forma muy rápida, para responder en tiempo real. Los PLC son utilizados donde se requieran tanto controles lógicos como secuenciales o ambos a la vez. [11]

Campos de aplicación

El PLC por sus especiales características de diseño tiene un campo de aplicación muy extenso. La constante evolución del hardware y software amplía constantemente este campo, para poder satisfacer las necesidades que se detectan en el espectro de sus posibilidades reales.

Su uso está difundido en el campo industrial, por la fiabilidad de su diseño y su funcionalidad es usado en ambientes hostiles, en donde es necesario el monitoreo y muestra de resultados en una manera eficiente e independiente.

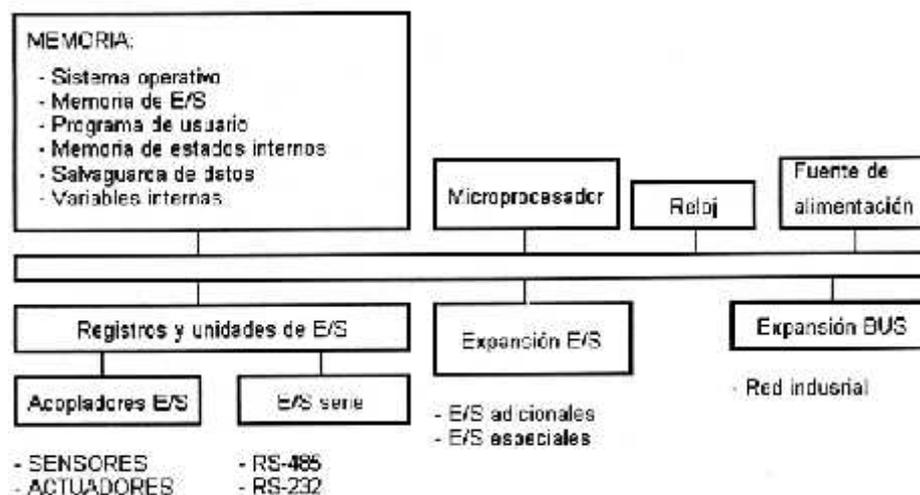
Gracias a las distintas formas de comunicación, es la forma más factible de realizar sistemas SCADA y ser controlado a distancia.

Sus principales características y aplicaciones son las siguientes

- Espacio mínimo
- Maquinaria de procesos variables.
- Procesos secuenciales.
- Procesos de fabricación habitualmente variables.
- Verificación de programación agrupada de las partes del proceso.
- Instalaciones de procesos complejos y amplios. [12]

2.15.2. ESTRUCTURA DEL PLC

Figura II 23. Estructura de PLC



Fuente: <http://www.profesormolina.com.ar/tecnologia/plc/introd/image002.gif>

Un controlador lógico programable está constituido por un conjunto de tarjetas o circuitos impresos, sobre los cuales están ubicados componentes electrónicos. [13]

La estructura básica del hardware está conformada por los siguientes elementos:

- **Fuente de alimentación.** La fuentes de alimentación, es uno de los principales componentes del controlador lógico programable, ya que su papel fundamental no es solo suministrar energía, para los requerimientos de operación del PLC (es decir al procesador, memoria, entradas/salidas), sino también al sistema de transmisión de datos, para un correcto envío/recepción de datos. Además debe proporcionar el voltaje correcto suministrado y debe alertar al CPU de posibles fallas en el el envío del voltaje. alertar al C.P.U. si todo marcha bien.
- Unidad de procesamiento central (CPU)
- **Módulos de memoria.**
 - EPROM: Siglas de Erasable Programmable Read-Only Memory son las siglas de (ROM borrable programable).
 - ROM: (read-only memory), Es la memoria de solo lectura, aquí se almacena la programación básica del plc para ejecutar su rutina, dentro de esta se encuentras su sistema operativo el cual es solo de lectura, destinado a ser solo leído y no sobrescrito.
 - RAM: Es la memoria de almacenamiento aleatorio, se usa para el almacenaje temporal de datos, los cuales se están ejecutando en el PLC, este tipo de memoria es temporal.
- **Unidad de programación.**
 - **Unidad Lógica** El núcleo de un PLC es la unidad lógica, basada en un microprocesador. Esta encargada de ejecutar las instrucciones programadas en la memoria.
 - **Unidad de memoria.** La memoria almacena el código de mensajes o instrucciones que tiene que ejecutar la unidad lógica del PLC.

En la actualidad las estructuras más representativas son tres en el mercado:

- Estructura compacta
- Estructura semimodular. (Estructura Americana)
- Estructura modular. (Estructura Europea)

2.15.2.1. ESTRUCTURA COMPACTA

Figura II 24. PLC Compacto



Fuente: <http://automationint.com/wp-content/uploads/2012/05/FX1S.jpg>

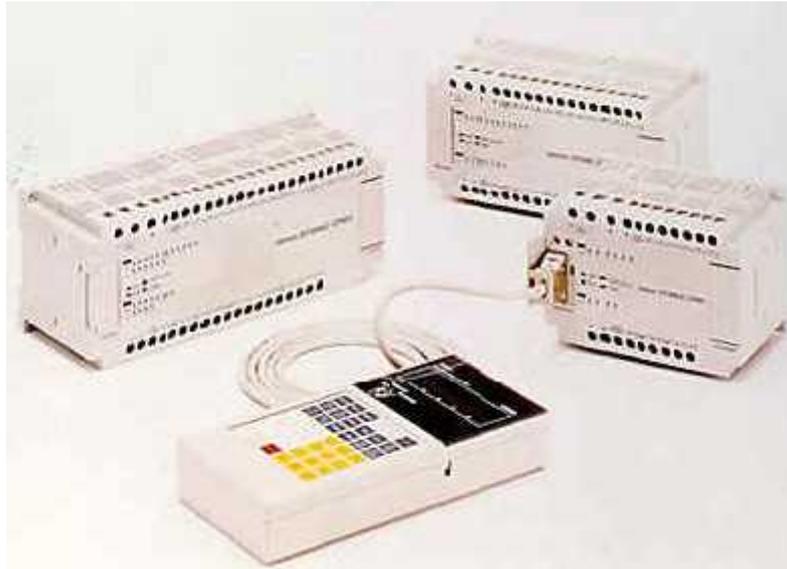
Este controlador lógico programable se diferencia por presentar en un solo conjunto todos sus componentes, siendo estos, fuente de alimentación CPU, memoria, entradas/salidas.

Son PLC de gama baja, siendo estos nanoautomatas, teniendo entre sus principales características una potencia de procesamiento de datos muy limitadas, dedicándose a controlar maquinas muy pequeñas o un uso netamente didáctico.

2.15.2.2. ESTRUCTURA SEMIMODULAR

Esta estructura básicamente, separa el módulo de entradas/salidas, del resto del controlador lógico programable, de tal manera que en un bloque compacto están la fuente de alimentación, CPU, separadamente del módulo de entradas/salidas.

Figura II 25. Estructura Semimodular



Fuente: <http://goo.gl/Ug9P9k>

Son controladores de gama media, usados por su capacidad de procesamiento media/alta

2.15.2.3. ESTRUCTURA MODULAR

Figura II 26. Estructura PLC modular



Fuente: <http://www.mitsubishielectric.in/images/fa/advanced-modular-plc.jpg>

Su principal característica es la de contar un módulo, predefinido para cada uno de sus componentes o elementos que componen el PLC, como puede ser el módulo de entradas y salidas, CPU, etc.

El emparejamiento de estos se realiza por el riel DIN, incorporado en cada uno de los módulos, en donde va alojado el Bus externo de unión, de los distintos módulos que lo componen.

Son PLC de gama alta, los que suelen tener este tipo de estructura, ya que permiten una gran flexibilidad y manipulación en su construcción.

2.15.3. CICLO DE FUNCIONAMIENTO DE UN PLC

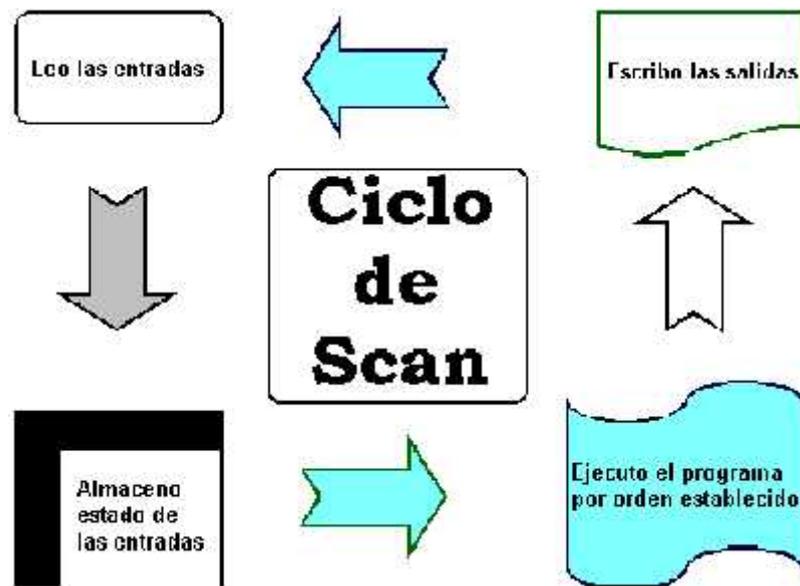
Son los pasos a ejecutar por el autómata, llevando a cabo el programa diseñado por el usuario, el cual está determinado por la longitud del programa y su programación lógica.

Esto debido al retardo de ejecución por cada instrucción realizada, por lo que el tiempo de ejecución es un factor crítico en la ejecución del programa.

En un sistema de control realizado por el PLC se tendrán los siguientes tiempos:

- 1) Vigilancia y exploración de las entradas.
- 2) Retardo de entrada.
- 3) Ejecución del programa de usuario.
- 4) Retardo en salidas.
- 5) Transmisión de las salidas.

Figura II 27. Ciclo de funcionamiento de un PLC



Fuente: <http://imagenes.unicrom.com/CicloScan.gif>

La causa principal, es que no se manejan las entradas y salidas de una forma directa, sino una imagen de memoria adquirida al comienzo del ciclo del plc, y estas no se modifican sino hasta el final del ciclo.

La etapa de vigilancia se encarga de verificar si se entró en un bucle infinito, siendo esta la causa principal para que arroje una alerta de error. [14]

2.15.4. GRAFCET

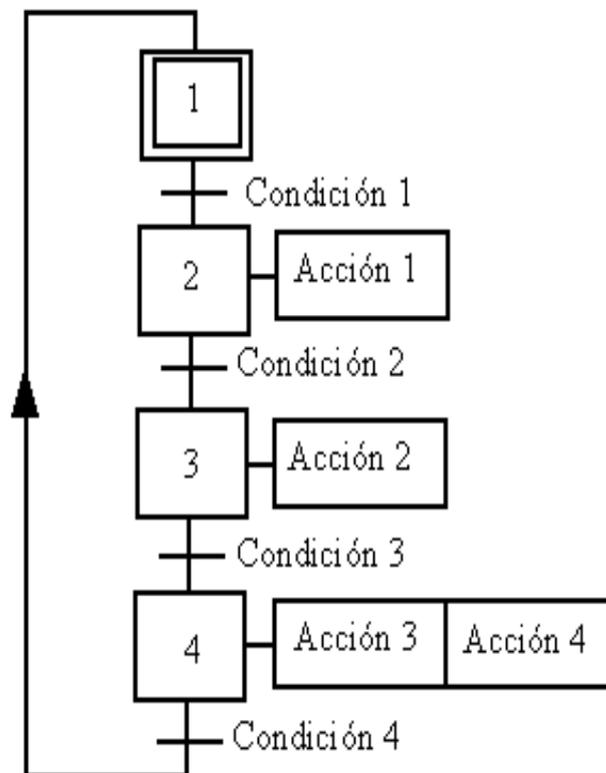
Creado a mediados de los años 70, por un conjunto de fabricantes de equipos industriales, resueltos a crear un lenguaje estándar para programar autómatas, que permita unificar las diferentes formas de programación.

El GRAFCET (Grphe Fonctionel de Commande Etape Transition o Grafico funcional de Etapas de Transición) es un diagrama sencillo que representa el progreso del proceso que se quiere automatizar, dividiendo las entradas y las operaciones a realizar, tomando en cuenta los procesos interludios o interrupciones que provocan esas tareas. Está definido por unos elementos gráficos y unas reglas de evolución que reflejan la dinámica del comportamiento del sistema.

Basados en el concepto anterior, GRAFCET usa elementos gráficos para representar las acciones a realizar, donde cada rectángulo representa la etapa a realizar y las líneas las condiciones necesarias a cumplir para avanzar a una nueva etapa, simplificando los métodos en un diagrama funcional, fácil de describir y entender las acciones que conllevaban a una transición y por ende al cambio de etapa.

Todo automatismo se puede representar en base a gráficos secuenciales, que representaran las etapas que el plc seguirá ejecutando automáticamente.

Figura II 28. Ejemplo de estructura GRAFCET



Fuente: <http://goo.gl/10jApQ>

2.15.4.1. ELEMENTOS GRÁFICOS

Etapas

Para representar el avance del proceso se simbolizan mediante gráficos, los cuales constituyen los estados de un proceso que se está realizando.

Figura II 29. Representación de Etapas

REPRESENTACION DE ETAPAS:



REPRESENTACION DE ETAPAS INICIALES:



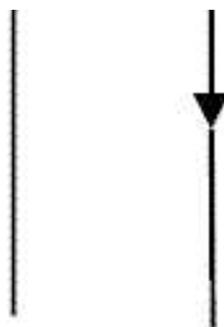
Fuente: https://www.uclm.es/profesorado/rcarcelen_plc/Prog3.htm

Estas etapas son la representación de los estados del sistema, así para poder representar las etapas, son usados cuadrados con un número o un símbolo en su interior que lo identifica. La etapa inicial se simboliza con un cuadrado de doble línea.

Las labores que llevan ligadas las etapas se representan con un rectángulo donde se indica el tipo de acción a realizar. Una etapa puede ser ligada a varias tareas.

Las líneas de evolución

Figura II 30. Líneas de Evolución

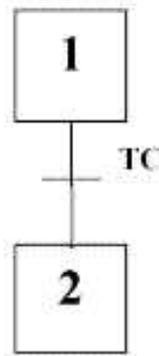


Fuente: https://www.uclm.es/profesorado/rcarcelen_plc/Prog3.htm

Estas líneas unen entre si la secuencia del programa, representando las futuras acciones y decisiones que se tomaran. Las líneas de evolución de evolución serán de arriba hacia abajo, estando dos líneas de evolución unidas no deben interpretarse que están unidas.

Las transiciones

Figura II 31. Transiciones



Fuente: https://www.uclm.es/profesorado/rcarcelen_plc/Prog3.htm

Las transiciones, representan las condiciones necesarias, para que la secuencia avance, en este caso los elementos asociados con esas transiciones son las entradas, que a su vez están conectados a sensores o finales de carrera. Gráficamente se representa como líneas entrecruzadas sobre las líneas de evolución.

2.15.4.2. REGLAS DE EVOLUCIÓN

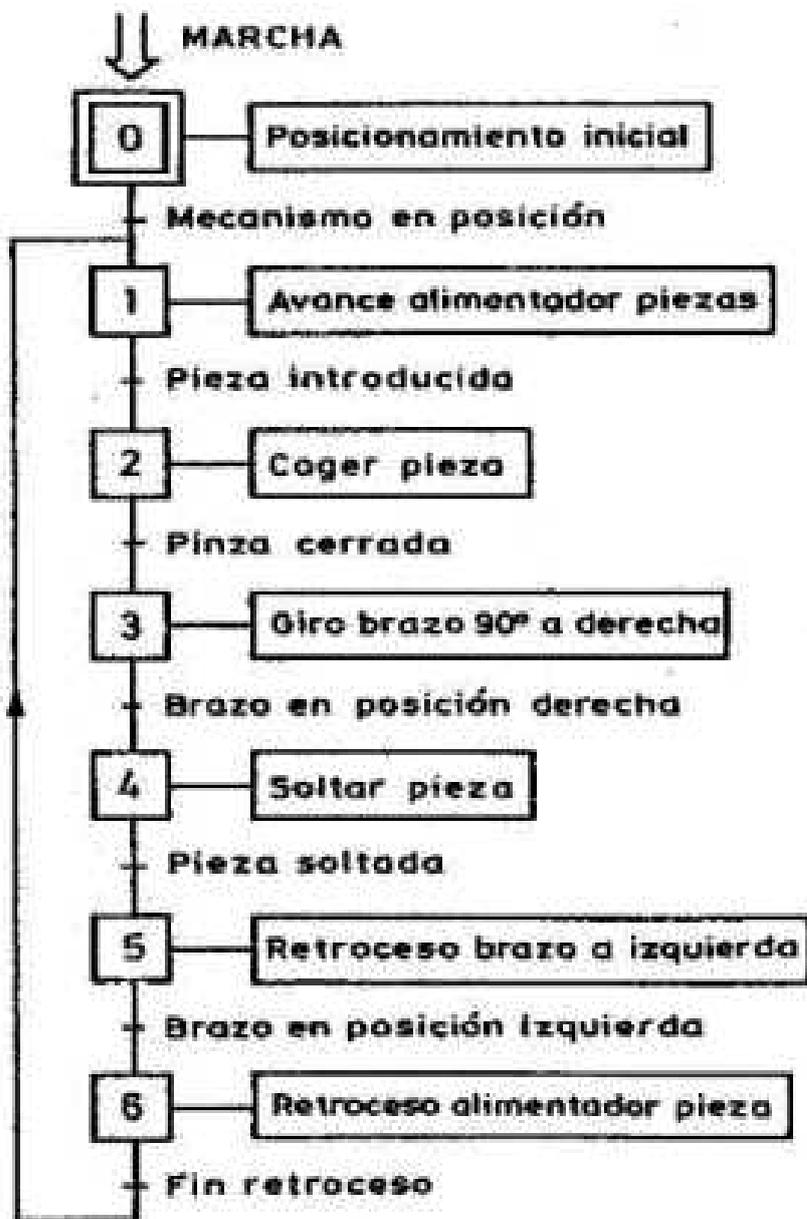
- Las etapas del proceso son activadas de forma secuencial.
- Las acciones se ligan a cada etapa, siendo estas activas solo cuando la etapa este activa.
- La etapa se activara cuando la anterior lo está, y la transición entre las dos haya sido cumplida.
- Al activarse la transición, implica que la etapa precedente debe desactivarse y activarse la etapa siguiente.
- La etapa inicial se activa antes que inicie el ciclo del programa.

2.15.4.3. ESTRUCTURAS DEL GRAFCET

Existen procesos que requieren estructuras más complejas en las que se representan bucles, tomas de decisiones o tareas simultáneas que deben sincronizarse. Para estos casos el GRAFCET dispone de otras estructuras básicas a partir de las cuales pueden generarse los diagramas de dichos procesos.

Secuencia lineal

Figura II 32. Secuencia Lineal

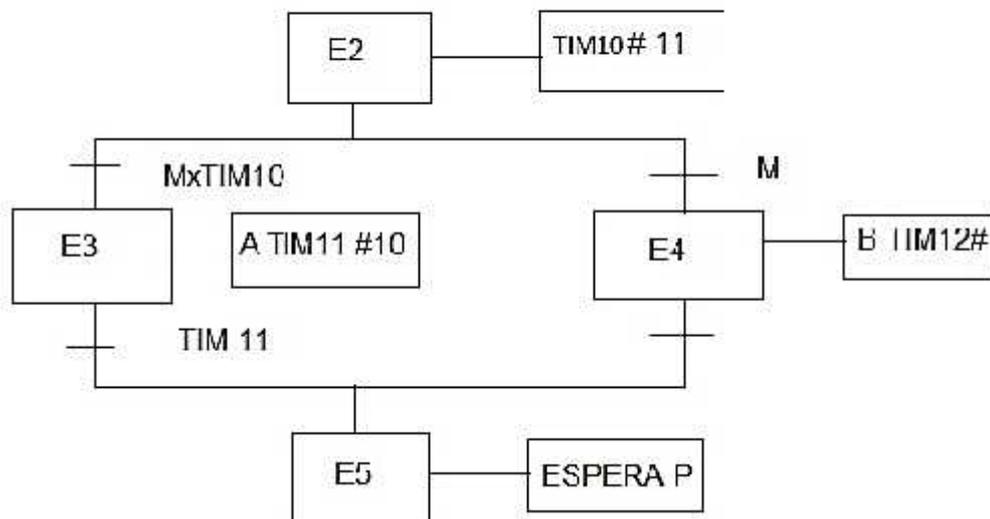


La secuencia lineal, es la estructura más simple posible, ya que consiste en una sucesión de etapas unidas seguidamente por líneas de evolución y condiciones de transición.

Donde solo una etapa determinada esta activa en un instante determinado, teniendo muy en cuenta las reglas de evolución antes establecidas.

Divergencia y convergencia en “o”

Figura II 33. Divergencia y Convergencia en "o"



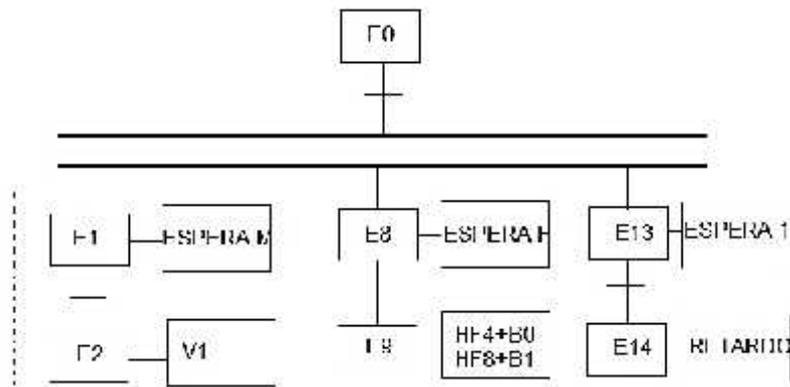
Fuente: https://www.uclm.es/profesorado/rcarcelen_plc/Prog3.htm

La divergencia y convergencia en “o”, o llamadas bifurcaciones, son aquellas en las que las alternativas o condiciones necesarias para cumplir una transición no son únicas y se pueden cumplir de una manera aleatoria.

Tanto la divergencia como la convergencia en “o”, están sujetas a las reglas de evolución antes expuestas.

Divergencia y convergencia en “y”

Figura II 34. Divergencia y Convergencia en "Y"



Fuente: https://www.uclm.es/profesorado/rcarcelen_plc/Prog3.htm

Son las llamadas bifurcaciones en “y”, aquellas en que los procesos y subcaminos se inician simultáneamente, cuando se cumpla determinada condición, como son temporizadores, coneros, entradas, etc.

Esta puede contener una estructura más compleja, anidando un camino dentro de un camino, que tendrá que ser ejecutado de manera simultánea con otros procesos.

Como en la divergencia y convergencia en “o”, está sujeta a las reglas generales de evolución del grafcet. [15]

2.15.5. DESCRIPCIÓN GENERAL DE LAS COMUNICACIONES

Los controladores lógicos programables de la serie TWIDO, poseen además de un puerto serie de comunicaciones, un segundo puerto opcional, que se usa para el monitoreo en tiempo real, como son Ethernet, RS 232 o RS 485.

Los servicios de los controladores en tiempo real, proporcionan funciones para intercambio de información con terminales de entrada/salida, así como funciones para comunicarse con terminales externos, pudiendo estos sistemas controlar y configurar nuestro plc por medio de TWIDO SOFT.

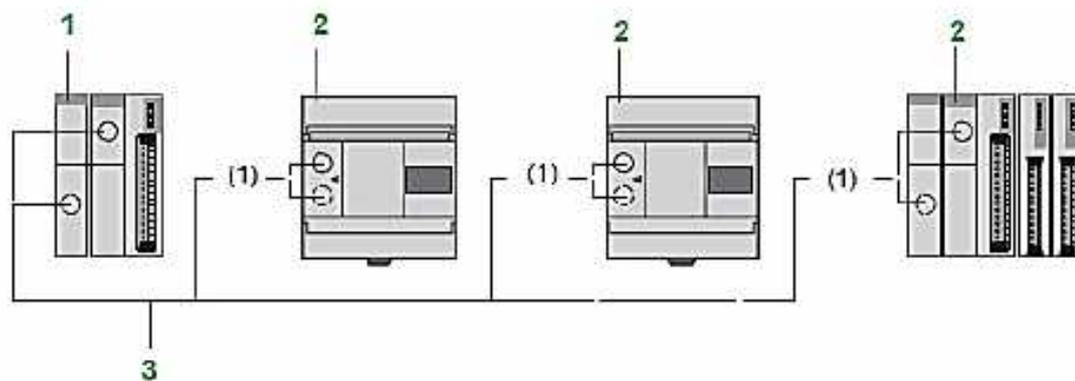
Cada puerto serie se utiliza para estos servicios, siendo configurado primero por el puerto serie, para poder usar sus funcionales opcionales.

Debemos tomar en cuenta tres protocolos explícitos disponibles en cada controlador.

Remote Link (Conexión remota): permite realizar una comunicación entre autómatas Twido vía RS-485, utilizado para ver **E/S a distancia** (sin programa en las CPUs deportadas) o para red de Twidos con programa, con una longitud máxima de **200 m** y hasta **8 equipos** en una red (**maestro + 7 esclavos**).

Estos equipos tiene la ventaja de ser altamente resistentes al ruido y , usados en entornos industriales por su alto nivel de robustez.

Figura II 35. Esquema conexión remota



- 1 Controlador de base compacta o modular denominado "maestro".
- 2 Bases compacta o modular Twido utilizadas en ampliación de E/S o "réflex" local.
- 3 Cable RS 485, 3 hilos, a partir del puerto de enlace serie Integrado o del 2.º puerto de enlace serie opcional.

Fuente: <http://eshop.schneider-electric.com/Download.aspx?infos=H377042.pdf+;3>

ASCII: permite comunicar el autómata, vía **RS-485 y RS-232**, con un gran número de dispositivos: **impresoras** (para la impresión periódica de reportes de producción), lectores de códigos de barras y módems.

Modbus: comunicación Modbus Maestro/Esclavo por ambos puertos (RS485 o 232), permite conectar Twido a un gran número de equipos industriales, basado en mensajería aperiódica entre equipos.

El controlador compacto TWDLCAE40DRF proporciona un puerto de comunicación RJ45 Ethernet integrado. Las comunicaciones **Ethernet** implementan el protocolo **TCP/IP Modbus**.

Además, se pueden implementar más comunicaciones agregando módulos de comunicación, existen varios tipos en función del protocolo que se desee.

- Módulo de comunicación Maestro **ASI**.
- Módulo de comunicación Maestro **CANOpen**.
- Módulo de comunicación Maestro/Esclavo **Modbus**.

Figura II 36. Módulos de comunicación



Fuente: <http://eshop.schneider-electric.com/Download.aspx?infos=H377042.pdf+;3>

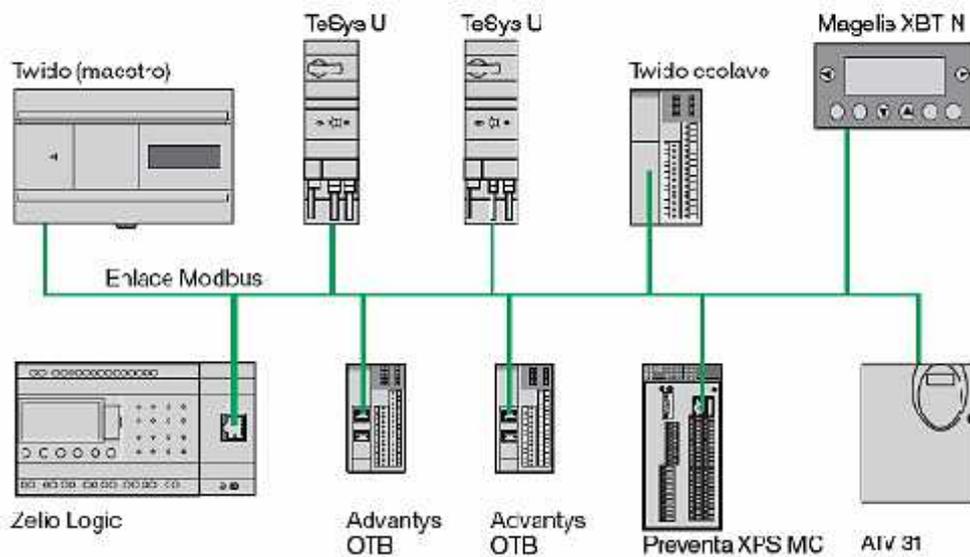
2.15.5.1 BUS DE COMUNICACIÓN MODBUS

El enlace serie Modbus permite responder a las arquitecturas maestro/esclavo (no obstante, es necesario comprobar que los servicios Modbus útiles para la aplicación se implanten en los equipos implicados).

El bus está constituido por una estación maestro y por estaciones esclavo. Sólo la estación maestro puede iniciar el intercambio (la comunicación directa entre estaciones esclavo no es posible). Existen dos mecanismos de intercambio:

- Pregunta/respuesta, las peticiones del maestro se dirigen a un esclavo determinado. El esclavo interrogado espera de vuelta la respuesta.
- Difusión, el maestro difunde un mensaje a todas las estaciones esclavo del bus. Éstas últimas ejecutan la orden sin emitir respuesta.

Figura II 37. Esquema de ejemplos de comunicación Modbus



Fuente: <http://eshop.schneider-electric.com/Download.aspx?infos=H377042.pdf+;3>

Modo maestro de Modbus: el modo maestro de Modbus permite que el controlador pueda iniciar una transmisión de solicitudes Modbus, esperando una respuesta desde un esclavo Modbus.

Modo esclavo Modbus: el modo esclavo Modbus permite que el controlador pueda responder a las solicitudes de Modbus desde un maestro Modbus.

Se trata del modo de comunicación predeterminado si no existe ninguna comunicación configurada.

La comunicación Modbus Maestro/Eslavo se puede realizar por ambos puertos (RS485 o 232). Este protocolo permite conectar Twido a un gran número de equipos industriales, como variadores de velocidad, arrancadores de motor, sensores...etc.

2.15.5.2. BUS DE COMUNICACIÓN ETHERNET

TwidoPort ConneXium aporta conectividad Ethernet a la línea de productos Twido. Es la pasarela entre un único dispositivo Twido Modbus/RTU (RS-485) y la capa física de las redes Modbus/TCP en el modo esclavo.

Este módulo de pasarela sólo admite el modo esclavo.

Al ser un módulo de comunicación Ethernet, tiene sus limitación y ventajas entre los cuales podemos citar los más importantes:

- Comunicación Remota.
- Uso compartido de Información.

Figura II 38. Ejemplo de Arquitectura Ethernet



Fuente: <http://eshop.schneider-electric.com/Download.aspx?infos=H377042.pdf+;3>

Las principales características del módulo TwidoPort son las siguientes:

- Clase **A10** (Sin servidor web, ni I/O scanning).
- **Mensajería** Modbus (lectura/escritura de palabras de datos). Función **BOOTP**.

- Admite la configuración manual utilizando Telnet.
- Interface físico 10BASE-T/100BASE-TX, con conector de tipo RJ45.
- Caudal binario **10/100 Mbits/s**, con reconocimiento de par trenzado. [16]

Los controladores Twido, ofrecen una solución integrada, un controlador con puerto Ethernet integrado como lo es el TWIDO LCAE 40 DRF, con una interfaz amigable con el usuario y de fácil configuración para su correcta integración en la red local LAN.

2.15.6. ACTUADORES NEUMÁTICOS

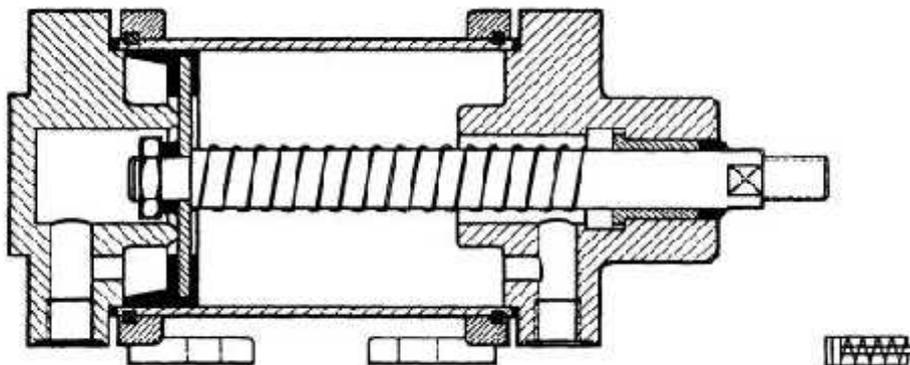
Los actuadores neumáticos son aquellos que usan el aire comprimido, para poder crear movimiento lineal o rotativo, siendo controlado por válvulas neumáticas o electroválvulas activadas con finales de carrera o sensores magnéticos que nos da la posición del actuador lineal referente a un punto.

Cilindros de simple y doble efecto

Los cilindros de simple efecto son aquellos que realizan un esfuerzo en un solo sentido, es decir realizan el trabajo en una sola carrera del ciclo. El retroceso del vástago se realiza al evacuar el aire comprimido en el cilindro, a lo que el vástago regresara a su posición de partida.

En la siguiente figura II 39. Se aprecia un cilindro de simple efecto y sus partes.

Figura II 39. Cilindro de simple efecto



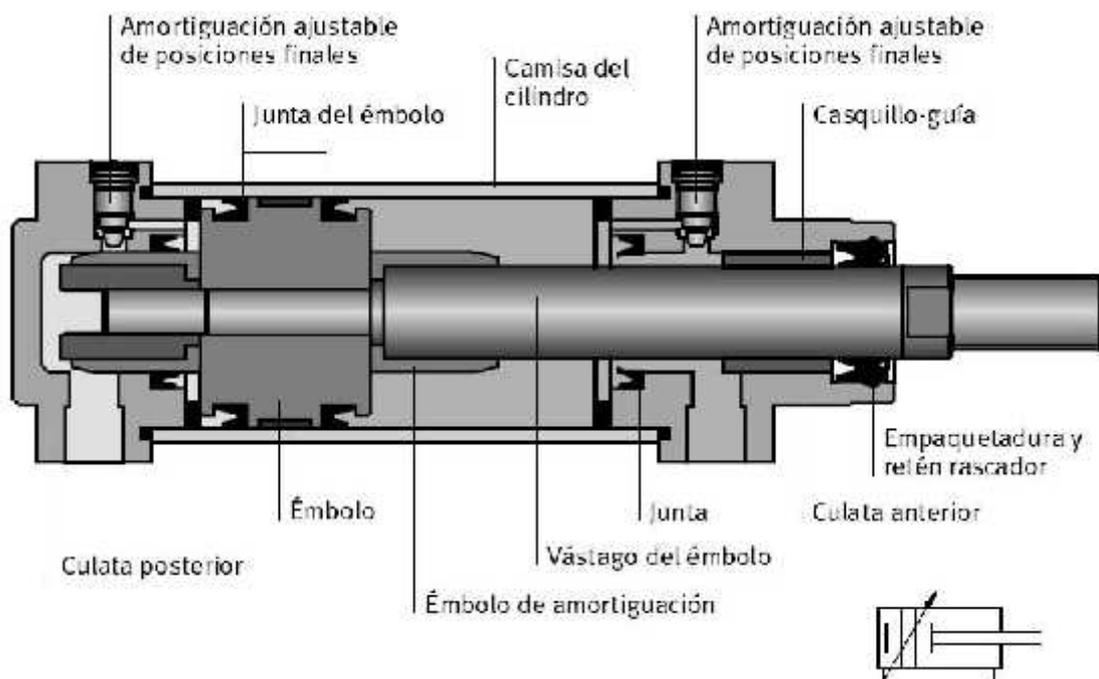
Fuente: <http://www.sapiensman.com/neumatica/images/neumat69.jpg>

Los cilindros de doble efecto pueden realizar un esfuerzo en ambos sentidos, por lo que pueden realizar un trabajo más eficiente tanto en su avance como en su retroceso.

Poseen dos tomas de entrada para el aire comprimido, cada una de ellas situada en las etapas inicial y final del cilindro, siendo la carrera más larga (200 mm), teniendo muy en cuenta el pandeo o curvamiento del vástago en su posición externa.

Cuando el aire comprimido ingresa por la parte inferior del cilindro realiza un movimiento de avance, siendo en el movimiento contrario cuando el aire ingresa por la entrada superior del cilindro, realizando un movimiento de retroceso, estos movimientos son controlados por electroválvula o válvulas que permitirán el paso del aire tanto para el avance como para el retroceso. [17]

Figura II 40. Cilindro de doble efecto



Fuente: http://4.bp.blogspot.com/_qjfmaGawmmQ/THd6LXJpvI/Cs/MWTtpvsV7BU/s1600/DobleEfecto.JPG

Electroválvulas 5/2

Figura II 41. Electroválvula 5/2



Fuente: <http://yusari.co.id/image-product/img314.jpg>

Son dispositivos que reaccionen a dispositivos eléctricos, mediante un solenoide que hace las veces de paso para la entrada o salida de flujos.

Al pasar un flujo de corriente por el solenoide genera un campo magnético que atrae al núcleo móvil, y al terminar el paso del flujo magnético acaba su efecto regresando a su posición normal, en la mayoría de los casos mediante un resorte. [18]

2.15.7. SENSORES MAGNÉTICOS

Figura II 42. Sensor Magnético



Fuente: <http://en2.airtac.com/upload/201107180201530151.JPG>

Los sensores cilíndricos magnéticos se utilizan principalmente para vigilar la posición del pistón en cilindros y manipuladores. El sensor detecta el campo del imán integrado en el pistón a través de la pared del actuador. Gracias a la detección de posición sin

contacto, nuestros sensores cilíndricos magnéticos funcionan de forma fiable y libre de desgaste, no se produce ninguna quemadura por roce, rebote o adhesión y solo hay un punto de actuación. La posición del pistón se detecta de forma fiable también a altas velocidades de desplazamiento. [19].

2.16. MÉTODOS DE NAVEGACIÓN

La navegación permite guiar a un robot móvil a través de un entorno con obstáculos. El objetivo es llevar al robot móvil a su destino en forma segura. Las tareas involucradas en la navegación son: la percepción del entorno a través de sus sensores, la planificación de una trayectoria libre de obstáculos, y el guiado del vehículo a través de la referencia construida. Entonces, la navegación es el proceso de usar los datos de los sensores para producir una representación del entorno. La navegación tiene tres componentes principales: cartografía, planificación, y acción. La cartografía utiliza los datos de los sensores para formar un modelo medioambiental (mapa). El mapa es usado por la etapa de planificación para calcular la dirección del destino evitando los obstáculos de la forma más segura. Cuando la dirección es encontrada, la acción entra en funcionamiento y el robot comienza a moverse hacia el lugar de destino.

2.16.1. CARTOGRAFÍA.

Los mapas son muy importantes para el proceso de navegación, son necesarios para la autonomía del robot móvil. El tipo de información que representa el entorno puede ser utilizada para clasificar los mapas como mapas de: caminos, espacio libre, orientado a objetos, o compuesto.

Los mapas pueden ser de naturaleza local o global. Los mapas locales definen un área finita alrededor del robot móvil. La frontera de un mapa local pueden ser las paredes de un cuarto o un espacio alrededor del robot móvil. Los mapas globales son mucho más amplios. La extensión de un mapa global puede ser varios pisos en una edificación o una extensión entera en un camino con obstáculos.

Los mapas globales son usados para colocar una dirección final de la meta o destinación. Los mapas locales incluyen información detallada, tal como obstáculos inesperados. La navegación reactiva suele contar únicamente con mapas locales.

El planeamiento de la navegación requiere cartografía global, por lo menos a tal grado de memorizar mapas locales.

La incertidumbre y el ruido en los sensores es algo muy importante a tener en cuenta en todos los tipos de mapas. La posición detectada de un obstáculo caerá dentro de un rango de la posición actual debido a la resolución actual del sensor. Esto es especialmente significativo cuando se usan sensores de baja resolución. La cartografía a menudo confía en los datos de los sensores que son relativos a la localización del robot móvil.

La localización del robot móvil es a su vez, relativa a un sistema coordinado global. Otros mapas confían en los datos de múltiples sensores que son alineados, según el movimiento relativo del robot móvil sobre un periodo de tiempo. [9]

Mapas de caminos. Contienen listas de caminos o movimientos basados en programación humana u otros métodos. Los mapas son guardados internamente y usados para guiar al robot móvil con la ayuda de indicadores externos. Los mapas de caminos son usados extensivamente en aplicaciones industriales donde hay poca variación en la trayectoria del robot móvil. Los mapas de caminos son típicamente de poco uso en robots móviles completamente autónomos.

Mapas de espacio libre. Se interesa por los espacios entre obstáculos. Ya que un robot móvil prosigue a través de un entorno desconocido, los datos concernientes a la extensión del espacio libre son recolectados y la localización de las áreas ocupadas, u obstáculos, son grabados en un gráfico espacial.

Mapas orientados a objetos. Se interesa por la localización de los obstáculos en el entorno. Los mapas de espacio libre están determinados por implicación (por ejemplo, áreas sin obstáculos). Los objetos son típicamente grabados como una colección de vértices (x,y) relativos al marco de referencia global. Alternativamente, la posición y

orientación de cada objeto puede estar especificada en una lista enlazada de (x,y) y . Los mapas orientados a objetos pueden producir una muy pequeña colección de datos para un entorno dado. Estos mapas son muy efectivos en situaciones donde el entorno es bien conocido.

Mapas compuestos. Ninguno de los anteriores mapas representa completamente el entorno. Los mapas de espacio libre ignoran los objetos, mientras que los mapas orientados a objetos están solamente interesados en los objetos y en el espacio libre implicado. Los mapas compuestos abarcan los datos concernientes de los objetos y del espacio libre. [9]

2.17. DISEÑO Y FUNCIÓN DEL ROBOT

El objetivo es crear un robot móvil de alta calidad, plenamente funcional con accionamiento omnidireccional. Las tres unidades de accionamiento permiten realizar movimientos en todas direcciones adelante, atrás y lateralmente de manera instantánea. Además, el robot también puede girar sobre un punto. También está equipado con una webcam y varios tipos de sensores analógicos para medición de distancias, sensores digitales para detectar la velocidad real.

El chasis consiste en una plataforma construida en base a un panel de aluminio compuesto. [20].

Figura II 43. Panel de aluminio compuesto

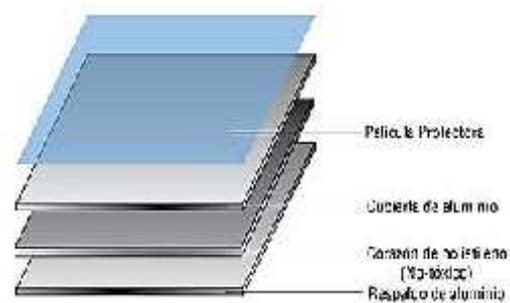


Fuente: http://quito.anunico.ec/panel_de_aluminio_compuesto-1237.html

El panel de aluminio compuesto consta de tres capas, dos láminas de aluminio recubriendo el corazón con material nanométrico como se indica en la figura II 44. Las

dos capas de aluminio son adheridas mediante aditivos que le dan su alta resistencia, apariencia tersa y agradable.

Figura II 44. Estructura aluminio compuesto



Fuente: <http://www.napresa.com.mx/PREFABRICADOSLIGEROS>

En la figura II 45. se muestra la estructura que conforma el chasis del robot, tiene una forma circular de 37 cm de diámetro.

Figura II 45. Estructura chasis



Fuente: Lopez J., Santillan G.

En el chasis se colocara las baterías recargables, las unidades de accionamiento y la cámara se hallan montadas en el chasis, en el que también se hallan situados los sensores de medición de distancia.

El chasis ofrece un espacio adicional y opciones de montaje para otros añadidos, sensores y/o actuadores. Para el caso de transporte de materiales se instalara un dispositivo que simule un elevador de carga en la parte frontal del robot, las baterías se

sujetaran en los costados del robot sobre los soportes de forma rectangular creados con ese fin.

2.17.1. UNIDAD DE ACCIONAMIENTO

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

Cada una de las 3 unidades de accionamiento consta de los siguientes componentes:

- Motor DC
- Caja reductora con una relación de reducción de 131:1
- Rodillos omnidireccionales
- Encoder incremental

Figura II 46. Unidad de accionamiento



Fuente: Lopez J., Santillan G.

El rodillo omnidireccional es puesto en movimiento en una determinada dirección por medio de su eje de accionamiento y también es capaz de desplazarse en cualquier dirección si se ve forzado por otros accionamientos en direcciones diferentes. Como resultado de la interacción con las otras dos unidades de accionamiento, es posible obtener un recorrido en una dirección que difiere de la dirección de cada uno de los respectivos accionamientos. [20]

2.17.2. LA UNIDAD DE CONTROL

La unidad de control puede configurarse de forma flexible utilizando el cable usb tipo B estándar que accede directamente al Arduino Mega 2560 permitiendo cargar y modificar el código para controlar los motores, recibir la información obtenida de los sensores digitales y análogos.

La unidad de control está compuesta por:

- El arduino mega 2560
- Placa puente H 298L
- Punto de acceso inalámbrico
- Sensores digitales y análogos.

2.17.3. FUENTE DE ALIMENTACIÓN

La alimentación eléctrica es suministrada por dos baterías recargables de 12 VCD con una capacidad de 5 Ah. Ambas baterías recargables están montadas en el chasis serán las encargadas de suministrar energía a los motores.

Figura II 47. Baterías 12 voltios



Fuente: López J, Santillán G

Además de una batería más pequeña de 6 VCD para suministrar energía a las partes electrónicas del robot como el Arduino Mega, puentes H, sensores, motores paso a paso.

2.17.4. SENSORES

En el Robot se ha integrado sensores para la medición de distancias a objetos y para detectar la velocidad del motor. Está equipado con nueve sensores de medición de distancia por infrarrojos, que se hallan montados en el chasis formando un ángulo de 40° entre sí. Con estos sensores, puede detectar objetos en las zonas circundantes.

Con ello 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. La conexión del sensor es especialmente sencilla e incluye tan sólo una señal de salida analógica y la alimentación. La electrónica de evaluación del sensor determina la distancia, que puede leerse como una señal analógica [20]

Sensor de proximidad inductivo

El sensor inductivo de proximidad se suministra como un componente adicional. Sirve para detectar objetos metálicos en las estaciones y se utiliza para el control de trayectorias. Lee señales de diferente intensidad dependiendo de si se halla el palet sobre una estación es decir la presencia o ausencia de un objeto. Con ello puede controlarse el recorrido de forma perfectamente diferenciada.

Tabla II II. Características técnicas sensor inductivo IME12

| Datos técnicos | |
|---------------------------|----------------|
| Tensión de funcionamiento | 15–30 V DC |
| Tensión de salida | 0–10 V |
| Tipo | 5 EA-M12B-UI-S |
| Número de artículo | 538292 |
| Diámetro | M12 |
| Margen de detección | 0 a 6 mm |
| Montaje | Casi enrasable |
| Frecuencia de conmutación | 1'000 Hz |
| Temperatura ambiente | -25 a +70° C |
| Protección | IP 67 |
| Material de cuerpo | Latón cromado |
| Par de apriete máximo | 10 Nm |
| Repetibilidad | 0,01 mm |

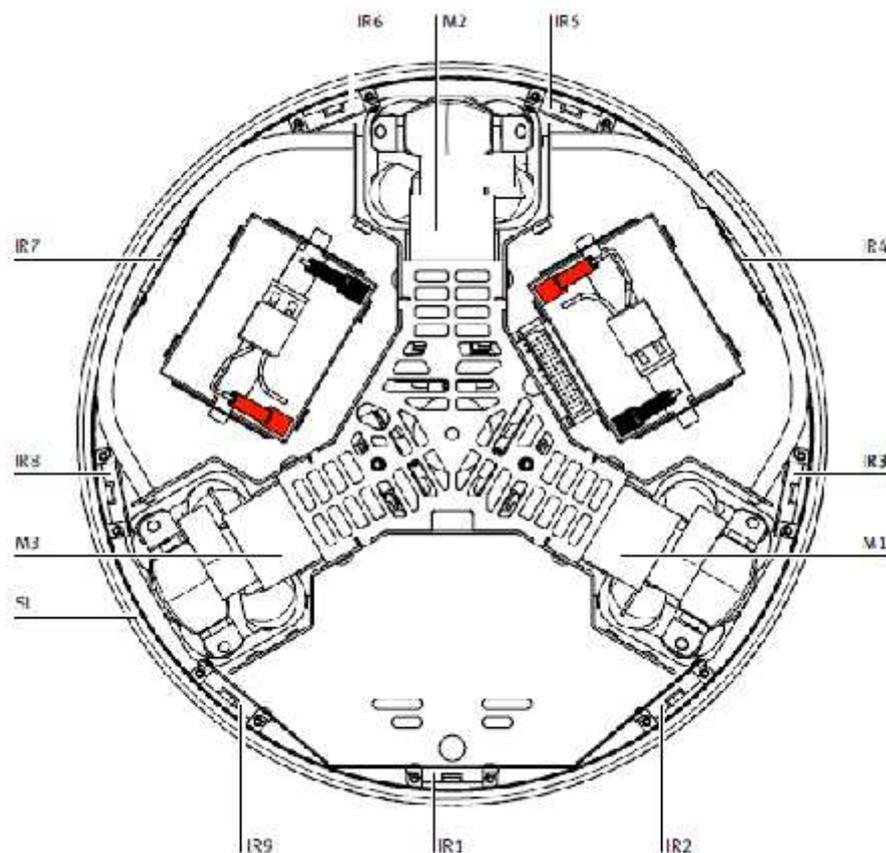
Fuente: www.Sick.com

Posiciones de los sensores y sus denominaciones

Los sensores de medición de distancia y el encoder incremental deben ser identificados inequívocamente para poder ser capaces de direccionarlos en la programación de forma objetiva. Las respectivas denominaciones se muestran en la figura inferior. IR1 es direccionado como “Distancia 1” en el software. De forma similar, los otros sensores de infrarrojos se denominan “Distancia 2” hasta “Distancia 9”. [20]

Los encoders incrementales están asignados a sus respectivas unidades de accionamiento.

Figura II 48. Ubicación de los sensores infrarrojos



Fuente: Festo didactic

2.17.5. PUNTO DE ACCESO INALÁMBRICO BLUETOOTH

El punto de acceso inalámbrico bluetooth es un componente que permite la comunicación con el robot móvil por medio un enlace por radiofrecuencia en la banda de los 2,4 GHz. Se caracteriza por:

- Facilitar las comunicaciones entre equipos móviles.
- Eliminar los cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.
- La especificación de Bluetooth define un canal de comunicación a un máximo 720 kbit/s.
- Un rango óptimo de 10 m .
- Opera en la frecuencia de radio de 2,4 a 2,48 GHz con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex.

2.18. PLANIFICACIÓN DE LA TRAYECTORIA

La idea general en la planificación es encontrar una trayectoria segura capaz de conducir al robot móvil desde un punto de partida hasta un punto de llegada.

El concepto de ruta segura implica el cálculo de un camino al menos continuo en posición, que sea libre de obstáculos. En virtud de esta ruta, el generador construirá las referencias que se le entregan al control de movimientos. Por ello, en la planificación de esta ruta se obvian las características cinemáticas y dinámicas del vehículo, ya que el cómputo de una referencia adecuada que cumpla con estos atributos es tarea del generador de caminos.

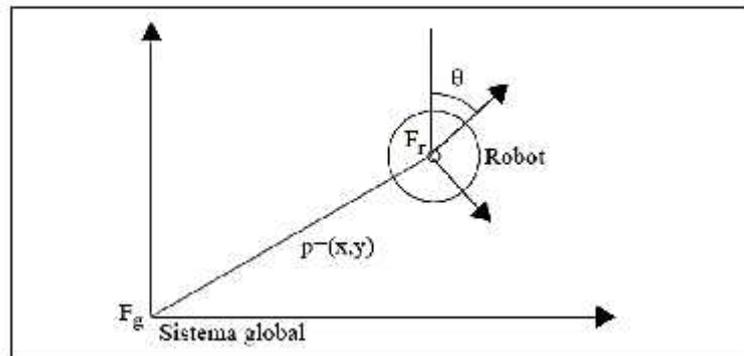
Por tanto, la ruta al tan sólo asegurar continuidad en posición, supone que únicamente los robots móviles omnidireccionales puedan seguir una referencia de tales características. [21]

El entorno en donde el robot realizará su tarea se lo puede considerar como un subconjunto de configuraciones en las cuales en cualquier instante de tiempo puede encontrarse el robot, igualmente se tendría un subconjunto inalcanzable que llegan a ser los obstáculos.

Se define una configuración \mathbf{q} de un robot como un vector cuyas componentes proporcionan información completa sobre el estado actual del mismo. Un robot es un

objeto rígido al cual se le puede asociar un sistema de coordenadas móvil. La localización del vehículo en un determinado instante de tiempo queda definida por la relación existente entre el sistema de coordenadas global F_g en virtud del cual está definido todo el entorno de trabajo y su sistema de coordenadas locales asociado F_r . [21]

Figura II 49. Sistema de coordenadas global, local asociado al robot



Fuente: Autor

La expresión que proporciona el estado actual del robot se define como:

$$q = (p, \theta) = (x, y, \theta)$$

Donde p es la posición y θ la orientación.

2.19. GENERACIÓN DE CAMINOS

El camino se lo construye en base a la planificación de la ruta y debe estar libre de obstáculos, la importancia de un camino con buenas propiedades se basa en la capacidad del seguidor de caminos para ejecutar la navegación con el menor error posible. La función del generador es convertir una ruta en un camino, llevar al robot de una posición inicial a una final tal que se elimine la restricción de omnidireccionalidad inherente a la definición de ruta.

El camino se define como la discretización de una curva continua que interpola ciertos puntos elegidos de la ruta calculada por el planificador. Por tanto, el problema de la definición de un camino con buenas propiedades pasa por la construcción de la función camino adecuado que las posea.

CAPÍTULO III

3. IMPLMETACIÓN DEL ROBOT

3.1. SENSORES UTILIZADOS

Sensores de infrarrojo para medir distancias.

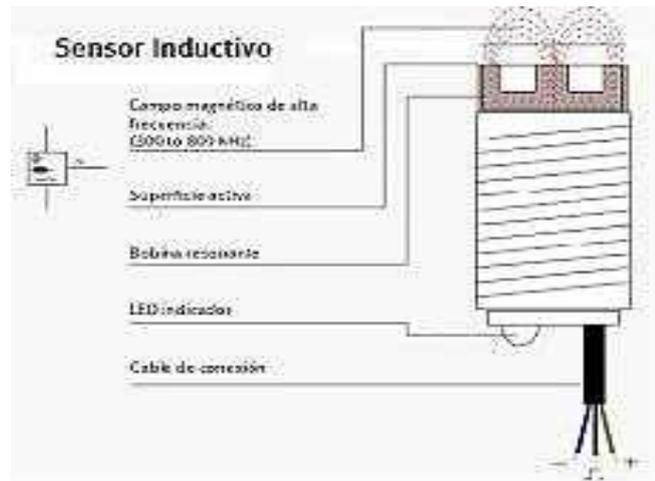
Utilizando estos sensores se pueden estimar las distancias a las que se encuentran los objetos en el entorno. Este tipo de sensores poseen una gran precisión, debido a que son muy direccionales al ser muy pequeña su longitud de onda. La distancia de medida depende de la potencia que se aplica al rayo de salida del infrarrojo y de su curva de funcionamiento a la cual fueron diseñados.

Sensores inductivos.

Son sensores basados en un cambio de inductancia debido a la presencia de un objeto metálico, están entre los sensores de proximidad industriales de uso más frecuente. La figura III 50 se muestra un sensor inductivo, que consiste en una bobina enrollada, situada junto a un imán permanente empaquetado en un receptáculo simple y robusto.

Hay una relación entre la amplitud de la tensión y la distancia sensor-objeto. La sensibilidad cae rápidamente al aumentar la distancia, el sensor solo es eficaz para fracciones de un milímetro.

Figura III 50. Sensor Inductivo



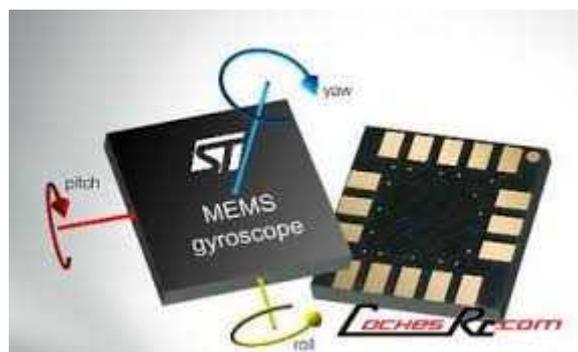
Fuente: sensoresdeproximidad.blogspot.com

Debido a que el sensor requiere movimiento para generar una forma de onda de salida, un método para producir una señal binaria es integrar esta forma de onda. La salida binaria se mantiene a nivel bajo en tanto que el valor integral permanezca por debajo de un umbral especificado, y luego se conmuta a nivel alto (indicando la proximidad de un objeto) cuando se supera el umbral.

Giroscopio.

Son brújulas de medida incremental; es decir, miden cambios en la orientación del robot. Se basan en medir la aceleración usando las leyes de Newton. En la figura III 51 se muestra un giroscopio electrónico.

Figura III 51. Giroscopio electrónico



Fuente: ericaparrasmartphones.blogspot.com

Los giroscopios tienen varias limitaciones:

- Acumulan el error con el paso del tiempo.
- Si se quieren tener precisiones aceptables (de décima de grado por segundo) son muy caros.
- Los giroscopios baratos tienen errores mayores de un grado por minuto.

Existen unos giroscopios ópticos basados en láser, que permiten medir con mucha precisión la orientación del robot (errores que van desde varios grados por hora, hasta 0.0001 grado por hora, según el precio).

Los encoders

Los codificadores ópticos o encoders incrementales, constan en su forma más simple de un disco transparente con una serie de marcas opacas colocadas radialmente y equidistantes entre sí, de un sistema de iluminación en el que la luz es colimada de forma adecuada, y de un elemento fotorreceptor. El eje cuya posición se quiere medir va acoplado al disco transparente.

Con esta disposición, a medida que el eje gire se irán generando pulsos en el receptor cada vez que la luz atraviese cada marca, y llevando una cuenta de estos pulsos es posible conocer la posición del eje y la velocidad de rotación.

Por otro lado las principales limitaciones con las que cuenta son:

Siempre es necesario un circuito contador para obtener una salida digital compatible con el puerto de entrada/salida de un microcontrolador. Otra posible forma de hacerlo se basaría en software especial según sea la aplicación específica, como por ejemplo, alguna interrupción o programación de alta velocidad, tiempo real, para obtener el tiempo de cambio entre un sector y otro.

Se pueden presentar problemas mecánicos debido a la gran precisión que se debe tener en su fabricación. Son dispositivos particularmente sensibles a golpes y vibraciones, estando su margen de temperatura de trabajo limitado por la presencia de componentes electrónicos.

Para incrementar la resolución del encoder y saber el sentido de giro se dispone de otra señal pero desplazada de la existente, de manera que el tren de pulsos B que se genere esté desplazado 90° eléctricos con respecto al generado por la primera franja, A.

De esta manera, con un circuito relativamente sencillo, es posible obtener una señal adicional que indique cual es el sentido de giro y que actúe sobre el contador correspondiente indicándole que incremente o reduzca la cuenta que se esté realizando. Es necesario además disponer de una marca de referencia Z sobre el disco que indique que se ha dado una vuelta completa y que, por tanto, se ha de empezar la cuenta de nuevo. Esta marca sirve también para poder comenzar a contar tras recuperarse de una caída de tensión.

La resolución de estos sensores depende directamente del número de marcas que se pueden poner físicamente en el disco. Un método sencillo para aumentar esta resolución es contabilizar los flancos de subida de los trenes de pulsos y contabilizar también los flancos de bajada, incrementando así la resolución pudiéndose llegar a 6400 pulsos; como se presenta en los encoders que se utilizan en este estudio, si se utiliza con la ayuda de circuitos adicionales, estos sensores pueden alcanzar hasta 100.000 pulsos por vuelta.

El modelo de encoders más utilizado son los ópticos. Estos constan de diferentes sectores que pueden ser opacos y transparentes, reflejantes y no reflejantes. Otro tipo también muy usado es el magnético, el cual está equipado con un sistema de detección magnética sin contacto.

La variación de campo magnético provocada por los dientes de una rueda de medida produce una onda sinusoidal de tensión. La electrónica del encoder transforma esta señal senoidal en una señal de onda cuadrada, triangular, etc. A pesar de no ser los más usados presentan ventajas sobre los ópticos:

- Altamente resistentes al polvo, los golpes o las vibraciones.
- Funcionamiento en un amplio rango de temperaturas: -20°C a 85°C.

El funcionamiento para cualquiera de estos casos es el mismo. Se implementa mediante un disco delgado metálico con una serie de ranuras equidistantes y homogéneas que se

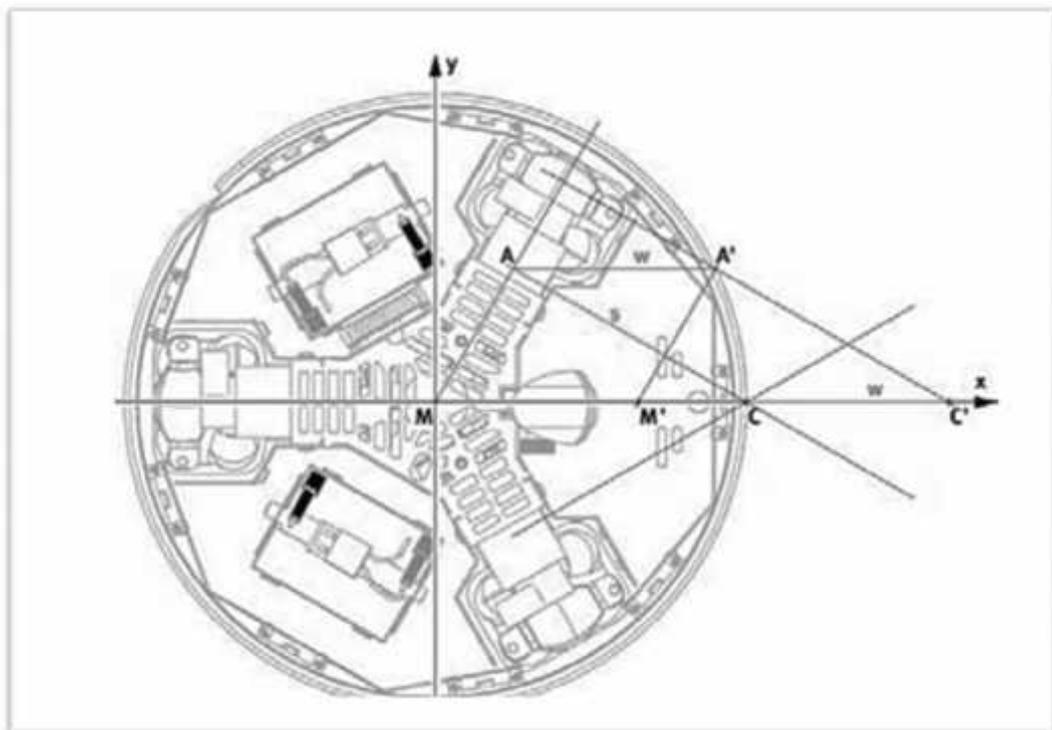
acopla directamente al eje del motor. Este gira dentro de un switch óptico, que deja pasar e interrumpe un haz de luz infrarroja. Este haz de luz está suministrado por el led infrarrojo de la cabeza lectora fija.

3.2. MOVIMIENTOS LINEALES Y POSICIONAMIENTO DE UN SISTEMA ROBÓTICO

El robot posee un sistema accionado por 3 ejes distribuidos a 360 grados equitativamente.

Este tipo de distribución da la ventaja de tener grados de libertad independientes en nuestro sistema robótico, pero se presenta otro problema a resolver, que será: determinar la distancia recorrida por la unidad, en relación al número de giros que den los motores. Considerando que el robot ejecuta un movimiento hacia delante y recorre un tramo “W” y sus ruedas recorren un tramo “S”

Figura III 52. Distancia real recorrida del robot



Fuente: www.festo.com

En la figura III 52, se analiza geoméricamente el comportamiento de esta distribución.

Analizando la imagen se obtiene:

$$S=W * \sin (60^\circ)$$

Para medir la distancia recorrida por cada rueda se toma en cuenta el diámetro de la rueda y el número de pulsos que entrega por vuelta cada encoder.

La distancia que recorre una llanta por cada revolución es igual al perímetro de la misma para lo cual utilizamos la siguiente formula:

$$DISTANCIA1vuelta = \pi * \phi rueda$$

La resolución nos indica la distancia que se recorre por cada pulso del encoder; se calcula mediante la siguiente formula:

$$Resolución\ encoder = \frac{DISTANCIA1vuelta}{\# pulsos vuelta}$$

La fórmula utilizada en el programa desarrollado en arduino para medir la distancia total recorrida por cada llanta es la siguiente:

$$D.rueda = \frac{DISTANCIA1vuelta}{\# pulsos vuelta} * (\#pulsos recorridos)$$

Tabla II III. Características físicas rodillo omnidireccional

| Rodillo omnidireccional, accionado (ARG 80) | |
|---|-------|
| Diámetro Ø | 80 mm |
| Máxima capacidad de carga | 40 kg |

Fuente: www.festo.com

Figura III 53. Rodillo omnidireccional



Fuente: <https://encryptedtbn2.gstatic.com/images?q=tbn:ANd9Gc>

Descripción

- Posibilidad de transporte en cualquier dirección
- Montaje en ejes normales o hexagonales
- Aplicaciones accionadas y no accionadas
- Omniwheels acoplables entre sí
- Estructura libre de corrosión: De poliamida y acero inoxidable
- Diámetro de rodillo 48 y 80 mm
- Capacidad de carga máx.: 50 N (48 mm); 250 N (80 mm)

Para el movimiento de cargas rígidas ligeras o medias

- 6 semirrodillos de poliamida constituyen 1 rodillo principal.

Tabla III IV. Dimensiones rodillo omnidireccional

| técnico | Datos |
|---------------|-------------------|
| Material | Acero y poliamida |
| Fuerza | 50 kg |
| Rodillo Ø | 80 mm |
| Duje Ø | 12.2 mm |
| Anchura total | 61 mm |

Fuente: Autor

Como el diámetro de las llantas omnidireccionales es 8 cm entonces:

$$DISTANCIA1vuelta = \pi * (8 \text{ cm})$$

$$DISTANCIA1vuelta = 25.13 \text{ cm}$$

Obtenemos que en una vuelta del rodillo omnidireccional se desplaza una distancia de 25.13 cm; para calcular la resolución tenemos como datos que el 64 CPR Encoder permite obtener 64 conteos por revolución del eje del motor contando tanto los flancos ascendentes y descendentes de tanto las salidas A y B. El uso de un solo borde da 16 pulsos por revolución del eje del motor.

Para el caso se utiliza los dos canales A y B del encoder dando un conteo de 4200 veces por revolución del eje del motor.

$$\text{Resolución encoder} = \frac{25.13 \text{ cm}}{4200 \text{ pulsos}}$$

$$\text{Resolución encoder} = 0.00598 \text{ cm/pulso}$$

Se obtiene como constante la resolución del encoder; dato que servirá para calcular la distancia recorrida por cada llanta dentro de la programación del arduino. Mediante la siguiente formula:

$$D_r = (0.0 \frac{c}{p}) (\#P_r)$$

Como se trata de un robot omnidireccional la distancia real de desplazamiento se calcula utilizando la fórmula: $S=W * \sin (60^\circ)$

- Donde S es la Distancia recorrida por la rueda (D.rueda).
- W es el desplazamiento del robot

Por lo tanto reemplazando se obtiene:

$$W = \frac{S}{\sin(60^\circ)}$$

$$W = \frac{(0.00598 \frac{cm}{pulsos}) * (\#pulsos recorridos)}{\sin(60^\circ)}$$

$$W = (0.0069 \frac{cm}{pulsos}) * (\#pulsos recorridos)$$

3.3. COMPONENTES DEL ROBOT MOVIL PALETIZADOR

Al diseñar el robot móvil paletizador, se tomó en consideración la velocidad, el torque requerido, el material necesario para poder realizar un correcto diseño, y así evitar posibles fallas en su funcionamiento, tanto en su parte física como en su parte estructural, al mismo tiempo especificando los parámetros de diseño y dimensiones, utilizados para la ejecución del proyecto.

3.3.1. GEARMOTOR 131:1 METAL 37DX57L MM WITH ENCODER

Figura III 54. Gearmotor 37Dx57L



Fuente: <http://www.pololu.com/product/1447>

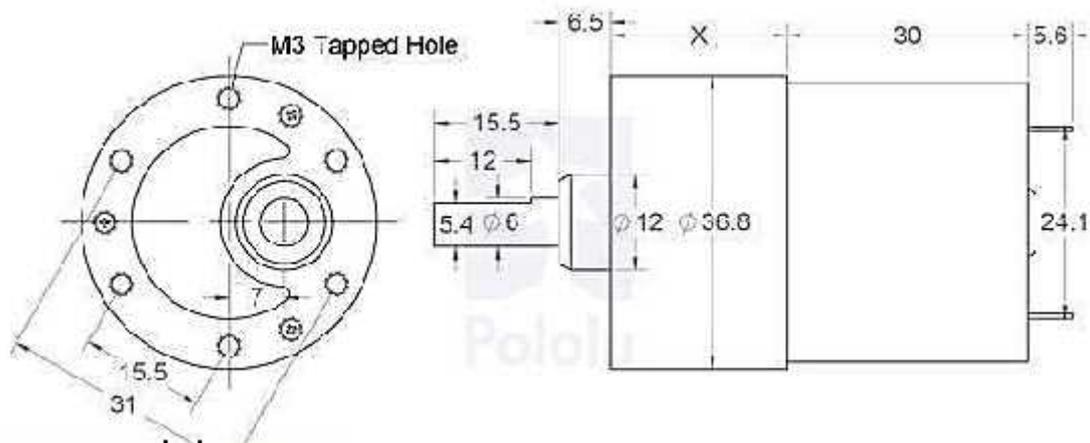
Este 2,71" x 1,45" x 1,45" motorreductor es un potente motor de corriente continua de 12V cepillado con un 131.25:1 caja de engranajes de metal y un codificador de cuadratura integrada que proporciona una resolución de 64 pulsos por revolución del eje del motor, que corresponde a 8400 conteos por revolución del eje de salida de la caja de cambios.

Estas unidades tienen un eje de salida de 0,61", 6 mm de diámetro -long en forma de D-. Características clave:

12 VCD : 80 RPM y 300 mA libre de gestión,

Torque: 220 oz-in (16 kg-cm).

Figura III 55. Dimensiones motorreductores (mm) de metal 37D



Fuente: <http://www.pololu.com/product/1447>

El siguiente diagrama de la figura III 55. Se muestra las dimensiones (en mm) de la línea 37D mm de motorreductores. El valor de X es 26,5 mm para el 100: 1 37Dx57L mm. El conjunto del encoder se extiende un adicional de 12,5 mm más allá de la parte trasera del motor.

3.3.2 ENCODER

Figura III 56. CPR encoder 64



Fuente: <http://www.pololu.com/product/1447>

Una de dos canales de efecto

Hall codificador se utiliza para detectar la rotación de un disco magnético en una protuberancia posterior del eje del motor. El codificador de cuadratura proporciona una resolución de 64 pulsos por revolución del eje del motor cuando se cuentan ambos bordes de ambos canales.

Para calcular los conteos por revolución de la salida de la caja de cambios, multiplicar la relación de transmisión por 64. El motor/codificador tiene seis pines de conexión codificadas por colores: (28 cm) conduce terminada por un conector hembra 1 × 6 con un 0,1" de tono, como se muestra en la FIGURA III 56.

Esta cabecera trabaja con estándar de 0,1 "conectores macho y nuestro macho puente y alambres precrimped. Si esta cabecera no es conveniente para su aplicación, se puede tirar de los hilos metálicos ondulados de la cabecera o cortar la cabecera fuera.

En la siguiente tabla se describen las funciones de alambre:

Tabla III V. Conexión de cables

| Color | Función |
|----------|--|
| Red | Potencia del motor (se conecta a un terminal de motor) |
| Negro | Potencia del motor (se conecta a la otra terminal del motor) |
| Verde | GND codificador |
| Azul | codificador Vcc (3,5 a 20 V) |
| Amarillo | salida del encoder A |
| Blanco | salida del codificador B |

Fuente: <http://www.pololu.com/product/1447>

Por tanto contando los flancos ascendentes y descendentes de tanto las salidas A y B, es posible obtener 64 conteos por revolución del eje del motor. Uso de un solo borde de uno Resultados de canal en 16 conteos por revolución del eje del motor, por lo que la frecuencia de la salida A es 16 veces la frecuencia de rotación del motor.

Si se utilizan los flancos de subida y de bajada de ambos canales se obtiene un conteo de 8400 pulsos por revolución, si se utiliza solo los flancos de subida o de bajada los conteos se reducen a 4200 pulsos por revolución.

3.3.3 ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel Processing. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino.

3.3.3.1 ARDUINO MEGA 2560

El Arduino Mega 2560 es una placa electrónica basada en el Atmega2560; se muestra en la figura III 57.

Figura III 57. Arduino Mega 2560 R3



Fuente: <https://paruro.pe/productos/tarjetas-de-desarrollo/tarjetas-de-microcontroladores/arduino-mega-2560-rev-3>.

Tabla III VI. Datos Técnicos

| | |
|---|---|
| Microcontroladores | Atmega2560 |
| Tensión de funcionamiento | 5V |
| Voltaje de entrada (recomendado) | 7-12V |
| Voltaje de entrada (límites) | 6-20V |
| Digital pines I / O | 54 (de los cuales 15 proporcionan salida PWM) |
| Pines de entrada analógica | 16 |
| Corriente DC por Pin I / O | 40 Ma |
| Corriente DC de 3.3V Pin | 50 mA |
| Memoria Flash | 256 KB de los cuales 8 KB utilizadas por bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Velocidad de reloj | 16 MHz |

Fuente: <http://arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardMega>

Este dispositivo o microcontrolador posee 54 pines digitales que pueden ser de entrada / salida; 15 se pueden utilizar como salidas PWM, 16 entradas analógicas, 4 UARTs (puertas seriales), un oscilador de cristal de 16 MHz, conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio.

Contiene todo lo necesario para apoyar el microcontrolador; basta con conectar el dispositivo a un ordenador con un cable USB o mediante un cable de poder con un adaptador de corriente continua o a una batería DC para energizar.

Entrada y salida

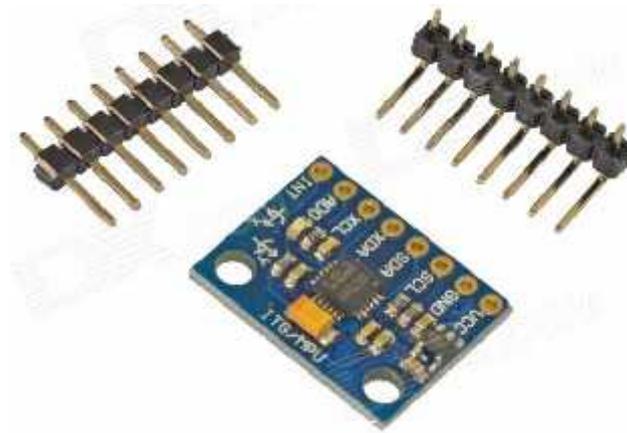
Cada uno de los 54 pines digitales en el Mega se puede utilizar como una entrada o salida, utilizando `pinMode ()`, `digitalWrite ()`, y `digitalRead ()` funciones. Operan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectada por defecto) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas:

- **Serial: pin 0 (RX) y pin 1 (TX); Serie 1: pin 19 (RX) y pin 18 (TX); Serie 2: pin 17 (RX) y pin 16 (TX); Serie 3: pin 15 (RX) y pin 14 (TX).** Se utiliza para recibir (RX) y transmitir datos en serie (TX) TTL.
- **Interrupciones externas:** contiene 6 interrupciones externas: pin 2 (interrupción 0), pin 3 (interrupción 1), pin 18 (interrupción 5), pin 19 (interrupción 4), pin 20 (interrupción 3), y pin 21 (interrupción 2) de interrupción. Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor.
- **PWM:** 2 a 13 y 44 a 46 para salidas PWM de 8 bits con el `analogWrite`
- **TWI:** 20 (SDA) y 21 (SCL). La comunicación usando la librería `Wire`.

El Mega 2560 tiene 16 entradas analógicas, cada una de las cuales proporcionan 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto se miden desde el suelo a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el pin AREF y función `analogReference ()`.

3.3.4. GIROSCOPIO GY-521 MPU6050 MODULO 6DOF

Figura III 58. GY-521

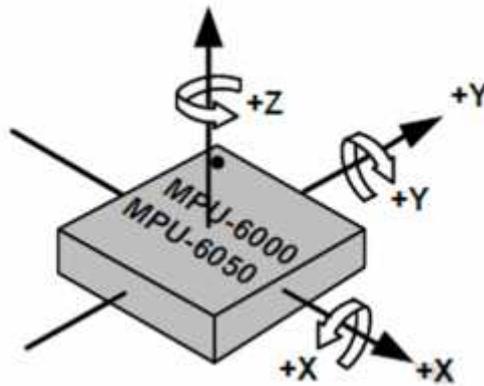


Fuente: <http://www.dx.com/es/p/gy-521-mpu6050-3-axis-acceleration-gyroscope-6dof-module-blue-154602>

- Modelo: GY-521
- Material: Plástico + PCB de cobre
- Chip: MPU-6050
- Fuente de alimentación: 3 ~ 5 V
- Modo de comunicación: estándar I2C protocolo de comunicación
- Chip integrado en el convertidor AD de 16 bits, 16 bits de datos de salida
- Giroscopios rango: + / - 250 500 1000 2000 grados / segundo
- Aceleración rango: + / - 2 g, + / - 4 g, + / - 8 g, + / - 16 g
- Pin paso: 2,54 mm
- Dimensiones: (2,1 cm x 1,6 cm x 0,3 cm)
- Peso: 0,18 oz (5 g)

El InvenSense sensor MPU-6050 contiene un acelerómetro MEMS y un giroscopio MEMS en un solo chip. Es muy preciso, ya que contiene 16 bits de analógico a hardware de conversión digital para cada canal. Para ello capta la x, y, z y el canal al mismo tiempo. El sensor utiliza el I2C parada de microbus para interactuar con el Arduino.

Figura III 59. Orientación de ejes de sensibilidad y polaridad de rotación.



Fuente: deerme.org/electronica/probando-el-sensor-gy-521-mpu6050

El MPU-6050 no es caro, especialmente teniendo en cuenta el hecho de que combina tanto un acelerómetro y un giroscopio.

El MPU-6050 combina un giroscopio de 3-ejes y un acelerómetro de 3-ejes en el mismo chip. Asimismo incorpora un procesador de movimiento capaz de procesar algoritmos MotionFusion de movimiento en 9 ejes. Se comunica a través de la interfaz I2C y posee una librería muy difundida para su utilización inmediata.

Este sensor puede entregar 6 grados de libertad (DOF) e incorpora un regulador de tensión a 3.3V y resistencias pull-up para su uso directo por I2C.

3.3.5. INFRARED DISTANCE SHARP GP2Y0A41SK (4-30cm)

Figura III 60. Sensor (GP2Y0A41SK0F)



Fuente: <http://www.robotistan.com/Sharp-GP2Y0A41SK-4-30cm.html>

GP2Y0A41SK0F es una unidad de sensor de medición de distancia, compuesto por una combinación integrada de PSD (posición detector sensible), IR-LED (infrarrojo diodo emisor de luz) y el circuito de procesamiento de señales. La variedad de la reflectividad del objeto, la temperatura ambiental y la duración de funcionamiento no son influenciados fácilmente a la detección de distancia debido a la adopción del método de triangulación.

Este dispositivo da salida a la tensión correspondiente a la distancia de detección. Así que este sensor también puede ser utilizado como un sensor de proximidad.

Características

1. Sensor de medición de distancia se une con PSD, LED infrarrojo y un circuito de procesamiento de señales
2. Ciclo de medición a corto plazo (16.5ms)
3. Distancia rango de medición: de 4 a 30 cm
4. Tamaño del paquete (29,5 × 13,0 × 13,5 mm)
5. Tipo de salida analógica

En resumen es un sensor de proximidad de infrarrojos fabricado por Sharp. Tiene una salida analógica que varía de 3.1V a 4 cm a 0,3 V a 30 cm, con una tensión de alimentación entre 4,5 y 5.5VDC. El sensor tiene una soldadura Terminal (JST) Conector japonés.

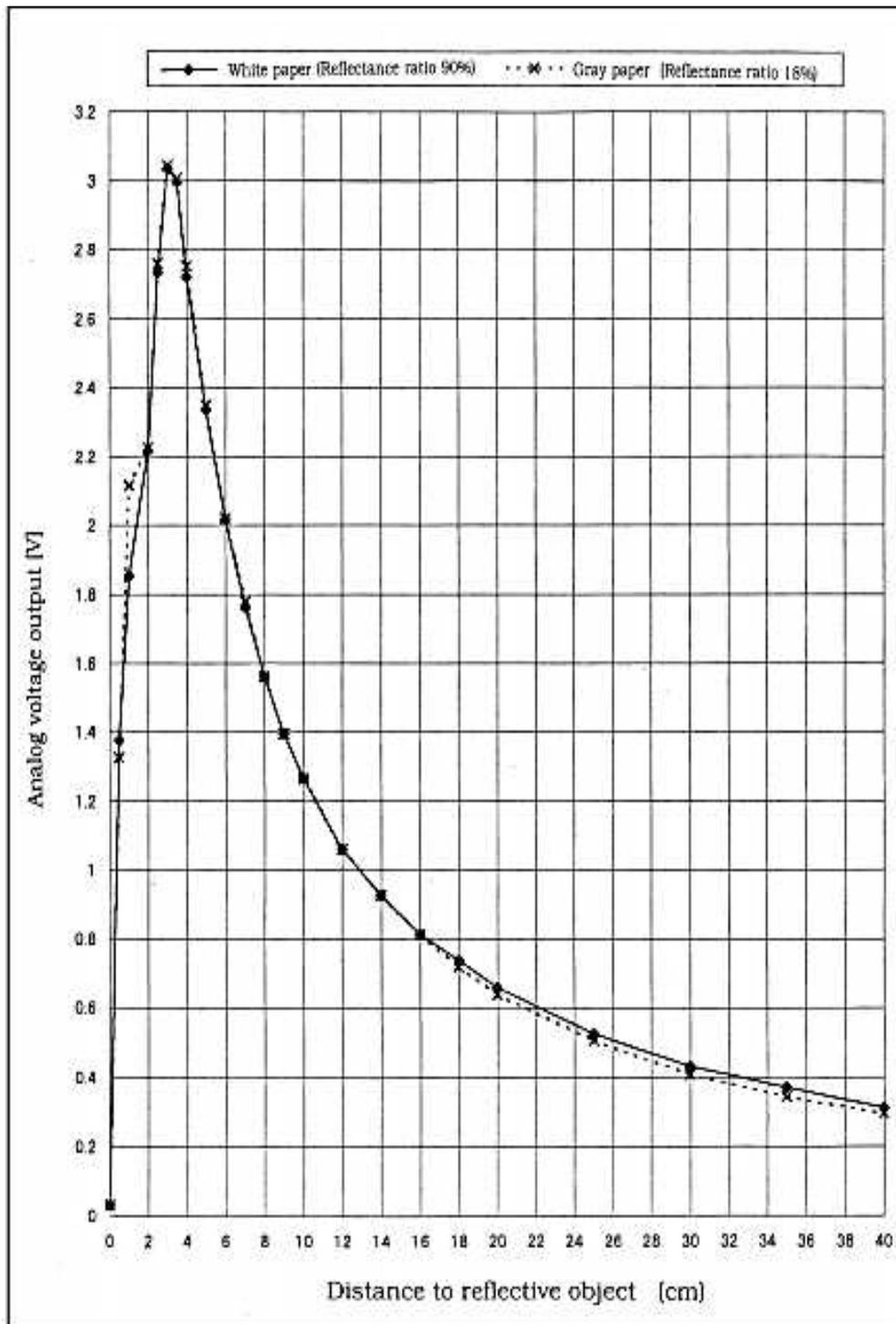
La tensión de salida puede verse afectada por muchas condiciones ambientales diferentes. La más obvia es hasta qué punto un objeto se encuentra del sensor es decir la distancia máxima del sensor. Pero muchas otras condiciones ambientales afectan el sensor, tales como la temperatura, la humedad, el ángulo de los objetos de superficie que el sensor ve, y el color de la superficie del objeto.

Ejemplo de salida característica de distancia

Esta grafica representa la relación que existe entre el voltaje y la distancia a la que se encuentra el objeto en el cual rebota la luz infrarroja la cual se utiliza para determinar la

distancia de un obstáculo; la estimación del objeto se lo hace con valores análogos en base a la medición de voltaje de salida del sensor mediante el cual se puede estimar la distancia a la que se encuentra un objeto utilizando la curva característica de funcionamiento de dicho sensor, tal como se muestra en la figura III 61.

Figura III 61. Relación voltaje-distancia sensor GP2Y0A41SK0F



Fuente: datasheet GP2Y0A41SK0F.pdf

3.3.6. PUENTE H DOBLE L298N MODULO

Figura III 62. Módulo L298N



Fuente: <http://www.techmake.com/rob-00108.html>

El módulo controlador L298N, usa un chip L298N que puede controlar directamente 2 motores de 3-30 VDC, tiene una salida de interfaz de 5V, soporta control desde sistemas de microcontroladores de 3.3V y 5V, con este módulo se puede controlar fácilmente la velocidad y dirección de motores de DC.

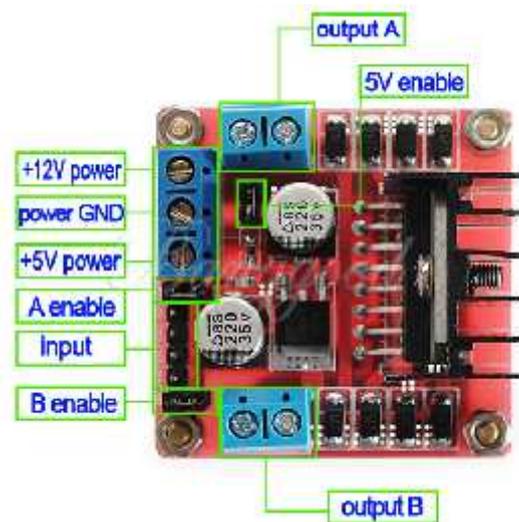
Este módulo puede ser fácilmente integrado a plataformas móviles, mecanismos lineales, etc.

Especificaciones:

- Chip controlador de motor: L298N puente H doble
- Voltaje de alimentación del chip: +5.5V a +24V
- Pico de corriente soportado: 2A
- Voltaje de alimentación de las terminales de control: 3 - 5.5V
- Voltaje de control: Niveles Alto = 5.5V, Bajo = 0V
- Potencia máxima de consumo: 20W
- Tamaño PCB: 53.5mm * 45mm * 30mm
- Distancia entre agujeros de sujeción: 47.5mm y 39mm
- Peso: 26 gramos

El módulo cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador LM7805 que suministra 5V a la parte lógica del integrado L298N. Cuenta con jumpers de selección para habilitar cada una de las salidas del módulo (A y B). La salida A está conformada por OUT1 y OUT2 y la salida B por OUT3 y OUT4. Los pines de habilitación son ENA y ENB respectivamente.

Figura III 63. Partes del módulo L298N



Fuente: <http://www.makieletronic.com/detalle.php?productoid=251>

Cuando el jumper de selección de 5V se encuentra activo, el módulo permite una alimentación de entre 6V a 12V DC. Como el regulador se encuentra activo, el pin marcado como +5V tendrá un voltaje de 5V DC. Este voltaje se puede usar para alimentar la parte de control del módulo ya sea un microcontrolador o un Arduino, pero se recomienda que el consumo no sea mayor a 500 mA.

Cuando el jumper de selección de 5V se encuentra inactivo, el módulo permite una alimentación de entre 12V a 35VDC. Como el regulador no está funcionando, tendremos que conectar el pin de +5V a una tensión de 5V para alimentar la parte lógica del L298N. Usualmente esta tensión es la misma de la parte de control, ya sea un microcontrolador o Arduino.

3.3.7. BLUETOOTH HC-06 MODULE

Figura III 64. Bluetooth hc-06 module



Fuente: <http://botscience.net/store/products/MODCOM/mlm-o-2636188463.jpg>

El bluetooth HC-06 es un módulo maestro esclavo, esto implica que no solo realiza conexiones entrantes, siendo apto también para realizar conexiones a otros dispositivos bluetooth.

Esto nos facilita la comunicación entre diferentes dispositivos que usen la misma tecnología.

3.3.8. FOTSENSOR RETRORREFLECTANTE IBEST M18

Figura III 65. Sensor PESI-R18POC3MD



Fuente: ibestchina.en.alibaba.com

Beneficios

- Reducción del tiempo de inactividad de la máquina
- Reducción de daños mecánicos
- Menos costes de mantenimiento debido a la vida útil más larga
- Alta resistencia a golpes y vibraciones

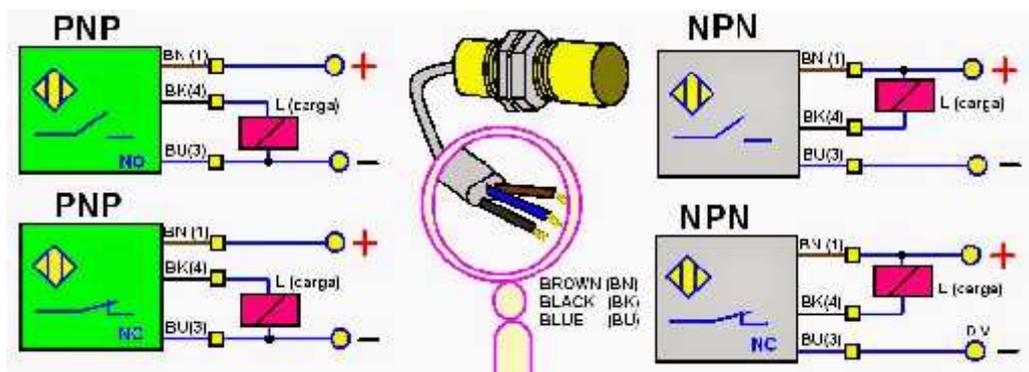
Características

- Diseño: cilíndrico
- Tamaño de la rosca: M18 x 1
- Rango de sensado: 3 metros
- Voltaje de funcionamiento: 10-30 VCD
- Corriente de funcionamiento: < 100 mA.
- Tipo de salida: PNP
- Función de salida: NO y NC
- El cableado eléctrico: DC 2 hilos.
- Angulo de trabajo: de 1 a 5 grados.
- Tipo de conexión: Cable, 3 hilos, 2 m

Conexión del fotosensor de 3 hilos

Es un interruptor de estado sólido que detecta la presencia o ausencia de un color sin necesidad de contacto físico. Lo hace por medio de detección electrónica de la variación de un campo.

Figura III 66. Diagramas de conexiones de sensores polarizados



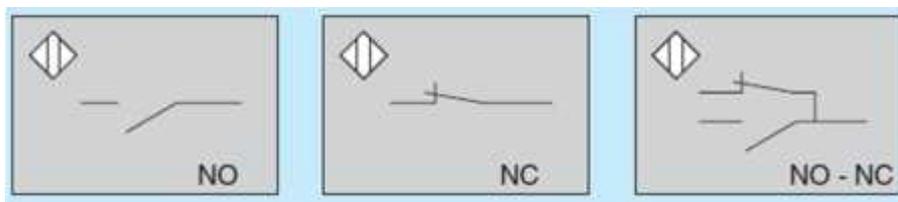
Fuente: http://coparoman.blogspot.com/2014_08_01_archive.html

Contactos de salida.

Los contactos de salida podrán ser:

- NO. El transistor o tiristos de salida se activa en presencia de una pantalla.
- NC. El transistor o tiristos de salida se bloquea en presencia de una pantalla.
- NO-NC. Dos salidas complementarias una activada y la otra bloqueada en presencia de una pantalla.

Figura III 67. Contactos en un detector inductivo.



Fuente: <https://automatismoindustrial.files.wordpress.com/2012/10/ab20.png>

3.3.9. CÁMARA WEB OMEGA 8 MEGAPIXELES-3636K

Figura III 68. Cámara web de 8 Mpx.



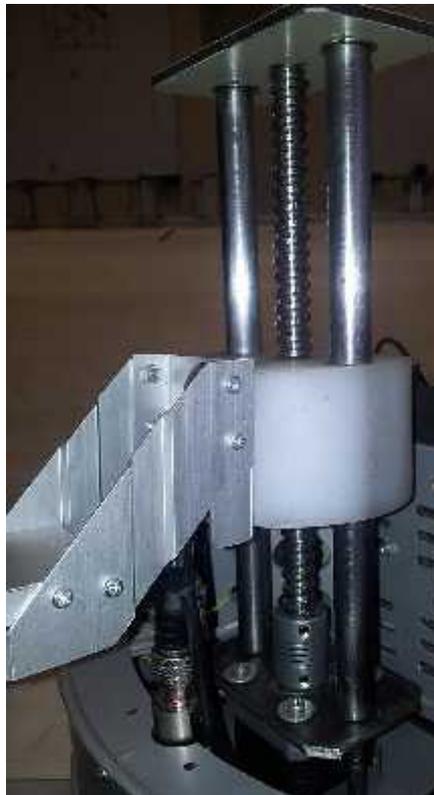
Fuente: <http://www.preciolandia.com/ve/camara-web-omega-8mp-con-microfono-model-72dctg-a.html>

Características

- Fotos de alta definición.
- Cámara de 8 Megapíxeles.
- Mayor rendimiento megapíxeles.
- Hasta 30 fps de video en directo.
- Efectos de video.
- Ajuste manual de 360° enfoque horizontal.
- Conector USB, plug and play.
- Micrófono incorporado.
- Auto balance de blancos y control de color.
- Resolución de 1600x1200.

3.3.10. ACTUADOR LINEAL

Figura III 69. Actuador lineal



Fuente: Autor

Un actuador lineal eléctrico es un dispositivo que convierte el movimiento de rotación de un motor de corriente continua de baja tensión en movimiento lineal, es decir, los movimientos de empuje y halado. De esta manera es posible elevar, ajustar, inclinar, empujar o halar objetos pesados o difíciles de alcanzar.

Adicionalmente, los actuadores ofrecen seguridad, movimiento silencioso, limpio de control preciso. Ellos son energía eficiente de larga vida útil con poco o ningún mantenimiento. La instalación de un actuador es muy fácil en comparación con los sistemas hidráulicos y el espacio requerido es mucho menor, ya que no tiene bombas o mangueras.

3.4. SOFTWARE UTILIZADO

Para el desarrollo de la tesis se optó por utilizar la interfaz de programación de Arduino 1.0.5. que maneja un lenguaje basado en C; el cual permite controlar el dispositivo Arduino Mega 2560; el cual contiene un microprocesador ATMEL 2560 e interfaces de entrada y salida para conectar sensores y actuadores que serán los encargados de controlar el robot omnidireccional.

Además se creó una aplicación en Labview 2013 mediante la cual se tomara imágenes del el ambiente cerrado en el cual se va a desplazar el robot mencionado y determinara la mejor trayectoria a seguir desde un punto inicial hasta un punto final según los requerimientos del sistema; además de estar en la capacidad de esquivar obstáculos de ser el caso necesario para cumplir con el objetivo de transporte de piezas; para los cual se utilizó la combinación de algoritmos que permitan la búsqueda de una mejor trayectoria.

3.4.1. ARDUINO ENVIROMENT 1.0.5

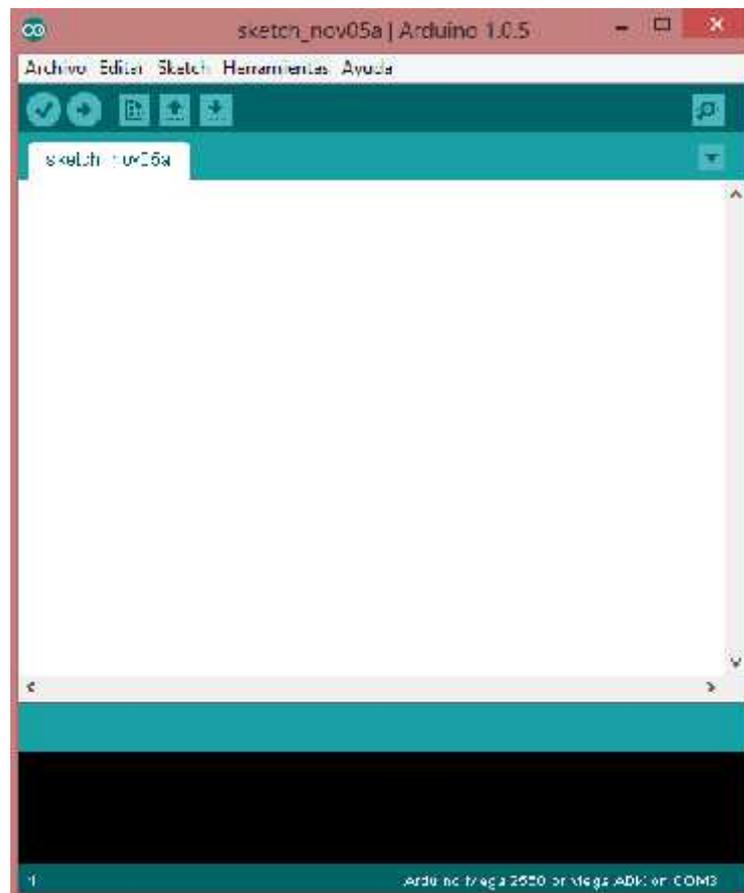
El entorno de desarrollo Arduino contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús.

El entorno contiene librerías de hardware existente en el mercado el cual facilita la programación y la correcta utilización de los mismos permitiendo obtener los valores de sensores de la manera más exacta posible.

Además contiene un monitor serial que permite verificar los datos de serie que se envían y reciben en el Arduino mega el cual facilita la opción de poder corregir errores de manera más fácil; basta con cargar el programa mediante la opción carga para guardar el programa escrito dentro del microcontrolador lo que permite un ahorro de tiempo al no tener que cargar el código al microcontrolador mediante quemadores sino que se lo puede hacer de manera directa mediante el cable de conexión USB.

En la figura III 70. se visualiza la interfaz del programa Arduino 1.0.5.

Figura III 70. Entorno Arduino 1.0.5.



Fuente: Lopez J., Santillan G.

3.4.1.1. DISEÑO DEL PROGRAMA EN ARDUINO

El programa creado mediante el entorno de programación de Arduino es el encargado de interactuar o intercambiar información desde los sensores hacia la plataforma HMI creada en LABVIEW y viceversa; el programa toma los datos en bruto y los transforma a datos legibles para el usuario mediante fórmulas matemáticas y librerías existentes; de manera que se puedan entender los datos obtenidos y decidir que se quiere hacer con estos.

Además el programa será el encargado de interactuar con los motores, decidir el sentido de giro y la velocidad a la que se moverán estos; además medir la distancia recorrida por cada llanta mediante los encoders y sensar los infrarrojos para el cálculo de distancia o proximidad de objetos u obstáculos y el sensor inductivo para el seguimiento de una trayectoria o línea ubicada en el piso.

Las partes fundamentales del código se describen a continuación:

En la función **setup** se declaran cuáles van a ser los pines de salida y de entrada a utilizarse en la placa del Arduino Mega para recibir los datos de los sensores y para activar los actuadores y motores del robot.

```
void setup () {
```

```
// Declaración entradas y salidas sensores
```

Se asigna el pin 40 denominado a1 como entrada de la señal del sensor inductivo que inicializara la función de seguidor de línea

```
pinMode(a1,INPUT);
```

```
// Declaración de los pines de salida para la activación de los motores
```

```
pinMode(26,OUTPUT);
```

```
pinMode(27,OUTPUT);
```

```
pinMode(28,OUTPUT);
```

```
pinMode(29,OUTPUT);
```

```
pinMode(30,OUTPUT);
```

```
pinMode(31,OUTPUT);
```

//Encoder

Se asignan los pines de entrada para sensar los datos provenientes de los encoders.

```
pinMode(encoder1PinA, INPUT);
pinMode(encoder2PinA, INPUT);
pinMode(encoder1PinB, INPUT);
pinMode(encoder2PinB, INPUT);
digitalWrite(encoder1PinA, HIGH);
digitalWrite(encoder1PinB, HIGH);
digitalWrite(encoder2PinA, HIGH);
digitalWrite(encoder2PinB, HIGH);
attachInterrupt(0, rencoder1, CHANGE);
attachInterrupt(2, rencoder2, CHANGE);
attachInterrupt(4, rencoder3, CHANGE);
attachInterrupt(5,seguidor,LOW);
```

Sensores de proximidad

Se asignan los pines analógicos como entradas para las señales de los sensores infrarrojos de proximidad de acuerdo al orden especificado en capítulo II figura 28 Ubicación de los sensores infrarrojos; por lo tanto los sensores quedarían en el siguiente orden:

```
IR1=A0
IR2=A1
IR3=A2
IR4=A3
IR5=A4
IR6=A5
IR7=A6
IR8=A7
IR9=A8
```

Y se declaran en el programa de la siguiente manera:

```
D1.begin(A0);  
D2.begin(A1);  
D3.begin(A2);  
D4.begin(A3);  
D5.begin(A4);  
D6.begin(A5);  
D7.begin(A6);  
D8.begin(A7);  
D9.begin(A8);
```

3.4.2. TWIDO SUITE 2.31

Twido Suite es el software nativo de programación del PLC Twido, presenta funcionalidades tanto para el usuario avanzado como para el principiante, gracias a su entorno intuitivo y de fácil manejo. Su entorno de programación es variado, haciendo posible la declaración de memorias de almacenamiento tanto en bits como arrays, siendo la programación más usada en ladder o escalera, un ambiente amigable que hace que la programación sea dinámica y eficiente.

Figura III 71. Pantalla inicio Twido 2.31



Fuente: Lopez J., Santillan G.

Al iniciar nuestra aplicación tuvimos que tomar en cuenta varios factores, entre los cuales los principales se pueden destacar los siguientes:

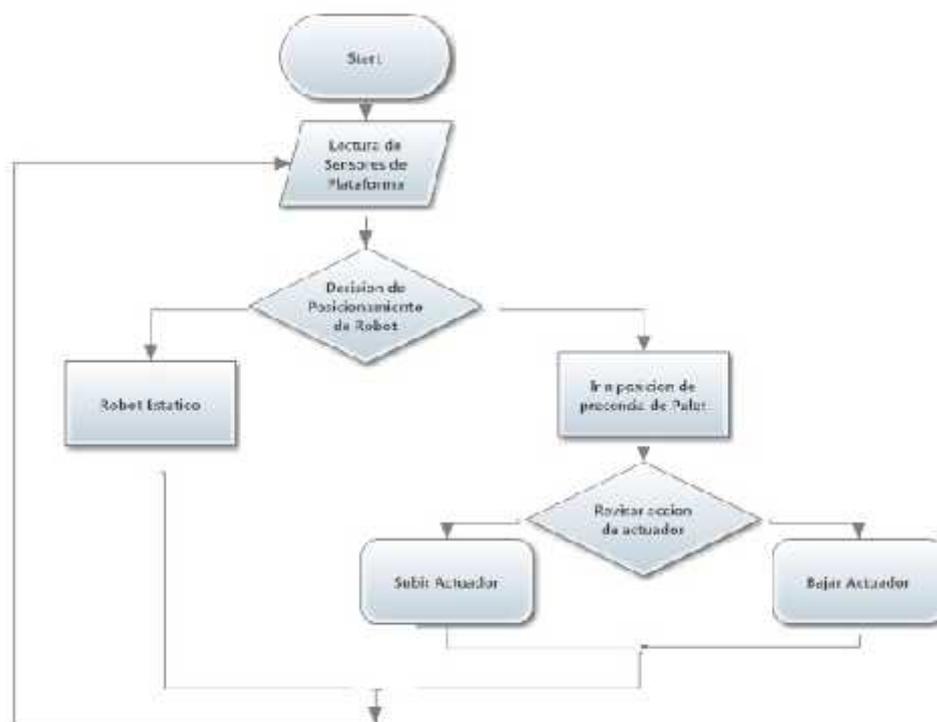
- Diseño en diagrama de contactos (ladder) compatible con las necesidades del proyecto propuesto.
- Recolección de datos desde el MPS
- Implementación de un HMI compatible con el PLC twido 40 CAE DRF.

3.4.2.1. DISEÑO EN DIAGRAMA DE CONTACTOS (LADDER)

El robot paletizador, deberá cumplir funciones específicas, entre las que podemos destacar el transporte de palets de una estación a otra, para poder realizar los diferentes procesos asignados a cada una de las estaciones, ya sea de almacenamiento o procesamiento de los mismos.

A continuación se muestra el algoritmo básico que seguirá el robot paletizador para recolectar los materiales.

Figura III 72. Algoritmo Robot Paletizador



Fuente: Lopez J., Santillan G.

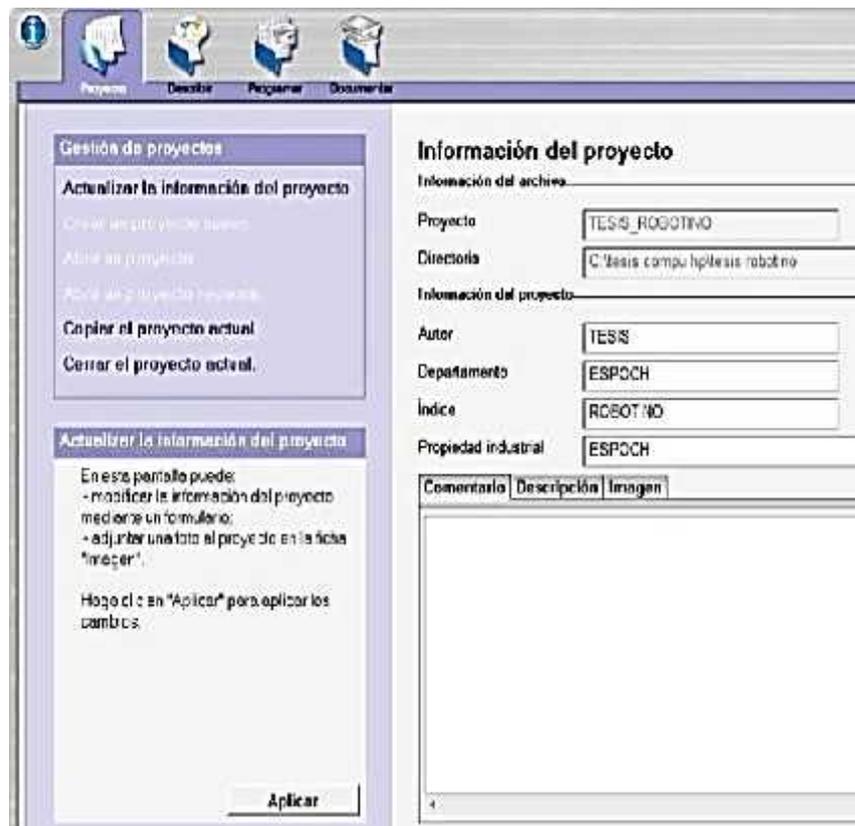
3.4.2.2. CONFIGURACIÓN DEL PLC Y RECOLECCIÓN DE DATOS DESDE EL MPS

Para poder recolectar los datos desde el sistema MPS hacia nuestro Sistema, usamos el protocolo modbus Ethernet, que usara el plc, mostrándose la configuración del puerto Ethernet para poder utilizar este protocolo y posteriormente implementarlos al HMI.

La configuración del PLC se realizó en primera instancia desde el puerto serie, para ajustar los parámetros, como son dirección de red, mascara de subred y demás configuraciones que se muestran a continuación.

Creación de un nuevo Proyecto

Figura III 73. Creación de nuevo proyecto Twido Soft



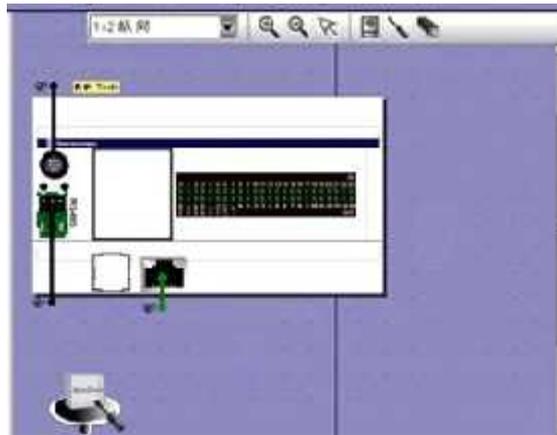
Fuente: Lopez J., Santillan G.

En Gestión de proyectos, pulsamos sobre crear un proyecto nuevo y llenamos los campos, para tener un control sobre el proyecto que realizamos y sus modificaciones realizadas, creando así un nuevo proyecto en blanco para poder usarlo.

Descripción del Hardware

Para nuestra aplicación se escogió el PLC TWIDO CAE-40DRF, que se ajusta a los parámetros requeridos para desarrollar nuestra aplicación. La cual contiene las entradas y salidas necesarias, y el puerto ethernet, necesario para realizar la comunicación modbus con Labview.

Figura III 74. Descripción del Hardware

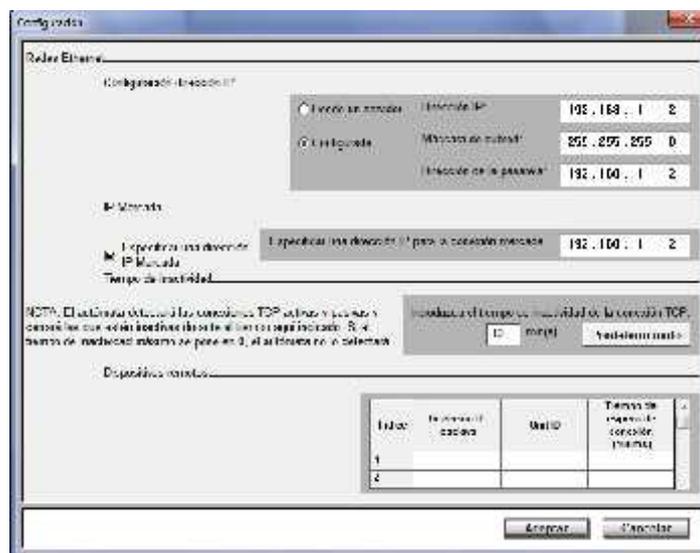


Fuente: Lopez J., Santillan G.

Configuración puerto de red en TWIDO soft

Al configurar por primera vez el plc, debemos establecer los parámetros de red a ser transmitidos por única vez mediante el cable serial, para poder usarlo desde la siguiente vez en la red de área local.

Figura III 75. Configuración de red PLC

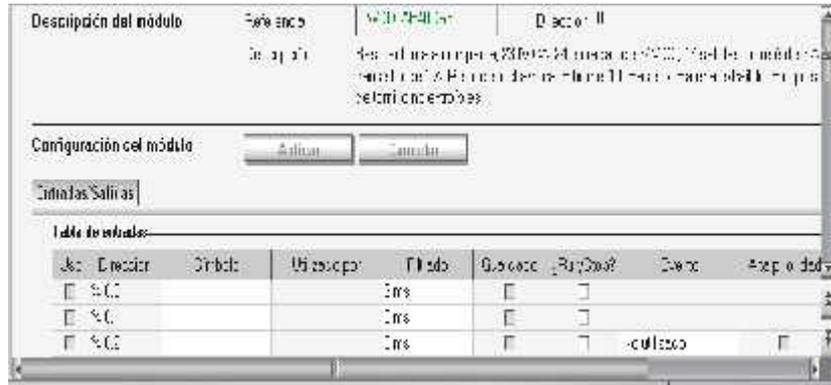


Fuente: Lopez J., Santillan G.

Direccinamiento y edicion de simbolos

El automata interactua con el exterior mediante señales de entrada y salida que son configuradas a traves de la interfaz, las cuales seran necesarias para realizar la supervision mediante Modbus ethernet con Labview.

Figura III 76. Asignación de Alias



Fuente: Lopez J., Santillan G.

En la configuración de los datos, debemos definir el tipo de datos a transmitir en este caso usamos memoria simples, que son lugares de almacenamiento de un solo bit, a transmitir mediante modbus.

Figura III 77. Configuración de memorias del HMI



Fuente: Lopez J., Santillan G.

Identificación de las entradas y salidas del PLC

Para poder realizar el programa se procedió a la identificación de las entradas y salidas de la plataforma donde se ubicará el robot paletizador.

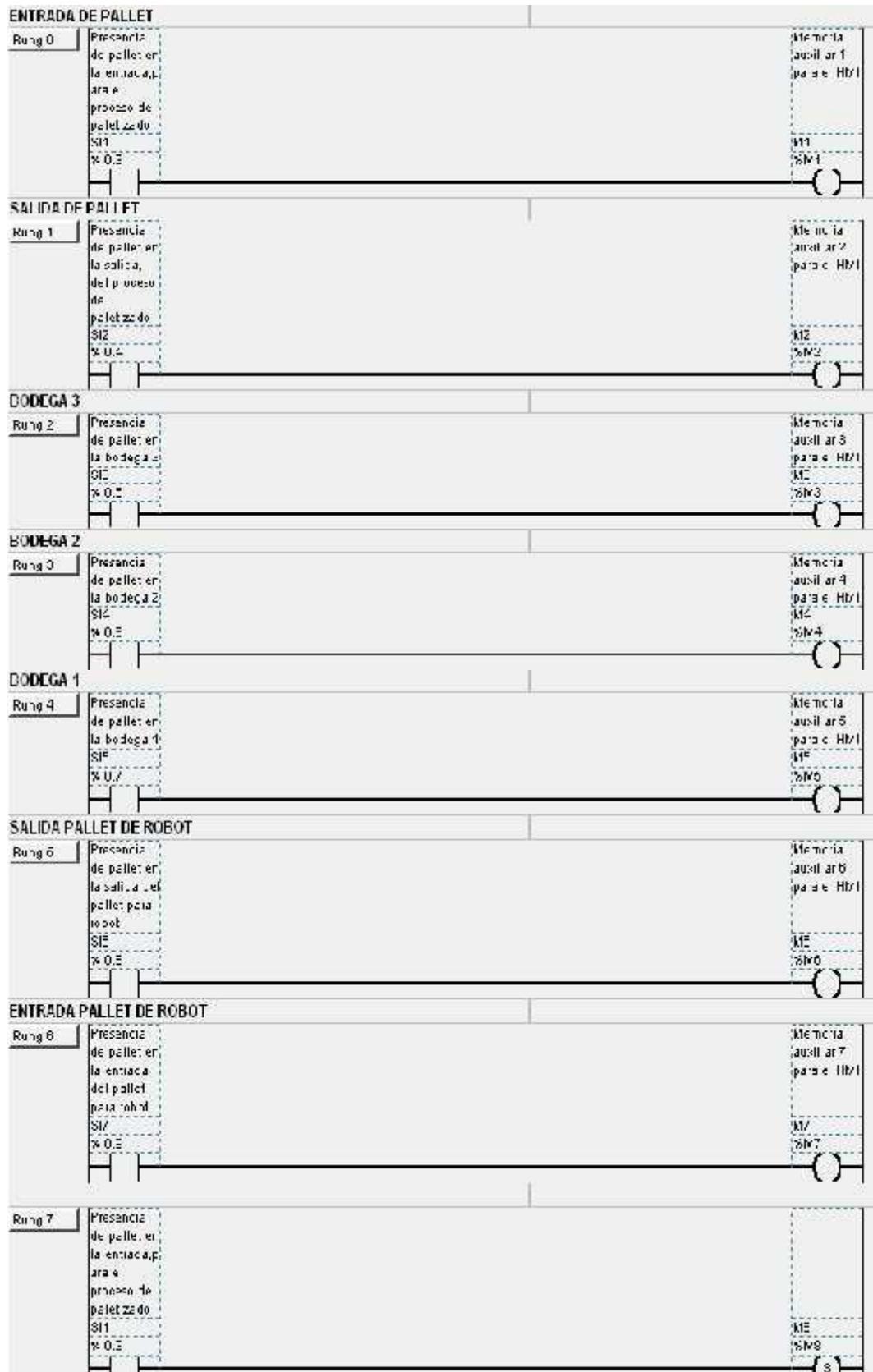
Tabla III VII Tabla de entradas y salidas de PLC TWIDO CAE 40 DRF

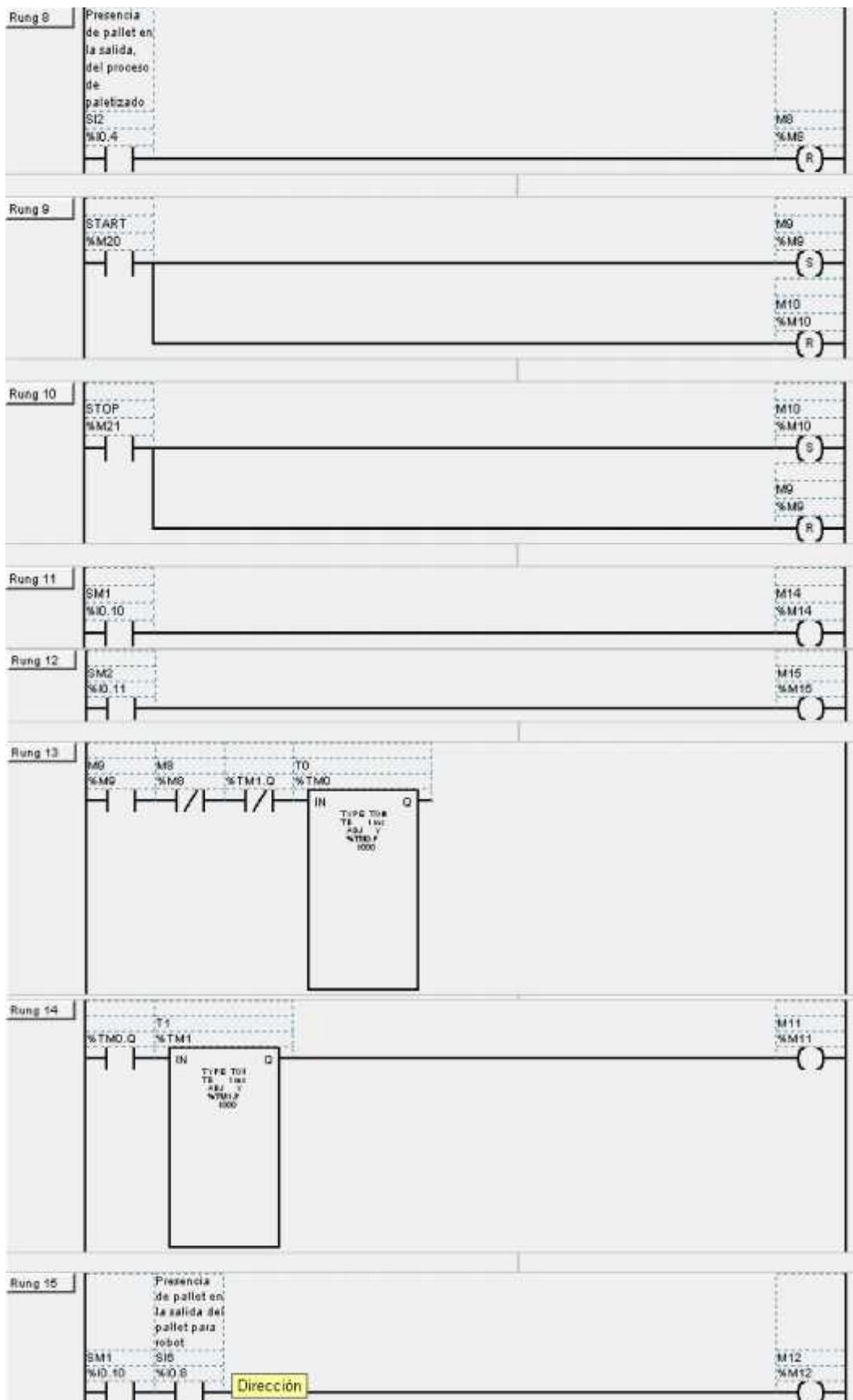
| | | | |
|---------|-------|------------------|--|
| Entrada | I0.9 | SI7 | Sensor inductivo de salida de pallets hacia el brazo robótico. |
| Entrada | I0.10 | SM1 | Sensor magnético que detecta la posición del vástago del cilindro 1. |
| Entrada | I0.11 | SM2 | Sensor magnético que detecta la posición del vástago del cilindro 2. |
| Salida | Q0.2 | Electroválvula_1 | Electroválvula que acciona el cilindro 1 para desplazar el pallet con productos. |
| Salida | Q0.3 | Electroválvula_2 | Electroválvula que acciona el cilindro dos para desplazar el pallet vacío. |
| Salida | Q0.4 | Foco_Rojo | Señal luminosa de paro del proceso. |
| Salida | Q0.5 | Foco_Amarillo | Señal luminosa que indica una anomalía. |
| Salida | Q0.6 | Zumbador | Señal acústica que se activa cuando las bodegas se encuentren ocupadas. |
| Salida | Q0.7 | Foco_Verde | Señal luminosa que se indica cuando el proceso se encuentra ejecutando. |
| Salida | Q0.8 | Baliza_Giratoria | Señal luminosa de presencia de pallets en el bloque de entrada. |
| Entrada | I0.3 | SI1 | Sensor inductivo para entrada de pallets. |
| Entrada | I0.4 | SI2 | Sensor inductivo para salida de pallets. |
| Entrada | I0.5 | SI3 | Sensor inductivo para la bodega 3. |
| Entrada | I0.6 | SI4 | Sensor inductivo para la bodega 2. |
| Entrada | I0.7 | SI5 | Sensor inductivo para la bodega 1. |
| Entrada | I0.8 | SI6 | Sensor inductivo para la entrada de pallets hacia el robot móvil. |

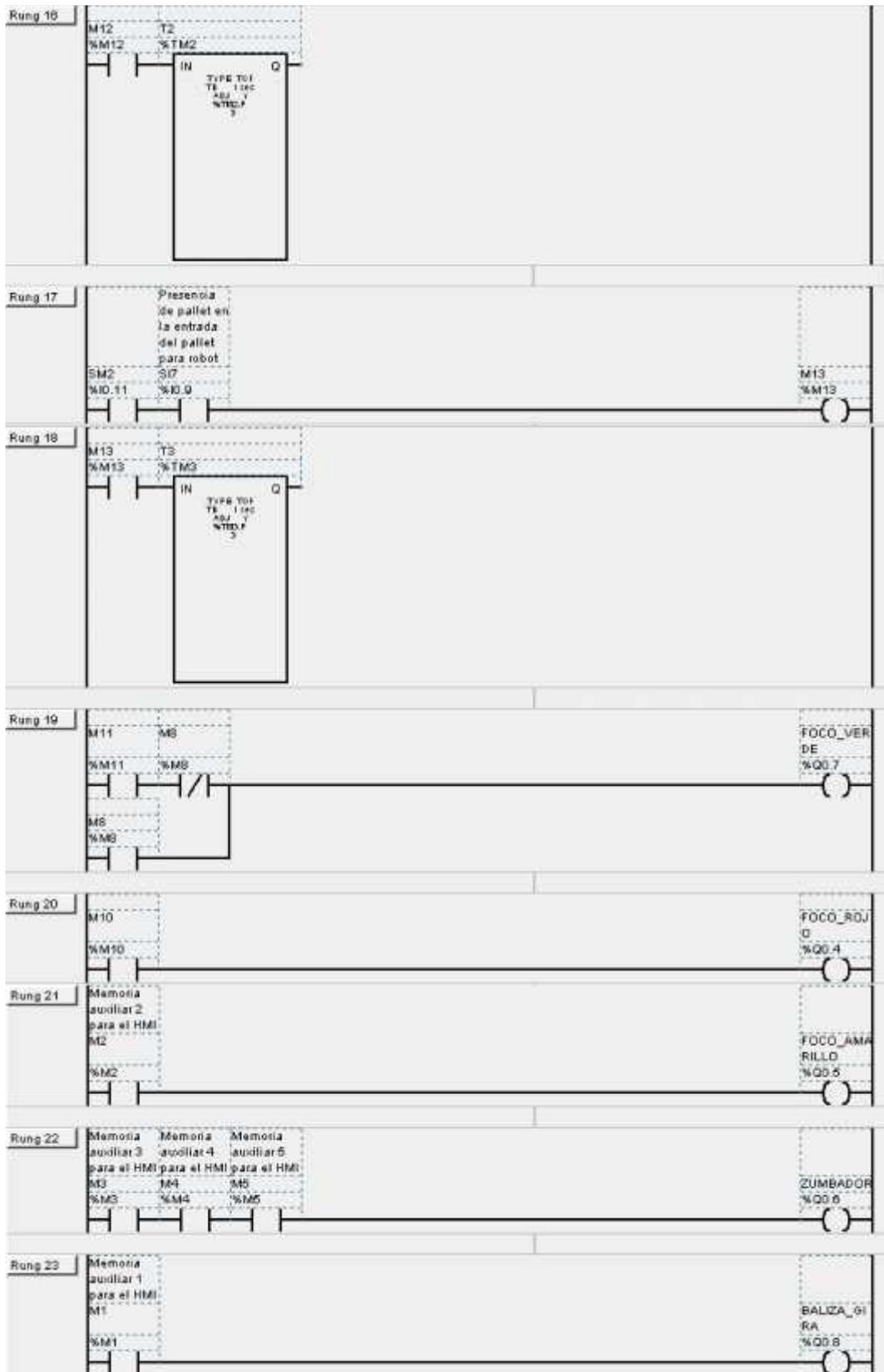
Fuente: Lopez J., Santillan G.

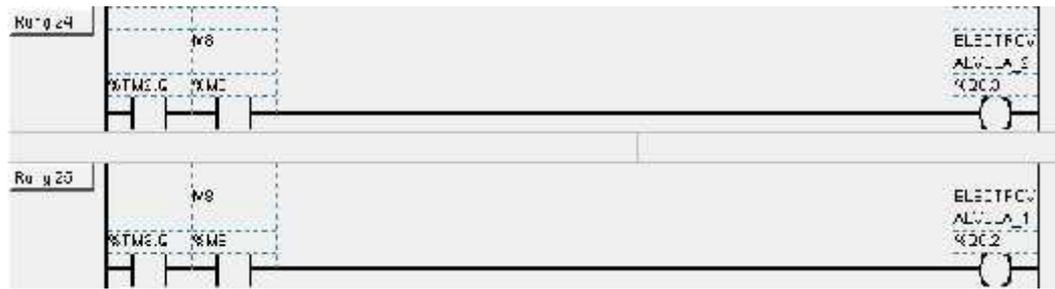
3.11.2.3 EDICIÓN DEL PROGRAMA

Figura III 78. Edición del Programa en Twido









Fuente: Lopez J., Santillan G.

Al transferir el programa se genera un archivo binario, que será ejecutado por el plc, para su posterior uso y conexión a Labview, mediante modbus Ethernet.

Figura III 79. Transmisión de programa a PLC TWIDO

Conexión para la puesta a punto

Este modo de conexión permite conectarse directamente a un autómata o transferir una aplicación entre el PC y un autómata.

Seleccione una Conexión:

| Tipo | Nombre | Modo de conexión | Dirección IP/Número |
|---------|-----------------|------------------|---------------------|
| Project | Modbus Ethernet | Ethernet | 192.168.1.2, Direc. |
| Pc | COM1 | Serial | COM1, Punit |

Establecer la comunicación:

| Nombre | Modo de conexión | Dirección IP/Número | Dirección | Caudal | Paridad | Bits de parada | Timeout | Timeout de la pausa |
|-----------------|------------------|---------------------|-----------|--------|---------|----------------|---------|---------------------|
| Modbus Ethernet | Ethernet | 192.168.1.2 | Direc. | | | | 5000 | 5 |

Fuente: Lopez J., Santillan G.

3.4.3. LABVIEW 2013

LabVIEW es una plataforma de programación gráfica que ayuda a ingenieros a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas. Ofrece integración sin precedentes con software legado existente, IP y hardware al aprovechar las últimas tecnologías de cómputo. LabVIEW ofrece herramientas para resolver los problemas de hoy en día y la capacidad para la futura innovación, más rápido y de manera más eficiente.

Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido, pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico además de ofrecer la posibilidad de interactuar con otros programas y lenguajes.

Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, MAC y GNU/Linux.

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica (Instrumentación electrónica) sino también a su programación embebida, comunicaciones, matemáticas, etc.

Figura III 80. Pantalla de Inicio Labview 2013



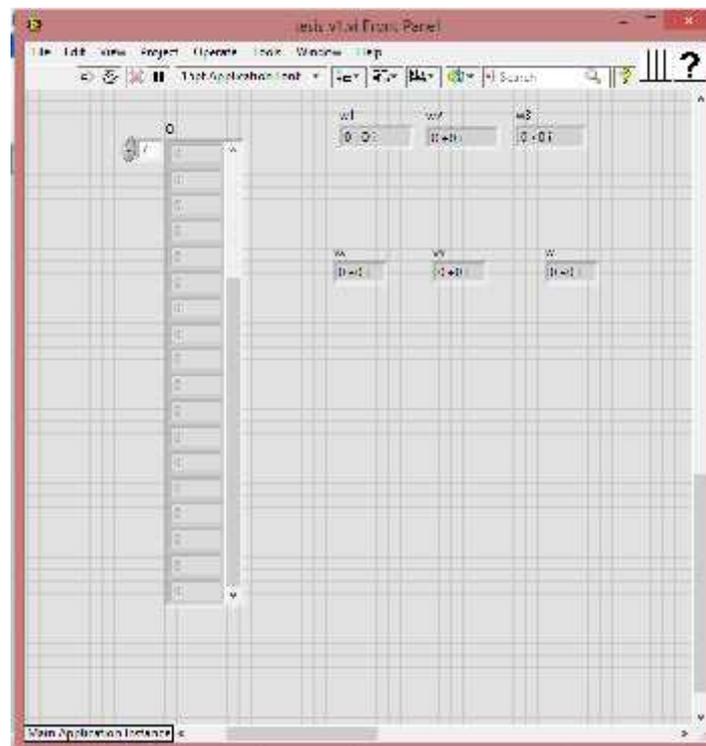
Fuente: Lopez J., Santillan G.

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación pueden hacer programas relativamente complejos, imposibles de hacer con lenguajes tradicionales.

Como se ha dicho es una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando su comprensión. Al tener ya pre-diseñados una gran cantidad de bloques, se le facilita al usuario la creación del proyecto, con lo cual en vez de estar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final. Cada VI consta de dos partes diferenciadas:

- **Panel Frontal:** es la interfaz con el usuario, la utilizamos para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real (como van fluyendo los datos). En esta interfaz se definen los controles (se usa como entradas, pueden ser botones, marcadores etc.) e indicadores (se usan como salidas, pueden ser gráficas).

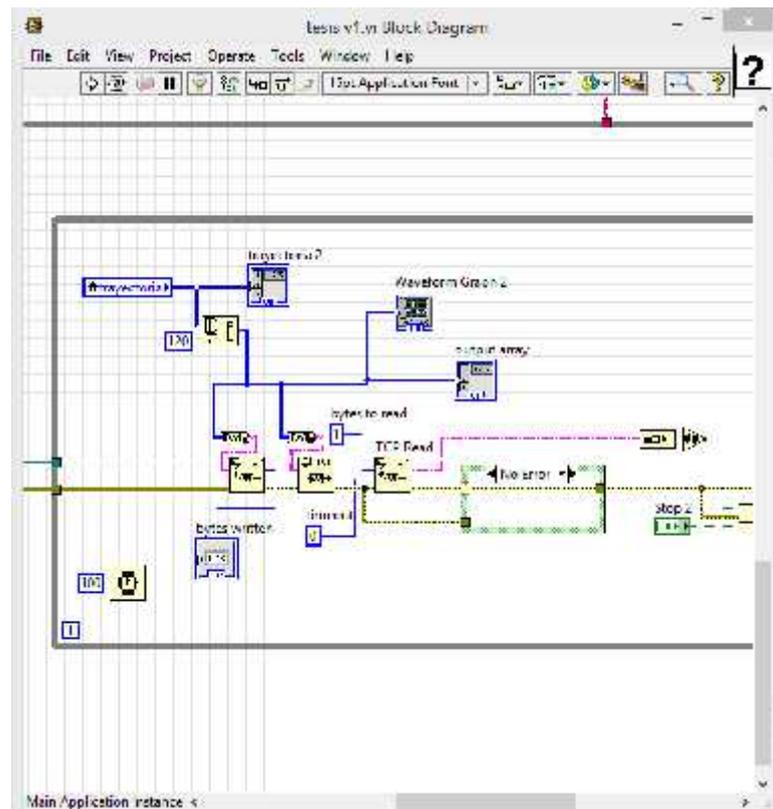
Figura III 81. Vista del panel frontal



Fuente: Lopez J., Santillan G.

- **Diagrama de Bloques:** es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan íconos que realizan una determinada función y se interconectan (el código que controla el programa. Suele haber una tercera parte icono/conector que son los medios utilizados para conectar un VI con otros VIs.

Figura III 82. Vista del diagrama de bloques



Fuente: Lopez J., Santillan G.

3.4.3.1. DISEÑO DEL PROGRAMA EN LABVIEW

El siguiente diseño está realizado en Labview 2013; con el objetivo de generar una trayectoria identificando el mejor camino desde un punto inicial hasta un punto final en una mesa de trabajo del sistema MPS del laboratorio de automatización; la misma que se utilizará para que se pueda desplazar el robot omnidireccional en toda el área de la mencionada mesa y en cualquier dirección; con el objetivo de transportar piezas desde

los diferentes pallets ubicados en la mesa de manera autónoma y sin la intervención humana simulando procesos industriales.

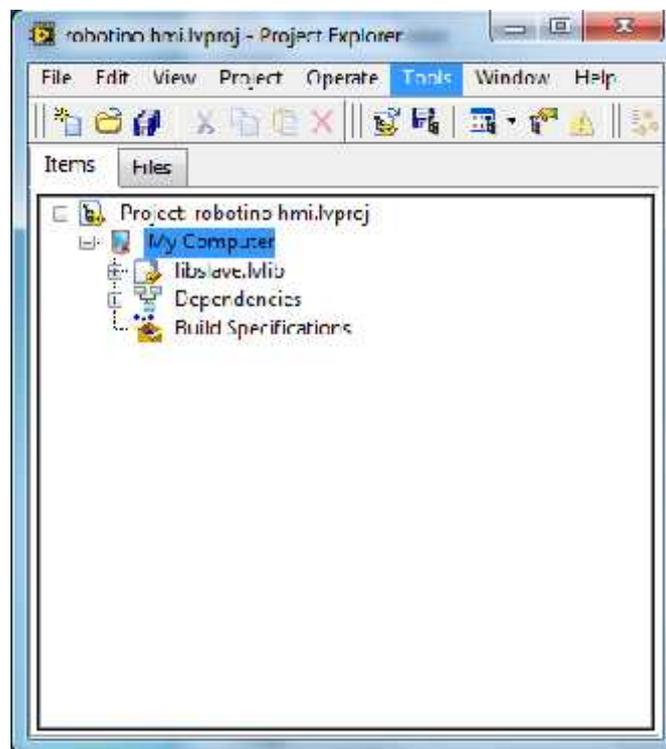
El programa está desarrollado en diferentes etapas la adquisición de imágenes; el tratamiento de las imágenes; la implementación de algoritmo AD; la obtención de la trayectoria; la integración del modelo cinemático del robot omnidireccional y la transmisión de la información.

Las etapas serán descritas a detalle a continuación y explicando el funcionamiento de cada una.

3.4.3.1.1. CREACIÓN DEL HMI EN LABVIEW

Al diseñar nuestro proyecto, debemos considerar la comunicación entre nuestro computador y la red local LAN, en la que está situado el PLC configurada anteriormente, con el objeto de mantener una comunicación estable entre el PLC Twido y Labview, mediante el protocolo de comunicaciones MODBUS Ethernet.

Figura III 83. Creación de nuevo Proyecto en Labview

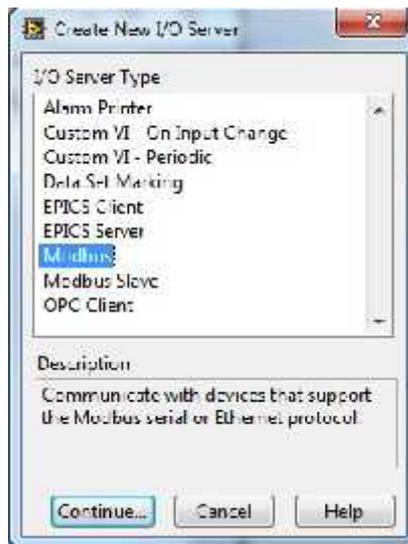


Fuente: Lopez J., Santillan G.

3.4.3.1.2. CREACIÓN DE LIBRERÍA DE COMUNICACIÓN

Para realizar la comunicación creamos un nuevo servidor de entrada y salida de datos (I/O server), que servirá como puente de comunicaciones entre el PLC Twido y nuestro computador.

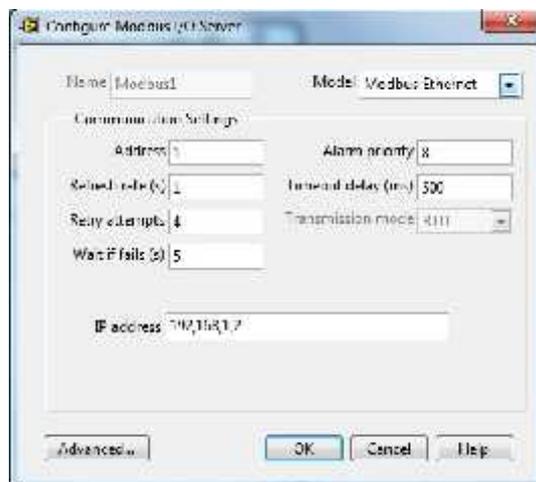
Figura III 84. Creación de servidor I/O



Fuente: Lopez J., Santillan G.

Para nuestra configuración usamos, Modbus ethernet, configuración que nos permite apuntar el servidor de entrada/salida a la dirección de nuestro PLC configurador anteriormente en TwidoSuite

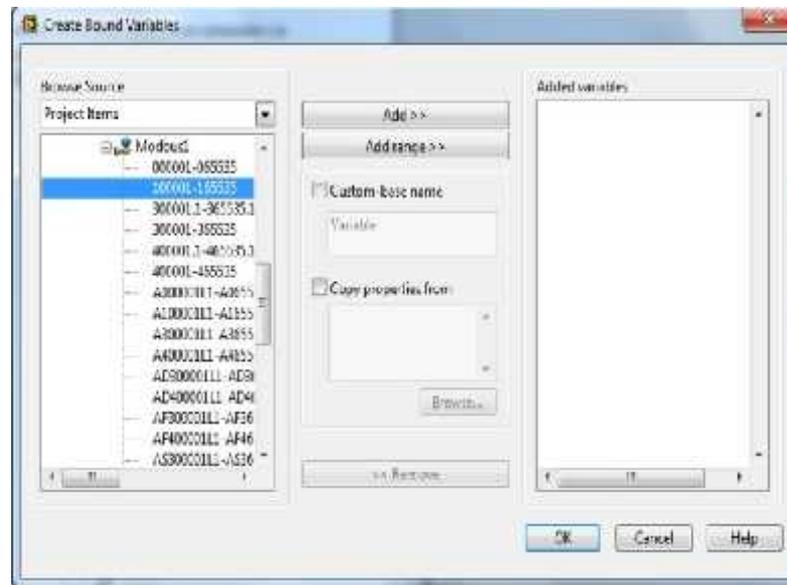
Figura III 85. Configuración de servidor Modbus I/O



Fuente: Lopez J., Santillan G.

Los rangos de valores a usar son 100001-1000015, de tipo booleano, correspondiente a los valores de las memorias del PLC, para crear variables compartidas a ser usadas en el HMI.

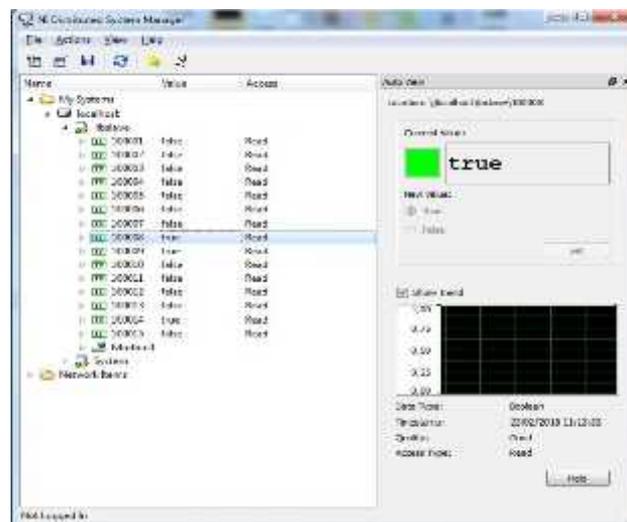
Figura III 86. Creación de variables compartidas



Fuente: Lopez J., Santillan G.

Al desplegar las variables compartidas en el NI Distributed System Manager, obtenemos una forma simple y eficiente de verificar la comunicación entre Labview y nuestro PLC TWIDO

Figura III 87. Consola de sistema distribuido

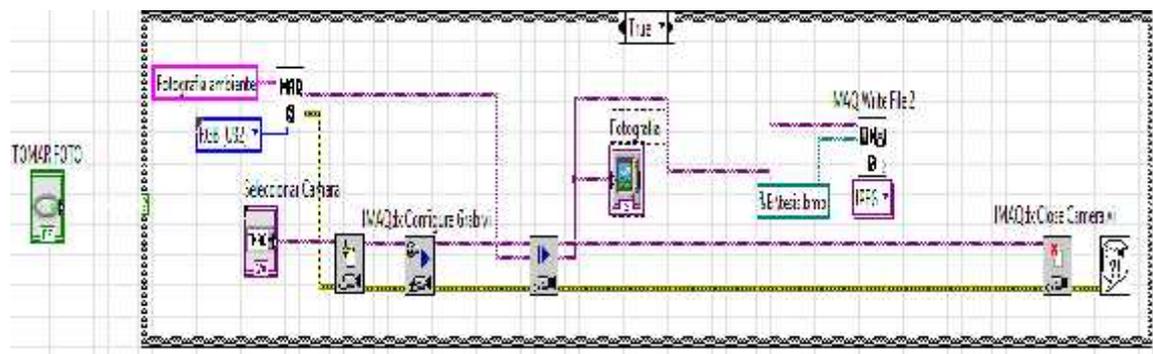


Fuente: Lopez J., Santillan G.

3.4.3.2. ADQUISICIÓN DE IMÁGENES

En esta etapa se procede a inicializar el proceso de toma de imágenes mediante una cámara web para que sea almacenarla en una ubicación de previamente definida por el usuario en formato BMP; el proceso se encuentra encapsulado en un bloque de Case Structure condicional q permite seleccionar el instante en el que se quiera tomar la imagen a través de un pulsador de tipo booleano.

Figura III 88. Adquisición de imagen



Fuente: Lopez J., Santillan G.

A continuación se detalla cada proceso de esta etapa:

3.4.3.2.1. IMAQ Create VI

Crea una ubicación de memoria temporal para almacenar una imagen; permite inicializar el espacio de memoria a utilizar para la captura de imagen.

Figura III 89. IMAQ CREATE



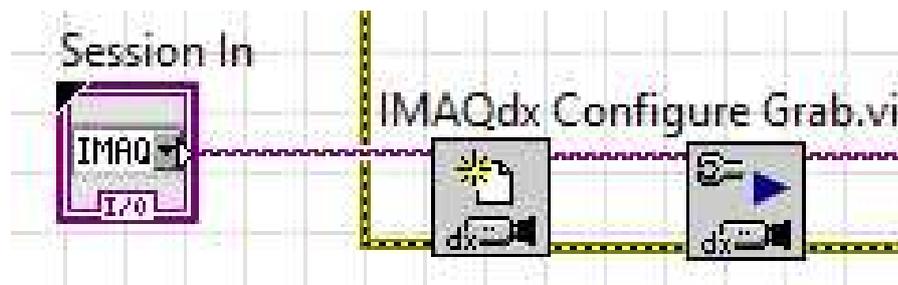
Fuente: Lopez J., Santillan G.

Se configura el nombre con el que se almacenara en la memoria temporal y el formato de la imagen que será RGB 32; es decir una imagen a colores de 32 bytes.

3.4.3.2.2. INICIALIZACIÓN DE LA CÁMARA

En este proceso se selecciona la cámara a utilizar en el panel frontal y se inicializa el driver de la cámara y el proceso de captura.

Figura III 90. Proceso de inicialización cámara



Fuente: Lopez J., Santillan G.

Sesión IN permite seleccionar la cámara a utilizar para tomar imágenes como se observa en la siguiente figura; el bloque IMAQdx localiza el driver e inicializa la cámara; e IMAQdx Configure Grab configura e inicia el proceso de toma de imágenes.

Figura III 91. Selección y toma de fotografía (Panel Frontal)

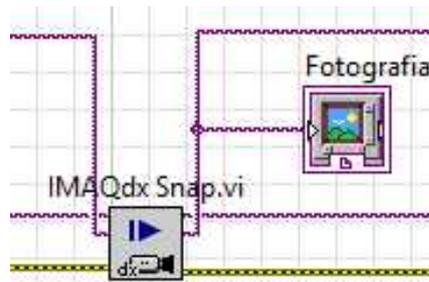


Fuente: Lopez J., Santillan G.

3.4.3.2.3. IMAQdx Snap

Configura, inicia, y adquiere una foto a la vez en cada iteración; además visualiza la imagen adquirida en el panel frontal mediante el bloque image display denominado Fotografía en este proceso.

Figura III 92. IMAQdx Snap

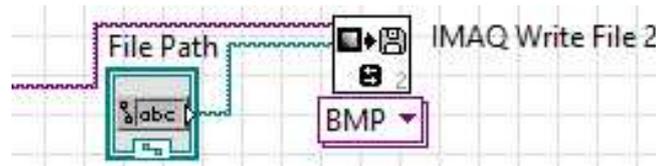


Fuente: Lopez J., Santillan G.

3.4.3.2.4. IMAQ write

Almacena la imagen tomada por la cámara web en formato BMP; mientras que file path permite especificar la ubicación en el disco duro en la que se desea guardar el archivo de imagen.

Figura III 93. IMAQ write

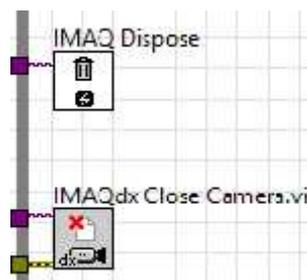


Fuente: Lopez J., Santillan G.

3.4.3.2.5. IMAQdx Close

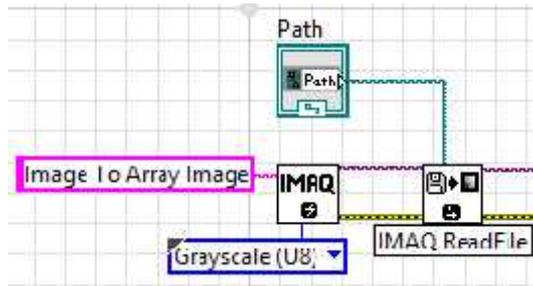
Detiene una adquisición en progreso, libera los recursos asociados con una adquisición, y se cierra la sesión de la cámara especificada; mientras IMAQ Dispose destruye una imagen y libera el espacio que ocupaba en la memoria para liberar recursos del sistema.

Figura III 94. IMAQdx Close



Fuente: Lopez J., Santillan G.

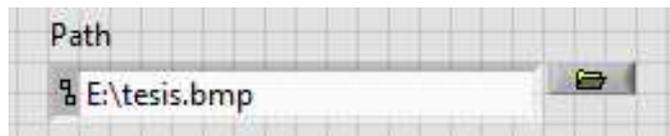
Figura III 96. IMAQ ReadFile



Fuente: Lopez J., Santillan G.

Mediante Path se agrega el destino o dirección en donde se encuentra almacenada la imagen del entorno adquirida por la cámara en el panel frontal tal como se muestra a continuación:

Figura III 97. Ruta de almacenamiento de la imagen

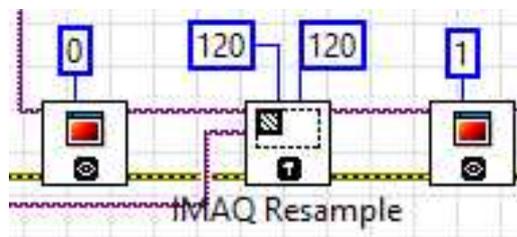


Fuente: Lopez J., Santillan G.

3.4.3.3.2. IMAQ Resample

Redimensiona la imagen a una resolución inferior de 120x120 píxeles.

Figura III 98. IMAQ Resample



Fuente: Lopez J., Santillan G.

Se realiza con propósitos matemáticos para que el cálculo de la trayectoria sea óptimo en tiempo; ya que si se usa la imagen real sin bajar la resolución los cálculos del algoritmo A* para la obtención de la trayectoria serán muy grandes y necesitarán de grandes recursos del sistema además de tomar mucho tiempo lo que no permitiría que el

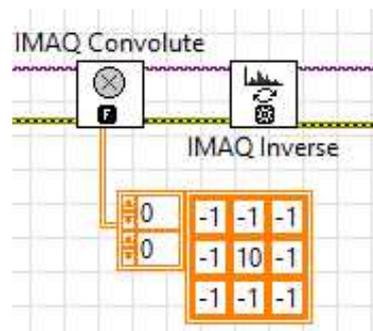
sistema sea en tiempo real ya que la latencia hasta finalizar los cálculos sería aproximadamente de 5 segundos. Al utilizar una resolución inferior se resuelve el problema de retardo.

La imagen real se visualiza automáticamente en la ventana 0 del primer bloque y la imagen reducida se visualiza en la ventana 1 del tercer bloque.

3.4.3.4. FILTRO E INVERSIÓN DE COLOR

La imagen reducida ingresa al bloque IMAQ Convolute que se utiliza para crear un filtro que resaltara los bordes de la imagen permitiendo identificar formas y objetos de mejor manera; luego pasa al bloque de IMAQ Inverse donde la imagen pasará a una inversión de colores dando como resultado una imagen como si fuera el negativo de la fotografía.

Figura III 99. Filtro e inversión de color



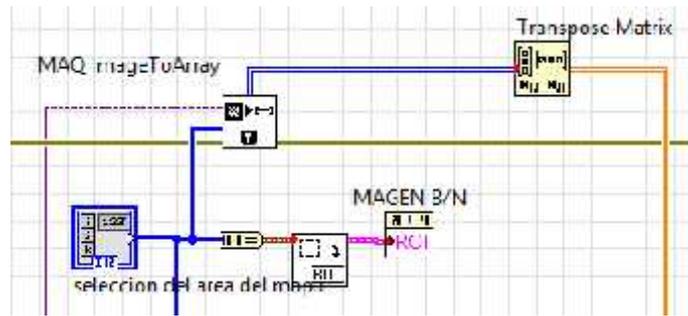
Fuente: Lopez J., Santillan G.

Esto se hace para facilitar el análisis de la imagen al solo tener que analizar dos colores blanco y negro con un alto nivel de contraste entre ambas que permite diferenciar objetos u obstáculos en el camino como se puede ver a continuación.

3.4.3.4.1. IMAQ IMAGETOARRAY

Es un proceso para cambiar el formato de la imagen de pixeles a un array.

Figura III 100. IMAQ ImageToArray



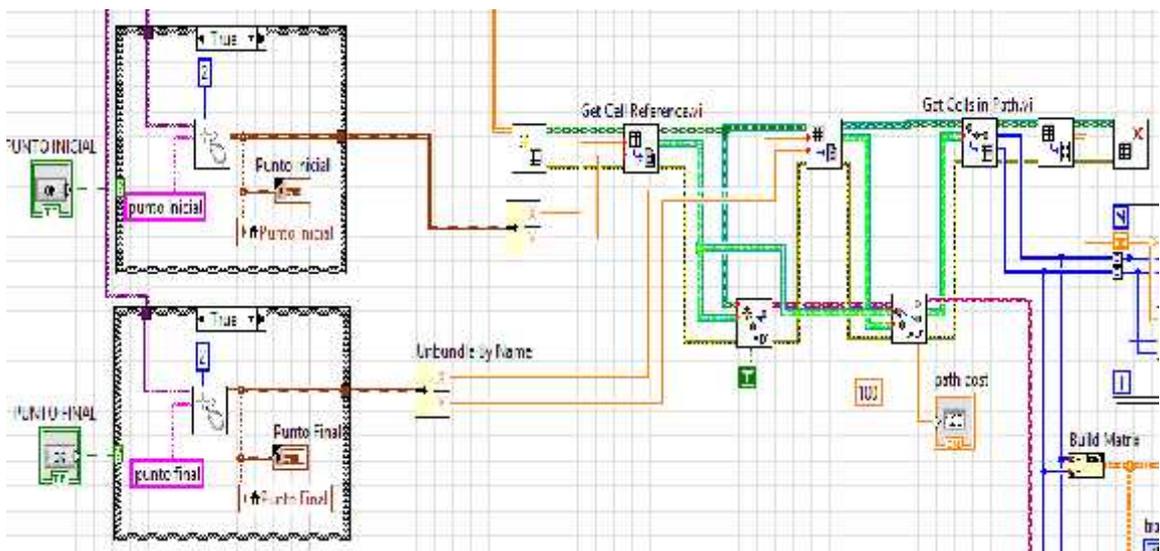
Fuente: Lopez J., Santillan G.

Toma la imagen que fue previamente reducida, filtrada e invertida de colores para transformarla en un formato de array de dos dimensiones (X, Y) que permitirá realizar los cálculos matemáticos para la obtención de la mejor trayectoria a seguir por el robot omnidireccional; además se tiene un bloque de selección del área del mapa donde se puede delimitar el espacio del área en la fotografía en donde se va mover el robot.

3.4.3.5. IMPLEMENTACIÓN DEL ALGORITMO A*

En la etapa de implementación del algoritmo A* se lleva a cabo procesos matemáticos para identificar el mejor camino desde un punto inicial hasta un punto final esquivando obstáculos que puedan haber en el camino; para lo cual se ingresa la fotografía tomada por las etapas previas y transformada a formato array.

Figura III 101. Implementación de algoritmo A*



Fuente: Autor

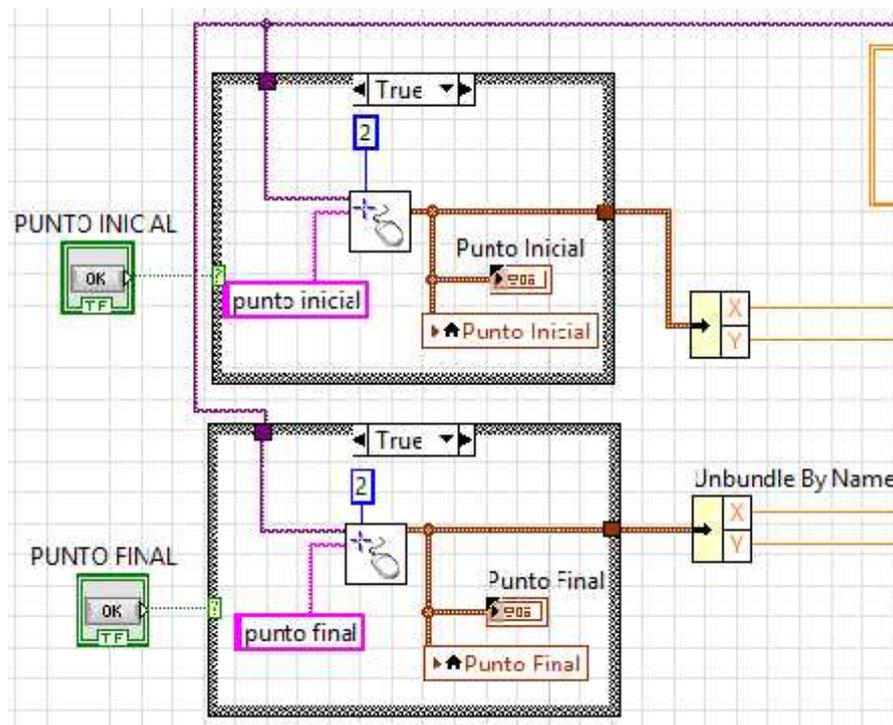
Entonces el programa ingresa la fotografía como mapa en el cual se visualiza la trayectoria de manera gráfica; y a la salida se obtendrá la trayectoria en formato de salida de puntos en el eje X y eje Y.

A continuación se describe cada bloque utilizado en esta etapa:

3.4.3.5.1. SELECCIÓN DE PUNTOS

En esta etapa se desarrolla bloques que permiten seleccionar el punto inicial y el punto final para el cálculo de la mejor trayectoria en el algoritmo A* estos son tomados de manera gráfica en la fotografía del entorno los datos son coordenadas geográficas (X, Y) que ingresan en el bloque de Get Cell Reference.

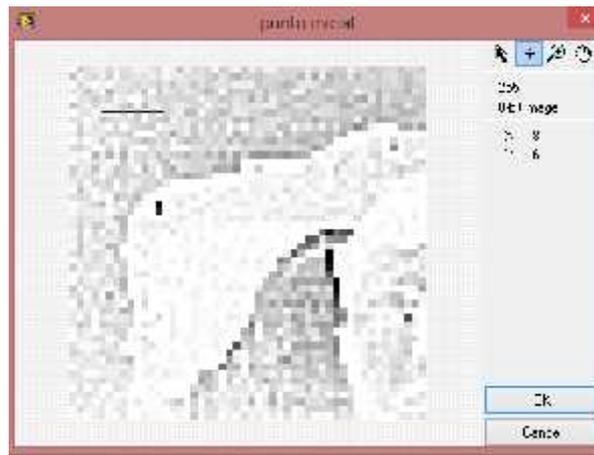
Figura III 102. Selección de puntos



Fuente: Lopez J., Santillan G.

Se utiliza un Case Structure condicional el cual mediante un botón en el panel frontal permite activar el bloque de selección de puntos.

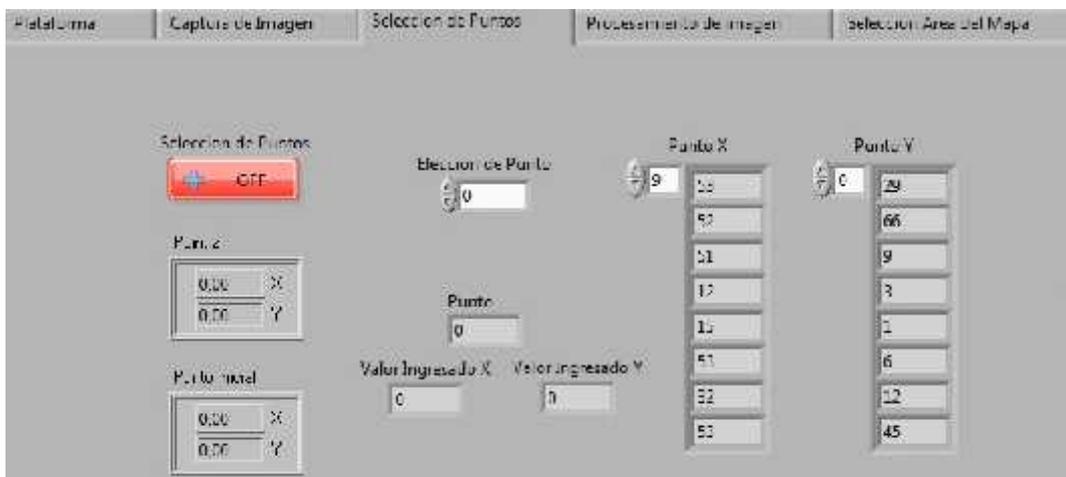
Figura III 103. Selección de punto inicial



Fuente: Lopez J., Santillan G.

Se puede observar los botones pulsadores en el panel frontal los cuales activaran la selección de puntos y su correspondiente en coordenadas (X, Y).

Figura III 104. Puntos de referencia (Panel Frontal)

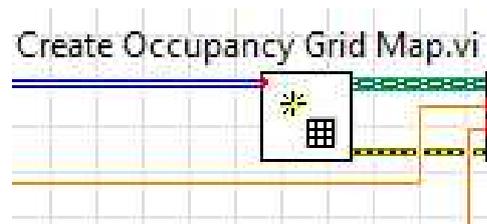


Fuente: Lopez J., Santillan G.

3.4.3.5.2. CREATE OCCUPANCY GRID MAPA

Genera un mapa en forma de una rejilla de variables, o células, que describen el entorno de robot. A la entrada tenemos la fotografía tratada y convertida en formato array que es representada por la línea azul; y a la salida tenemos la fotografía convertida en células que servirán para el cálculo del Algoritmo A* representada por la línea verde.

Figura III 105. Create Occupancy Grid Map

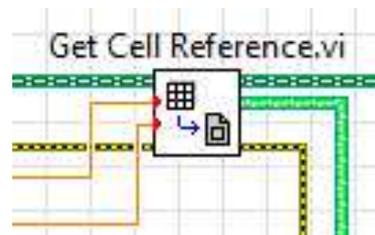


Fuente: Lopez J., Santillan G.

3.4.3.5.3. GET CELL REFERENCE

Devuelve una referencia a la celda en el mapa de la red de ocupación en la posición que especifique el usuario. En este caso se utiliza este bloque para especificar una posición específica dentro de la imagen permitiendo ingresar puntos de referencia; un punto inicial y final para la generación del camino; es en este bloque que ingresan los datos de los bloques de selección de la etapa anterior.

Figura III 106. Get Cell Reference



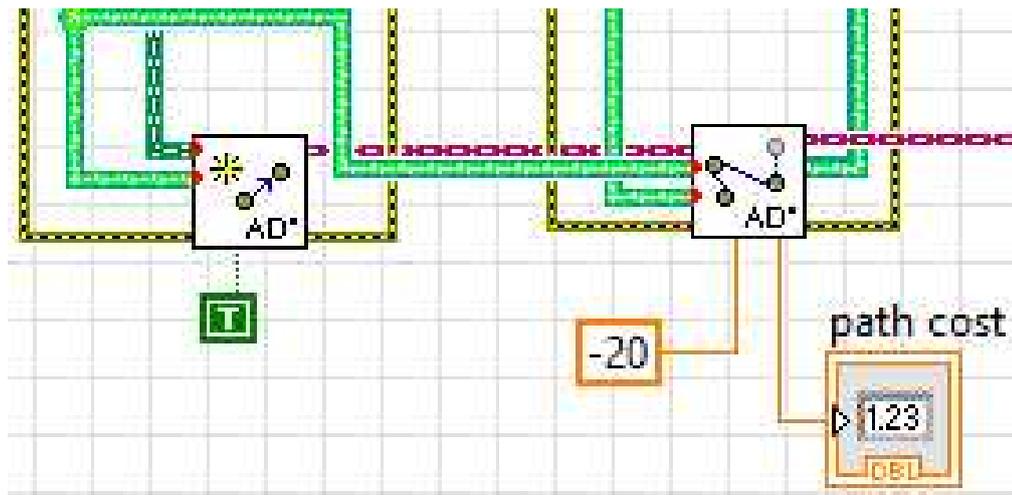
Fuente: Lopez J., Santillan G.

3.4.3.6. ALGORITMO A* DINÁMICO

El algoritmo AD* o algoritmo A dinámico es el encargado de calcular la mejor ruta o trayectoria mediante un costo; es decir se asigna un valor a cada celda, este es el algoritmo más usado y conocido de todos para encontrar el camino más corto entre un punto A y un punto B.

Mediante la aplicación del algoritmo A dinámico se accede a más memoria lo que permite almacenar una mayor cantidad de información y realizar los cálculos de manera más rápida; lo cual es necesario debido a que se toma fotografías constantemente para mantener el entorno del robot actualizado permitiendo que si llega a existir la presencia de un obstáculo el robot pueda esquivarlo y seguir con su trayectoria final.

Figura III 107. Algoritmo AD*



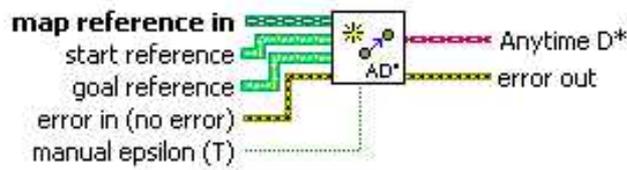
Fuente: Lopez J., Santillan G.

Usualmente el algoritmo A se utiliza de manera estática y mediante la aplicación de nodos a los cuales se les asigna un costo; para este proyecto se optó utilizar el algoritmo dinámico y la utilización de las celdas y que como se dijo anteriormente se actualizara el entorno constantemente mediante la adquisición de imágenes por la cámara web que tomara las imágenes de la mesa del sistema MPS superficie por la cual se moverá el robot omnidireccional; en la mesa se encuentra los pallets ubicados en los extremos de la misma los cuales entregan y reciben las piezas transportadas por el robot.

El algoritmo A se tratar de un algoritmo heurístico, ya que una de sus principales características es que hará uso de una función de evaluación heurística, mediante la cual etiquetará los diferentes nodos de la red y que servirá para determinar la probabilidad de dichos nodos de pertenecer al camino óptimo. El algoritmo A* utiliza una función de evaluación: $f(n) = g(n) + h'(n)$, donde $h'(n)$ representa el valor heurístico del nodo a evaluar desde el actual, n, hasta el final, y $g(n)$, el coste real del camino recorrido para llegar a dicho nodo, n, desde el nodo inicial.

Primero se inicializa una sesión de planificación de trayectoria con la que el AD* opera; hay que tener en cuenta que para la planificación de un camino óptimo puede requerirse de mucho tiempo para lo cual se coloca el valor de verdadero (TRUE) para manual epsilon con el fin de equilibrar el tiempo necesario para planificar con exactitud la ruta calculada.

Figura III 108. Initialize AD*



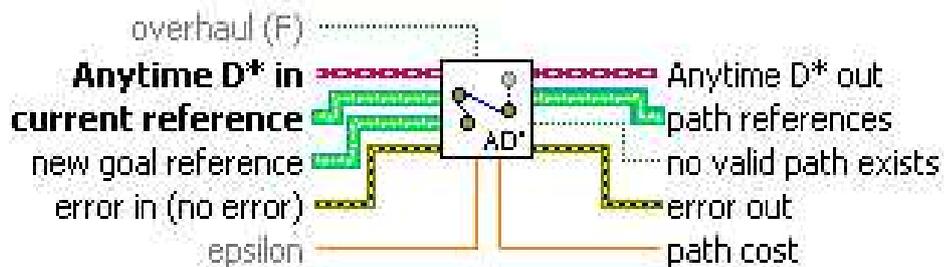
Fuente: Lopez J., Santillan G.

En este bloque se ingresa los siguientes datos:

- Map reference in: que es el mapa del entorno en formato de celdas; es decir la fotografía de la mesa que ha sido tratado por las diferentes etapas previas.
- Start reference: que es el punto inicial o punto de partida; este es el punto inicial para el cálculo de la mejor trayectoria. Esta información saldrá desde el primer Get Cell Reference de esta etapa.
- Goal reference: que es el punto final o punto de llegada; este es el punto final al cual debe llegar el robot.

Una vez inicializado y configurado los tiempos se ejecuta el algoritmo en si mediante el siguiente bloque:

Figura III 109. Bloque AD*



Fuente: Lopez J., Santillan G.

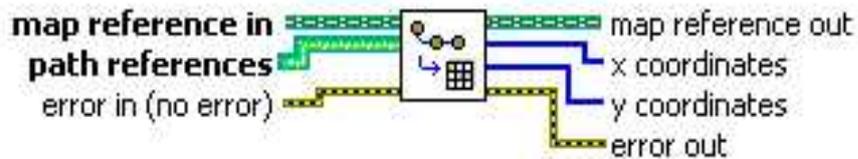
En este bloque se ingresa datos como la posición actual del robot; y se obtiene a la salida de path references un array de puntos de referencias sobre el mapa a lo largo del camino calculado.

Además se configura épsilon para regular la sensibilidad de colores de los objetos u obstáculos en el mapa.

3.4.3.6.1. GET CELLS IN PATH

Devuelve las coordenadas de la celda de todos los puntos a lo largo del camino calculado a través del mapa de la red de ocupación. En coordenadas cartesianas (X, Y).

Figura III 110. Get Cells in Path

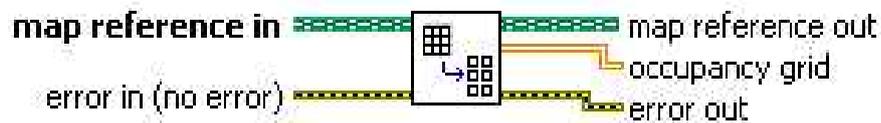


Fuente: Lopez J., Santillan G.

3.4.3.6.2. GET ALL OCCUPANCY GRID CELLS

Devuelve una matriz de las células en un mapa cuadrículado de ocupación.

Figura III 111. Get All Occupancy Grid Cells

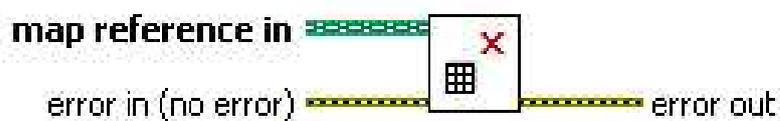


Fuente: Lopez J., Santillan G.

3.4.3.6.3. CLOSE OCCUPANCY GRID MAP

Cierra la sesión de la ocupación de grilla del mapa.

Figura III 112. Close Occupancy Grid Map

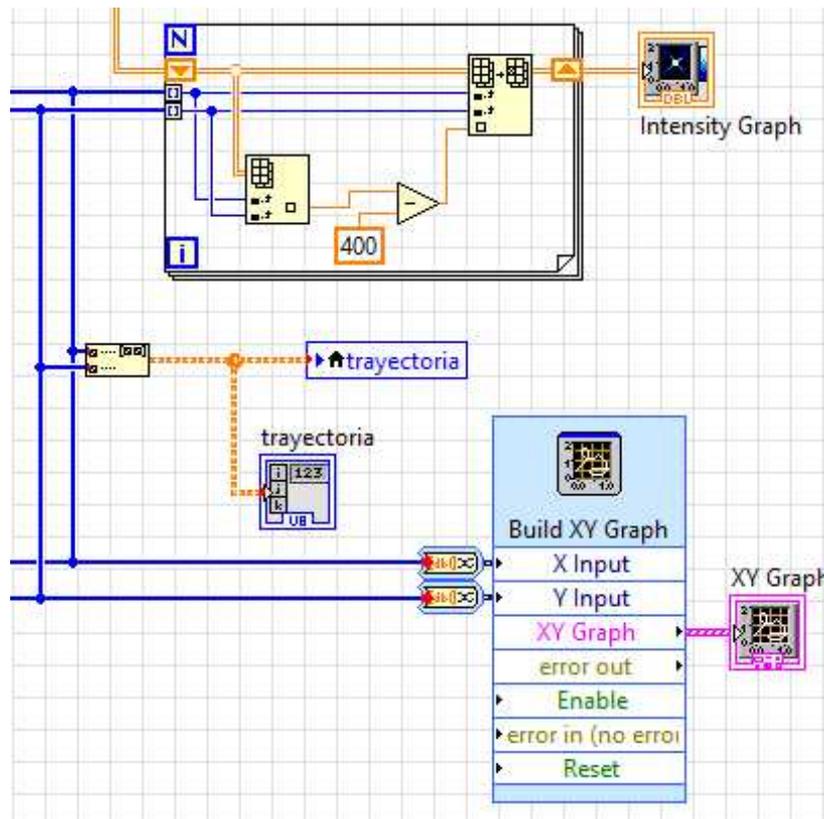


Fuente: Lopez J., Santillan G.

3.4.3.7. ADQUISICIÓN DE LA TRAYECTORIA

La obtención de la trayectoria se analiza en esta etapa se lo realiza mediante procesos matemáticos y algebraicos.

Figura III 113. Adquisición de la trayectoria



Fuente: Lopez J., Santillan G.

Se toma las coordenadas cartesianas (X, Y) desde el bloque Get Cells in Path representadas por las líneas azules en la figura III 110; que son las coordenadas de la trayectoria calculada que será la que tiene que seguir el robot en esta etapa se grafica la trayectoria a seguir sobre la fotografía del entorno físico de la mesa; además se crea y se visualiza un array de dos dimensiones con las coordenadas (X;Y) y por último se grafica solo la trayectoria para que pueda ser analizada de mejor manera.

Como se puede observar en el panel frontal; se obtiene la gráfica del mapa incluida la trayectoria en el bloque superior mediante la adaptación de las coordenadas cartesianas de la trayectoria más el array del mapa mediante el bloque Index Array Function y se envía a graficar en la función Intensity Graph.

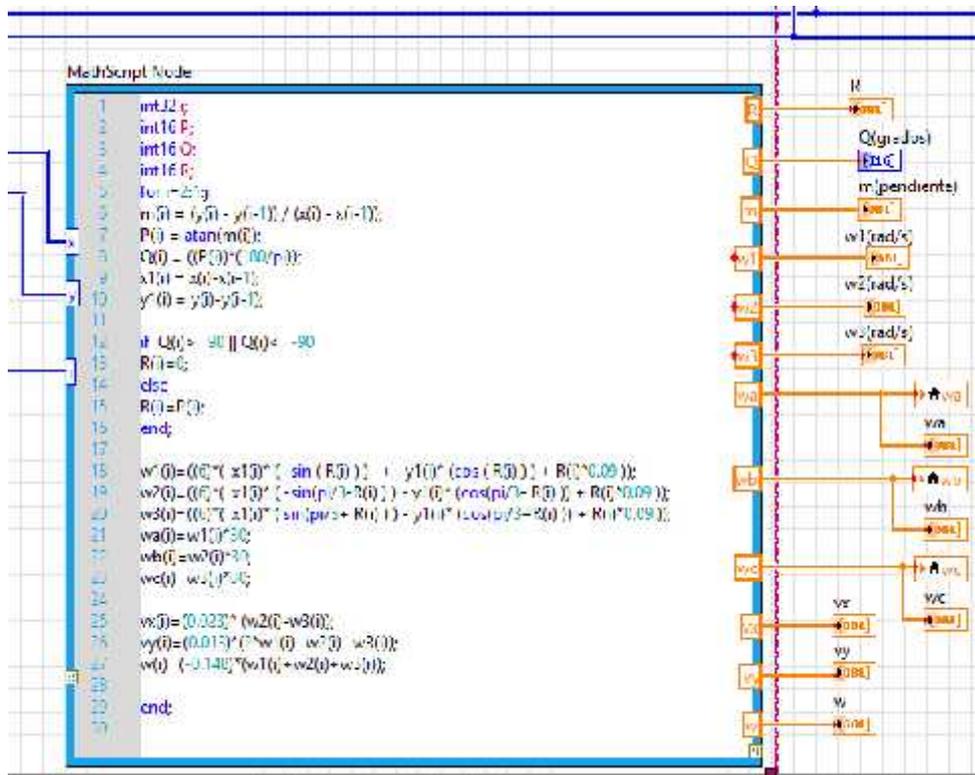
Utilizando la función Build Matrix se une las coordenadas X y las coordenadas Y representadas por las líneas azules que se encuentran de forma individual y se forma una matriz; la cual se visualiza y se almacena en la variable local con el nombre de trayectoria.

Y en el último bloque se grafica el camino obtenido mediante la función Build XY Graph para visualizar de manera gráfica la trayectoria en un plano X-Y.

3.4.3.8. INTEGRACIÓN DEL MODELO CINEMÁTICO

En la siguiente etapa se implementa el modelo matemático del robot omnidireccional utilizando lenguaje de programación de texto, con el objetivo de integrar la fórmulas matemáticas del modelo cinemático del robot las cuales definen el movimiento del mismo en un plano bidimensional X-Y, en este caso será la superficie de la mesa de trabajo; el movimiento estará basado en la trayectoria obtenida en la etapa anterior.

Figura III 114.Integración del modelo cinemático



Se ingresa las coordenadas de la trayectoria obtenida y el tamaño del array es decir la longitud del arreglo el cual representa el número de coordenadas a ser ingresadas y calculadas; en el nodo de programación Math Script están serán las variables de entrada, el código está basado en un lenguaje de programación de texto propio de Labview este bloque se encuentra en la librería Mathematics; en este nodo se ingresan las fórmulas de los modelos matemáticos analizados en los capítulos anteriores para el movimiento de un robot omnidireccional.

Como variables de salida tenemos la V_x (La velocidad lineal en X), V_y (La velocidad lineal en Y) y W_c (La velocidad angular de giro).

Y las velocidades angulares de cada una de las tres llantas omnidireccionales las que se encargaran de dar movimiento al robot, el resultado será un numero decimal que representa la velocidad en radianes por segundo puede ser positivo o negativo; el signo indicara el sentido de giro del motor.

Para el control de los motores tenemos las variables de salida w_1 , w_2 , w_3 ; las velocidades angulares de cada uno de los motores del robot omnidireccional.

Los valores de las velocidades de los motores se almacenaran en las variables locales para que puedan ser utilizadas con facilidad en otros procesos del programa.

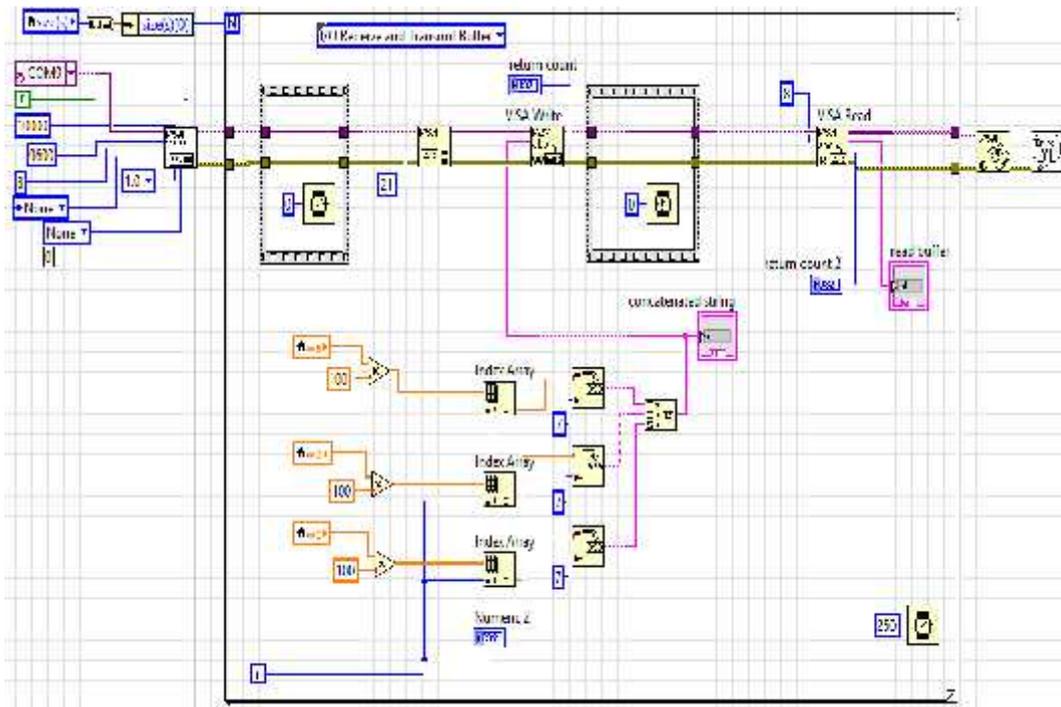
3.4.3.8. TRANSMISIÓN Y RECEPCIÓN DE INFORMACIÓN

Esta es la etapa en la cual se transmite los datos de manera serial e inalámbrica hacia el Arduino ubicado en el robot; el cual es el encargado de tomar esta información y convertirla en movimiento en las unidades de accionamiento, mediante la activación de los motores controlando la velocidad de estos con PWM y el sentido de giro cada uno de los tres motores incorporados en la estructura del robot mismo.

En este proceso se envía cada una de las velocidades angulares denominadas w_a , w_b , w_c , multiplicadas por 100 para convertirlas en valores enteros para facilitar el envío y recepción de estos datos.

El envío de los datos se logra a través del uso de la librería serial llamada VISA que utiliza puertos de comunicación serial (COM) del ordenador para permitir el envío y recepción de datos mediante un módulo inalámbrico acoplado al ordenador y un dispositivo bluetooth incorporado en el robot logrando una comunicación inalámbrica entre ambos.

Figura III 115. Etapa de transmisión y recepción de información



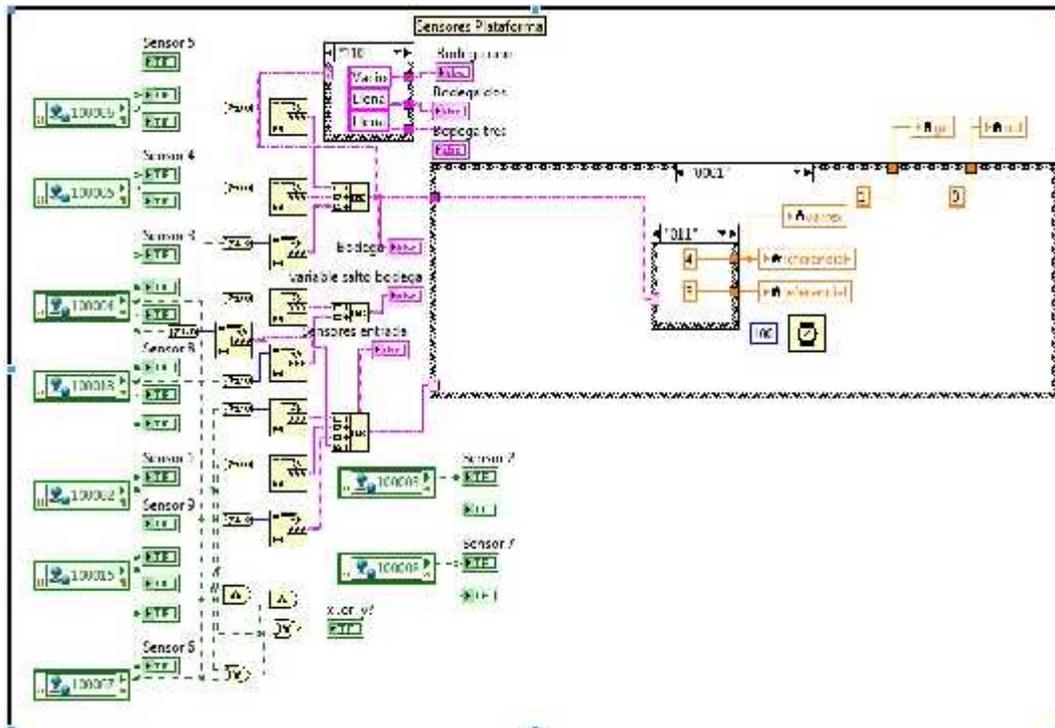
Fuente: Lopez J., Santillan G.

En el desarrollo esta etapa se hace uso de los bloques de escritura y lectura propios de la librería VISA; además de tener que agregar un bloque de asignación de buffer para almacenar y evitar errores con los datos recibidos; además hay que activar el modo síncrono en los bloques de escritura y lectura para que no ocurran problemas de conexión y de comunicación con los drivers del ordenador de esta manera se logra una comunicación óptima.

Los datos se transmiten uno a uno en cada iteración para lo cual se asigna un tiempo de 320 milisegundos que es lo que tomara al robot recorrer la distancia de un cuadro o pixel de la foto.

3.4.3.9. INDICACIÓN DE PUNTOS FINALES DE TRAYECTORIA AL ROBOT PALETIZADOR MEDIANTE PLATAFORMA CONTROLADA POR PLC.

Figura III 116. Sensores Plataforma controlada por PLC



Fuente: Lopez J., Santillan G.

Para indicar los puntos a los que se debe dirigir el robot paletizador, usamos el algoritmo propuesto anteriormente. Con este fin, empleamos la conexión realizada mediante modbus Ethernet con el PLC twido, recibiendo las señales de presencia de palet, para así indicar al robot paletizador la dirección donde debe ser entrega o recibido el palet.

El control se realizó mediante estructura case, evitando así la redundancia de órdenes, y el retardo de procesos, al ser usado la estructura if.

Los datos se transformaron de tipo boolean a tipo string para poder unirlos en una sola cadena, para compararlas con posiciones que se ingresaron anteriormente, realizando el control de una manera más eficiente.

CAPÍTULO IV

4. PRUEBAS Y RESULTADOS

4.1. INTRODUCCIÓN

Las pruebas estuvieron orientadas a determinar las funcionalidades, así como la precisión con que realiza las acciones, tanto como de paletizado y orientación del robot respecto al plano, tomando en cuenta que los procesos son realizados de manera automática, sobre la plataforma controlada por PLC.

Aquí se detallan las pruebas y resultados obtenidos del robot paletizador, para comprobar los resultados que se derivan del presente trabajo de titulación.

4.2. ANÁLISIS DEL ROBOT

Nuestro robot se clasifica como un robot industrial; debido a que se podría considerar como un manipulador multifuncional reprogramable, capaz de mover materias, piezas según trayectorias variables.

Según la AFRI; el robot se puede considerar dentro del tipo D, es decir un robot capaz de adquirir datos de su entorno, readaptando su tarea en función de estos.

El robot posee 4 articulaciones, tres articulaciones rotativas y una articulación de tornillo.

Se denomina grado de libertad a cada uno de los, movimientos independientes que puede realizar cada articulación con respecto a la anterior.

Para el movimiento del robot se optó realizarlo mediante trayectorias punto a punto, es decir que cada articulación evoluciona desde su posición inicial a la final sin realizar consideración alguna sobre el estado o evolución de las demás articulaciones; además utilizando un movimiento simultáneo de ejes, es decir las articulaciones del robot se mueven una o varias simultáneamente y cada una a velocidades específicas

En cuanto a la programación de un robot consiste en indicar paso a paso las diferentes acciones que deberá realizar durante su funcionamiento automático; para el desarrollo de este robot se utilizó programación textual de nivel robot, especificando cada uno de los movimientos que tiene que realizar el robot y descomponiendo la tarea global de transporte de materiales en varias subtareas:

1. El seguimiento de la trayectoria.
2. Giro en búsqueda de marcas.
3. Seguidor de línea.
4. Cuadrarse.
5. Accionamiento del actuador lineal.
6. Posicionamiento con giroscopio.

En cuanto a los tipos de datos que se utilizan son las coordenadas cartesianas (X-Y) que definirán la trayectoria a seguir calculada previamente, debido a que el robot se moverá sobre una superficie plana no posee eje Z.

El problema cinemático directo se reduce a encontrar una matriz de transformación que relaciona el sistema de coordenadas ligado al cuerpo con el sistema de coordenadas de referencia.

Se conoce que se debe utilizar matrices de transformación homogénea, sin embargo existe un problema con este método ya que se debería definir un total de 11 sistemas de coordenadas, de forma que se obtienen matrices demasiado grandes para trabajar con normalidad, por lo que este método se define como poco adecuado. Por lo tanto se utiliza como alternativa una recursividad cinemática que forma parte de la formulación dinámica de Newton-Euler, para obtener las relaciones cinemáticas de una forma mucho más sencilla.

Por la dificultad que se presentan en los robots móviles muchos autores emplean soluciones cinemáticas particularizadas para cada configuración específica de vehículos, dependiendo del número de llantas y del tipo.

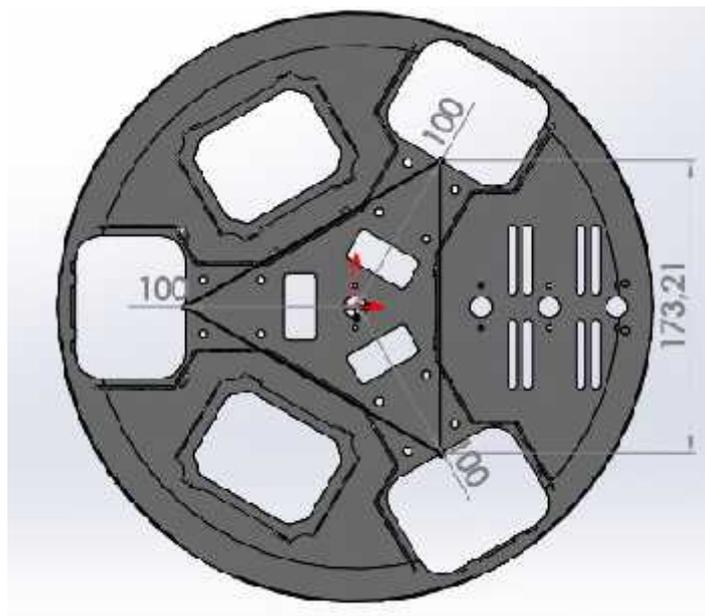
En definitiva, de forma independiente al tipo de rueda empleado, la cinemática directa tiene como objetivo el cálculo de la velocidad lineal y angular del robot a partir de las correspondientes aportaciones de cada una de sus ruedas. Y la cinemática inversa permite conocer a qué velocidad angular y el sentido de giro que debe tener cada una de las ruedas con respecto a las otras.

4.3. PRUEBAS MECÁNICAS

La estabilidad del robot paletizador está basada en el centro de gravedad del triángulo equilátero, que son los puntos de apoyo del sistema, que balancean la carga, ubicándola en el centro del robot, para así obtener un movimiento uniforme y lo más cercano a ser preciso con respecto al plano.

Al ser un triángulo equilátero el baricentro es el centro de gravedad del sistema, siendo este el punto medio del robot paletizador.

Figura IV 117. Configuración Omnidireccional



Fuente: Lopez J., Santillan G.

El material usado, nos permitió crear un prototipo estable, ya que el aluminio compuesto tiene la suficiente resistencia para soportar el peso tanto de la circuitería como de las baterías y el actuador lineal.

De las pruebas realizadas pudimos confirmar, que la estructura del robot está nivelada, en sus tres ejes del triángulo, en donde se ubica todo el peso que debe cargar el robot paletizador, siendo la plataforma al ser demasiada blanda un impedimento para que este, se estabilice de manera correcta.

Para esto se realizó pruebas en diferentes tipos de superficies, para así comprobar la eficiencia del modelo sobre la plataforma y otras superficies.

Tabla IV VIII. Comparación de Superficies

| Superficie | Porcentaje de Precisión |
|--------------------|-------------------------|
| Aluminio Compuesto | 60% |
| Cerámica | 75% |
| Parquet | 80% |
| Triplex 9 MM | 95% |

Fuente: Lopez J., Santillan G.

De las pruebas realizadas se determinó, que la superficie de la plataforma no es lo suficientemente rígida, para que el robot paletizador cumpla sus funciones a cabalidad, siendo este el motivo del cambio de superficie de la plataforma de aluminio compuesta a madera triplex de 9 mm.

4.4. PRUEBAS DE ADQUISICIÓN DE IMÁGENES

Para poder realizar las pruebas de adquisición de imagen, se tomaron en cuenta tres aspectos importantes.

La distancia focal, el ángulo, y la resolución de la cámara utilizada.

Tomando en cuenta estas consideraciones, se calibro la imagen a un ángulo en picado, para poder aumentar la profundidad de la imagen, así obviando la altura del techo que no permitió tomar una imagen perpendicular del plano.

Figura IV 118. Captura de Fotografía de la Plataforma



Fuente: Lopez J., Santillan G.

La distancia focal se corrigió ampliando la resolución de la cámara usada, así con una cámara de 1.3 Mpx se obtuvo un xx porcentaje de la plataforma, siendo inservible para poder realizar el procesamiento de imágenes de una manera óptima.

El mapa usado debe ser perpendicular al plano de una resolución óptima para poder obtener una trayectoria correcta, y así llegar a los puntos definidos.

En la siguiente tabla se ilustra el error al momento de ubicar los puntos, donde el robot paletizador tendrá que seguir la ruta.

4.5. PRUEBA Y CALIBRACIÓN DE SENSORES

Los sensores ultrasónicos, fueron calibrados para poder evitar posibles colisiones con objetos ubicados en la plataforma controlada, e incluso con el espacio circundante en ella.

Teniendo en cuenta la estructura física y el espacio requerido, se ubicaron los 9 sensores, a lo largo de la estructura externa., calibrándolos así a una distancia variable recurrente a nuestras necesidades.

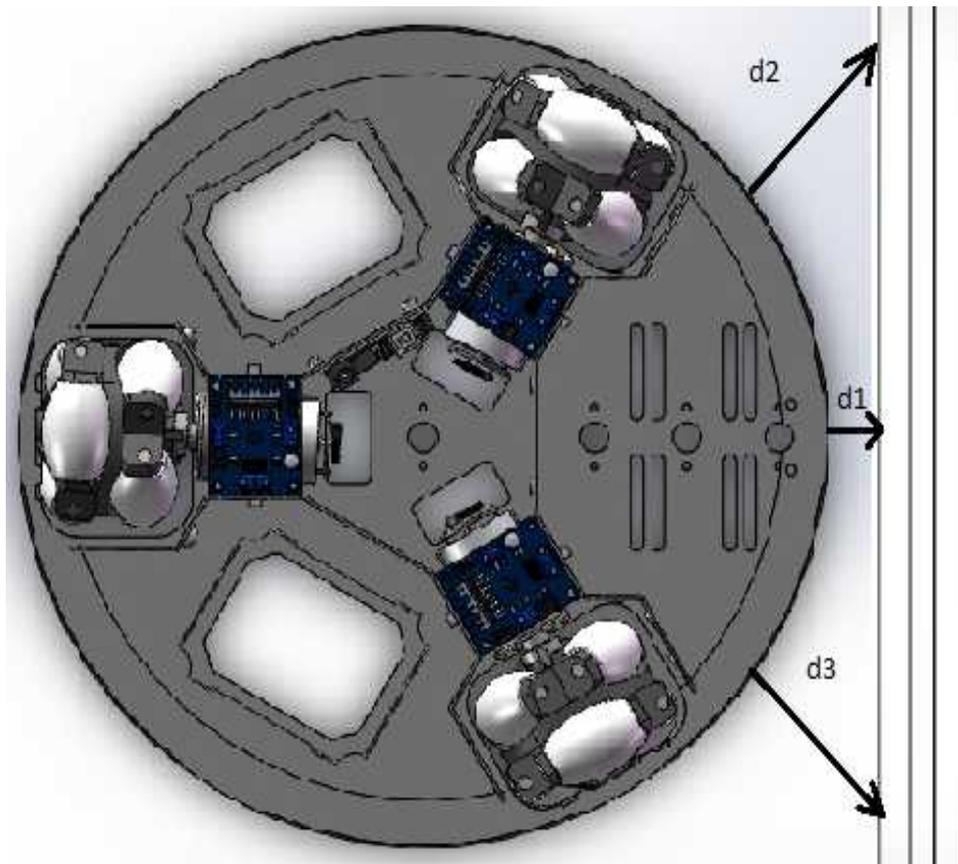
Para realizar la secuencia de seguidor, en un inicio se utilizó, un sensor magnético que gracias a sus características, nos ayudó en un principio a poder realizar las pruebas correspondientes en el proceso seguidor de línea, que se usó en conjunto con cinta bandit de acero inoxidable, de espesura de 1mm.

Al momento de calibrar el sensor se tomó en cuenta, la altura del robot paletizador con respecto a la mesa, ya que el material donde el robot está circulando es de aluminio compuesto, el cual fue detectado por el sensor inductivo provocando errores en la programación, y al ser este material demasiado maleable provocaba ondulaciones que forzaban la des calibración del sensor inductivo.

Se usó de remplazo un sensor de color, que discrimina el color negro, usándolo así para seguir el camino de entrada a la recolección del palet.

Al usar el proceso de seguidor de línea se usó un algoritmo combinado que usa el proceso del seguidor en conjunto con la lectura de los sensores Sharp frontales, que permiten la triangulación de la posición respecto a un eje de referencia, para así levantar y recoger el palet con precisión.

Figura IV 119. Configuración Seguidor de Línea



Fuente: Lopez J., Santillan G.

El sensor d1 medirá la distancia frontal del robot paletizador hacia el perfil de aluminio para acercarse a recoger el palet en la estantería, mientras que los sensores d2 y d3, calculan la distancia para centrar el robot paralelamente al perfil de aluminio.

4.6. PRUEBAS DE CONSUMO DE CORRIENTE

Para estimar el consumo de corriente y duración de baterías se evaluó cada uno de los componentes, siendo estos sensores y actuadores, revisando sus características técnicas y el consumo en vacío de los motores con su caja reductora.

Cabe recalcar que la alimentación de los motores es independiente al circuito principal, solo compartiendo el pin de GND (tierra común) para así cerrar el circuito de una manera correcta.

Enumerando cada uno de ellos se obtuvo la siguiente tabla, que especifica los consumos de corriente

Tabla IV IX. Censo de carga

| Cantidad | Descripción | Consumo Unitario | Consumo Total | Unidad |
|----------|--------------|------------------|---------------|--------|
| 9 | Sensor Sharp | 15 | 315 | mA |
| 1 | Sensor Color | 200 | 200 | mA |
| 1 | Arduino Mega | 40 | 40 | mA |
| 1 | MPU-6050 | 40 | 40 | mA |
| 1 | Bluethoth | 40 | 40 | mA |
| 3 | Encoder | 10 | 30 | mA |
| 3 | L298N | 36 | 108 | mA |
| | | Total | 773 | mA |

Fuente: Lopez J., Santillan G.

Para calcular la autonomía de nuestro circuito utilizamos la formula y el coeficiente de Peukert que nos da un resultado:

$$T = \frac{H}{\left(\frac{I}{C}\right)^K}$$

Dónde:

T: es el tiempo de horas de autonomía del robot

C: es la capacidad de la batería en amperios/hora

H: es el tiempo que está calculado la capacidad nominal, es decir el amperaje que puede entregar durante determinado tiempo

I: es la intensidad de corriente que consume nuestro circuito

K: es el exponente de Peukert, generalmente siendo este de 1.3 para baterías de plomo acido

Sabiendo que las baterías usadas son de plomo-acido recargables con una carga máxima de 12 V a 4.5 A, tendremos el resultado de autonomía.

$$T = \frac{20}{\left(\frac{0.773 \cdot 20}{4.5}\right) \cdot 1.1}$$

$$T = 4.02$$

Siendo el tiempo de autonomía de nuestro circuito principal de 4.02 horas.

4.7. PRUEBAS DE SOFTWARE DE CONTROL

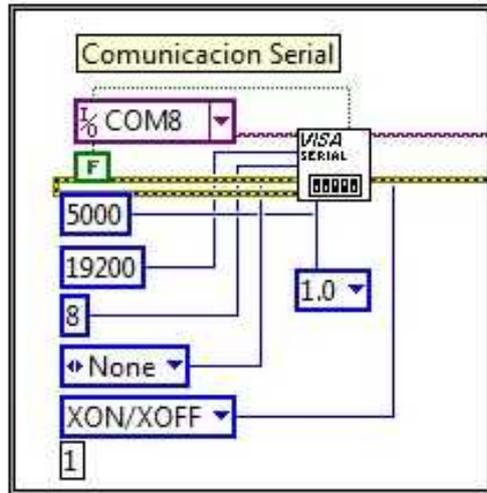
En nuestro software de control se tuvieron que implementar parámetros para establecer la comunicación entre el robot paletizador y Labview, así como también la concatenación de procesos entre Labview y la plataforma controlada por PLC.

Considerando que el robot paletizador realizara procesos de manera autónoma, se programó la plataforma, para que realice las acciones con temporizadores y contadores para así darnos flexibilidad en los tiempos que maneja el sistema.

En tanto que la comunicación vía Bluetooth del robot paletizador se configuro los siguientes parámetros:

- Velocidad de Transmisión.
- Nombre del Bluetooth y Password
- Numero de puerto serial

Figura IV 120. Parámetros de configuración del serial



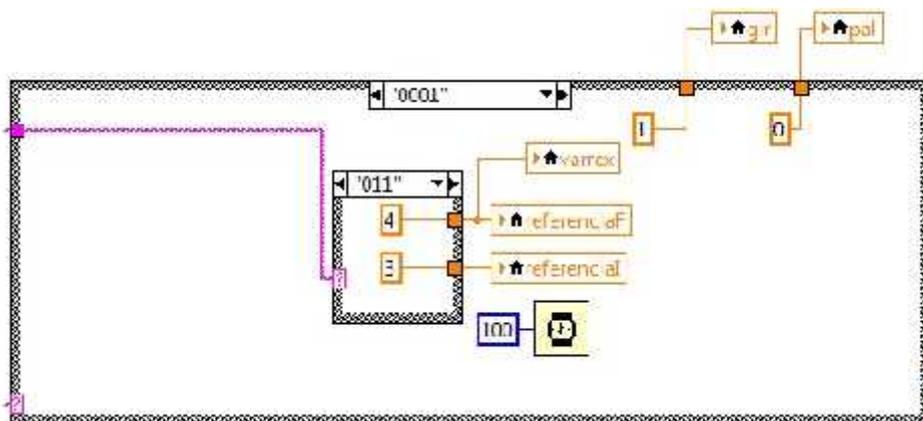
Fuente: Lopez J., Santillan G.

Las opciones de configuración son variadas, siendo las esenciales para la comunicación la enunciada anteriormente.

La velocidad de transmisión, se estableció a 19200, siendo uno de los principales motivos, el MPU-6065 (Giroscopio), que trabaja a la velocidad de transmisión antes expuesta.

En la misma línea de pruebas se realizó las pruebas de control del actuador lineal, sabiendo que la programación original se centra en Arduino, y que el control de giro, ya sea positivo o negativo se realiza desde Labview, enviando un cero o un uno para activar el giro positivo o negativo según sea el caso.

Figura IV 121. Control de Trayectoria



Fuente: Lopez J., Santillan G.

El actuador lineal, al estar constituido por un motor paso a paso y un tornillo sin fin de 20 cm con barras laterales, fue controlado mediante la secuencia full step, cuya librería esta en Arduino, sabiendo que bajo condiciones ideales, la secuencia de rotación del motor en vacío y sin carga, es precisa.

4.8. PRUEBAS DE SEGUIMIENTO DE TRAYECTORIAS

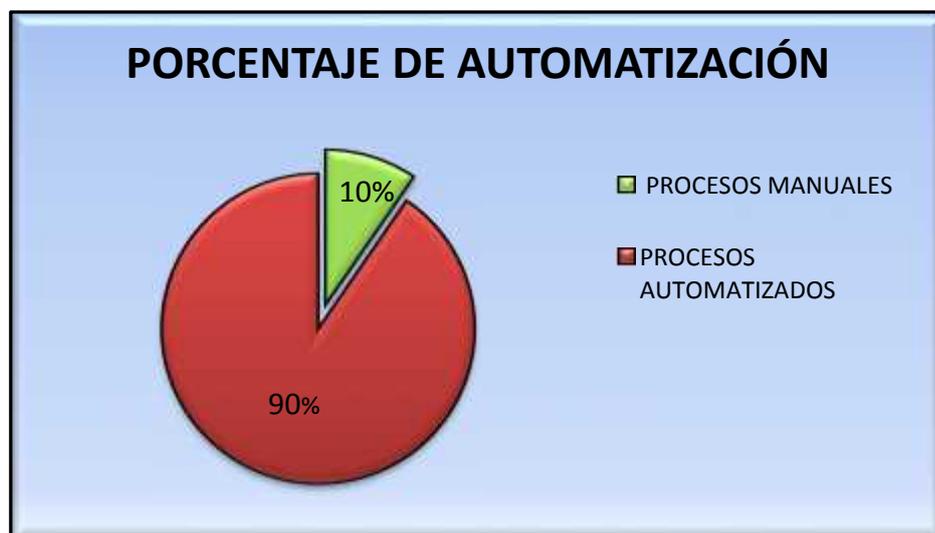
Las pruebas de seguimiento de trayectorias se realizaron de manera manual, para verificar el correcto posicionamiento del robot al finalizar el proceso, verificando el correcto funcionamiento del software de control y de los actuadores, que funcionan en conjunto al modelo cinemático inverso usado anteriormente.

Se determinó con las pruebas realizadas que ocasionalmente, el robot paletizador tiende a desfasarse en el plano, así verificando que se trata de imperfecciones de la plataforma, al cruzar entre lugares que no están firmes en su totalidad.

4.9. PORCENTAJE DE AUTOMATIZACIÓN

El sistema de transporte de materiales consta de dos partes que son la mesa y el robot, los cuales interactúan juntos para lograr el objetivo deseado, para lo cual se automatizo la mayoría de los procesos que intervienen en estas dos componentes.

Figura II 122. Porcentaje de automatización de los procesos del MPS



Fuente: Lopez J., Santillan G.

Tabla IV X. Procesos Automatizados

| N° | PROCESOS | TIPO DE PROCESOS | |
|--------------------------|--|------------------|---------------|
| | | MANUALES | AUTOMATIZADOS |
| 1 | Colocación de palets | x | |
| 2 | Comunicación entre PLC y PC | | X |
| 3 | Comunicación entre PC y robot | | X |
| 4 | Detección de palets | | X |
| 5 | Accionamiento de válvulas | | X |
| 6 | Temporizado de rampas | | X |
| 7 | Proceso llenado de bodega | | X |
| 8 | Toma de fotografías para generar mapa | | X |
| 9 | Selección de puntos para trayectoria | x | |
| 10 | Calculo de la trayectoria | | X |
| 11 | Envío de la trayectoria desde la pc al robot | | X |
| 12 | Accionamiento de los motores | | X |
| 13 | Seguimiento de trayectoria | | X |
| 14 | Rastreo de obstáculos a 360° | | X |
| 15 | Giro del robot para búsqueda de marcas | | X |
| 16 | Reconocimiento de marcas | | X |
| 17 | Activación del proceso seguidor de línea | | X |
| 18 | Activación del proceso de cuadrarse | | X |
| 19 | Accionamiento del actuador lineal | | X |
| 20 | Posicionamiento del robot con el eje de referencia mediante giroscopio | | X |
| 21 | Re inicialización del robot | | X |
| TOTAL DE PROCESOS | | 2 | 19 |

Fuente: Lopez J., Santillan G.

4.10. PRUEBA DE HIPÓTESIS

Al finalizar nuestro trabajo de titulación podemos realizar nuestro análisis de nuestra hipótesis planteado, teniendo como resultado las siguientes

- *Hi*: La incorporación del robot móvil paletizador al sistema MPS, mejorará el índice de automatización del sistema.
- *Ho*: La incorporación del robot móvil paletizador al sistema MPS, no mejorará el índice de automatización del sistema.

El nivel de significancia tomada fue del 5%, siendo obtenido en la tabla Chi cuadrado un valor de 5.991

Mencionado en estudios anteriores se pudo obtener que el índice de automatización fue de 42.85 que nos fue útil para contrastar los resultados de nuestro trabajo de titulación.

$$\text{Grados de Libertad} = (\text{fila} - 1)(\text{columna} - 1) = 1$$

Tablas de Frecuencias Esperadas

$$= \frac{\text{Total Columna(Para dicha Celda)} * \text{Total Fila (Par dicha Celda)}}{\text{Suma Total}}$$

$$\chi^2_{calc} = \sum \frac{(f_o - f_e)^2}{f_e}$$

Tabla IV XII. Tabla Chi Cuadrado

| | <i>Fo</i> | <i>Fe</i> | $(fo-fe)^2/fe$ |
|-------------------------------|-----------|-----------|----------------|
| Procesos Automatizados | 19 | 14 | 1,785714286 |
| | 2 | 7 | 3,571428571 |
| Procesos Manuales | 9 | 14 | 1,785714286 |
| | 12 | 7 | 3,571428571 |
| | | | 10,71428571 |

Fuente: Lopez J., Santillan G.

Realizando los cálculos respectivos se obtuvo que tengamos un Chi cuadrado de 10.714286, siendo este mucho mayor a nuestro valor de insignificancia, podemos aceptar la hipótesis *Hi*, siendo nuestro trabajo de titulación una herramienta útil para incrementar el índice de automatización del sistema MPS.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- El robot móvil omnidireccional fue diseñado y construido para acoplarse a un sistema de producción modular de paletizado cumpliendo con la función de transportar materiales de una estación a otra a lo largo de una mesa de trabajo de 2.4x2.4 metros; para lo cual desarrollamos un software dedicado a la toma de imágenes y tratado de las mismas; mediante la plataforma de LabVIEW con el fin de generar un mapa o entorno del robot.
- Usando el software de programación Twidosoft, se desarrolló un programa que automatice el robot paletizador, para así lograr la integración al sistema MPS.
- Se seleccionaron y acoplaron sensores internos en el robot para crear un sistema móvil autónomo capaz de realizar determinadas actividades sin la intervención de un operador; para lo cual se diseñó un software que permite acoplar y controlar dichos sensores, además de actuadores del robot utilizando la plataforma de programación de Arduino.
- Para cumplir con los requerimientos del movilización y localización del robot se combinaron diferentes métodos como la estimación explícita de la posición, sistemas odométricos, y estimación mediante percepción del entorno combinándolos con el algoritmo A* dinámico además de la cartografía; para generar un sistema que sea capaz de calcular y seguir la mejor trayectoria siendo además capaz de interactuar con el entorno que lo rodea.
- El acople e implementación del robot móvil en la mesa de MPS para el transporte de palets de una estación a otra de forma autónoma y precisa fue exitoso.
- La utilización del modelo cinemático del robot móvil omnidireccional logro un correcto funcionamiento del mismo mediante modelos matemáticos que permitieron controlar el giro de cada llanta dotando al robot de la capacidad de moverse en cualquier dirección de forma instantánea y en forma omnidireccional permitiendo ahorrar tiempo.

RECOMENDACIONES

- Hay que considerar como limitaciones que el robot debe moverse en una superficie plana y sin imperfecciones
- No deben existir elementos flexibles en la estructura del robot.
- El peso del robot debe estar equilibrado de tal manera que el centro de masa coincida con el centro del robot para que no existan.
- Las baterías del robot deberán ser recargadas regularmente, así evitando errores en la comunicación y la pérdida de datos.
- Incentivar a la utilización del sistema, para el aprendizaje de recursos necesarios en la vida profesional.
- Usar el robot con fines de competencia, gracias a sus funcionalidades tienen un alto grado de aplicabilidad tanto en categoría libre, como en seguidor de línea y futbolista en diversas competencias.

RESUMEN

Se diseñó un robot paletizador para el transporte de materiales de un Sistema de Producción Modular, para el laboratorio de Redes Industriales de la Escuela de Ingeniería Electrónica en Control y Redes Industriales en la Escuela Superior Politécnica de Chimborazo.

El Robot paletizador fue diseñado en base a estándares previamente obtenidos en estudios anteriores, movilizándose así sobre la plataforma controlada por el Controlador Lógico Programable, siendo éste, diseñado en el software de diseño 3D, SOLID WORKS, permitiendo visualizar de una manera precisa las dimensiones correctas, y la colocación de los componentes, que se forman el presente trabajo de titulación.

El método usado fue el Investigativo, ayudándonos a hallar una manera estable para realizar el control y posicionamiento del robot, así usando la odometría y cartografía, como procesos que nos ayudan a ubicar dicho robot en el plano X-Y, todo esto realizado en el software de Programación Labview. Usando el aluminio compuesto, como material base, para crear un modelo estable del robot, para que transporte palets en el plano.

Usamos el Algoritmo A, que realiza el proceso de detección de ocupación en el mapa y de búsqueda del mejor camino, o de la mejor trayectoria en base a los costos.

El presente trabajo fue diseñado para incrementar el índice de automatización del Sistema de Producción Modular, teniendo en consideración un índice de automatización anterior del 45% obtenido de estudios anteriores, siendo éste elevado a un 95% de automatismo, cuando se implementó el robot paletizador al sistema.

Concluimos que usando nuevas tecnologías, se puede llegar a automatizar diferentes sistemas en varios niveles, siendo el único impedimento el factor económico al momento de implementar el Sistema de Producción Modular

Recomendamos a los estudiantes, usuarios del robot móvil paletizador, tener el debido cuidado y mantenimiento al robot, evitando así posibles daños al sistema.

PALABRAS CLAVE: <ROBOT PALETIZADOR> <ALGORITMO A> <CONTROLADOR LÓGICO PROGRAMABLE> <POSICIONAMIENTO DEL ROBOT> <TRANSPORTE DE PALET> <CARTOGRAFÍA> <AUTOMATIZACIÓN> < ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES>

SUMMARY

It was designed a palletising robot for the material transport of a Production Modular system, for the Industrial Networks Lab of Electronic Engineering in Control and Industrial Networks in the ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO (higher education).

The palletising robot was designed in basis to standard previously obtained in previous studies, mobilising over the controller platform for the programmable logic controller, designed in the design software 3D, SOLID WORKS, allowing to display of exact way the correct dimensions, and the location of components, that form the present job of titulation.

The used method was the research, helping to find a stable way in order to realice the control and positioning robot, using the odometry and cartography, like processes that help to locate this robot in the X-Y matrix, all of this made in the programming software LABVIEW. Using the compound aluminium, like basic material, in order to create a stable model of robot, for the pallets transportation over a flat surface.

It were used algorithm A, thah makes the detection process of occupation, in the map and search the best way, or the best haul in basis to the cost.

The current work was designed to increase the index of automatization of Modular Production System, keeping in mind a previous indes of automatization of 45% obtained of previous studies, being raised to 90% of automatism, when the palletising robot was implemented to the system.

It is concluded that using new technologies, it is posible to automate different systems in various levels, being the unique obstacle, the economical factor at the momento of implmente the Modular Production System.

It is recommended to the students, users of mobile palletising robot, to have appropriate care and maintenance to the robot, avoiding failures to the system.

KEYWORDS: <PALLETISING ROBOT> <ALGORITHM A> <PROGRAMMABLE LOGIC CONTROLLER> <POSITIONING ROBOT> <PALLET TRANSPORTATION> <CARTOGRAPHY> <AUTOMATIZATION> <ELECTRONIC ENGENEERING SCHOOL IN CONTROL AND INDUSTRIAL NETWORKS>

BIBLIOGRAFÍA

- [1] **ALONSO, J.** Tecnologías de la Información y de la Comunicación, 8a Ed., Barcelona-España., Ra-Ma, 2010.,pp 20-30
- [2] **BARRIENTOS, A.** Fundamentos de robótica, 4ta ed., Madrid-España., Mcgraw Hill., 2007.,pp 80-110.
- [3] **DORF, R. C. y BISHOP, R. H.** Sistemas de control moderno. , Traducido del ingles por Sebastian Dormido Canto y Raquel Dormido Canto. ,10a ed., Madrid-España, Pearson Educación., 2005., pp 1-24.
- [4] **KALPAKJIAN, S. y SCHMID S. R.** Manufactura, ingeniería y tecnología. , traducido por Jaime Espinosa Limón., 5a ed., México-México, Pearson Educación., 2008., pp 1085-1236.
- [5] **CLASIFICACIÓN DE ROBOTS**
<http://www.adripx.es/9.html>.
12-03-2014.
- [6] **CICLO DE TRABAJO**
<http://www.industriaynegocios.cl/Academicos/AlexanderBorger/Docts%20Docencia/Seminario%20de%20Aut/trabajos/vespertino%202005/PLC/documentos/CICLO%20DE%20TRABAJO%20DE%20UN%20PLC.htm>.
11-01-2015.
- [7] **CONTROLADOR LÓGICO PROGRAMABLE (PLC)**
datateca.unad.edu.co/contenidos/299013/Manual061ControladorLgicoProgramablePLC.pdf.
07-12-2014.

- [8] **DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO PARA LA SIMULACIÓN**
<http://dSPACE.esPOCH.edu.ec/bitstream/123456789/2817/1/108T0040.pdf>.
11-01-2015.
- [9] **GRAFCET**
http://www.uclm.es/profesorado/rcarcelen_plc/Prog3.htm
11-01 2015.
- [10] **ROBOTINO**
<http://goo.gl/pSQRI4>
21-01-2014.
- [11] **ROBOT OMNIDIRECCIONAL**
<http://bibdigital.epn.edu.ec/bitstream/15000/3772/1/CD-3550.pdf>.
21-01-2014.
- [12] **SISTEMAS NEUMATICOS**
<https://neumaticabasicaepp.wordpress.com/44-2/receptores-neumaticos/cilindros-de-simple-y-doble-efecto/>.
15-12-2014.
- [13] **SENSORES MAGNÉTICOS**
<http://www.balluff.com/balluff/MES/es/products/overview-magnetic-cylinder-sensors.jsp>.
12-12-2014.
- [14] **TWIDO**
<http://repositorio.espe.edu.ec/handle/21000/7302>
19-01 2015.
- [15] **VÁLVULAS**
<http://www.distritec.com.ar/detalleNovedad.php?titulo=%BFQU%C9%20ES%20UNA%20ELECTROV%C1LVULA%20Y%20PARA%20QU%C9%20SIRV>
16-12-2014.

ANEXO 1

MANUAL TÉCNICO

ROBOT MÓVIL

PALETIZADOR

MANUAL TÉCNICO ROBOT PALETIZADOR

INTRODUCCIÓN

El robot paletizador, es el complemento final del sistema MPS, que ofrece varias funcionalidades, siendo una herramienta útil de aprendizaje, tanto como para análisis en robótica y para el sistema de MPS al cual esta acoplado.

El siguiente manual técnico, ilustra las facultades tanto manuales como automáticas, brindando una información clara y precisa, sobre los usos y limitaciones del robot paletizador.

Antes de usar el robot paletizador se debe considerar el presente manual técnico, para evitar posibles fallas y averías en el sistema.

Instrucciones de Seguridad

- El uso sin cuidado del robot paletizador puede causar accidentes
- Polarice los componentes de manera correcta para evitar posibles daños a los componentes eléctricos y electrónicos.
- Evite maltratar al robot, el uso inadecuado del mismo puede causar daños en su estructura mecánica
- Recargue las baterías de manera habitual a un voltaje correcto.
- No lavarlas ni echarle líquidos
- Al momento de manipular el robot, asegúrese que sus botones de encendido estén en la posición off, y su alimentación desconectada.

Fuente de alimentación

El robot paletizador está alimentado por tres baterías, dos de 6 voltios y una de 12 voltios, cada una teniendo un uso diferente en el sistema, teniendo en cuenta que no se deberá polarizar mal estas, evitando así daños en su integridad física.

Batería (1) de 6v, Alimentación básica del circuito

La batería tiene como usa alimentar los siguientes componentes:

- Arduino Mega
- Sensores Sharp
- Sensor Magnético
- Placas de control L298N

Batería (2) de 6v, Actuador Lineal

Esta tiene como uso la alimentación del motor paso a paso, que compone el actuador lineal.

Batería (3) de 12V, Alimentación de Motores

Los conectores de las baterías están diferenciados para su mayor entendimiento y evitar posibles errores al momento de su conexión

Tabla: Configuración de Pines de Baterías

| Componente | Positivo (+) | Negativo (-) |
|--------------------|---------------------|---------------------|
| Actuador Lineal | Amarillo | Azul |
| Circuito Principal | Rojo | Azul |
| Motores | Rojo | Azul |

Conexión de componentes

Los sensores, actuadores y demás dispositivos están conectados al circuito principal que en este caso es la tarjeta Arduino Mega 2560, se detallan a continuación en la siguiente gráfica y tabla su respectiva conexión.

Cabe recalcar que aunque siendo el voltaje máximo de alimentación de 12 V, el cicui8to no se debe alimentar a dicho voltaje, ya que incurre a posibles fallas del circuito y acortamiento de la vida útil de todos sus componentes electrónicos.

TABLA DE CONEXIÓN

| PINES | CONEXIÓN Y FUNCIÓN |
|-----------------|--|
| Pin 3.3V | Se conecta al pin de alimentación del giroscopio |
| Pin 5V | Se conecta al pin de alimentación del bluetooth |
| Pin GND | Se conecta al pin de GND del bluetooth |
| Pin A0 | Se conecta a la señal del sensor Sharp identificado como IR1 |
| Pin A1 | Se conecta a la señal del sensor Sharp identificado como IR2 |
| Pin A2 | Se conecta a la señal del sensor Sharp identificado como IR3 |
| Pin A3 | Se conecta a la señal del sensor Sharp identificado como IR4 |
| Pin A4 | Se conecta a la señal del sensor Sharp identificado como IR5 |
| Pin A5 | Se conecta a la señal del sensor Sharp identificado como IR6 |
| Pin A6 | Se conecta a la señal del sensor Sharp identificado como IR7 |
| Pin A7 | Se conecta a la señal del sensor Sharp identificado como IR8 |
| Pin A8 | Se conecta a la señal del sensor Sharp identificado como IR9 |
| Pin 21 | Se conecta al pin de SCL del giroscopio para comunicación |
| Pin 20 | Se conecta al pin de SDA del giroscopio para comunicación |
| Pin 19 | Se conecta al pin señal del encoder de la llanta 1 |
| Pin 18 | Se conecta al pin señal del encoder de la llanta 2 |
| Pin 17 | Se conecta al pin de TX del Bluetooth para comunicación |

| | |
|-------------------|--|
| Pin 16 | Se conecta al pin de RX del Bluetooth para comunicación |
| Pin 2 | Se conecta al pin INT(interrupción) del giroscopio |
| Pin 3 | Se conecta al pin señal del encoder de la llanta 3 |
| Pin 40 | Se conecta al pin de señal de sensor óptico |
| Pin 26 | Se conecta al pin de activación de la llanta 1 del puente H1 |
| Pin 27 | Se conecta al pin de activación de la llanta 1 del puente H1 |
| Pin 28 | Se conecta al pin de activación de la llanta 2 del puente H2 |
| Pin 29 | Se conecta al pin de activación de la llanta 2 del puente H2 |
| Pin 30 | Se conecta al pin de activación de la llanta 3 del puente H3 |
| Pin 31 | Se conecta al pin de activación de la llanta 3 del puente H3 |
| Pin 7 | Se conecta al pin de Enable del puente H1 (envía señal de PWM) |
| Pin 8 | Se conecta al pin de Enable del puente H2 (envía señal de PWM) |
| Pin 9 | Se conecta al pin de Enable del puente H3(envía señal de PWM) |
| Pin 5V (1) | Se conecta a la alimentación de los sensores infrarrojos. |
| Pin GND(1) | Se conecta a la tierra (GND) de los sensores infrarrojos. |

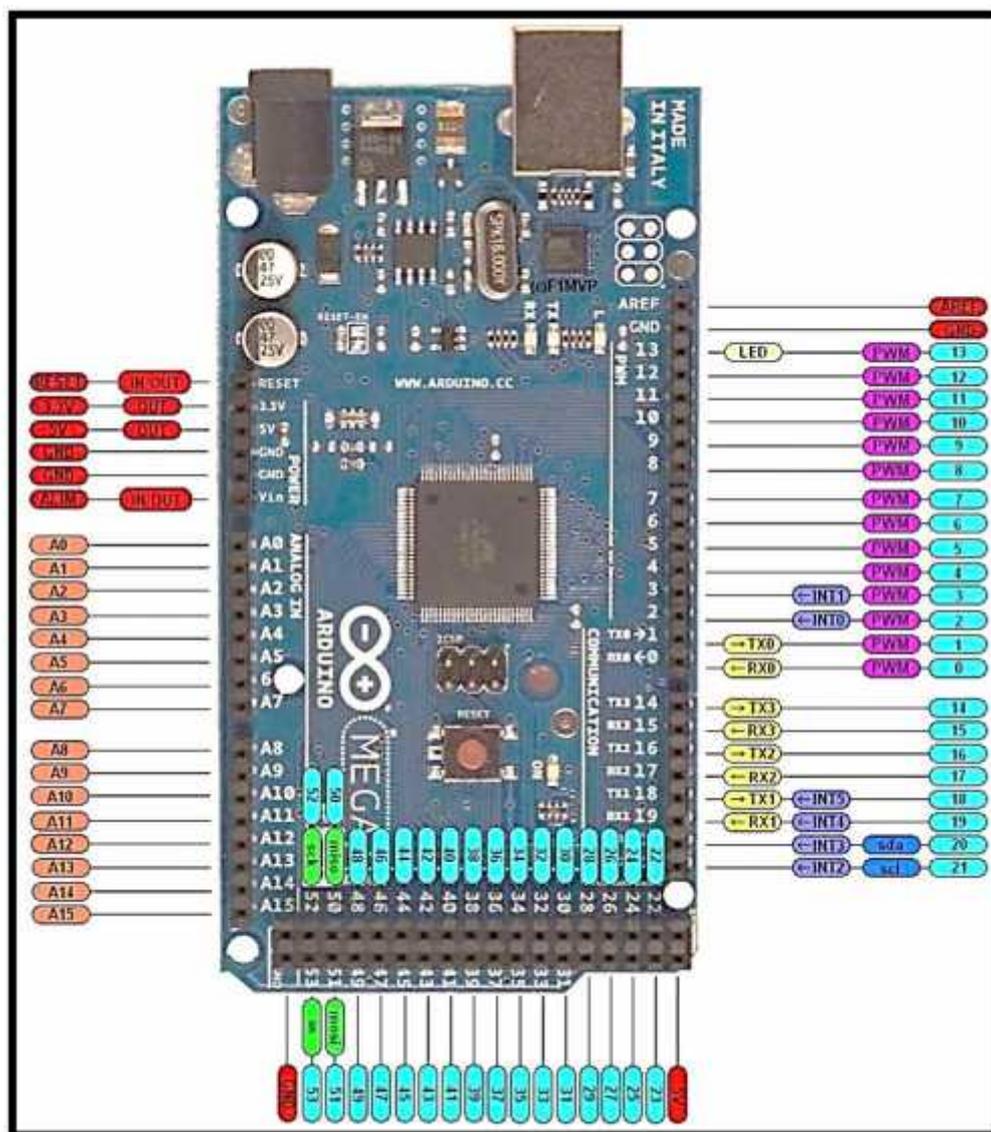


Fig. Pines Arduino Mega 2560 R3

HMI Robot Paletizador

Antes de comenzar tenemos que tener en consideración las configuraciones básicas del computador, tanto como de software como de hardware, siendo estas las siguientes.

- Bluetooth para PC
- Computador con sistema operativo W7
- Software Bluesoleil
- Labview 2013

Configuraciones INICIALES

Para obtener un correcto funcionamiento debemos seguir los siguientes pasos.

1. Iniciar Bluesoleil
2. Emparejar el dispositivo, la clave de emparejamiento es “0000”

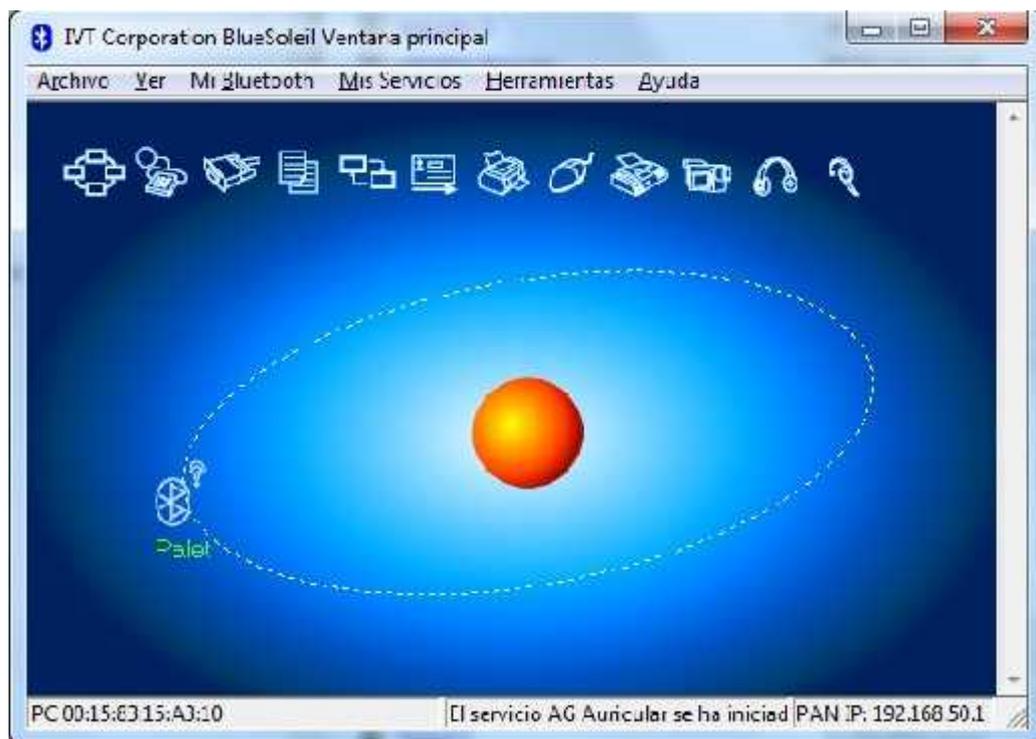


Fig. Pantalla Principal BlueSoleil

3. Buscar servicios
4. Conectar al servicio bluetooth
5. Verificar el puerto de transmisión de datos



Fig. Puerto de Transmisión Bluetooth

Al obtener los datos ingresar a la pantalla hmi_version_final.vi que se adjunta al manual, e ingresar el puerto de transmisión obtenido.

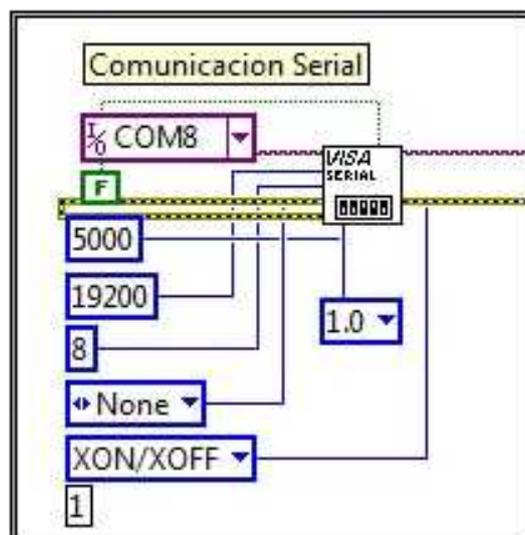


Fig. Configuración de Comunicación Serial

Configurar los datos de entrada de la matriz de puntos

- Debemos cambiar el path de la ubicación de los archivos x.lvm e y.lvm

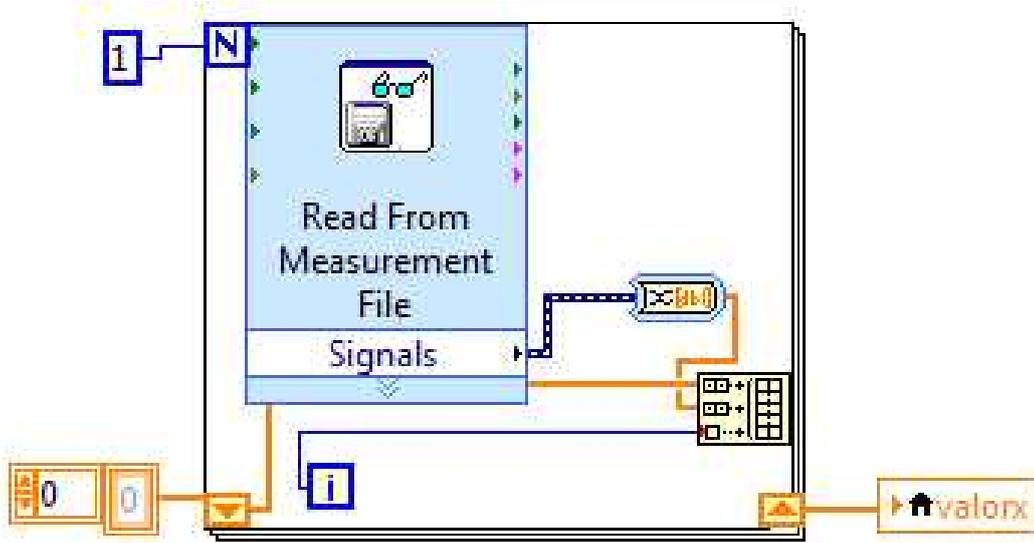


Fig. Configuración de path de datos

Configuración del PATH de la imagen tomada desde la cámara

- Indicar al programa el mapa donde el robot paletizador se movilizara

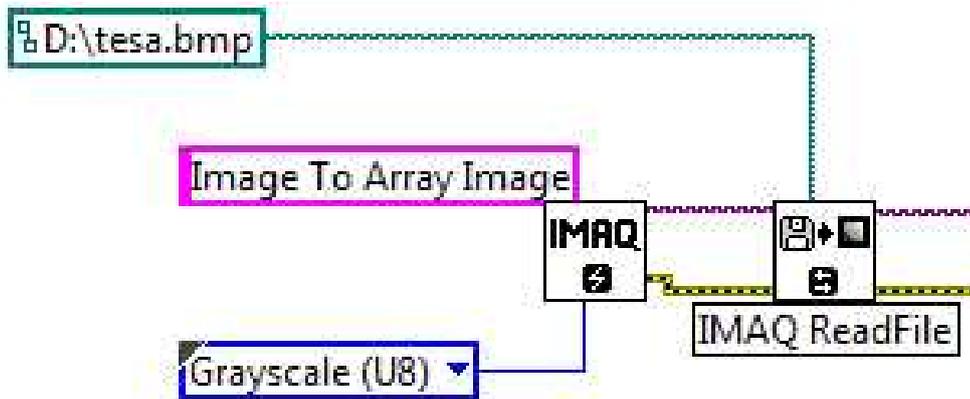


Fig. Configuración de path de datos

Terminando así las configuraciones iniciales, y listo para manejar el robot paletizador

HMI

Para iniciar el HMI, debemos, ejecutar en primera instancia el archivo 1ra_pantalla.vi, y ejecutar el programa, de manera normal y no en bucle.



Fig. Pantalla de Ingreso a HMI

Siendo esta la pantalla de ingreso a nuestro sistema, y las opciones del robot, se pueden elegir libremente desde la siguiente pantalla



Fig. Pantalla de Elección de función

Teniendo cuatro posibilidades a escoger

- Transporte de materiales
- Seguidor de Línea
- Configuración de sensores

Transporte de materiales

En la siguiente pantalla se explican los componentes expuestos en cada pestaña

- Plataforma controlada por PLC.- Se muestra el estado de los sensores de la plataforma y la dirección de la trayectoria que el robot debe seguir.



Fig. Plataforma

Esta pestaña contiene indicadores visuales, que muestran la ubicación del palet dentro del sistema. Y el estado de la bodega.

- Captura de Imagen.- Esta pestaña nos ayuda a tomar una foto del mapa donde el robot se quiera se movilice.

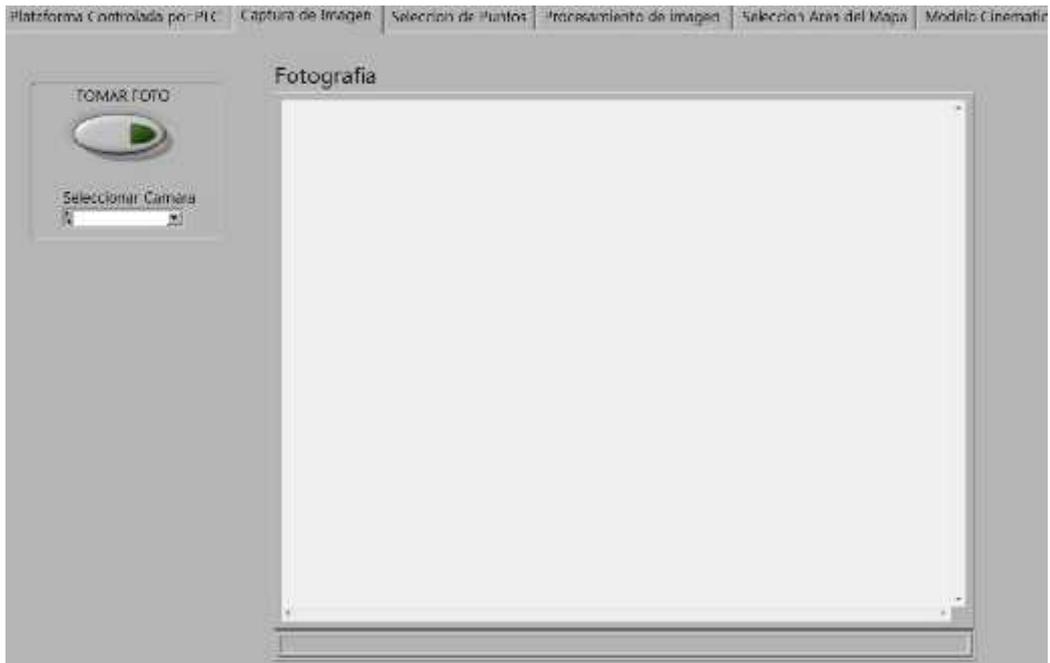


Fig. Captura de Imagen

- Selección de Puntos.- Los controles de la pantalla sirven para poder manipular los puntos iniciales y finales donde se moverá el robot paletizar.

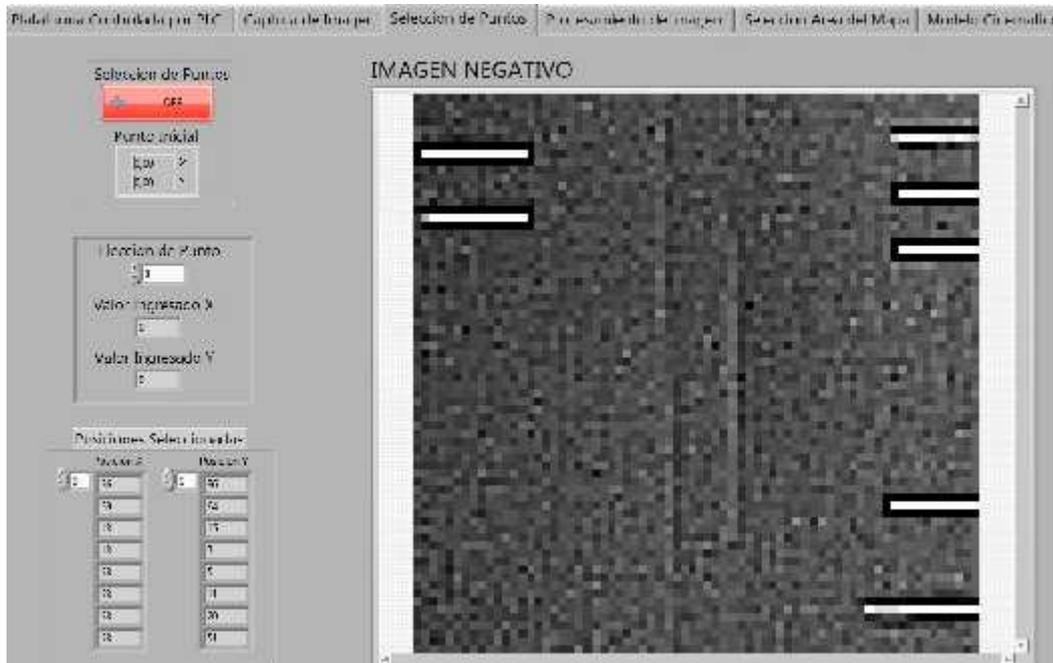


Fig. Selección de Puntos

El control de dirección de punto modificara el punto mostrado.

- Procesamiento de Imagen.- Esta pestaña nos ayudara a visualizar el recorrido del robot paletizador

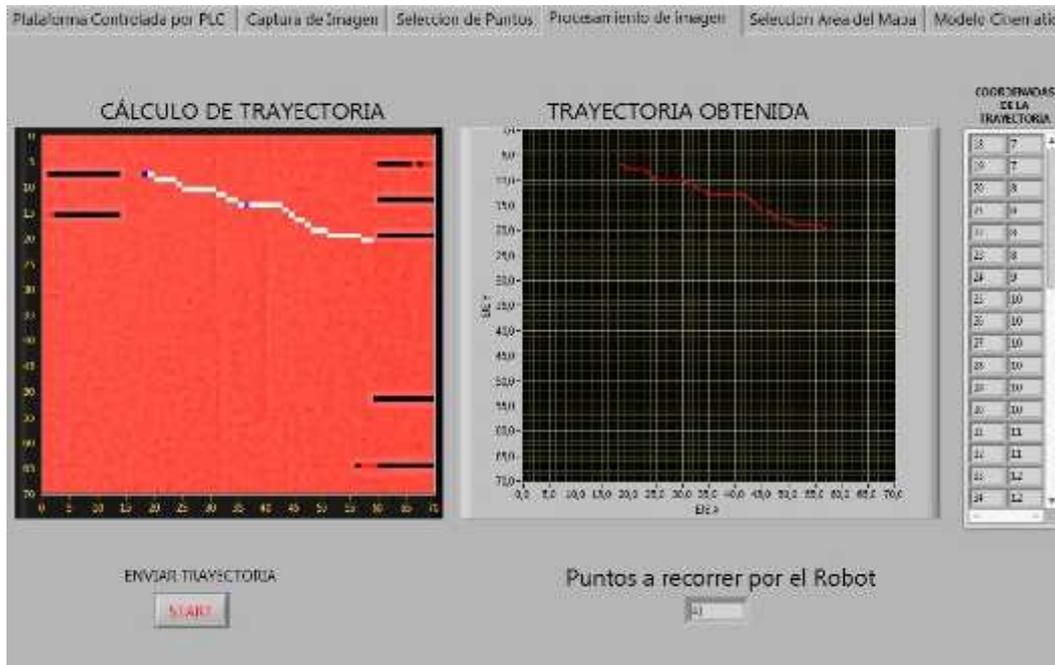


Fig. Planeación de Trayectoria

- Modelo Cinemático.- Visualización del cálculo del modelo cinemático del robot

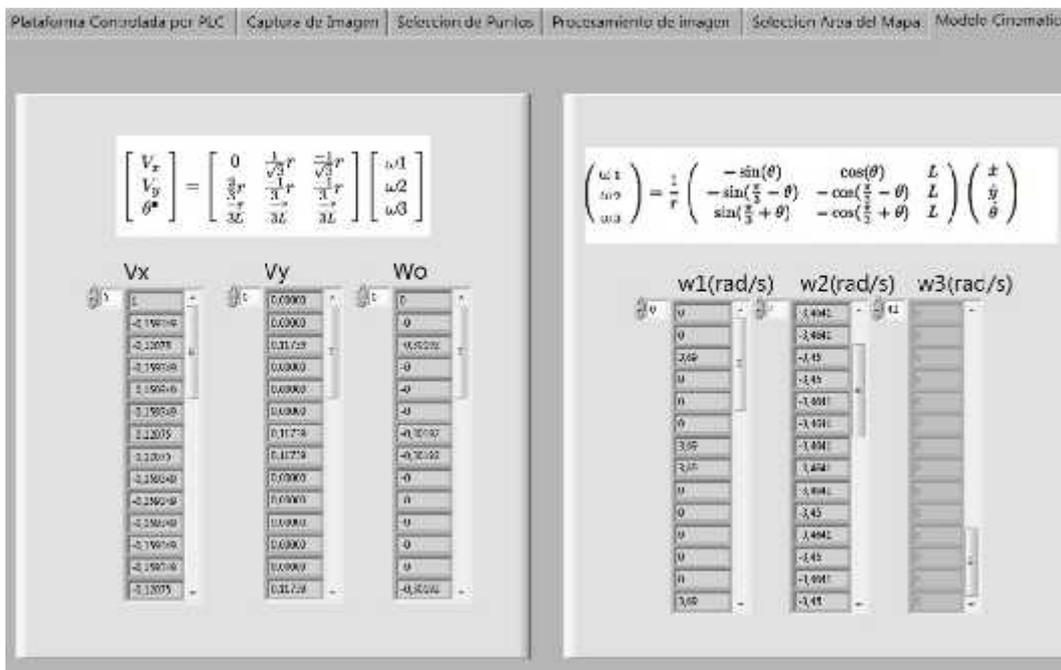


Fig. Planeación de Trayectoria

Seguidor de Línea

Al seleccionar esa función el robot entrara automáticamente en el modo seguidor de línea

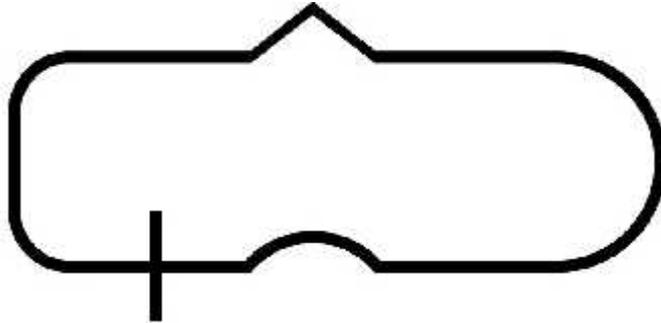


Fig. Pista Seguidor de Línea

Trayectoria Libre

Usando esta función, podemos seleccionar una trayectoria al azar, seleccionando el punto inicial y el punto final del robot.

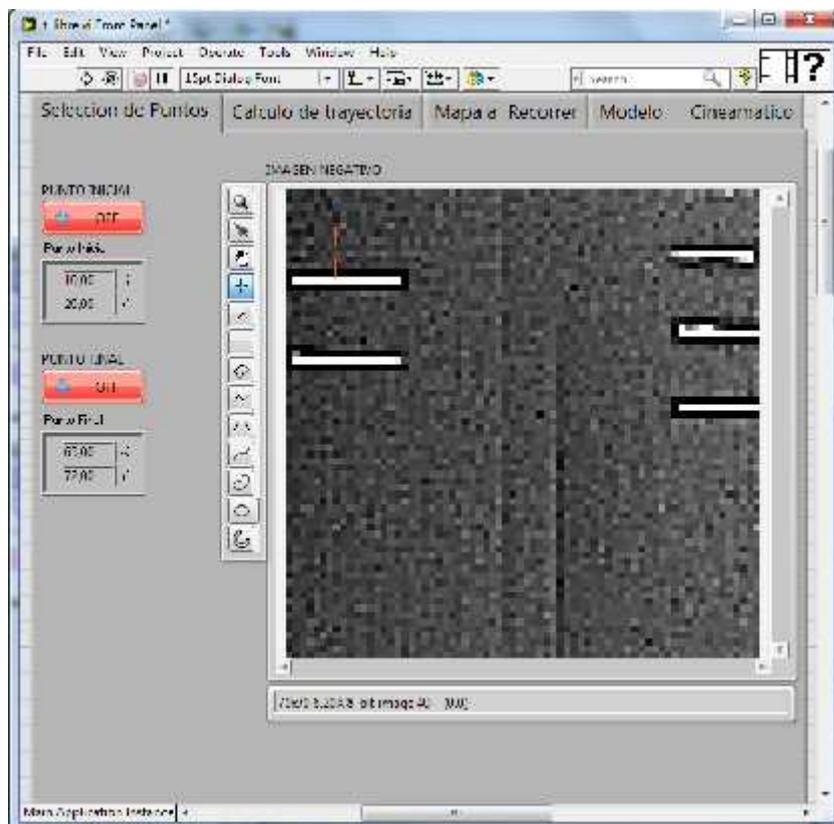


Fig. Selección de Trayectoria Libre

Configuración de sensores

En esta opción podemos controlar la distancia a que están configurados los sensores Sharp, que por defecto es 5 cm.



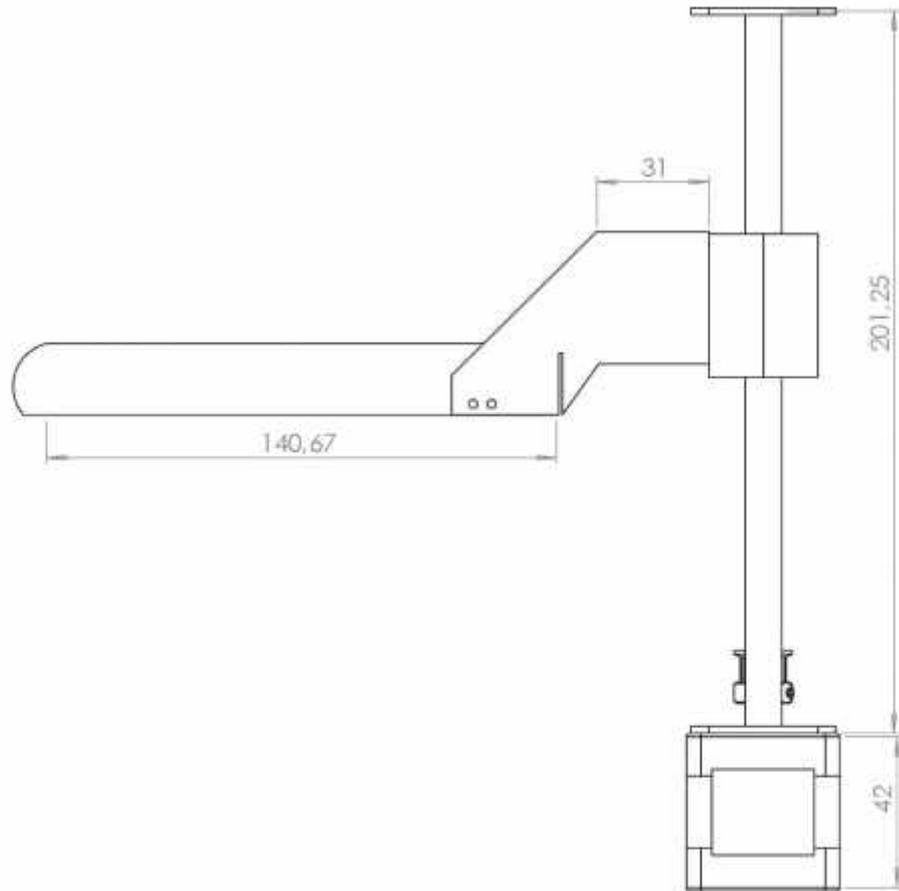
Fig. Configuración de Sensores

Se debe seleccionar el número de sensor infrarrojo a modificar, y se le asigna el valor deseado mediante la barra deslizante, pudiendo configurar cada sensor a la distancia requerida por el usuario; desde un valor mínimo de 4 centímetros hasta 40 centímetros.

ANEXO 2

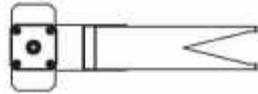
Diseño de piezas en Solid

Works

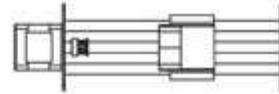


| | | | | | |
|--|----------------------------------|---------------------------|-------------------------------------|--|-----------------------|
| (INDICAR SI HAY QUE SERVICIAR) LAS COTAS DE ESPESOR EN MM (SERVICIO) (SERVICIO) (SERVICIO) | | ACABADO Ninguno | SERVICIO Y REPARACIONES VIGAS | HOJAS DE LA ESCALA Robot Paletizador | REVISOR |
| DISEÑADOR Jorge Lopez | DISEÑADOR Gala Soriano | DISEÑADOR | DISEÑADOR | TITULO Actuador Lineal de 20 cm | |
| CALIFICACION | CALIFICACION | CALIFICACION | CALIFICACION | MATERIAL Aluminio y Metal | NÚMERO DE 1 |
| PESO 1kg | ESCALA | ESCALA | ESCALA | ESCALA | ESCALA A4 |

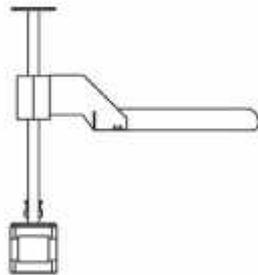
Vista Superior



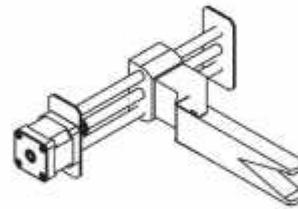
Vista Posterior



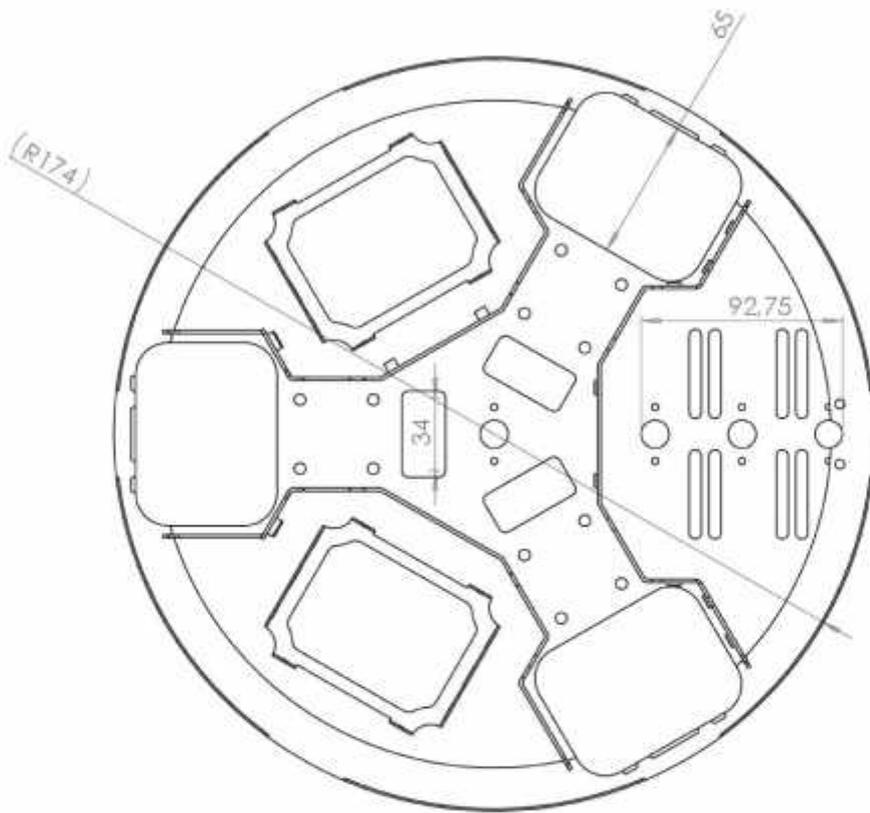
Vista Lateral



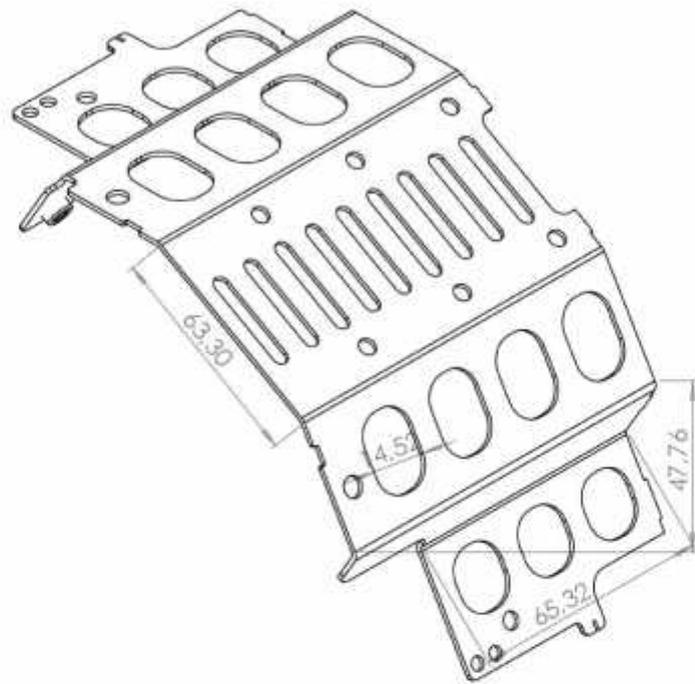
Vista Dimetrica



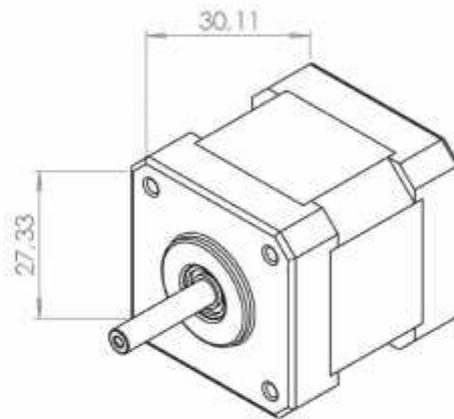
| | | | | | | | |
|---|---------------------------------------|--------------------|--------|--------------------------------------|--|-------------------------------------|-------------|
| TIPO DE BOCAL (OJO DE BUEY) LAS COTAS DE ESPESOR EN MM ACABADO SUPERFICIAL (SERVICIOS) ESPECIAL INICIAL REGULAR | | ACABADO Ninguno | | REFINADO Y REFINAR AREAS VITAL | | HOJAS DE LA ESCALA REVISOR | |
| | | | | | | Robot Paletizador | |
| | | | | | | TITULO Vistas de actuador Lineal | |
| DISEÑ. VERIF. APROB. FIRM. CALIF. | NOMBRE Jorge Lopez Gola Sorlián | FECHA | RECIBI | MATERIAL Aluminio y Metal | | N° DE DISEÑO 2 | A4 |
| | | | | PESO 1 kg | | ESCALA 1:1 | HOJA 1 DE 1 |



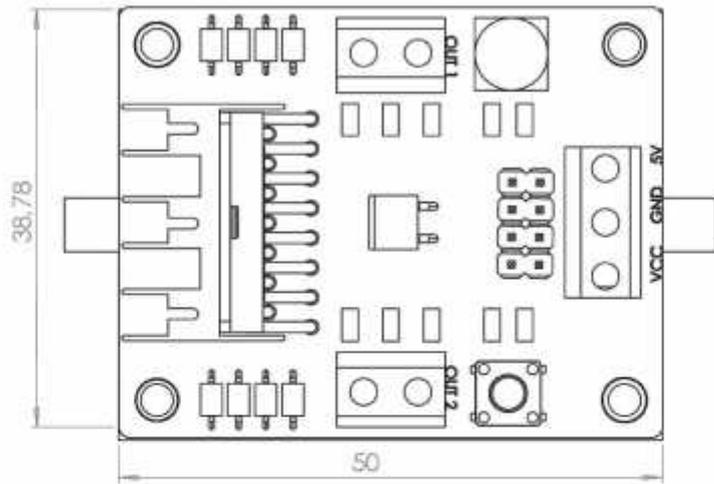
| | | | | | | |
|---|---------------------|---------------------------------------|---|-------------------------|--------------------|-----------|
| (TIPO DE BORDO (D/C/O/R/H/H)) (LÍNEA CONTINUA (E/P/RES/BI) (E/H/M)) ACABADO SUPERFICIAL (CERAMICO) (S/PA) (N/COLOAR) | | ACABADO | (PROFUNDIDAD) (COMPLETAR ÁREAS) (VÁLID) | | (NOMBRE LA ESCALA) | (REVISOR) |
| | | | | | Robot Paletizador | |
| (DISEÑO) (VERIFICACIÓN) (AUTORIZACIÓN) (FECHA) | (NOMBRE) (CARGO) | (FECHA) | (FECHA) | (FECHA) | Base Robot | |
| | | MATERIAL Aluminio Compuesto | | (NÚMERO DE) 3 | A4 | |
| | | PESO 0.8 kG | | (ESCALA) | (PÁGINA DE) | |



| | | | | | | | | | |
|--|--|--------------------------|--|---------------------------------------|--|---|--|----------|--|
| (INDICAR CÓDIGO DE IDENTIFICACIÓN) (LUGAR DE EMISIÓN) (FECHA DE EMISIÓN) (NOMBRE) (CÓDIGO) | | ACABADO: Ninguno | | REVISIONES: REVISIONES: VÁLIDAS | | NOMBRE DE LA ESCALA: Robot Paletizador | | REVISOR: | |
| DISEÑO: JORGE LOPEZ | | VERIFICADO: GILIO SANWON | | MATERIAL: Aluminio Compuesto | | NÚMERO DE PIEZAS: 4 | | A4 | |
| PESO: 0.8 kg | | ESCALA: 1:1 | | HOJA 1 DE 1 | | | | | |

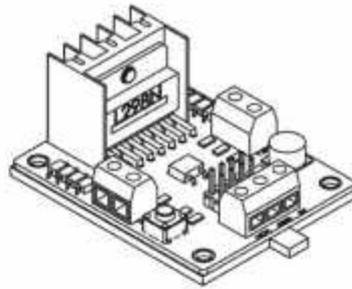


| | | | | | | |
|--|----------------------------|---------|--------------------------------------|--|---|--------------|
| TITULO DE BOCAL (O C/30-30-30) LAS COLAS SE ENTREGAN EN PAQ. IDENTIFICACION ESPECIAL REGULAR | | ACABADO | PREPARADO / REAJUSTADO / VIVAS | | NOMBRE DE LA ESCALA Robot Paleizador | REVISION |
| Ninguno | | | | | Robot Paleizador | |
| DISEÑADOR Jorgo Lopez | DISEÑADA Celia Santolan | REVISOR | | | TITULO Vistas de motor paso a paso | |
| AREA | | | | | MATERIAL Cobre, Metal | |
| PAIS | | | | | NÚMERO DE DISEÑO 5 | A4 |
| CALIBRE | | | | | PESO 0.5 kg | ESCALA (1:1) |
| | | | | | HOJA 1 DE 1 | |

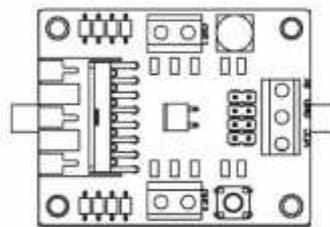


| | | | | | | |
|---|----------------|----------|------------------|------------------------------------|---------------------|----------|
| (INDICAR SI HAY QUE COLOCAR EN EL DISEÑO CONTACTOS DE PUESTA EN COMÚN ACORDANDO SUPERFICIALES) (INDICAR SI HAY QUE COLOCAR EN EL DISEÑO CONTACTOS DE PUESTA EN COMÚN ACORDANDO SUPERFICIALES) | | AC/BAJOS | Ninguno | RESERVAR Y COMPLETAR ÁREAS VÁLIDAS | NOMBRE DE LA ESCALA | REVISIÓN |
| | | | | | Robot Paletizador | |
| | | | | | Driver L298 | |
| DISEÑADOR | HOMER | FECHA | RECIBO | | | |
| DISEÑADOR | Jorge Lopez | | | | | |
| VERIFICADOR | Dago Santillan | | | | | |
| AREA | | | | | | |
| PAIS | | | | | | |
| CATEGORIA | | | | MATERIAL | Nº DE DISEÑO | A4 |
| | | | Aluminio y Metal | 6 | | |
| | | | PESO 1kg | ESCALA 1:1 | HOJA 1 DE 1 | |

Vista Dimetrica



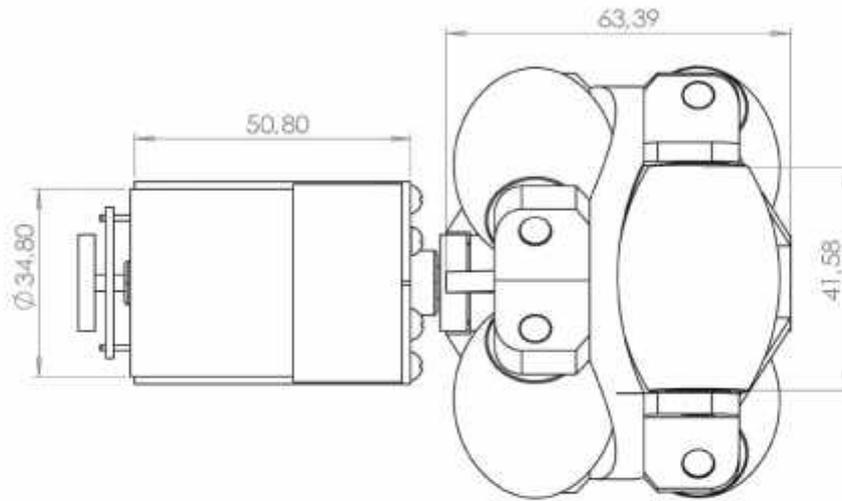
Vista Frontal



Vista Posterior

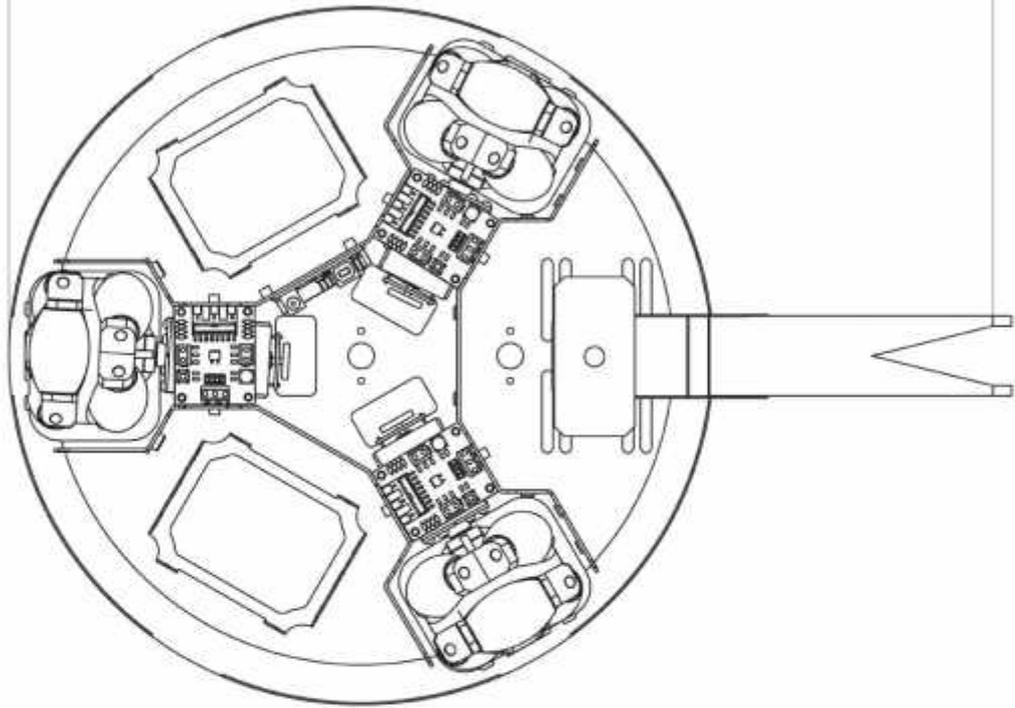


| | | | | | |
|---|---------------------------------|---|--|---|--------------------------|
| (INDICAR CÓDIGO DE IDENTIFICACIÓN) (LUGAR DE CONTACTO CON EL PRODUCTO) (CÓDIGO DE IDENTIFICACIÓN) (CÓDIGO DE IDENTIFICACIÓN) (CÓDIGO DE IDENTIFICACIÓN) | | ACABADO: Ninguno | PRESENTACIÓN: PACKING: AREA 45 VTA | CÓDIGO DE LA ESCALA: Robot Paletizador | REVISIÓN: Driver L298 |
| DISEÑO: Jorgé Lopez | VERIFICACIÓN: Gladys Samirán | DISEÑO: REVISIÓN: APROBACIÓN: FIRMAS: CALIFICACIONES: | MATERIAL: Aluminio y Metal | NÚMERO DE DISEÑO: 7 | A4 |
| | | PESO: 1 kg | DESCRIPCIÓN: (DESCRIBIR EL PRODUCTO) | PÁGINA 1 DE 1 | |



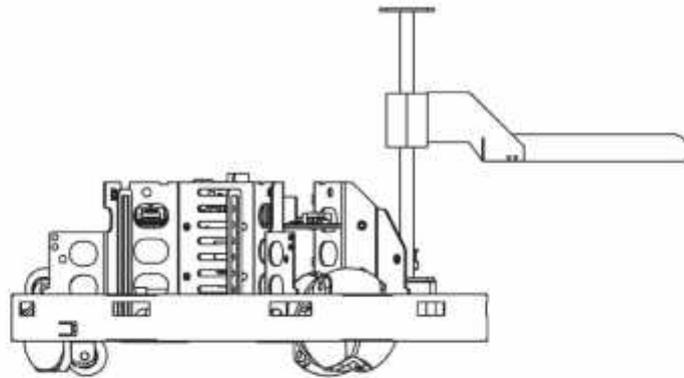
| | | | | | |
|--|----------------|--------------------|---|--|------------------------------|
| (INDICAR CÓDIGO DE COLORES) LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL (SERVICIOS) (SEAL) (ACABAR) | | ACABADO | SERVICIOS (SERVICIOS) (SERVICIOS) | | HOCAMBI LA ESCALA REVISOR |
| | | Largura | | Robot Paletizador | |
| | FORMA | SEAL | RECH | TIPO | |
| DESE: | ADG01 L0242 | | | Motor montado en rueda omnidireccional | |
| VERE: | Claro Sanfilan | | | | |
| ARCE: | | | | | |
| RAER: | | | | | |
| CALE: | | MATERIAL | | PL/DE/DE/JO | 8 |
| | | Aluminio Compuesto | | | A4 |
| | | PESO 0.8 KG | | ESCALA: 1:1 | PÁGINA 1 DE 1 |

488,67

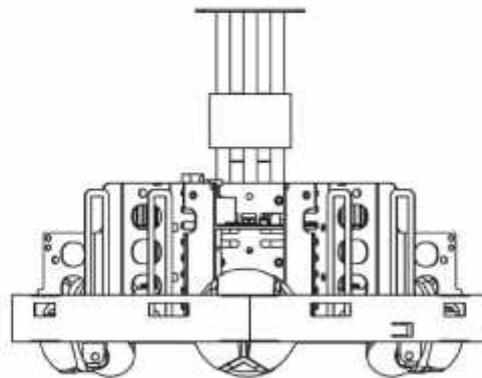


| | | | | | | | |
|---|-----------------------|------------|---------------|--|---------------------------------------|------------------------|-------------|
| (INDICAR CÓDIGO DE CONTROL) (LUGAR DE FABRICACIÓN) (ACABADO SUPERFICIAL) (TOLERANCIAS) (UNIDAD) (MATERIAL) | | ACABADO | Fingura | | PROCESADO / REQUISITOS ADICIONALES | NO CAMBIAR LA ESCALA | REVISIÓN |
| | | | | | | Robot Paleizador | |
| | | | | | | Placa Lateral | |
| DISEÑO: VERIFICADO: APROBADO: FECHA: CALIFICACIÓN: | NOMBRE: AYDA LOPEZ | FECHA: | REVISIÓN: | | | NÚMERO DE DISEÑO: 9 | A4 |
| MATERIAL: Aluminio Compuesto | | | | | | PESO: 0,8 KG | HOJA 1 DE 1 |

Vista Lateral



Vista Frontal



| | | | | |
|---|---|--|--------------------------------------|--|
| TIPO DE BOCAL (D/C/O/S/HH/H): LAS COTAS DE EJEMPLO EN GRAS ACABADO SUPERFICIAL (SERIE/NO/SI) TOLERANCIAS (SERIAL/NO/SI) INICIAL/NO/SI | | ACABADO Ninguno | REFERENCIAS ROMPER AREAS VITAL | HOJAS DE LA ESCALA REVISOR Robot Paletizador |
| HOMBRE DISEÑO: Jorge Lopez VERIFICACIÓN: Gato Sanfeliu AREA: PAIS: CALIFICACIÓN: | SERIA: RECIBIÓ: MATERIAL: Aluminio, Plástico y metal | TIPO: Vistas Robotino | NÚMERO DE HOJA: 10 | A4 |
| FECHAS: B E G | | ESCALAS: HOJA 1 DE 1 | | |

ANEXO 3

Datasheet

Arduino Mega 2560

Technical Specification

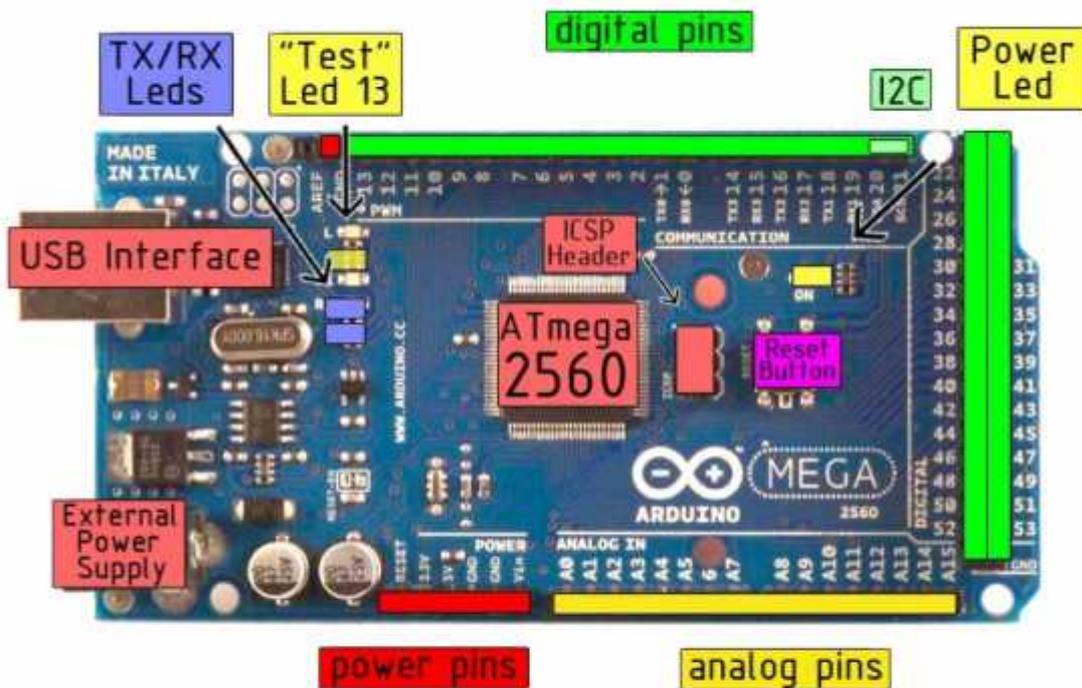


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

| | |
|-----------------------------|---|
| Microcontroller | ATmega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 54 (of which 14 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |

the board



radiospares **RADIONICS**



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



ANEXO 4

Datasheet

Gear Motor Pololu 37D

RB-Pol-124
Pololu 12V, 29:1 Gear Motor w/Encoder



This 2.54" × 1.45" × 1.45" gearmotor is a powerful 12V motor with a **29:1** metal gearbox and an integrated quadrature encoder that provides a resolution of 64 counts per revolution of the motor shaft, which corresponds to **1856 counts per revolution** of the gearbox's output shaft. These units have a 0.61"-long, 6 mm-diameter D-shaped output shaft. This gearmotor is also available without an encoder.

Key specs at **12 V**: 350 RPM and 300 mA free-run, 110 oz-in (8 kg-cm) and 5 A stall.

Gearmotor Options

This powerful gearmotor is available in six different gear ratios and features an integrated quadrature encoder with 64 counts per revolution (CPR) of the motor shaft. A version without the encoder is also available.

| Gear Ratio | With Encoder | Without Encoder |
|-------------------|----------------------------|----------------------------|
| 19:1 | Metal Gearmotor 37Dx52L mm | Metal Gearmotor 37Dx52L mm |
| 29:1 | Metal Gearmotor 37Dx52L mm | Metal Gearmotor 37Dx52L mm |
| 50:1 | Metal Gearmotor 37Dx54L mm | Metal Gearmotor 37Dx54L mm |
| 67:1 | Metal Gearmotor 37Dx54L mm | Metal Gearmotor 37Dx54L mm |
| 100:1 | Metal Gearmotor 37Dx57L mm | Metal Gearmotor 37Dx57L mm |
| 131:1 | Metal Gearmotor 37Dx57L mm | Metal Gearmotor 37Dx57L mm |

Note: The version we used to call "122:1 metal gearmotor 37Dx57L mm" is now "131:1". The gear ratio is now more accurately specified, but the other specifications have not changed.

These motors are intended for use at 12 V, though the motor can begin rotating at voltages as low as 1 V.

Gearmotor Dimensions

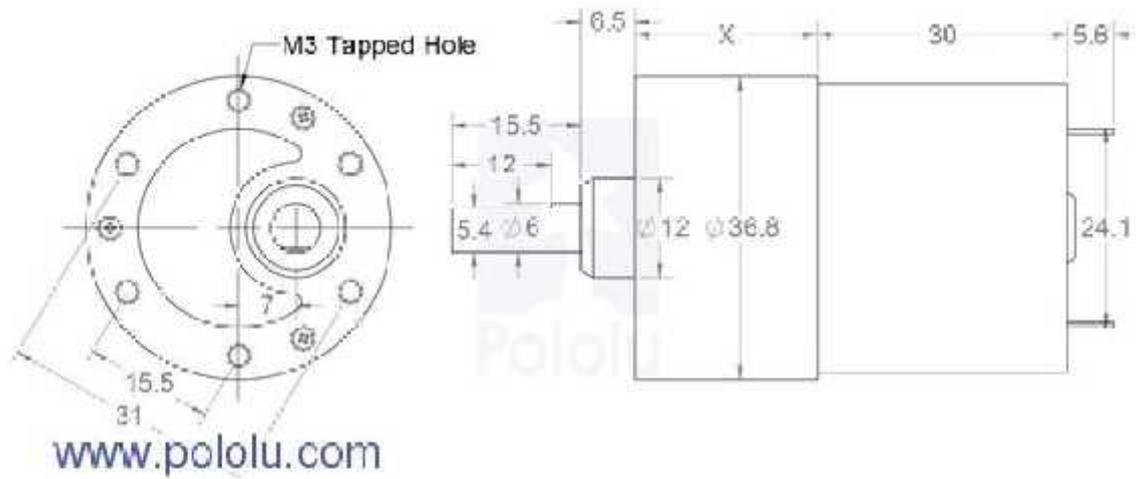
The face plate has six mounting holes evenly spaced around the outer edge threaded for M3 screws. These mounting holes form a regular hexagon and the centers of neighboring holes are 15.5 mm apart. You can use our custom 37D mm metal gearmotor bracket (shown in the left picture below) to mount the gearmotor to your project via these mounting holes and the screws that come with the bracket.



Gearmotor with bracket and hub.



37D mm metal gear motor with 64 CPR encoder and Pololu 90x10mm wheel.



37D mm metal gearmotor dimensions (units in mm).

Warning: Do not screw too far into the mounting holes as the screws can hit the gears. We recommend screwing no further than 3mm (1/8") into the screw hole.

Using the Encoder

A two-channel Hall effect encoder is used to sense the rotation of a magnetic disk on a rear protrusion of the motor shaft. The quadrature encoder provides a resolution of 64 counts per revolution of the motor shaft. To compute the counts per revolution of the gearbox output, multiply the gear ratio by 64. The motor/encoder has six color-coded, 11" (28 cm) leads:



37D mm metal gear motor with 64 CPR encoder.

| Color | Function |
|--------|------------------------------|
| Black | motor power |
| Red | motor power |
| Blue | Hall sensor Vcc (3.5 – 20 V) |
| Green | Hall sensor GND |
| Yellow | Hall sensor A output |
| White | Hall sensor B output |

These leads have stripped, unterminated ends that can be soldered or crimped to make solder free connectors. The Hall sensor requires an input voltage, Vcc, between 3.5 and 20 V and draws a maximum of 10 mA. The A and B outputs are square waves from 0 V to Vcc approximately 90° out of phase. The frequency of the transitions tells you the speed of the motor, and the order of the transitions tells you the direction. The following oscilloscope capture shows the A and B (yellow and white) encoder outputs using a motor voltage of 12 V and a Hall sensor Vcc of 5 V:

ANEXO 5

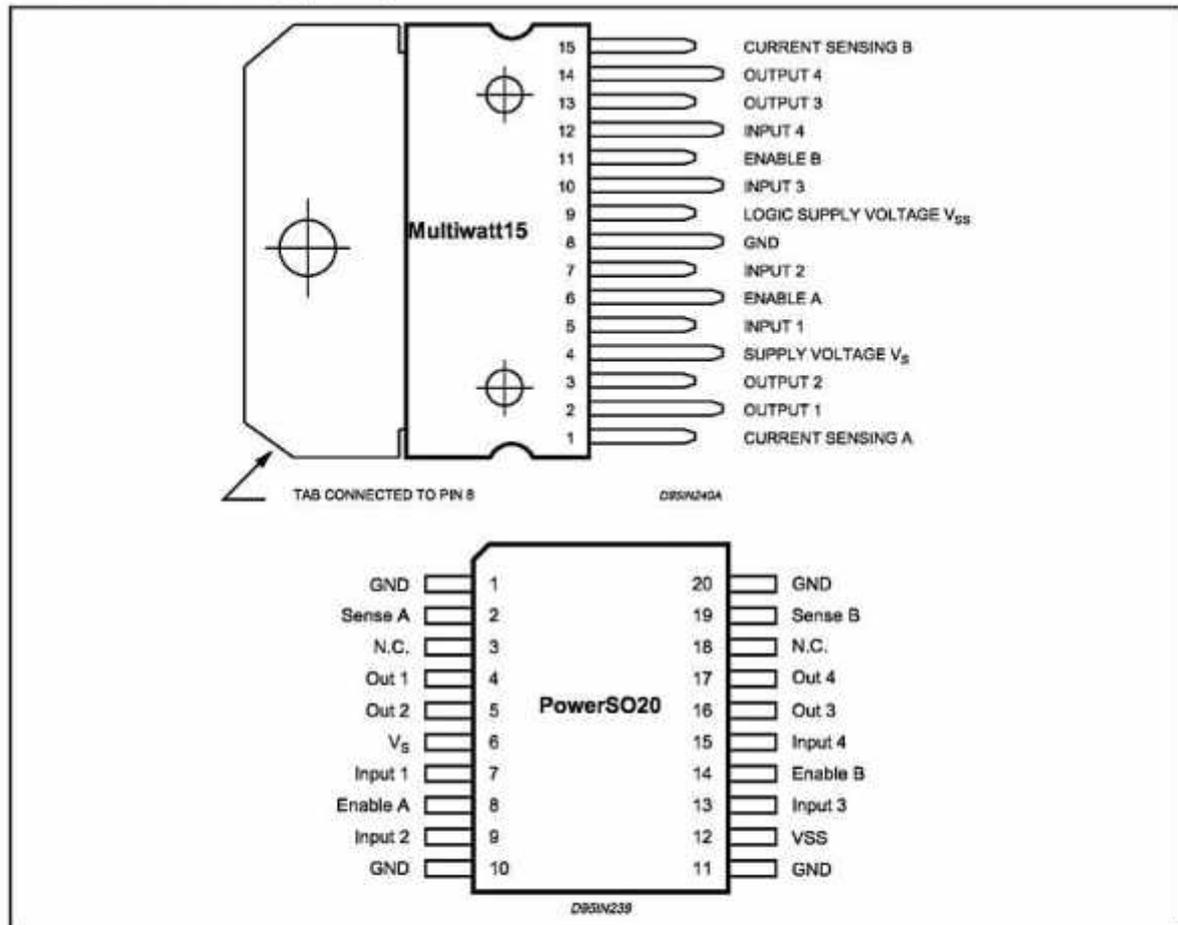
Datasheet

Puente H L298N

ABSOLUTE MAXIMUM RATINGS

| Symbol | Parameter | Value | Unit |
|----------------|---|---------------|-------------|
| V_S | Power Supply | 50 | V |
| V_{SS} | Logic Supply Voltage | 7 | V |
| V_i, V_{en} | Input and Enable Voltage | -0.3 to 7 | V |
| I_o | Peak Output Current (each Channel) - Non Repetitive ($t = 100\mu s$) - Repetitive (80% on -20% off; $t_{on} = 10ms$) - DC Operation | 3 2.5 2 | A A A |
| V_{sens} | Sensing Voltage | -1 to 2.3 | V |
| P_{tot} | Total Power Dissipation ($T_{case} = 75^\circ C$) | 25 | W |
| T_{op} | Junction Operating Temperature | -25 to 130 | $^\circ C$ |
| T_{stg}, T_j | Storage and Junction Temperature | -40 to 150 | $^\circ C$ |

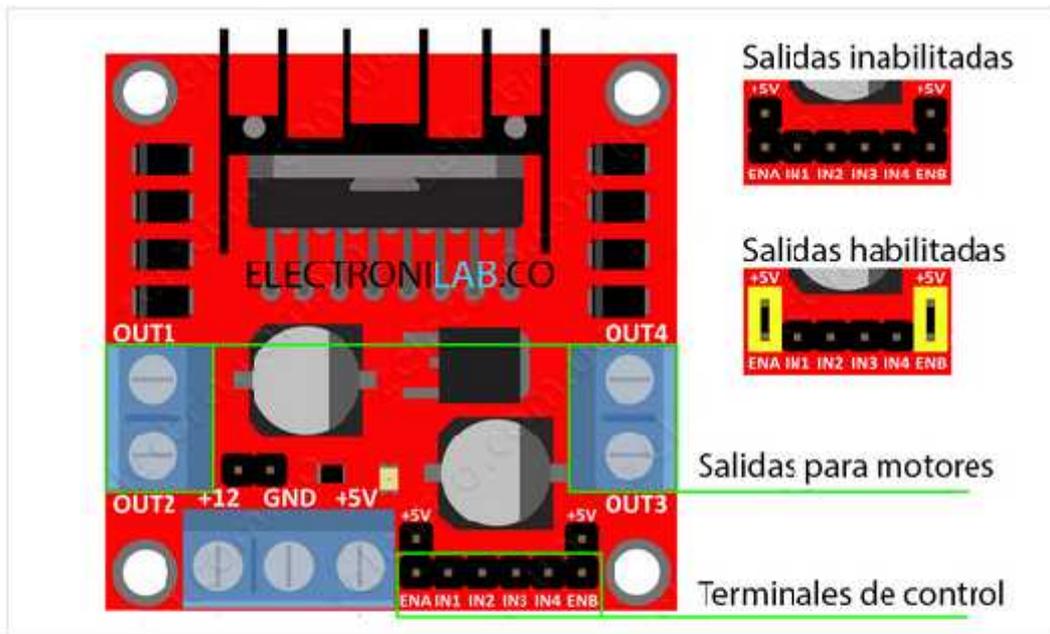
PIN CONNECTIONS (top view)



THERMAL DATA

| Symbol | Parameter | PowerSO20 | Multiwatt15 | Unit |
|------------------|-------------------------------------|-----------|-------------|--------------|
| $R_{th\ j-case}$ | Thermal Resistance Junction-case | Max. | 3 | $^\circ C/W$ |
| $R_{th\ j-amb}$ | Thermal Resistance Junction-ambient | Max. | 13 (*) | $^\circ C/W$ |

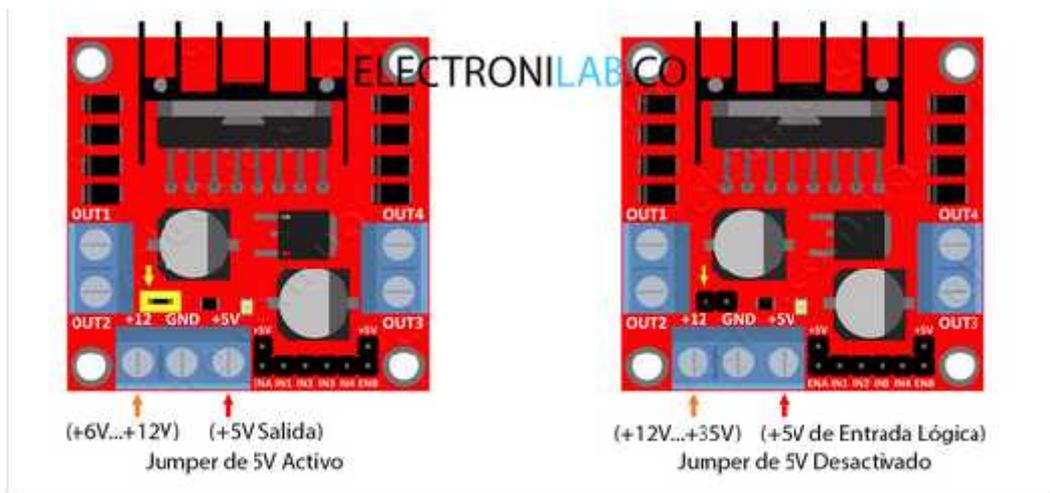
(*) Mounted on aluminum substrate



En la parte inferior se encuentran los pines de control del módulo, marcados como IN1, IN2, IN3 e IN4.

Conexión de alimentación

Este módulo se puede alimentar de 2 maneras gracias al regulador integrado LM7805



ANEXO 6

Datasheet

Sensor Sick IME 08

IME08-1B5PSZT0S | IM Standard

INDUCTIVE PROXIMITY SENSORS



Ordering information

| Model name | Part no. |
|-----------------|----------|
| IME08-1B5PSZT0S | 1040838 |

Other models and accessories → www.sick.de/IM_Standard



Detailed technical data

Features

| | |
|-----------------------------|----------------------|
| Housing | Cylindrical standard |
| Thread size | M8 x 1 |
| Sensing range S_s | 1.5 mm |
| Assured sensing range S_a | 1.215 mm |
| Installation type | Flush |
| Switching frequency | 4,000 Hz |
| Connection type | Connector M8, 3-pin |
| Output type | PNP |
| Output function | NO |
| Electrical wiring | DC 3-wire |
| Enclosure rating | IP 67 ¹⁾ |

¹⁾ According to EN 60529.

Mechanics/electronics

| | |
|--------------------------------|---------------------------|
| Supply voltage | 10 V DC ... 30 V DC |
| Ripple | ≤ 10 % |
| Voltage drop | ≤ 2 V ²⁾ |
| Current consumption | ≤ 10 mA ²⁾ |
| Time delay before availability | ≤ 100 ms |
| Hysteresis | 5 % ... 15 % |
| Repeatability | ≤ 2 % ^{3) 4)} |
| Temperature drift (of S_s) | ± 10 % |
| EMC | According to EN 60947-5-2 |

²⁾ At I_b max.

³⁾ Without load.

⁴⁾ U_b and T_a constant.

⁵⁾ Of S_r .

| | |
|-------------------------------|-----------------------------------|
| Output current I_a | ≤ 200 mA |
| Short-circuit protection | ✓ |
| Reverse polarity protection | ✓ |
| Power-up pulse protection | ✓ |
| Shock/vibration | 30 g, 11 ms/10 Hz ... 55 Hz, 1 mm |
| Ambient operating temperature | -25 °C ... +75 °C |
| Housing material | Metal, Nickel-plated brass |
| Housing cap material | Plastic, PA6 |
| Housing length | 50 mm |
| Thread length | 34 mm |
| Tightening torque, max. | Typ. 5 Nm |
| UL File No. | NRKH.E181493 |

¹⁾ At I_a max.

²⁾ Without load.

³⁾ I_{a0} and T_a constant.

⁴⁾ D² S².

Reduction factors

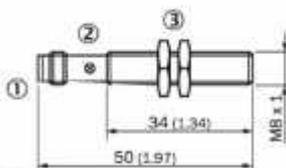
| Note | The values are reference values which may vary |
|----------------------------|--|
| Carbon steel St37 (Fe) | 1 |
| Stainless steel (V2A, 304) | Approx. 0.8 |
| Aluminum (Al) | Approx. 0.45 |
| Copper (Cu) | Approx. 0.4 |
| Brass (Br) | Approx. 0.4 |

Installation note

| Remark | Associated graphic see "Installation" |
|--------|---------------------------------------|
| B | 8 mm |
| C | 8 mm |
| D | 4.5 mm |
| F | 12 mm |

Dimensional drawing (Dimensions in mm (inch))

IME08 Standard, connector, flush



- ① Connection
- ② LED indicator
- ③ Fastening nuts (2 x); width across 13, metal

ANEXO 7

Datasheet

PLC TWIDO CAE40DRF

TWDLCAE40DRF

compact PLC base Twido - 100..240 V AC
supply - 24 I 24 V DC - 16 O



Main

| | |
|--------------------------------|---|
| Range of product | Twido |
| Product or component type | Compact base controller |
| Discrete I/O number | 40 |
| Discrete input number | 24 |
| Discrete input voltage | 24 V |
| Discrete input voltage type | DC |
| Discrete output number | 2 transistor 14 relay |
| Number of I/O expansion module | 7 |
| [Us] rated supply voltage | 100...240 V AC |
| Use of slot | Memory cartridge |
| Data backed up | Internal RAM external battery TSXPLP01 3 years |
| Integrated connection type | Ethernet TCP/IP RJ45 10/100 Mbit/s 1 twisted pair transparent ready class A10 Non isolated serial link mini DIN Modbus/character mode master/slave RTU/ASCII RS485 half duplex 38,4 kbit/s Power supply Serial link interface adaptor RS232C/RS485 |
| Complementary function | PID Event processing |

Complementary

| | |
|---|---|
| Concept | Transparent Ready |
| Discrete input logic | Sink or source |
| Input voltage limits | 20,4...26,4 V |
| Discrete input current | 7 mA I0,2 to I0,5 7 mA I0,8 to I0,23 11 mA I0,0 to I0,1 11 mA I0,6 to I0,7 |
| Input impedance | 2100 Ohm I0,0 to I0,1 2100 Ohm I0,6 to I0,7 3400 Ohm I0,2 to I0,5 3400 Ohm I0,8 to I0,23 |
| Filter time | 35 µs + programmed filter time for I0,0 to I0,5 at state 1 40 µs + programmed filter time for I0,0 to I0,5 at state 0 40 µs + programmed filter time for I0,6 to I0,23 at state 1 150 µs + programmed filter time for I0,6 to I0,23 at state 0 |
| Insulation between channel and internal logic | 1500 Vrms for 1 minute |
| Insulation resistance between channel | None |
| Minimum load | 0,1 mA |
| Contact resistance | ≤ 30000 µOhm |
| Load current | 2 A 240 V AC resistive 30 cyc/mn relay outputs 2 A 30 V DC resistive 30 cyc/mn relay outputs 2 A 240 V AC inductive 30 cyc/mn relay outputs 2 A 30 V DC inductive 30 cyc/mn relay outputs |
| Mechanical durability | ≥ 20000000 cycles relay outputs |
| Electrical durability | ≥ 100000 cycles relay outputs |

| | |
|--------------------------------|--|
| Current consumption | 5 mA 24 V DC at state 0 90 mA 5 V DC at state 1 128 mA 24 V DC at state 1 128 mA 24 V DC state 1 + input ON 170 mA 5 V DC at state 0 240 mA 5 V DC state 1 + input ON |
| I/O connection | Non-removable screw terminal block |
| Input/Output number | ≤ 152 removable screw terminal block with I/O expansion module ≤ 208 spring terminal block with I/O expansion module ≤ 264 HE-10 connector with I/O expansion module |
| Network frequency | 50/60 Hz |
| Supply voltage limits | 85...264 V |
| Network frequency limits | 47...63 Hz |
| Power supply output current | 0.4 A 24 V DC sensors |
| Power supply input current | 790 mA |
| Inrush current | ≤ 35 A |
| Protection type | Power protection internal fuse |
| Power consumption in VA | 65 VA 100 V 77 VA 264 V |
| Insulation resistance | > 10 MOhm at 500 V, between supply and earth terminals > 10 MOhm at 500 V, between I/O and earth terminals |
| Program memory | 3000 instructions |
| Exact time for 1 K instruction | 1 ms |
| System overhead | 0.5 ms |
| Memory description | Internal RAM 256 internal bits, no floating, no trigonometrical Internal RAM 3000 internal words, no floating, no trigonometrical Internal RAM 128 timers, no floating, no trigonometrical Internal RAM 128 counters, no floating, no trigonometrical Internal RAM double words, no floating, no trigonometrical Internal RAM floating, trigonometrical |
| Free slots | 1 |
| Realtime clock | With ≤ 30 s/month 30 days |
| Port Ethernet | 10BASE-T/100BASE-TX |
| Communication service | BOOTP client Ethernet TCP/IP Modbus messaging Ethernet TCP/IP |
| Positioning functions | PWM/PLS 2.7 kHz |
| Counting input number | 2 20000 Hz 32 bits 4 5000 Hz 16 bits |
| Analogue adjustment points | 1 point adjustable from 0...1023 1 point adjustable from 0 to 511 points |
| Marking | CE |
| Status LED | 1 LED green PWR 1 LED green RUN 1 LED red module error (ERR) 1 LED user pilot light (STAT) 1 LED Ethernet status (LAN ST) 1 LED 10 or 100 Mbit/s rate (LACT) 1 LED per channel green I/O status |
| Product weight | 0.525 kg |

Environment

| | |
|---------------------------------------|---|
| Immunity to microbreaks | 10 ms |
| Dielectric strength | 1500 V for 1 minute, between supply and earth terminals 1500 V for 1 minute, between I/O and earth terminals |
| Product certifications | CSA UL |
| Ambient air temperature for operation | 0...55 °C |
| Ambient air temperature for storage | -25...70 °C |
| Relative humidity | 30...95 % without condensation |
| IP degree of protection | IP20 |
| Operating altitude | 0...2000 m |
| Storage altitude | 0...3000 m |

ANEXO 8

Datasheet

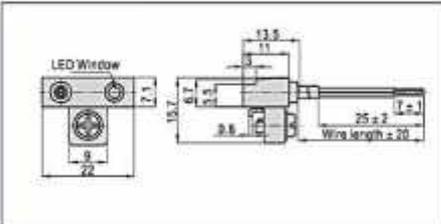
CS1-J Series

Sensor switch

CS1-J Series



Dimensions



Ordering code

Model can be changed Ordering code. Example:
 Specification of sensor switch: CS1-J type
 Model of sensor switch: two-line magnetic spring pipe with contact, without indicator light/normally opened
 Connecting way: length of wire is 2m
Model: CS1-J X-020
Ordering code: CS1 J X 020

| | |
|-------------------------|---|
| Number of sensor switch | Connecting way |
| CS1: Sensor switch | C08: M8 quick joint, length of wire is 150mm |
| | C12: M12 quick joint, length of wire is 150mm |
| | 020: length of wire is 2m |
| | 030: length of wire is 3m |
| | 050: length of wire is 5m |
| | 100: length of wire is 10m |

| | |
|------------------------|--|
| Model of sensor switch | Blank: two-line magnetic spring pipe with contact/normally opened |
| | X: two-line magnetic spring pipe with contact, without indicator light/normally opened |

Specification

| Item/Type | CS1-J | CS1-JX |
|------------------------------|--|----------------------|
| Switch logic | STSP | Normally opened type |
| Switch type | Reed switch with contact | |
| Operating voltage(V) | 5-240V AC/DC | |
| Max. Switching current(mA) | 100 | |
| Switching rating(W) | Max. 10 | |
| Current consumption | No | |
| Voltage drop | 2.5V Max. @100mA DC | |
| Cable | Φ3.32C Gray oil resistant PVC (Flame retarded) | |
| Indicator | Red LED | No |
| Leakage current | No | |
| Sensitivity(Gauss) | 60-75 | |
| Max. Frequency(Hz) | 200 | |
| Shock(m/s ²) | 300 | |
| Vibration(m/s ²) | 90 | |
| Temperature range(°C) (1) | -10-20 | |
| Enclosure classification | IP67(NEMA6) | |
| Protection circuit | No | |

(1) Note: Please contact us for high temperature resistant(125°C), low temperature resistant(-40---25°C) and explosion-proof sensor switch.

Explain of model

CS1-J X-020

| | | |
|-------------------------|--------------------|---|
| Number of sensor switch | CS1: Sensor switch | (1) Connecting way |
| | | C08: M8 quick joint, length of wire is 150mm |
| | | C12: M12 quick joint, length of wire is 150mm |
| | | 020: length of wire is 2m |
| | | 030: length of wire is 3m |
| | | 050: length of wire is 5m |
| | | 100: length of wire is 10m |

| | |
|--------------------------------|--------------------------|
| Specification of sensor switch | Product Series |
| J: J type | SDA/TN/TWH/TWM/AOC32-100 |
| | TWQ32-50/QCK32-63 |

Model of sensor switch
 Blank: two-line magnetic spring pipe with contact/normally opened
 X: two-line magnetic spring pipe with contact, without indicator light/normally opened

(1) Note: The quick joint that is attached at the end of wire is three-needle-male joint-linear-rotary screw thread type. The female joint plug has to be ordered additionally. Please refer to P428 for the specific data.

Mounting

| CS1-J, CS1-JX(AQC, SDA, TN, TWH, TWM, TWQ Series) | |
|---|---|
| Installation example | Installation method |
| | <p>No additional accessories are necessary for the sensor switch of CS1-J, CS1-JX series. It can be directly fixed onto the cylinder, which is convenient and fast.</p> <p>1. Loosen the clamping screw, slide the inductive switch to the slot and adjust it to the proper position. Tighten the clamping screw to fix.</p> |
| | |

ANEXO 9

Datasheet

GP2Y0A41SK0F

SHARP (4-30cm)

■Absolute maximum ratings

(Ta=25°C, Vcc=5V)

| Parameter | Symbol | Ratings | Unit | Remark |
|-------------------------|--------|-----------------|------|--------|
| Supply voltage | Vcc | -0.3 to +7 | V | - |
| Output terminal voltage | Vo | -0.3 to Vcc+0.3 | V | - |
| Operating temperature | Topr | -10 to +60 | °C | - |
| Storage temperature | Tstg | -40 to +70 | °C | - |

■Operating supply voltage

| Symbol | Rating | Unit | Remark |
|--------|------------|------|--------|
| Vcc | 4.5 to 5.5 | V | - |

■Electro-optical Characteristics

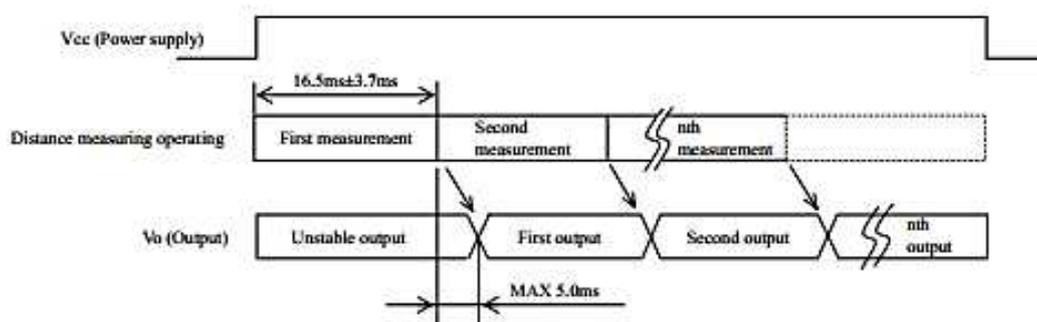
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---------------------------|-------------|--|------|------|------|------|
| Measuring distance range | ΔL | (Note 1) | 4 | - | 30 | Cm |
| Output terminal voltage | Vo | L=30cm (Note 1) | 0.25 | 0.4 | 0.55 | V |
| Output voltage difference | ΔVo | Output change at L change (30cm → 4cm) (Note 1) | 1.95 | 2.25 | 2.55 | V |
| Average supply current | Icc | L=30cm (Note 1) | - | 12 | 22 | mA |

※L : Distance to reflective object

(Note 1) Using reflective object : White paper

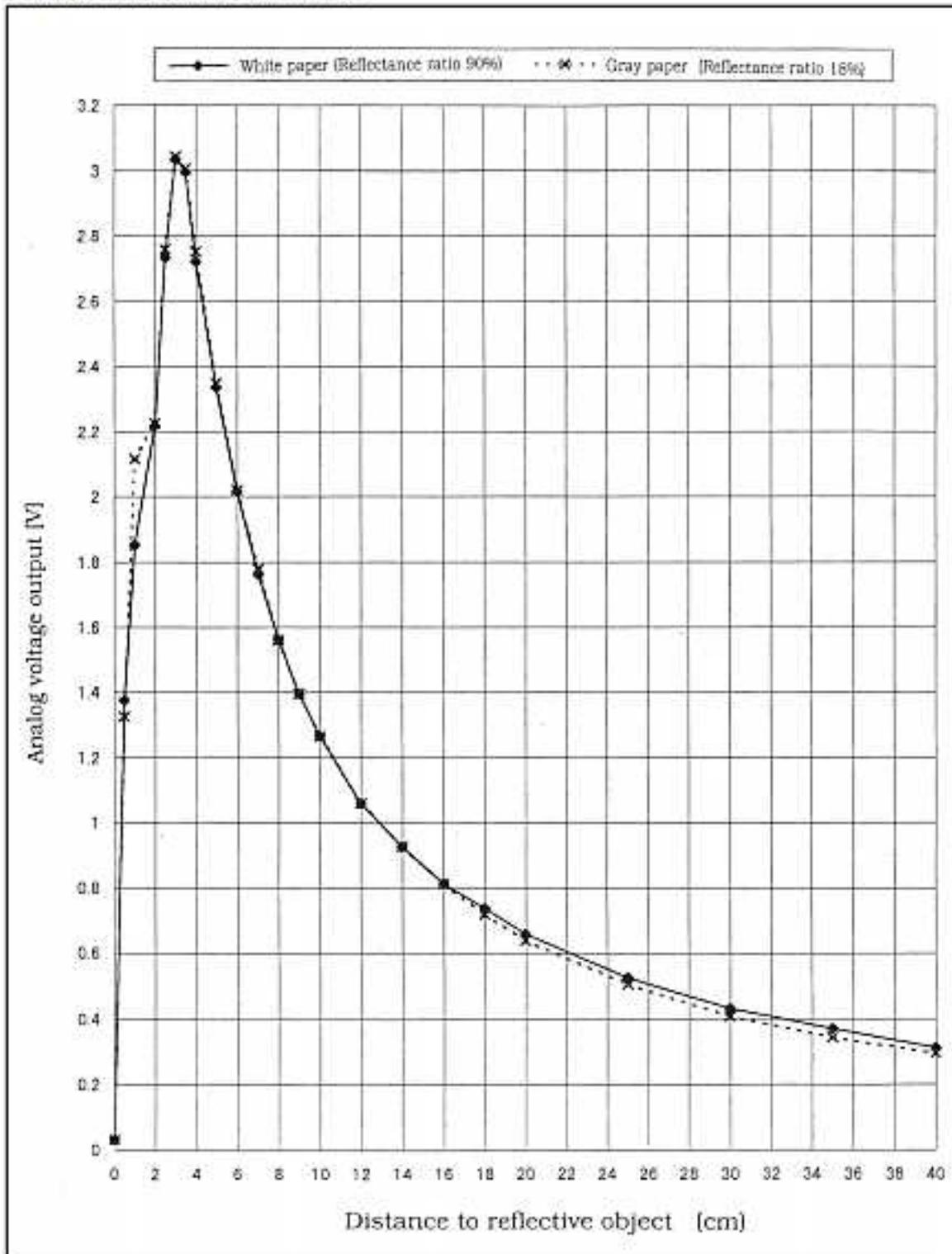
(Made by Kodak Co., Ltd. gray cards R-27 • white face, reflective ratio ; 90%)

■Timing Chart



Supplements

• Example of output distance characteristics



ANEXO 10

CÓDIGO DE

PROGRAMA ARDUINO

```

#define encoder1PinA 2
#define encoder1PinB 3
#define encoder2PinA 21
#define encoder2PinB 20
#define encoder3PinA 19
#define encoder3PinB 18
#include <DistanceGP2Y0A21YK.h>

```

```
//variables sensores
```

```
int a1=40;
int val=0;
int v1,v2,v3,v4,v5,v6,v7,v8,v9;
```

```
String instring1=" ";
String instring2=" ";
String instring3=" ";
char str [22];
char str1 [7];
char str2 [7];
char str3 [7];
float tiempo;
float d1;
float rev1,rev2,rev3;
float S1,S2,S3; //distancia recorrida
volatile unsigned int count1 = 0;
volatile unsigned int count2 = 0;
volatile unsigned int count3 = 0;
int i;
int Wa,Wb,Wc;
float W1,W2,W3;
DistanceGP2Y0A21YK
D1,D2,D3,D4,D5,D6,D7,D8,D9;
```

```

void setup () {
// declaracion entradas y salidas sensores
//seguidor
pinMode(a1,INPUT);
//motores
pinMode(26,OUTPUT);
pinMode(27,OUTPUT);
pinMode(28,OUTPUT);
pinMode(29,OUTPUT);
pinMode(30,OUTPUT);
pinMode(31,OUTPUT);
//encoder
pinMode(encoder1PinA, INPUT);
pinMode(encoder2PinA, INPUT);
pinMode(encoder1PinB, INPUT);
pinMode(encoder2PinB, INPUT);
digitalWrite(encoder1PinA, HIGH);
digitalWrite(encoder1PinB, HIGH);

```

```

digitalWrite(encoder2PinA, HIGH);
digitalWrite(encoder2PinB, HIGH);

attachInterrupt(0, rencoder1, CHANGE);
attachInterrupt(2, rencoder2, CHANGE);
attachInterrupt(4, rencoder3, CHANGE);
// attachInterrupt(5,seguidor,LOW);

```

```
//sensores
```

```
D1.begin(A0);
D2.begin(A1);
D3.begin(A2);
D4.begin(A3);
D5.begin(A4);
D6.begin(A5);
D7.begin(A6);
D8.begin(A7);
D9.begin(A8);
```

```

Serial.begin(9600);
Serial2.begin(9600);
}

```

```
//calculo encoder 1
```

```

void rencoder1() {

if (digitalRead(encoder1PinA)==LOW &&
digitalRead(encoder1PinB)==HIGH){
count1++;
}
else if(digitalRead(encoder1PinA)==LOW &&
digitalRead(encoder1PinB)==LOW){
count1--;
}
else if(digitalRead(encoder1PinA)==HIGH
&& digitalRead(encoder1PinB)==HIGH){
count1--;
}
else {
count1++;
}
//pos=((count/1000)*360); // divide count,
multiply by 360 to get degrees.
}

```

```
//calculo encoder 2
```

```
void rencoder2() {
```

```

    if (digitalRead(encoder2PinA)==LOW &&
digitalRead(encoder2PinB)==HIGH){
    count2++;
}
else if(digitalRead(encoder2PinA)==LOW &&
digitalRead(encoder2PinB)==LOW){
    count2--;
}
else if(digitalRead(encoder2PinA)==HIGH
&& digitalRead(encoder2PinB)==HIGH){
    count2--;
}
else {
    count2++;
}
//pos=((count/1000)*360);
// divide count, multiply by 360 to get
degrees.
}

//calculo encoder 3
void rencoder3() {
    if (digitalRead(encoder3PinA)==LOW &&
digitalRead(encoder3PinB)==HIGH){
    count3++;
}
else if(digitalRead(encoder3PinA)==LOW &&
digitalRead(encoder3PinB)==LOW){
    count3--;
}
else if(digitalRead(encoder3PinA)==HIGH
&& digitalRead(encoder3PinB)==HIGH){
    count3--;
}
else {
    count3++;
}
}
//pos=((count/1000)*360);
// divide count, multiply by 360 to get
degrees.
}

//movimientos motores
void adelante () {
    digitalWrite(26,LOW);
    digitalWrite(27,LOW);
    digitalWrite(28,HIGH);
    digitalWrite(29,LOW);
    digitalWrite(30,HIGH);
    digitalWrite(31,LOW);
}

}

void paro () {
    digitalWrite(26,HIGH);
    digitalWrite(27,HIGH);
    digitalWrite(28,HIGH);
    digitalWrite(29,HIGH);
    digitalWrite(30,HIGH);
    digitalWrite(31,HIGH);
}

void atras () {
    digitalWrite(26,LOW);
    digitalWrite(27,LOW);
    digitalWrite(28,LOW);
    digitalWrite(29,HIGH);
    digitalWrite(30,LOW);
    digitalWrite(31,HIGH);
}

void giroderecha () {
    digitalWrite(26,HIGH);
    digitalWrite(27,LOW);
    digitalWrite(28,LOW);
    digitalWrite(29,HIGH);
    digitalWrite(30,HIGH);
    digitalWrite(31,LOW);
}

void giroizquierda () {
    digitalWrite(26,LOW);
    digitalWrite(27,HIGH);
    digitalWrite(28,HIGH);
    digitalWrite(29,LOW);
    digitalWrite(30,LOW);
    digitalWrite(31,HIGH);
}

void avanceizquierda () {
    digitalWrite(26,LOW);
    digitalWrite(27,LOW);
    digitalWrite(28,LOW);
    digitalWrite(29,LOW);
    digitalWrite(30,HIGH);
    digitalWrite(31,LOW);
}

void avancederecha () {
    digitalWrite(26,LOW);
    digitalWrite(27,LOW);
    digitalWrite(28,LOW);
    digitalWrite(29,HIGH);
    digitalWrite(30,LOW);
    digitalWrite(31,LOW);
}
}

```

```

void aizqd () {
  digitalWrite(26,HIGH);
  digitalWrite(27,LOW);
  digitalWrite(28,LOW);
  digitalWrite(29,HIGH);
  digitalWrite(30,LOW);
  digitalWrite(31,LOW);
}
void aderd () {
  digitalWrite(26,LOW);
  digitalWrite(27,HIGH);
  digitalWrite(28,LOW);
  digitalWrite(29,LOW);
  digitalWrite(30,HIGH);
  digitalWrite(31,LOW);
}
void rizqd () {
  digitalWrite(26,HIGH);
  digitalWrite(27,LOW);
  digitalWrite(28,LOW);
  digitalWrite(29,LOW);
  digitalWrite(30,LOW);
  digitalWrite(31,HIGH);
}
void rderd () {
  digitalWrite(26,LOW);
  digitalWrite(27,HIGH);
  digitalWrite(28,LOW);
  digitalWrite(29,HIGH);
  digitalWrite(30,LOW);
  digitalWrite(31,LOW);
}

void seguidor(){
seguidor:

if (v1<=20) {
  if (digitalRead(a1)== HIGH ){
    avancederecha();
    delay (40);
  }
  else if(digitalRead(a1)== LOW){
    avanceizquierda();
    delay (40);
  }
}
goto seguidor;
}
else {
  paro();
}
}

// === MAIN PROGRAM LOOP ===

void loop() {

//inicio sensores
val=digitalRead (a1);
v1=D1.getDistanceCentimeter();
v2=D2.getDistanceCentimeter();
v3=D3.getDistanceCentimeter();
v4=D4.getDistanceCentimeter();
v5=D5.getDistanceCentimeter();
v6=D6.getDistanceCentimeter();
v7=D7.getDistanceCentimeter();
v8=D8.getDistanceCentimeter();
v9=D9.getDistanceCentimeter();

//sensores lectura
Serial2.println(v1);
Serial2.println(v2);
Serial2.println(v3);
Serial2.println(v4);
Serial2.println(v5);
Serial2.println(v6);
Serial2.println(v7);
Serial2.println(v8);
Serial2.println(v9);

if(v1<=14 || v2<=14 || v3<=14 || v4<=14 ||
v5<=14 || v6<=14 || v7<=14 || v8<=14 )
{
  paro();
}
else
{
  if (Serial2.available())>0){
    Serial2.readBytes(str,21);
// if (Serial.read() == ' '){

    for (i = 0; i < 7; i++){
      str1[i] = str[i];
      str2[i] = str[i+7];
      str3[i] = str[i+14];
    }

    Serial2.println ("datos");
    Serial2.println (str);

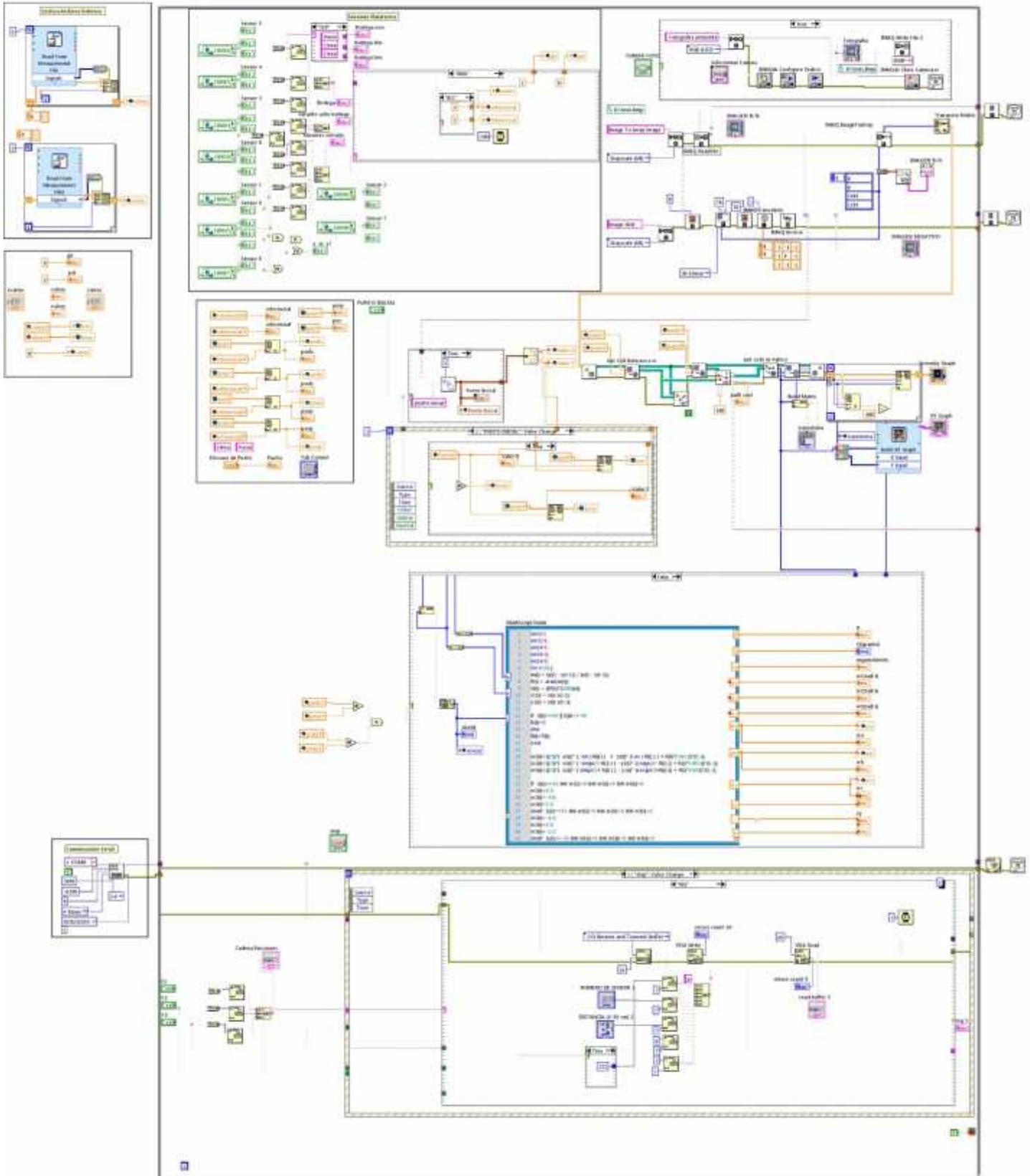
    instring1=str1;
    instring2=str2;
    instring3=str3;

```


ANEXO 11

CÓDIGO DE

PROGRAMA LABVIEW



Debug Deployment

hmi version 1.3 - copia - copia.vi

G:\prueba tesis\hmi version 1.3 - copia - copia.vi

Last modified on 19/04/2015 at 15:15

Printed on 19/04/2015 at 15:15

