UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO-CAMPUS SUR

CARRERA DE INGENIERÍA ELECTRÓNICA

MENCIÓN SISTEMAS INDUSTRIALES

TESIS PREVIA A LA OBTENCIÓN DEL TITULO DE INGENIERO ELECTRÓNICO

"Implementación de un prototipo para el control y monitoreo de un proceso industrial mediante el acoplamiento sincronizado del robot móvil Robotino de Festo y los brazos robóticos articulados RV-2AJ de Mitsubishi."

AUTORES:

LISBETH KARINA MENA LÓPEZ

JOHANNA PATRICIA SALVADOR LÓPEZ

DIRECTOR: ING. RODRIGO TUFIÑO

QUITO, FEBRERO 2012

DECLARACIÓN

Nosotras, Lisbeth Karina Mena López y Johanna Patricia Salvador López, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional, y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Lisbeth Karina Mena	**Johanna Patricia Salvador**

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Lisbeth Karina Mena López y Johanna Patricia Salvador López, bajo mi dirección.

Ing. Rodrigo Tufiño
DIRECTOR DE TESIS

AGRADECIMIENTO

A Dios por su infinita bondad; a mis padres, por su sacrificio, entrega y amor incondicional; a mi hermano, por sus enseñanzas, su apoyo y por ser la fuerza que me ha permitido aceptar retos y vencer obstáculos; a mi amor, por acompañarme en mis alegrías y tristezas.

Al Ing. Germán Arévalo por su apoyo y confianza.

A mis amigas, en especial a Annie con quien hemos compartido gratos momentos.

Infinitas gracias para el Ing. Rodrigo Tufiño quien ha brindado su dedicación a este proyecto, mi gratitud por haber sido un excelente guía y amigo.

Lisbeth Mena

DEDICATORIA

A quienes comparten mis triunfos:

A mis padres Don Gabrielito y Doña Lucita, a mi Ñañito y a Gabito Garzón.

Lisbeth Mena

AGRADECIMIENTO

A Dios por cada día de vida y enseñanza, a mi Papi porque con su ejemplo, fuerza y perseverancia me ha enseñado a luchar y me ha apoyado para cumplir mis metas y sueños, a mi mami Marci que me impulsa cada día a ser mejor, a Tita que más que mi hermana es mi amiga y confidente y ha estado junto a mi sobrellevando las adversidades. A mi hermano Santy que me inculco la electrónica desde la escuela. A mis sobrinos Gabo e Isaac que con una sonrisa alimentan mi alma. A Tommy por estar junto a mí cuando menos lo esperaba, gracias por su apoyo incondicional. A Gina, Dianita, Denisse, Gabo, que han sido grandes amigos de la Universidad. Un especial agradecimiento a Lis que juntas hemos impulsado y sobrellevado el presente trabajo, gracias por confiar en mí, por tu inmensa paciencia y gran corazón.

A Rodry por su inmenso apoyo y confianza.

Al Ing. Germán Arévalo, profesor René y Ramirito porque han sido de gran aporte para mi formación personal, espiritual y educativa.

Johanna Salvador

DEDICATORIA

La presente tesis se la dedico a mi familia que gracias a su apoyo y confianza pude concluir con éxito mi carrera. A quienes han compartido mis triunfos Andy, Tomy, Lis, Gia, Dianita, Gabo.

A todos quienes colaboraron en mi formación técnica educativa.

Johanna Salvador

CONTENIDO

INTRODUCCIÓN 1

CAPITULO 1 3

1. PLANTEAMIENTO DEL PROBLEMA 3

1.1 HIPÓTESIS 3

1.2 TESIS 3

1.3 OBJETIVOS 3

1.3.1 Objetivo general 3

1.3.2 Objetivos específicos 4

CAPITULO 2 5

2. ESTADO DEL ARTE 5

2.1 ROBÓTICA 5

2.1.1 Manipuladores 13

2.1.2 Manipulador RV-2AJ 17

2.1.3 Robótica móvil. 19

2.1.4 Robot móvil Robotino® de Festo 25

2.1.5 Robots híbridos. 27

2.2 PLC (CONTROLADOR LÓGICO PROGRAMABLE) 28

2.3 SENSORES 29

2.4 COMUNICACIÓN OPC. 35

2.5 INTERFAZ HMI. 38

CAPITULO 3 43

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN 43

3.1 DISEÑO 43

3.1.1 Diseño previo y diagrama de bloques del sistema. 43

3.1.2 Entorno de trabajo 44

3.1.3 Diseño Electrónico 48

3.1.4 Circuitos de control 50

3.1.5 Diseño del Software 52

3.1.5.1 Algoritmo del sistema 52

3.1.5.2 Algoritmo del PLC 54

3.1.5.3 Algoritmo del Brazo Robótico (estación de ensamblaje) 55

3.1.5.4 Algoritmo del Brazo Robótico (estación de clasificación y almacenamiento) 64

3.1.5.5 Algoritmo del robot móvil Robotino 67

3.1.6 Diseño de la comunicación OPC 70

3.1.7 Diseño de la interfaz de usuario 71

3.2 DESARROLLO 73

3.2.1 Programación del robot móvil Robotino de Festo. 73

3.2.2 Programación de los brazos robóticos RV-2AJ. 77

3.2.2.1 Brazo 1 Ensamblaje de Piezas 77

3.2.2.2 Brazo2 Clasificación de Piezas 90

3.2.3 Programación del PLC. 94

3.3 IMPLEMENTACIÓN 102

3.3.1 Implementación de los circuitos electrónicos. 102

3.3.2 Implementación de la comunicación OPC 105

3.3.3 Implementación de la interfaz HMI. 117

3.3.4 Puesta en Marcha 127

3.3.1 Pruebas y Ajustes 135

3.3.2 Integración total del sistema 136

CAPITULO 4 149

4. ANALISIS DE RESULTADOS 149

4.1 IDENTIFICACIÓN DE PARÁMETROS PARA COTEJAR RESULTADOS. 149

4.2 PRESENTACIÓN DE RESULTADOS. 150

4.3 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS OBTENIDOS. 155

CAPITULO 5 158

5. CONCLUSIONES Y RECOMENDACIONES 158

5.1 CONCLUSIONES 158

5.2 RECOMENDACIONES 159

BIBLIOGRAFIA 160

ANEXOS 162

ANEXO 1 162

DATOS TÉCNICOS 162

ROBOT MÓVIL ROBOTINO DE FESTO 162

BRAZO ROBÓTICO RV-2AJ: 164

PLC S7-200 CPU 222: 165

EASYPORT USB 168

ANEXO 2 169

ACTUALIZACION ROBOTINO 169

ANEXO 3 175

INDICE DE FIGURAS Y TABLAS

INTRODUCCIÓN 1

CAPITULO 1 3

1. PLANTEAMIENTO DEL PROBLEMA 3

CAPITULO 2 5

2. ESTADO DEL ARTE 5

TABLA 2.1 HISTORIA DE LOS ROBOTS 7

FIGURA 2.1 PATO ARTIFICIAL QUE REALIZÓ VAUCANSON EN 1738 BEBÍA, COMÍA Y DIGERÍA. 8

[FUENTE: "ATLAS ILUSTRADO DE LOS ROBOTS DE LEONARDO DA VINCI ", TADDEI, MARIO] 8

FIGURA 2.2 USO DE ROBOTS INDUSTRIALES DURANTE 2008 – 2009 10

[FUENTE: HTTP://WWW.IFR.ORG/UPLOADS/MEDIA/2010_EXECUTIVE_SUMMARY_REV.PDF] 10

FIGURA 2.3 POBLACIONES ESTIMADAS EN CADA CONTINENTE DE ROBOTS INDUSTRIALES 11

[FUENTE: HTTP://WWW.IFR.ORG/UPLOADS/MEDIA/2010_EXECUTIVE_SUMMARY_REV.PDF] 11

TABLA 2.2 CIFRAS ESTIMADAS DE EXISTENCIA DE ROBOTS INDUSTRIALES A NIVEL MUNDIAL 12

[FUENTE: HTTP://WWW.IFR.ORG/UPLOADS/MEDIA/2010_EXECUTIVE_SUMMARY_REV.PDF] 12

FIGURA 2.4 MANIPULADOR QUIRÚRGICO 13

[FUENTE: HTTP://WWW.INTRAMED.NET] 13

FIGURA 2.5 ESTRUCTURA MECÁNICA DEL MANIPULADOR RV 2AJ 14

[FUENTE: MANUAL BRAZO MITSUBISHI RV 2AJ] 14

FIGURA 2.6 TIPOS DE ARTICULACIONES 15

[FUENTE: "FUNDAMENTOS DE ROBÓTICA" BARRIENTOS, ANTONIO] 15

FIGURA 2.7 TIPOS DE CONFIGURACIONES MÁS FRECUENTES 16

[FUENTE: "FUNDAMENTOS DE ROBÓTICA" BARRIENTOS, ANTONIO] 16

FIGURA 2.8 ROBOT PARA SOLDADURA 17

[FUENTE: HTTP://INNOVATIONTECNOLOGY.BLOGSPOT.COM] 17

FIGURA2.9 BRAZO ROBÓTICO RV-2AJ 18

[FUENTE: MANUAL RV-2AJ, MITSUBISHI] 18

FIGURA 2.10 BOTONERA DE MANDO 19

[FUENTE: "MANUAL, TEACHINGPENDANT (T/B)".] 19

FIGURA 2.11 EL CARRITO AUTOMOTOR VS. ROBOT MÓVIL DE LA NASA..... 20

[FUENTE: "ATLAS ILUSTRADO DE LOS ROBOTS DE LEONARDO DA VINCI", TADDEI, MARIO] 20

FIGURA 2.12 LEÓN ROBÓTICO DE LEONARDO VS. PERRO AIBO DE SONY 21

[FUENTE: "ATLAS ILUSTRADO DE LOS ROBOTS DE LEONARDO DA VINCI", TADDEI, MARIO] 21

TABLA 2.3 SISTEMAS DE LOCOMOCIÓN..... 23

FIGURA 2.13 APLICACIONES DE ROBOTS MÓVILES 25

[FUENTE: HTTP://MEDGADGET.ES/2011/05/_INVESTIGADORES_DESARROLLAN_UN.HTML] 25

FIGURA 2.14ROBOTINO 26

[FUENTE: HTTP://WWW.FESTO.COM/NET/STARTPAGE/ROBOTINO]..... 26

FIGURA 2.15 ROBOT ASIMO UNO DE LOS ROBOTS HUMANOIDES MÁS AVANZADOS 27

[FUENTE: HTTP://WWW.UNIVERSIDADPERU.COM/ROBOT-CAMINADOR-DE-HONDA-BLOG.PHP] 27

FIGURA 2.16 ARQUITECTURA INTERNA DEL PLC..... 29

[FUENTE: HTTP://WWW.EL.BQTO.UNEXPO.EDU.VE/LAB_PLC/PAGINAS/ESTRUCTURA.HTML] 29

FIGURA 2.17 DIAGRAMA DE BLOQUES DE LOS ELEMENTOS DE UN SENSOR 30

[FUENTE: HTTP://TODOPRODUCTIVIDAD.BLOGSPOT.COM/2010/03/SOBRE-LOS-SENSORES-INTELIGENTES.HTML]

..... 30

FIGURA 2.18 CLASIFICACIÓN DE SENSORES 33

[FUENTE: "INTRODUCCIÓN A LA ROBÓTICA" DE SAHA, SUBIR.] 33

FIGURA 2.19 GENERACIONES DE SENSORES INTELIGENTES 35

[FUENTE: J. E. BRIGNELL, "SMART SENSORS. A COMPREHENSIVE SURVEY," UNIVERSITY OF SOUTHAMPTON,

UK.ISBN. 1989] 35

FIGURA 2.20 PROBLEMA DE COMUNICACIÓN SIN TECNOLOGÍA OPC. 36

[FUENTE: HTTP://ES.WIKIPEDIA.ORG/WIKI/OPC]..... 36

FIGURA 2.21SOLUCIÓN AL PROBLEMA AL CONTAR CON TECNOLOGÍA OPC. 37

[FUENTE: HTTP://ES.WIKIPEDIA.ORG/WIKI/OPC]..... 37

FIGURA 2.22 COMUNICACIÓN OPC 38

[FUENTE: AUTORAS TESIS] 38

FIGURA 2.23XEROX ALTO Y SU INTERFAZ DE USUARIO..... 39

[FUENTE: HTTP://WWW.TEXTUALMENTEACTIVO.COM/ARCHIVES/4228] 39

FIGURA 2.24 DIVERSOS EQUIPOS ELECTRÓNICOS CON INTERFACES DE USUARIO. 39

[FUENTE: HTTP://COMPLEMENTOSINFORMATICOS.BLOGSPOT.COM/2011/07/CONSEJOS-PARA-ELEGIR-UNA-TABLET.HTML] 39

FIGURA 2.25INTERFAZ HMI DEL SOFTWARE IN TOUCH 41

[FUENTE: HTTP://PROCESOSLAGOS.BLOGSPOT.COM/2010/05/CHILE-EN-EL-MUNDIAL_05.HTML] 42

CAPITULO 3 43

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN 43

FIGURA 3.1 DIAGRAMA DE BLOQUES DEL SISTEMA 43

[FUENTE: AUTORAS TESIS] 43

FIGURA 3.2 DISEÑO DEL ENTORNO DEL SISTEMA. 44

[FUENTE: AUTORAS TESIS] 44

FIGURA 3.3 ESTACIÓN DE ENSAMBLAJE. 45

[FUENTE: AUTORAS TESIS] 45

FIGURA 3.4 UBICACIÓN DE ÉMBOLO, MUELLE Y TAPA EN EL CILINDRO. 46

[FUENTE: CÉLULA DE FABRICACIÓN FLEXIBLE, MPS-C ESTACIÓN 6: ROBOT FESTO DIDACTIC-2003] 46

FIGURA 3.5 ROBOT MÓVIL 46

[FUENTE: HTTP://WWW.FESTO.COM/NET/STARTPAGE/ROBOTINO] 46

FIGURA 3.6 ESTACIÓN DE CLASIFICACIÓN. 47

[FUENTE: AUTORAS TESIS] 47

FIGURA 3.7 DISEÑO DE ACOPLAMIENTO DE SENSORES Y MOTORES REALIZADA EN PROTEUS 7 PROFESSIONAL 49

[FUENTE: AUTORAS TESIS] 49

FIGURA 3.8 DISEÑO DE LAS CONEXIONES PARA PLC. 51

[FUENTE: AUTORAS TESIS] 51

TABLA 3.1 ENTRADAS Y SALIDAS DE PLC. 51

[FUENTE: AUTORAS TESIS] 51

FIGURA 3.8 DIAGRAMA DE BLOQUES DE COMUNICACIÓN OPC. 70

[FUENTE: AUTORAS TESIS] 70

FIGURA 3.9 DISEÑO DE PANTALLA DE INICIO DE INTERFAZ HMI..... 72

[FUENTE: AUTORAS TESIS] 72

FIGURA 3.10 DISEÑO DE PANTALLA PROCESO DE INTERFAZ HMI 73

[FUENTE: AUTORAS TESIS] 73

FIGURA 3.11 LISTA DE POSICIONES PARA LA ESTACIÓN DE ENSAMBLAJE 79

[FUENTE: AUTORAS TESIS] 79

FIGURA 3.12 LISTA DE POSICIONES PARA ESTACIÓN DE CLASIFICACIÓN 90

[FUENTE: AUTORAS TESIS] 90

FIGURA 3.13 RUTEADO DE PISTAS DE CIRCUITO DE ACONDICIONAMIENTO DE SENSORES Y MOTORES EN PROTEUS 7 PROFESSIONAL..... 103

[FUENTE: AUTORAS TESIS]..... 103

FIGURA 3.14 RUTEADO DE PISTAS DE CIRCUITO DE LADO 1..... 104

[FUENTE: AUTORAS TESIS]..... 104

FIGURA 3.15 RUTEADO DE PISTAS DE CIRCUITO DE LADO 2. 104

[FUENTE: AUTORAS TESIS]..... 104

FIGURA 3.16 PISTAS IMPRESAS Y CIRCUITO FINAL DE ACONDICIONAMIENTO DE SENSORES Y MOTORES DEL PALLET 105

[FUENTE: AUTORAS TESIS] 105

FIGURA 3.17 VENTANA DE PROGRAMA EASYPORT DEMO APLICATION 106

[FUENTE: AUTORAS TESIS] 106

FIGURA 3.18 EASYPORT CONECTADO Y ENLAZADO AL MÓDULO 3 DEL COM4..... 107

[FUENTE: AUTORAS TESIS] 107

FIGURA 3.19 VENTANA DE EZOPC V5.3 108

[FUENTE: AUTORAS TESIS] 108

FIGURA 3.20 VENTANA DEL SIMULADOR S7-PLCSIM 108

[FUENTE: AUTORAS TESIS] 108

FIGURA 3.21 VIRTUAL CONTROLLER, EASYPORT Y S7-PLCSIM CONECTADOS. 109

[FUENTE: AUTORAS TESIS] 109

FIGURA 3.22 EASYPORTENLAZADOCON EL SERVIDOR 110

[FUENTE: AUTORAS TESIS] 110

FIGURA 3.23 MENÚ DE ROBOTINOVIEW PARA LIBRERÍA OPC CLIENTE..... 111

[FUENTE: AUTORAS TESIS] 111

FIGURA 3.24 VENTANA DE OPC CLIENTE DE ROBOTINOVIEW. 112

[FUENTE: AUTORAS TESIS] 112

FIGURA 3.25 OPC CLIENTE CONECTADO 112

[FUENTE: AUTORAS TESIS] 112

FIGURA 3.26 VERIFICACIÓN DE CONECTIVIDAD EN ROBOTINOVIEW. 113

[FUENTE: AUTORAS TESIS] 113

FIGURA 3.27 VERIFICACIÓN DE CONECTIVIDAD EN EZOPC. 114

[FUENTE: AUTORAS TESIS] 114

FIGURA 3.28 VERIFICACIÓN DE ENVÍO Y RECEPCIÓN DE DATOS EN ROBOTINOVIEW. 115

[FUENTE: AUTORAS TESIS] 115

FIGURA 3.29 VERIFICACIÓN DE ENVÍO Y RECEPCIÓN DE DATOS EN EZOPC. 115

[FUENTE: AUTORAS TESIS] 115

FIGURA 3.30 CONEXIONES FÍSICAS PARA COMUNICACIÓN OPC..... 116

[FUENTE: AUTORAS TESIS] 116

FIGURA 3.31 EASYPORT ENVIANDO Y RECIBIENDO DATOS A TRAVÉS DE LA COMUNICACIÓN OPC. 117

[FUENTE: AUTORAS TESIS] 117

FIGURA 3.32 EDUTRAINER EN FUNCIONAMIENTO. 117

[FUENTE: AUTORAS TESIS] 117

FIGURA 3.33 VENTANAS CREADAS EN INTOUCH. 118

[FUENTE: AUTORAS TESIS] 118

FIGURA 3.34 VENTANA PRINCIPAL 118

[FUENTE: AUTORAS TESIS]..... 118

FIGURA 3.35 VENTANA DE OPCIONES PARA ANIMACIÓN. 119

[FUENTE: AUTORAS TESIS]..... 119

FIGURA 3.36 VENTANA PARA MOSTRAR Y DIRECCIONAR A OTRAS VENTANAS. 119

[FUENTE: AUTORAS TESIS]..... 119

FIGURA 3.37 VENTANA DE PROCESO 120

[FUENTE: AUTORAS TESIS] 120

FIGURA 3.38 VENTANA DEL DICCIONARIO DE TAGNAMES. 122

[FUENTE: AUTORAS TESIS] 122

FIGURA 3.39 VENTANA PARA CREAR ACCESS NAME 122

[FUENTE: AUTORAS TESIS] 122

FIGURA 3.40 VENTANA PARA CONFIGURACIÓN DE ACCESS NAME. 123

[FUENTE: AUTORAS TESIS] 123

FIGURA 3.41 VENTANA DEL DICCIONARIO DE TAGNAMES. 123

[FUENTE: AUTORAS TESIS] 123

TABLA 3.2 MAPA DE LAS DIRECCIONES DE PLC, INTOUCH Y BRAZOS ROBÓTICOS 125

[FUENTE: AUTORAS TESIS] 125

FIGURA 3.42 LISTA DE TAGNAMES. 125

[FUENTE: AUTORAS TESIS] 125

FIGURA 3.43 VENTANA DE OPCIONES PARA ANIMACIÓN. 126

[FUENTE: AUTORAS TESIS] 126

FIGURA 3.44 VENTANA PARA CONFIGURAR CONDICIONES DE VISIBILIDAD. 126

[FUENTE: AUTORAS TESIS] 126

FIGURA 3.45 CONTADOR DE PIEZAS ARMADAS 127

[FUENTE: AUTORAS TESIS] 127

FIGURA 3.46 IO SERVER PARA COMUNICAR PLC S7200 CON INTOUCH 128

[FUENTE: AUTORAS TESIS] 128

FIGURA 3.47 MENÚ DE CONFIGURACIÓN DE S7200PPI 128

[FUENTE: AUTORAS TESIS] 128

FIGURA 3.48 VENTANA DE CONFIGURACIÓN DEL PUERTO DE COMUNICACIÓN DE IO SERVER 129

[FUENTE: AUTORAS TESIS] 129

FIGURA 3.49 VENTANA DE CONFIGURACIÓN MODEM SETUP DE IO SERVER 129

[FUENTE: AUTORAS TESIS] 130

FIGURA 3.50 VENTANA DE TOPICDEFINITION DE IO SERVER 130

[FUENTE: AUTORAS TESIS] 130

FIGURA 3.51 VENTANA DE CONFIGURACIÓN DEL ACCESO DE COMUNICACIÓN DE IO SERVER..... 130

[FUENTE: AUTORAS TESIS] 130

FIGURA 3.52 VENTANA DE CONFIGURACIÓN PARA DDE SERVER DE IO SERVER 131

[FUENTE: AUTORAS TESIS] 131

FIGURA 3.53 CLIENTE DE IO SERVER PARA COMUNICAR PLC S7200 CON INTOUCH 131

[FUENTE: AUTORAS TESIS] 131

FIGURA 3.54 MENÚ PARA COMUNICACIÓN ENTRE CLIENTE Y SERVIDOR S7200PPI 132

[FUENTE: AUTORAS TESIS] 132

FIGURA 3.55 VENTANA DE CONFIGURACIÓN PARA INICIAR COMUNICACIÓN. 132

[FUENTE: AUTORAS TESIS] 132

FIGURA 3.56 COMUNICACIÓN INICIALIZADA..... 133

[FUENTE: AUTORAS TESIS] 133

FIGURA 3.57 VENTANA DE REQUEST PARA COMPROBAR LA COMUNICACIÓN. 133

[FUENTE: AUTORAS TESIS] 133

FIGURA 3.58 LECTURA DE PUERTOS FÍSICOS DE PLC S7200. 134

[FUENTE: AUTORAS TESIS] 134

FIGURA 3.59 COMPROBACIÓN DE COMUNICACIÓN EN INTOUCH. 135

[FUENTE: AUTORAS TESIS] 135

TABLA 3.3 PRUEBAS Y AJUSTES PARA SINCRONÍA DEL SISTEMA. 136

[FUENTE: AUTORAS TESIS] 136

FIGURA 3.60 DIAGRAMA DE LA INTEGRACIÓN DEL PROCESO..... 137

[FUENTE: AUTORAS TESIS] 137

FIGURA 3.61 CONEXIONES PARA PLC S7200 138

[FUENTE: AUTORAS TESIS] 138

FIGURA 3.62 PANEL DE CONTROL 138

[FUENTE: WWW.FESTO-DIDACTIC.COM/INT-EN/LEARNING-SYSTEM]..... 138

FIGURA 3.63 DIAGRAMA DE LA INTEGRACIÓN DE LOS BRAZOS ROBÓTICOS AL PROCESO. 139

[FUENTE: AUTORAS TESIS] 139

FIGURA 3.64 CONEXIONES PARA BORNERA DE BRAZO 1, ESTACIÓN DE ENSAMBLAJE. 140

[FUENTE: AUTORAS TESIS] 140

FIGURA 3.65 CONEXIONES PARA CONCENTRADORA DE BRAZO 1, ESTACIÓN DE ENSAMBLAJE..... 140

[FUENTE: AUTORAS TESIS] 140

FIGURA 3.66 CONEXIONES FÍSICAS PARA LA CONCENTRADORA BRAZO 1. ESTACIÓN DE ENSAMBLAJE 141

[FUENTE: AUTORAS TESIS] 141

FIGURA 3.67 CONEXIONES PARA BORNERA DE BRAZO 2, ESTACIÓN DE CLASIFICACIÓN. 141

[FUENTE: AUTORAS TESIS] 141

FIGURA 3.68 CONEXIONES PARA CONCENTRADORA DE BRAZO 2, ESTACIÓN DE CLASIFICACIÓN. 142

[FUENTE: AUTORAS TESIS] 142

FIGURA 3.69 CONEXIONES FÍSICAS PARA CONCENTRADORA DE BRAZO 2, ESTACIÓN DE CLASIFICACIÓN. 143

[FUENTE: AUTORAS TESIS] 143

FIGURA 3.70 DIAGRAMA DE LA INTEGRACIÓN DE ROBOTINO AL PROCESO. 144

[FUENTE: AUTORAS TESIS] 144

FIGURA 3.71 CONEXIONES PARA I/O DIGITALES DE EDUTRAINER, ESTACIÓN DE TRANSPORTE. 144

[FUENTE: AUTORAS TESIS] 144

FIGURA 3.72 ACOPLAMIENTO ESTACIÓN DE ENSAMBLAJE VS ENTORNO..... 145

[FUENTE: AUTORAS TESIS] 145

FIGURA 3.73 BANDEJA ESTACIÓN DE CLASIFICACIÓN 146

[FUENTE: AUTORAS TESIS] 146

FIGURA 3.74 ACOPLAMIENTO ESTACIÓN DE CLASIFICACIÓN VS ENTORNO 146

[FUENTE: AUTORAS TESIS] 146

FIGURA 3.75 PALLET 147

[FUENTE: AUTORAS TESIS] 147

FIGURA 3.76 SOPORTE PALLET..... 147

[FUENTE: AUTORAS TESIS] 147

FIGURA 3.77 ESTRUCTURA PARA CHASIS DEL ROBOTINO 148

[FUENTE: AUTORAS TESIS] 148

FIGURA 3.78 PALLET IMPLEMENTADO EN ROBOTINO. 148

[FUENTE: AUTORAS TESIS]..... 148

CAPITULO 4 149

4. ANALISIS DE RESULTADOS 149

TABLA 4.1 DATOS TOMADOS DE LA ESTACIÓN DE ENSAMBLAJE. 151

[FUENTE: AUTORAS TESIS]..... 151

TABLA 4.2 DATOS TOMADOS DE LOS MOTORES DEL PALLET. 151

[FUENTE: AUTORAS TESIS]..... 151

TABLA 4.3 DATOS TOMADOS DEL SEGUIDOR DE LÍNEA DE ROBOTINO. 152

[FUENTE: AUTORAS TESIS]..... 152

TABLA 4.4 DATOS TOMADOS DEL SEGUIDOR DE LÍNEA DE ROBOTINO. 153

[FUENTE: AUTORAS TESIS]..... 153

TABLA 4.5 DATOS TOMADOS DE LA ESTACIÓN DE CLASIFICACIÓN. 154

[FUENTE: AUTORAS TESIS]..... 154

TABLA 4.6 TIEMPO. 154

[FUENTE: AUTORAS TESIS]..... 154

TABLA 4.7 RESULTADOS DE LA ESTACIÓN DE ENSAMBLAJE. 155

[FUENTE: AUTORAS TESIS]..... 155

TABLA 4.8 RESULTADOS DE LOS MOTORES DEL PALLET. 156

[FUENTE: AUTORAS TESIS]..... 156

TABLA 4.9 RESULTADOS SEGUIDOR DE LÍNEA DE ROBOTINO. 156

[FUENTE: AUTORAS TESIS]..... 156

TABLA 4.10 RESULTADOS DE LA ESTACIÓN DE CLASIFICACIÓN..... 157

[FUENTE: AUTORAS TESIS]..... 157

TABLA 4.11 TIEMPO DE PRODUCCIÓN. 158

[FUENTE: AUTORAS TESIS]..... 158

CAPITULO 5 158

5. CONCLUSIONES Y RECOMENDACIONES..... 158

BIBLIOGRAFIA 160

ANEXOS 162

ANEXO 1 162

TABLA A.1 DATOS TÉCNICOS DE ROBOTINO 163

TABLA A.2 DATOS TÉCNICOS MOTOR DE ROBOTINO 163

TABLA A.3 DATOS TÉCNICOS REDUCTOR DE ROBOTINO 163

TABLA A.4 DATOS TÉCNICOS RUEDA OMNIDIRECCIONAL DE ROBOTINO 164

TABLA A.5 DATOS TÉCNICOS BRAZO ROBÓTICO RV-2AJ 165

TABLA A.6 DATOS TÉCNICOS GENERALES DE LA CPU DEL PLC S7-200 166

TABLA A.7 DATOS DE ALIMENTACIÓN DE LAS CPU 167

TABLA A.8 DATOS TÉCNICOS DE EASYPORT USB 168

ANEXO 2 169

ANEXO 3 175

FIGURA A3.1 IMPLEMENTACIÓN DEL SISTEMA INDUSTRIAL 175

[FUENTE: AUTORAS TESIS] 175

RESUMEN

La automatización de las plantas industriales es un aspecto muy importante en el crecimiento de las empresas, ya que se ven en la necesidad de incrementar la demanda del producto, ofrecer productos de mejor calidad, optimizar el consumo de energía.

Para realizar un proceso de automatización industrial se requiere del uso de sistemas o elementos computarizados y electromecánicos para controlar maquinarias que realicen cierto proceso de producción en una planta, la automatización reduce ampliamente la necesidad sensorial y mental del humano.

En el Ecuador existe un déficit en el desarrollo de tecnologías lo cual afecta directamente en la economía y desarrollo industrial, debido a la baja productividad, altos costos de manufactura, y con calidad no competitiva para mercados internacionales. Por lo cual es necesario incluir sistemas controlados automáticamente en procesos industriales. La misión como ingenieros en la especialización de Sistemas Industriales es diseñar y desarrollar nuevas aplicaciones y modificar las existentes para cualquier proceso productivo.

En el presente proyecto se realizará la implementación de un prototipo para el control y monitoreo de un proceso industrial genérico, es decir aplicable a los requerimientos de un entorno industrial real, esto se logra mediante el acoplamiento sincronizado del robot móvil Robotino de Festo y los brazos robóticos articulados RV-2Aj de Mitsubishi existentes en los laboratorios de la carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana sede Quito.

Cada uno de estos robots trabajan independientemente uno de otro, para fines de aprendizaje, pero en este proyecto se pretende realizar una labor conjunta, para ello se utilizará un PLC Siemens S7-200 CPU 222, el cual controlará a los dispositivos dentro del ambiente industrial creado , además se podrá visualizar el proceso desde una PC mediante una interfaz HMI.

En el desarrollo de este documento se encuentran cinco capítulos que sintetizan el trabajo realizado para conseguir la temática propuesta.

El primer capítulo detalla el planteamiento del problema en conjunto con los objetivos pertinentes para conseguir que el tema propuesto se llegue a cumplir.

El estado del arte del problema planteado será estudiado en el capítulo II., por lo que se realiza un análisis cronológico del desarrollo de la robótica móvil y fija; de las interfaces de usuario, controladores lógicos programables, de sensores y una revisión de lo que es la comunicación OPC.

Dentro del capítulo III se encuentra el proceso del diseño y la implementación del sistema de estudio, en el cual se detallan las tareas efectuadas para la obtención del entorno industrial requerido, y de su correcto funcionamiento.

El capítulo IV presenta el análisis de los resultados obtenidos de implementación y funcionamiento del prototipo para un proceso industrial.

Finalmente a partir los resultados se muestran las conclusiones logradas del trabajo realizado.

INTRODUCCIÓN

La automatización de las plantas industriales es un aspecto muy importante en el crecimiento de las empresas, ya que se ven en la necesidad de incrementar la demanda del producto, ofrecer productos de mejor calidad, optimizar el consumo de energía.

Para realizar un proceso de automatización industrial se requiere del uso de sistemas o elementos computarizados y electromecánicos para controlar maquinarias que realicen cierto proceso de producción en una planta, la automatización reduce ampliamente la necesidad sensorial y mental del humano.

En el presente proyecto se realizará la implementación de un prototipo para el control y monitoreo de un proceso industrial genérico, es decir aplicable a los requerimientos de un entorno industrial real, esto se logra mediante el acoplamiento sincronizado del robot móvil Robotino de Festo y los brazos robóticos articulados RV-2Aj de Mitsubishi existentes en los laboratorios de la carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana sede Quito.

Cada uno de estos robots trabajan independientemente uno de otro, para fines de aprendizaje, pero en este proyecto se pretende realizar una labor conjunta, para ello se utilizará un PLC Siemens S7-200 CPU 222, el cual controlará a los dispositivos dentro del ambiente industrial creado , además se podrá visualizar el proceso desde una PC mediante una interfaz HMI.

En el desarrollo de este documento se encuentran cinco capítulos que sintetizan el trabajo realizado para conseguir la temática propuesta.

El primer capítulo detalla el planteamiento del problema en conjunto con los objetivos pertinentes para conseguir que el tema propuesto se llegue a cumplir.

El estado del arte del problema planteado será estudiado en el capítulo II., por lo que se realiza un análisis cronológico del desarrollo de la robótica móvil y fija; de las interfaces de usuario, controladores lógicos programables, de sensores y una revisión de lo que es la comunicación OPC.

Dentro del capítulo III se encuentra el proceso del diseño y la implementación del sistema de estudio, en el cual se detallan las tareas efectuadas para la obtención del entorno industrial requerido, y de su correcto funcionamiento.

El capítulo IV presenta el análisis de los resultados obtenidos de implementación y funcionamiento del prototipo para un proceso industrial.

Finalmente a partir los resultados se muestran las conclusiones logradas del trabajo realizado.

CAPITULO 1

1. PLANTEAMIENTO DEL PROBLEMA

En el Ecuador existe un déficit en el desarrollo de tecnologías lo cual afecta directamente en la economía y desarrollo industrial, debido a la baja productividad, altos costos de manufactura, y con calidad no competitiva para mercados internacionales. Por lo cual es necesario incluir sistemas controlados automáticamente en procesos industriales. La misión como ingenieros en la especialización de Sistemas Industriales es diseñar y desarrollar nuevas aplicaciones y modificar las existentes para cualquier proceso productivo. Por lo tanto, en el siguiente trabajo se plantea un prototipo de control para entornos industriales mediante el acoplamiento sincronizado del robot móvil Robotino de Festo y los brazos robóticos articulados RV-2AJ de Mitsubishi.

1.1 Hipótesis

La elaboración de un proceso industrial autónomo se logra a través de la sincronización de la robótica fija conjuntamente con la robótica móvil, mediante el uso de un controlador lógico programable.

1.2 Tesis

Para elaborar un proceso industrial autónomo se debe realizar la sincronización de la robótica fija conjuntamente con la robótica móvil, a través del uso de un controlador lógico programable y de acoples mecánicos dentro de un entorno controlado.

1.3 Objetivos

1.3.1 Objetivo general

Implementar un prototipo para el control y monitoreo de un proceso industrial mediante el acoplamiento sincronizado del robot móvil Robotino de Festo y los brazos robóticos articulados RV-2AJ de Mitsubishi, utilizando una interfaz HMI.

1.3.2 Objetivos específicos

- Definir el entorno del proceso industrial a ser controlado.

- Determinar los requerimientos generales del sistema y las características más relevantes de control que se debe tener el entorno industrial.

- Diseñar los programas de control para cada uno de los brazos robóticos Mitsubishi RV-2AJ y el robot móvil Robotino de Festo.

- Programar algoritmos para el control requerido en el robot móvil Robotino de Festo y el brazo articulado RV-2AJ de Mitsubishi a través de su respectivo software.

- Realizar el control sincronizado entre los brazos robóticos y el Robotino mediante un PLC, implementando una comunicación OPC.

- Diseñar y construir una interfaz HMI para el control y monitoreo del sistema industrial

- Integrar todo el proceso a controlar y realizar pruebas del sistema final.

CAPITULO 2

2. ESTADO DEL ARTE

2.1 Robótica

La Robótica es una disciplina relativamente joven que está actualmente en auge y en continuo desarrollo junto con la tecnología. Por lo que es relevante en los estudios de ingeniería debido a las capacidades y aplicaciones de los robots para producir bienes y servicios o explotar recursos naturales, espaciales, submarinas, subterrestres y agrícolas en ambientes de difícil acceso, peligrosos o desagradables para el ser humano.

Historia

Existe una larga tradición de autómatas desde el mundo griego hasta nuestro siglo, pasando por los autómatas artesanos franceses y suizos del siglo XVIII. Donde ya incorporaban interesantes dispositivos mecánicos para el control automático de movimientos. [1]

El termino robot se usó por primera vez en 1921, en la obra de teatro *Rossum´s Universal Robots (RUR),* escrita por el checo KarelCapek. El origen de la palabra "robota" significa fuerza del trabajo o servidumbre.

A continuación se detalla una reseña cronológica de los desarrollos más destacados en robótica:

AÑO	AUTOR	AUTOMATA
1352	Desconocido	Gallo de la Catedral de Estrasburgo
1499	Leonardo Da Vinci	León mecánico

[1] Extraído del Capítulo 1 del libro "Robótica Manipuladores y Robots Móviles" de Ollero Aníbal

1525	Juanelo Turriano	Hombre de Palo
1738	Jacques Vaucanson	Flautista, Tamborilero, Pato, Muñecas mecánicas de tamaño humano
1769	W. Von Kempelen	Jugador de ajedrez
1770	Familia Droz	Escriba, Organista, Dibujante.
1805	Henry Maillardet	Muñeca mecánica capaz de dibujar.
1947	Desconocido	Teleoperador accionado por servomecanismos eléctricos.
1948	Goertz	Manipulados mecánico
1954	George Devol	Primer Robot Industrial
1954	Goertz	Manipulador Eléctrico
1954	C. W. Kenward	Patente de manipulador robótico
1961	Joseph Engelberger	Primer brazo articulado (Para General Motors)
1966	Trallfa	construyó e instaló un robot de pintura por pulverización
1971	StandfordUniversity	Brazo robótico de accionamiento eléctrico
1978	StandfordUniversity	Robot Puma
1981	Universidad de Yamanashi en Japón	Robot Scara
1985	Universidad de Waseda	WASUBOT robot músico
1986	Honda	Desarrollo siete robots bipedos
1989	Pacífico los Laboratorios BattelleNorthwest en	Robot antropomórfico a gran escala con 42

	Richland, Washington	grados de libertad
1993	Honda	Desarrolló P1 y P3.
1996	Universidad de Tokio	Robot humanoide de tamaño humano y de bajo costo
2000	Honda	Robot humanoide bípedo ASIMO
2001	Sony	Pequeños robots humanoides de entretenimiento llamado Sony Dream Robot
2005	Mitsubishi	Wakamaru , un robot doméstico destinado a ofrecer compañía a las personas mayores y discapacitados.
2007	TOSY Robótica JSC.	Topio , un robot jugando al ping pong.
2009	Instituto Nacional de Ciencia Industrial Avanzada y Tecnología	HRP - 4C Robot doméstico japonés
2010	Nasa y General Motors	Robonaut 2, caminante espacial.

TABLA 2.1 Historia de los robots

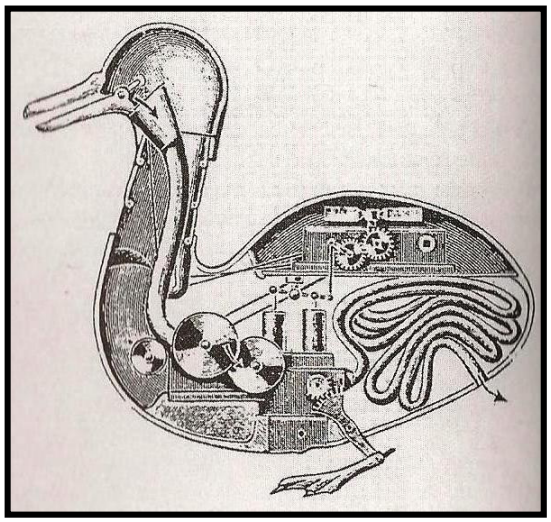


FIGURA 2.1 Pato Artificial que realizó Vaucanson en 1738 bebía, comía y digería.

[Fuente: "Atlas Ilustrado de los Robots de Leonardo Da Vinci ", TADDEI, Mario]

El máximo impulsor de la palabra robot, fue el escritor Isaac Asimov. En octubre de 1945 publicó en la revista GalaxyScienceFiction una historia en la que por primera vez enunció sus tres leyes de la robótica:

1. Un robot no puede perjudicar a un ser humano, ni con su inacción, permitir que un ser humano sufra daño.

2. Un robot debe obedecer las órdenes recibidas por un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.

3. Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o segunda ley.

Más tarde, Fuller 1999 introdujo la cuarta ley que dice:

4. Un robot podrá tomar el trabajo de un ser humano, pero no debe dejar a esta persona sin empleo.

Estas leyes deben ser acatadas al momento de diseñar y construir un robot según su aplicación.

Robots

Un robot se define como: un manipulador multifuncional reprogramable, capaz de mover materiales, piezas, herramientas, o dispositivos especiales a través de movimientos variables programados para el desempeño de tareas diversas.

Existen otras definiciones de robot por diferentes organizaciones pero entre todas coinciden en dos puntos básicos: primero la capacidad de reprogramación y la multifuncionalidad de los robots.

Para determinar el uso de un robot se debe tomar en cuenta las siguientes reglas:

1 La primera regla se conoce como las cuatro D de la robótica, es decir, si la tarea es sucia (Dirty), aburrida (Dull), peligrosa (Dangerous), o difícil (Difficult). Si se tiene una de estas características, un ser humano probablemente no será capaz de ejecutar la tarea de forma eficiente. Por lo tanto, es un trabajo apropiado para que se le automatice o para que lo realice un robot.

2 La segunda regla es que un robot no debe dejar a un ser humano desempleado. La robótica y la automatización debe servir para hacer nuestra vida más placentera, y no desagradable.

3 La tercera regla implica cuestionarse si es posible o no encontrar personas dispuestas a realizar el trabajo. De no ser así, dicho trabajo es candidato para la automatización o la robótica.

4 La cuarta regla general es que el uso de los robots o de automatización debe tener beneficios económicos a corto y largo plazo.

Considerando una tarea que tiene que realizarse una sola vez o unas cuantas veces y que no es peligrosa es mejor que la efectué un ser humano. Después de todo, él es la más flexible de todas las máquinas. En cambio, será mejor que una tarea que tenga que ser hecha cientos o cientos de miles de veces, sea ejecutada por una máquina automatizada flexible, como un robot industrial.[2]

Aplicaciones

Existen varios campos de aplicación para robots industriales, siendo la industria automovilística quien ha empleado la mayor parte de robots actualmente instalados, seguido por la fuerte demanda de la elaboración de componentes

[2]Extraído del capítulo 1 de "Introducción a la Robótica" de SAHA, Subir.

eléctricos y electrónicos como se puede apreciar en la Figura 2.1. Por otro lado el uso de robots para la producción de instrumentos médicos y ópticos, son casi nulos.

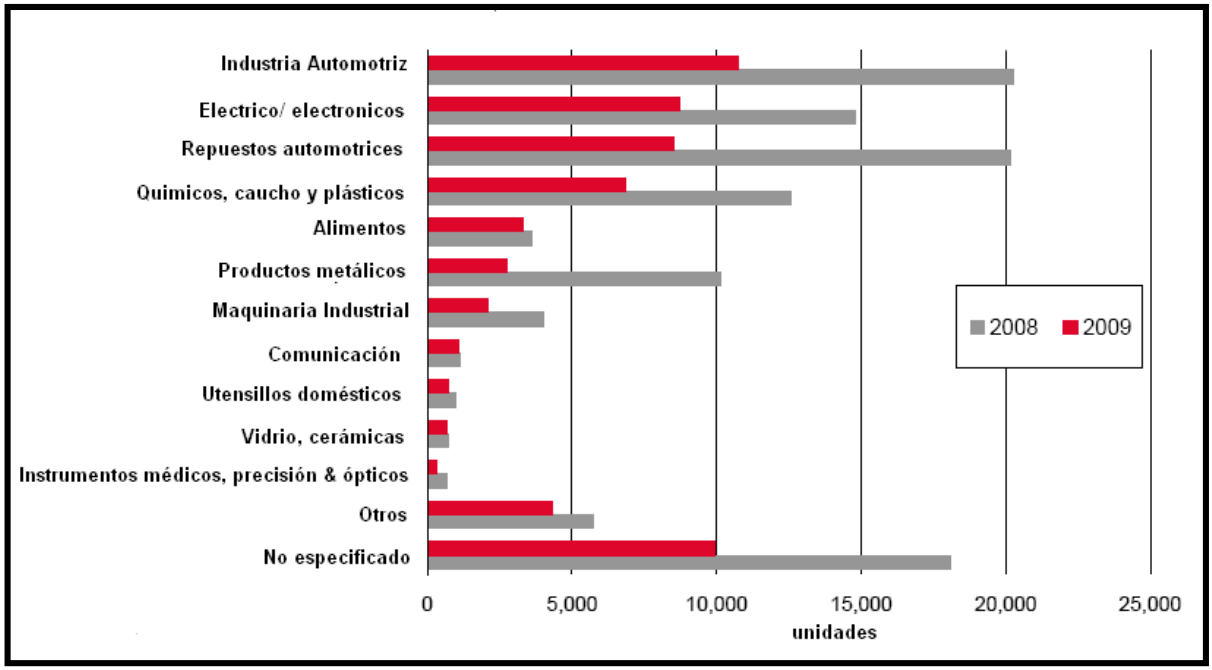


FIGURA 2.2 Uso de robots industriales durante 2008 – 2009

[Fuente: http://www.ifr.org/uploads/media/2010_Executive_Summary_rev.pdf]

Población

En la Figura 2.2 se muestra la población estimada de robots industriales en diferentes continentes, siendo siempre Asia el continente pionero en la utilización de robots.

La distribución de robots por países y los pronósticos para los próximos años se presentan en la Tabla 2.2.

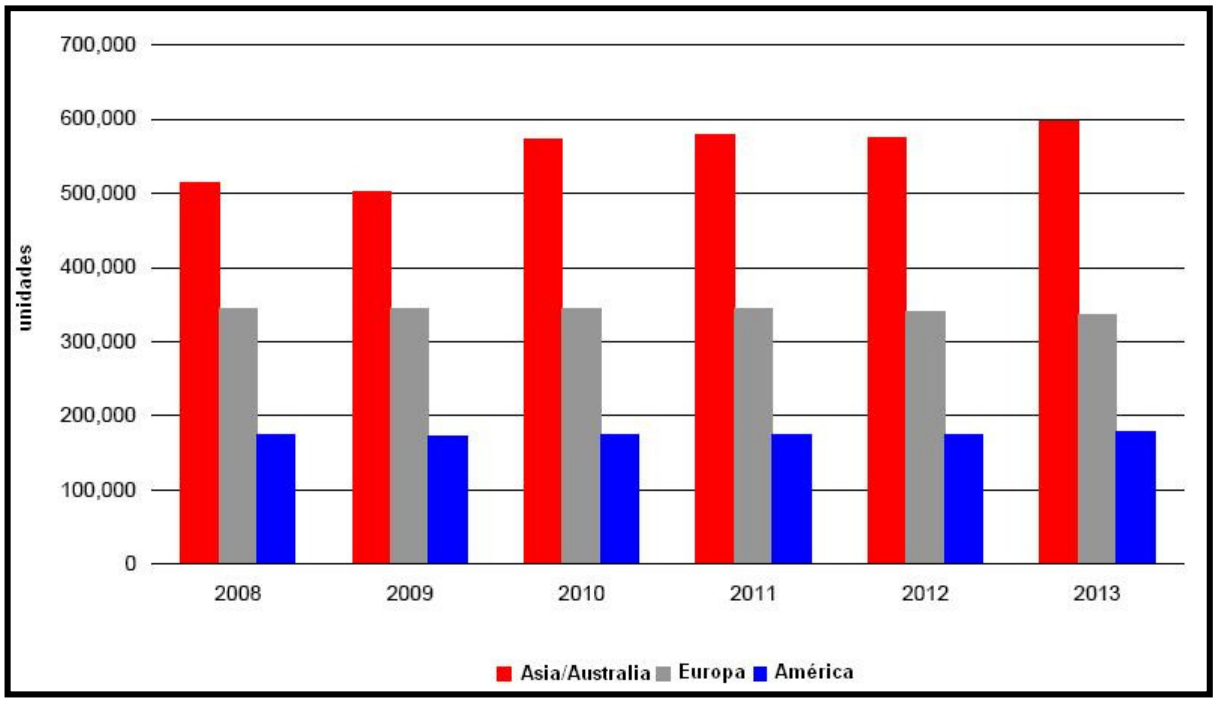


FIGURA 2.3 Poblaciones estimadas en cada continente de robots industriales

[Fuente: http://www.ifr.org/uploads/media/2010_Executive_Summary_rev.pdf]

PAIS	INSTALACIONES AL AÑO			FUNCIONAMIENTO A FIN DE AÑO		
	2009	2010	2013	2009	2010	2013
América	8.917	11.900	15.300	172.141	174.600	178.700
Norte América (Canadá, México, USA)	8.417	11.000	13.500	166.183	167.800	169.100
América Central y del Sur	500	900	1.800	5.958	6.800	9.600
Asia/Australia	30.117	40.500	57.100	501.422	574.000	597.900
China	5,525	8.500	16.500	37.312	45.800	84.500
India	363	600	1.500	4.079	4.700	8.100
Japón	12.767	17.000	21.000	332.720	315.900	263.700
República de Corea	7.839	9.800	12.000	79.003	87.400	109.800
Taiwán	1.474	2.000	2.800	24.365	25.600	31.000
Tailandia	774	1.000	1.400	7.185	8.200	11.900
Otros lados de Asia	866	1.600	1.900	10.061	86.400	88.900
Australia/ Nueva Zelanda	509			6.697		
Europa	20.483	23.000	29.100	343.661	344.400	337.200
Benelux	1.286			11.678		
Francia	1.450	1.700	2.000	34.099	34.200	29.500
Alemania	8.507	10.000	12.600	144.133	144.200	143.300
Italia	2.883	2.900	4.200	62.242	60.800	54.500
España	1.348	1.500	1.800	28.781	28.500	24.900

Suecia	587			9.396		
Reino Unido	635	650	900	13.923	13.300	10.800
Países Europeos del Centro Oriente	1.448			10.268		
Otras partes de Europa	2.339	6.250	7.600	29.141	63.400	74.200
África	196	300	500	1.973	2.300	3.300
TOTAL	60.018	76.000	102.300	1.020.731	1.097.100	1.119.800

TABLA 2.2 Cifras estimadas de existencia de robots industriales a nivel mundial

[Fuente: http://www.ifr.org/uploads/media/2010_Executive_Summary_rev.pdf]

Clasificación de Robots

Para la clasificación de robots se ha tomado en consideración tres categorías, las cuales se detallan a continuación:

1. Manipuladores, o brazos robóticos, están físicamente anclados en su lugar de trabajo. Los manipuladores son robots industriales con más unidades instaladas en el mundo.

2. La segunda categoría son los robots móviles. Los robots móviles se desplazan por su entorno utilizando ruedas, piernas o mecanismos similares. Ejemplos de estos robots son NAVLAB, vehículo terrestre sin tripulación, el ULV capaz de realizar un desplazamiento autónomo para la navegación de autopistas, vehículos aéreos sin tripulación UAV y los vehículos submarinos sin tripulación AUV.

3. El tercer tipo es un hibrido: un robot móvil equipado con manipuladores. Esto incluye a los robots Humanoides, cuyo diseño se asemeja al torso humano. Los robots híbridos pueden aplicar sus efectores en un campo más amplio que los robots anclados pero las tareas son más complejas porque no tienen la rigidez que el anclaje proporciona.

El campo de la robótica también incluye prótesis artificiales, entornos inteligentes, y sistemas multicuerpo[3].

2.1.1 Manipuladores

Los robots manipuladores o robots industriales fueron los encargados de inaugurar la era de los robots en los años 60, con la herencia adquirida de los primeros teleoperadores. Por ello, es el área de la robótica donde la investigación está más avanzada. Es difícil encontrar investigaciones que trabajen en los aspectos básicos de los robots manipuladores tradicionales y, las pocas que hay, están centradas en la inclusión de modernos sensores o actuadores. Un ejemplo de esto es la adición de cámaras para reconocimiento avanzado de imágenes que mejoren la efectividad de dichos robots.

El área más interesante, en donde la investigación está en auge es en la búsqueda de novedosas aplicaciones para los robots manipuladores, como es el caso de los robots quirúrgicos. En este caso, la investigación no está centrada en el robot sino en adaptar éste a las necesidades propias de la aplicación.

FIGURA 2.4 Manipulador quirúrgico

[Fuente: http://www.intramed.net]

[3]RUSSELL Stuart J. y NORVING Peter, "Inteligencia Artificial un Enfoque Moderno", 2da edición, pp. 1023- 1024.

Estructura Mecánica

Se conoce también al robot industrial como brazo robótico y esto se debe a que guarda cierta similitud con la anatomía del brazo humano. También por ello, para hacer referencia a los distintos elementos que componen el robot, se utilizan términos como cuerpo, brazo, codo y muñeca. Formalmente, se denomina base al punto de apoyo del robot, generalmente sujeto de forma fija al suelo. Los elementos o eslabones van unidos por medio de diferentes articulaciones, que permiten un movimiento relativo entre cada dos eslabones consecutivos. En la parte final, se sitúa el elemento terminal o efector final, que son los encargados de interaccionar directamente con el entorno del robot.

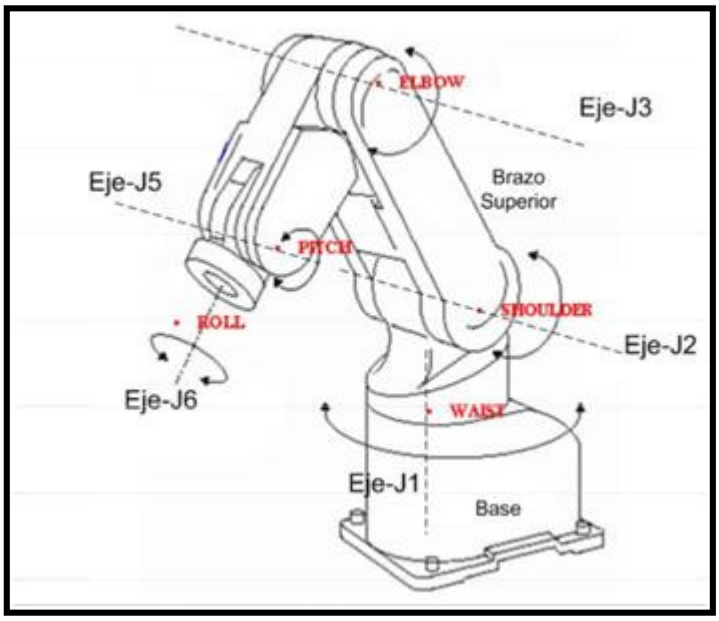


FIGURA 2.5 Estructura mecánica del manipulador RV 2AJ

[Fuente: Manual Brazo Mitsubishi RV 2AJ]

El movimiento de cada articulación puede ser de desplazamiento, de giro o de una combinación de ambos. De este modo son posibles los 6 tipos diferentes de articulaciones, con sus diferentes grados de libertad, que se muestran a continuación:

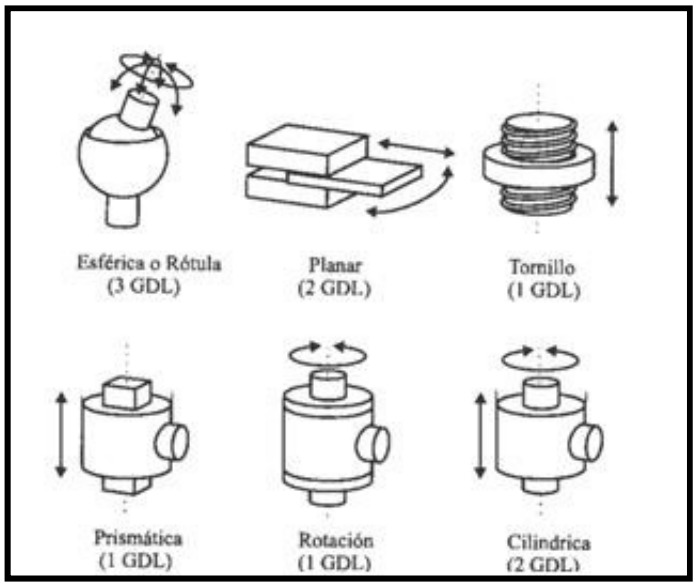


FIGURA 2.6 Tipos de articulaciones

[Fuente: "Fundamentos De Robótica" BARRIENTOS, Antonio]

Los grados de libertad son el número de movimientos independientes que puede realizar cada articulación con respecto a la anterior. El número de grados de libertad de un robot viene dado por la suma de los grados de libertad de cada una de sus articulaciones.

El empleo de diferentes combinaciones de articulaciones en un robot, da lugar a diferentes configuraciones con características para el diseño y construcción del robot según su aplicación.

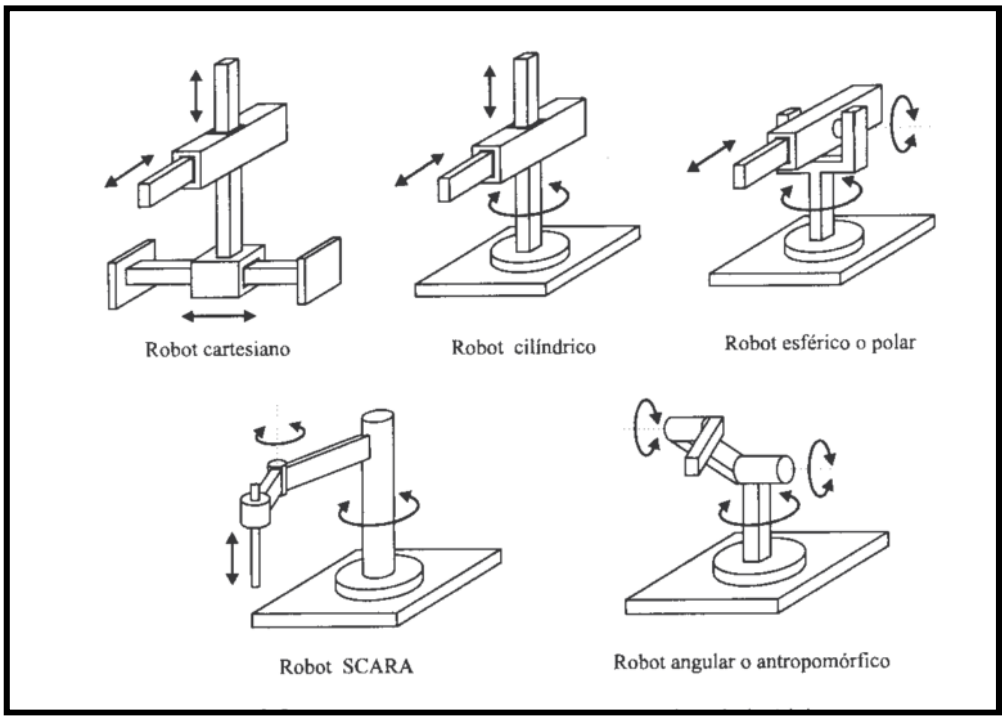


FIGURA 2.7 Tipos de Configuraciones más frecuentes

[Fuente: "Fundamentos De Robótica" BARRIENTOS, Antonio]

Aplicaciones de los Manipuladores

Los robots manipuladores tienen su principal foco de trabajo en la industria, automatizando los procesos de producción o almacenaje. Generalmente no trabajan de forma independiente sino en conjunto con otras máquinas y herramientas formando células de trabajo. Se enumeran a continuación algunos ejemplos:

- Operaciones de procesamiento, como soldadura, pintura, etc. Este tipo de robots son muy comunes en la industria de la automoción.

- Operaciones de ensamblaje, donde el trabajo repetitivo facilita el uso de este tipo de robots.

- Operaciones de empaque (en tarimas o pallets), agilizando el proceso y manejando grandes pesos.

- Otro tipo de operaciones como pueden ser remachados, estampados, corte por chorro de agua, sistemas de medición, etc.

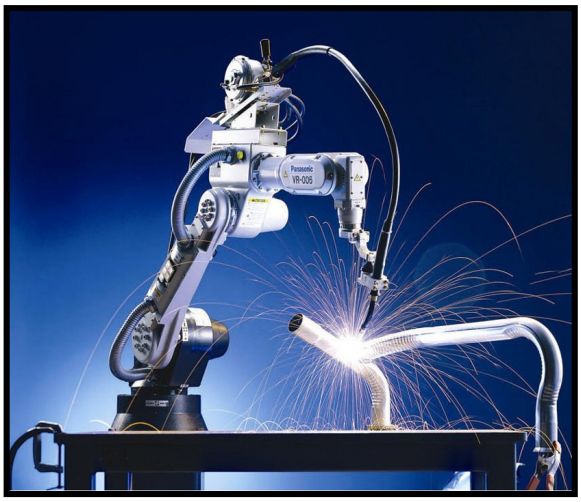


FIGURA 2.8 Robot para soldadura

[Fuente: http://innovationtecnology.blogspot.com]

2.1.2 Manipulador RV-2AJ

El brazo robótico RV-2AJ es un robot industrial compacto desarrollado con la tecnología avanzada de Mitsubishi. Este robot tiene un alcance (con la pinza hacia abajo) de 410mm y combina una velocidad máxima de 2,1m/s con una repetitividad de ±0.02mm. Los servomotores de corriente alterna, unidos a encoders de posición absoluta garantizan una fiabilidad y bajo mantenimiento que son difíciles de superar por otros robots de similares características.

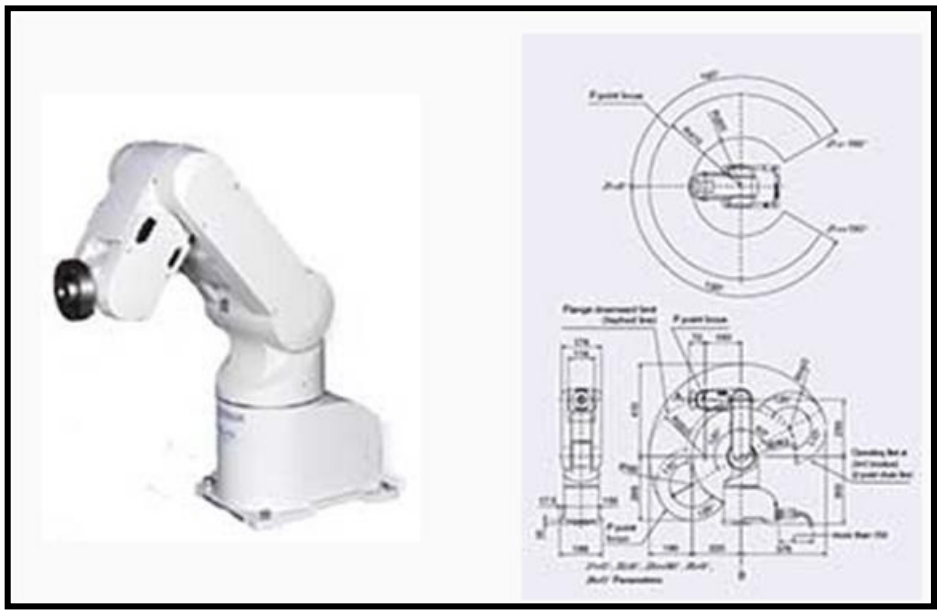


FIGURA2.9 Brazo robótico RV-2AJ

[Fuente: Manual RV-2AJ, Mitsubishi]

El brazo robot tiene integrados en su interior una serie de conductos que permiten la conexión de pinzas y ventosas neumáticas para su movimiento.

Dispone de un puerto RS232 y de 16 entradas y salidas digitales que son el equipamiento de serie para comunicar el robot con el exterior. Un adaptador permite comunicar el robot mediante red Ethernet (Protocolo TCP/IP). Otra opción, la red CC-Link, de Mitsubishi, permite el intercambio rápido de datos, sobre todo entre el robot y un PLC.

Para el control del brazo es necesario utilizar una controladora CR1-571 y una botonera para grabar y ejecutar los movimientos deseados.

Controlador Del Robot (Cr1-571)

En el controlador radica el sistema de control del Robot y también se conecta la botonera o teach. Tiene su propio lenguaje de programación llamado MelfaBasicIV y los datos se transmiten por puerto serial o TCP/IP desde una computadora.

El software que permite el envió de datos desde la PC a la controladora se llama Ciros Studio, mediante dicho programa se puede controlar el movimiento de los brazos en sus diferentes ejes y además se posee el control de los sensores del entorno.

El corazón del controlador es una CPU de 64 bits que permite la ejecución en paralelo de hasta 32 programas en modo multitarea, es decir, mientras se está moviendo puede recibir datos, activar o desactivar entradas y salidas, hacer cálculos y 28 tareas más.

Botonera De Mando (T/B)

También llamado teachpendant, se utiliza para determinar las posiciones del brazo robot. Para ayudar en la programación y para el control del robot, tiene integrado un visualizador LCD.

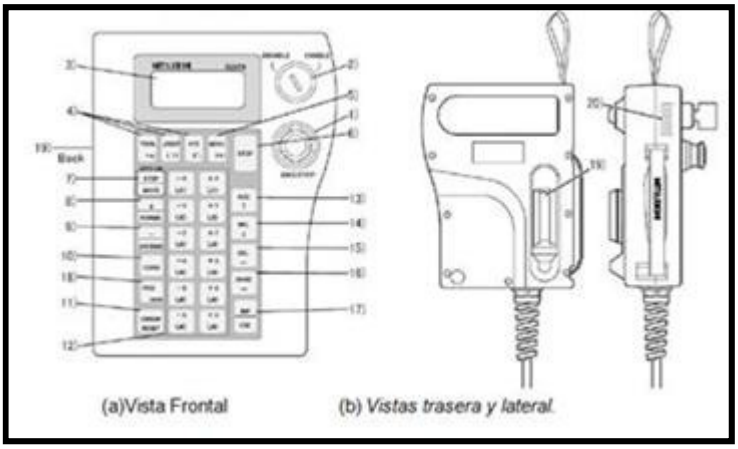


FIGURA 2.10 Botonera de mando

[Fuente: "Manual, Teachingpendant (T/B)".]

2.1.3 Robótica móvil.

Hasta la actualidad, los sistemas robóticos eran estacionarios. Los robots móviles constituyen el siguiente paso de desarrollo en robótica. Los sistemas móviles pueden ejecutar las mismas tareas que sus antecesores fijos, pero, además, son

capaces de desplazarse de un lugar a otro y cumplen las condiciones necesarias para la realización un sinnúmero de tareas adicionales.

En los años 1500, Leonardo Da Vinci, creo el "Carrito Automotor", haciéndolo funcionar como un sistema de arcos y tirantes, con dos ruedas dentadas engranando en dos ruedas que salen debajo y se deslizan por un riel, es curioso que este mecanismo se convierta en un dispositivo aplicable a cualquier mecanismo robótico, con lo que este sencillo sistema con ruedas sería el antepasado de los robots de la NASA.

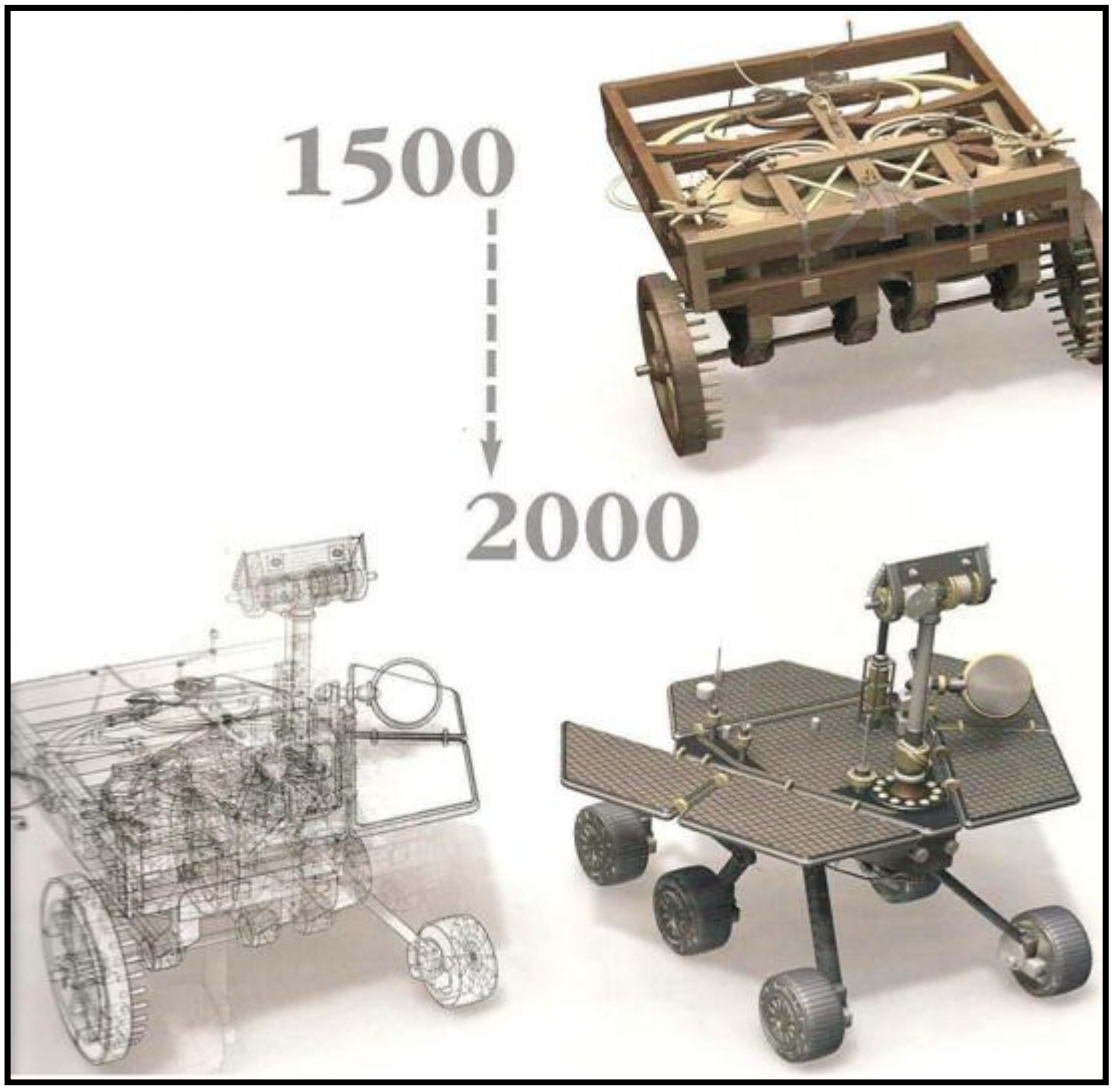


FIGURA 2.11 El Carrito Automotor vs. Robot móvil de la NASA

[Fuente: "Atlas Ilustrado de los Robots de Leonardo Da Vinci", TADDEI, Mario]

Otro ejemplo de los avances tecnológicos en la robótica móvil, es el comparativo entre el León Robótico de Leonardo y el Perro Aibo de Sony.

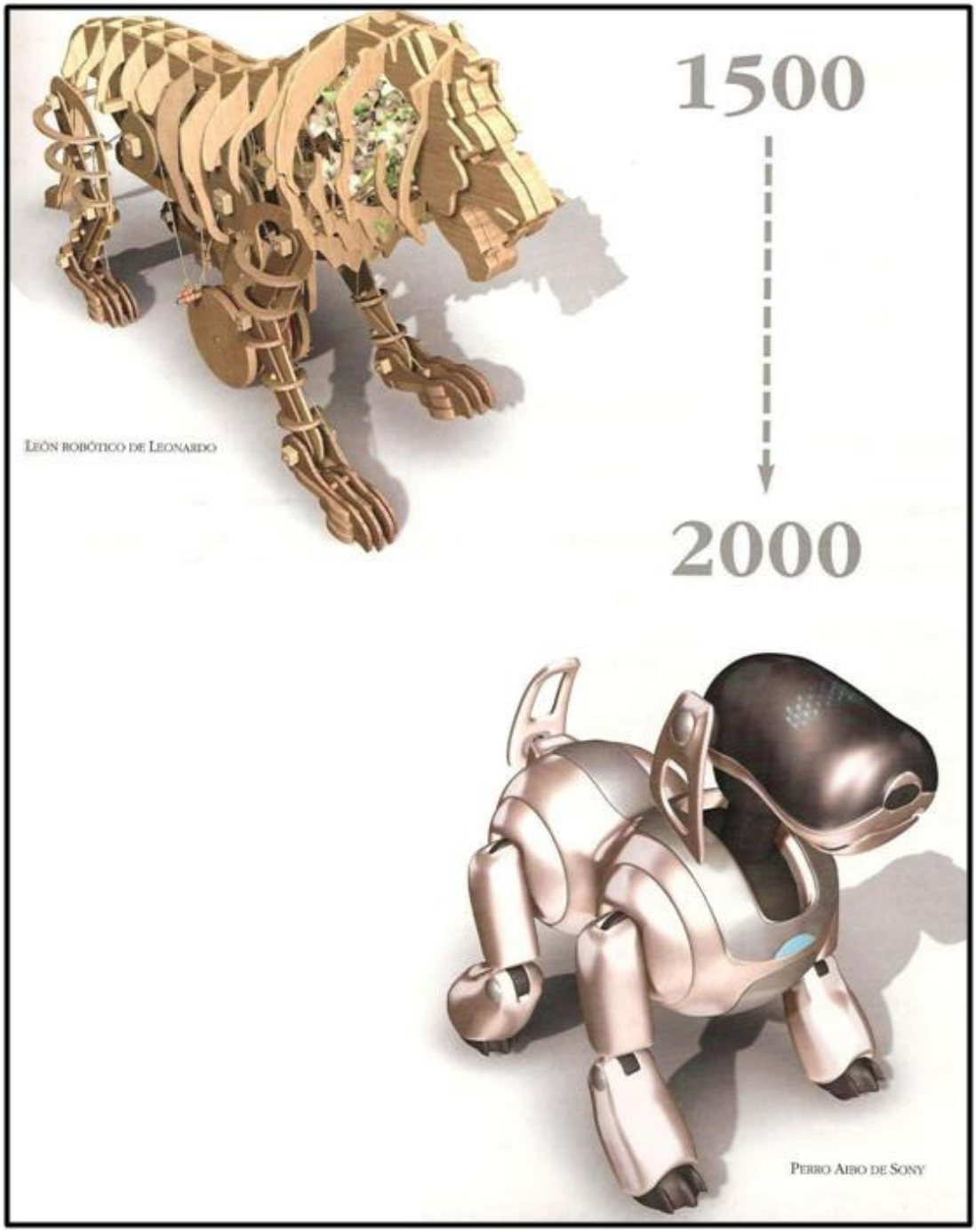
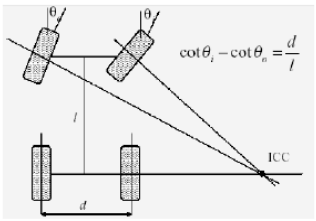
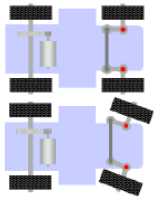
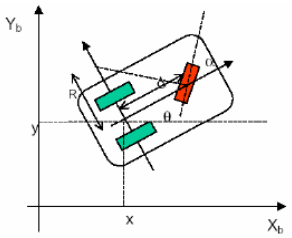
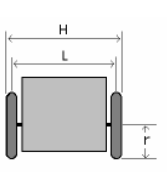
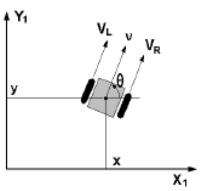
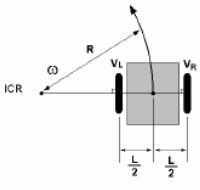


FIGURA 2.12 León robótico de Leonardo vs. Perro AIBO de Sony

[Fuente: "Atlas Ilustrado de los Robots de Leonardo Da Vinci", TADDEI, Mario]

El desplazamiento de los robots depende del tipo de locomoción que se emplee en su sistema, la palabra locomoción hace referencia al movimiento que realiza

una persona, un animal, un microorganismo, un aparato o una máquina para moverse de un lugar a otro, para trasladarse en el espacio. La locomoción varía en términos de forma, estructura, velocidad y otros elementos de acuerdo al tipo de sujeto al que se hagan referencia. Por otra parte, la locomoción puede realizarse mediante ruedas o patas.

SISTEMAS DE LOCOMOCIÓN	
Diseño Ackerman	 
Diseño Triciclo Clásico	
Diseño Direccionamiento Diferencial	  
Control Sincronizado	

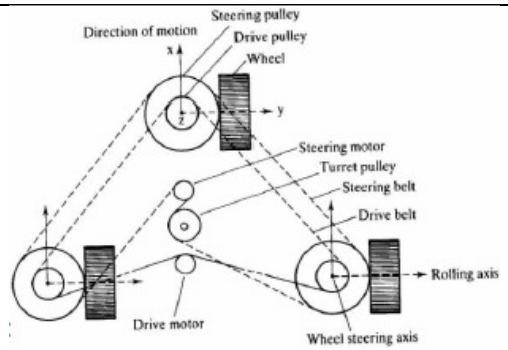
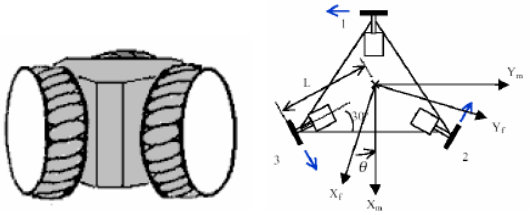
	
Omnidireccional	

TABLA 2.3 Sistemas de locomoción

Otro de los términos importantes dentro de la robótica móvil es la percepción, que es el proceso en el cual los sensores del robot elaboran un mapa del entorno utilizando una determinada representación interna. La percepción es complicada porque por lo general los sensores tienen ruido y el entorno es parcialmente observable, impredecible y dinámico.

Aplicaciones de la robótica móvil.

Los robots móviles son importantes ya que son capaces de detectar objetos según criterios tridimensionales y pueden localizar objetos en lugares inaccesibles para las personas, o donde las condiciones son inadecuadas. A continuación se citan algunas aplicaciones;

- Robot Espacial.- son usados para realizar investigaciones en la superficie de Marte, puede viajar durante más de un año y sobrevivir en condiciones imprescindibles para la supervivencia de los seres humanos.

- Estos robots permiten explorar científicamente sistemas de canalización, mundos subacuáticos y volcanes, es decir, lugares de difícil acceso para el ser humano.

- Para personas discapacitadas o de edad avanzada, un medio de transporte significa disfrutar de una mayor libertad de movimiento e independencia. En estos medios, los sistemas de orientación, navegación, detección y elusión de obstáculos tienen mucha importancia.

- El estudio de la robótica móvil y el desarrollo de las unidades respectivas se explica en buena parte por la necesidad o el deseo de utilizar robots para que ayuden a las personas en su trabajo diario y en su entorno cotidiano. Estos autómatas se utilizan en oficinas, hospitales, museos, bibliotecas, supermercados, instalaciones deportivas (p.ej. para cortar el césped), salas de exposición, aeropuertos, estaciones de tren, universidades, escuelas y, en un futuro no muy lejano, también en los hogares.

- En un gran número de fábricas, la movilidad es esencial para el transporte de materiales durante el proceso de fabricación y los robots móviles son utilizados con gran eficiencia en estos dominios.

- Para inspección de instalaciones, infraestructuras (puentes, presas) y grandes edificaciones; la detección y monitorización de catástrofes (incendios, inundaciones, erupciones de volcanes, terremotos), exploración; vigilancia, seguridad ciudadana; detección y desactivación de minas antipersonas y otras.

FIGURA 2.13 Aplicaciones de Robots móviles

[Fuente: http://medgadget.es/2011/05/_investigadores_desarrollan_un.html]

2.1.4 Robot móvil Robotino® de Festo

Ha sido diseñado para cubrir diferentes requerimientos de la formación profesional y continuada orientada a la industria. Su hardware está formado por componentes técnicos adecuados para fines didácticos.

Los robots móviles deben cumplir determinados criterios técnicos y ciertas características:

- Máquinas móviles provistas de sistemas propios de orientación, navegación, detección y evasión de obstáculos.

- Fuente de energía propia para el robot y alimentación propia para el ordenador.

- Inclusión de detectores y actuadores propios.

El sistema para la enseñanza Robotino® cumple todas estas condiciones y permite familiarizarse con los numerosos temas relacionados con la tecnología de la robótica móvil.

Robotino® se caracteriza por:

- Su tecnología está al descubierto y, por lo, tanto, visible.

- Su tecnología invita al estudiante a entender su forma de funcionamiento y a pensar en posibles aplicaciones.

- Es una solución semejante a las que se utilizan en la industria, ya que está compuesto por componentes que se emplean realmente.

- Es versátil, fácil de transportar y ocupa poco espacio.

FIGURA 2.14Robotino

[Fuente: http://www.festo.com/net/startpage/robotino]

Robotino® es un sistema de 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. Además, el robot puede girar sobre un punto. También está equipado con una webcam y varios tipos de sensores analógicos para medición de distancias, sensores binarios para protección de colisiones y encoders digitales para detectar la velocidad real.

El controlador de Robotino® consiste en un PC embebido con una tarjeta compact flash, en la cual se han instalado varias aplicaciones de demostración y el sistema operativo (Linux). Las aplicaciones de demostración pueden ejecutarse directamente desde el panel de control del Robotino®.

Robotino® puede programarse con el software Robotino®View en un PC a través del LAN inalámbrico. También están disponibles APIs Linux y C++ para la programación del Robotino®.

Robotino® es autónomo y puede accederse al controlador directamente a través de la LAN inalámbrica (WLAN). Además es posible conectar actuadores y sensores adicionales a través de un interface de E/S.[4]

2.1.5 Robots híbridos.

Intentan reproducir total o parcialmente la forma y el comportamiento del ser humano. Actualmente, los androides están bastante evolucionados, sobre todo en Japón, pero aún no tienen una utilidad práctica por sus propias limitaciones y por su precio de fabricación. Básicamente están destinados a la investigación y a tareas de marketing de las propias empresas desarrolladoras.

FIGURA 2.15 Robot ASIMO Uno de los robots humanoides más avanzados

[Fuente: http://www.universidadperu.com/robot-caminador-de-honda-blog.php]

[4]Manual del instructor Robotino, Festo

2.2 PLC (Controlador Lógico Programable).

El autómata industrial es un equipo electrónico, programable en lenguaje no informático, diseñado para controlar en tiempo real y en ambiente de tipo industrial, los procesos secuenciales.

Su historia se remonta a finales de la década de 1960 en respuesta a las necesidades de la automatización de la industria automotriz norteamericana por lo cual la industria buscó nuevas tecnologías electrónicas para dar una solución más eficiente y reemplazar los sistemas de control basados en circuitos eléctricos con relés, interruptores y otros componentes comúnmente utilizados para el control de los sistemas de lógica combinacional.

Este equipo electrónico de control tiene un cableado interno (hardware) independiente al proceso a controlar, que se adapta a dicho proceso mediante un programa específico (software) que contiene la secuencia de operaciones a realizar. Esta secuencia se define sobre señales de entrada y salida al proceso.

El autómata gobierna las señales de salida según el programa de control previamente almacenado en una memoria, a partir de las señales de entrada.

Una ventaja del PLC frente a otros sistemas de control programables es la estandarización de su hardware, que permite la configuración de sistemas de control, según las necesidades industriales.

El autómata se configura alrededor de una unidad central o de control, que unida por buses internos a las interfaces de entrada y salida y a las memorias, define la arquitectura interna del autómata.

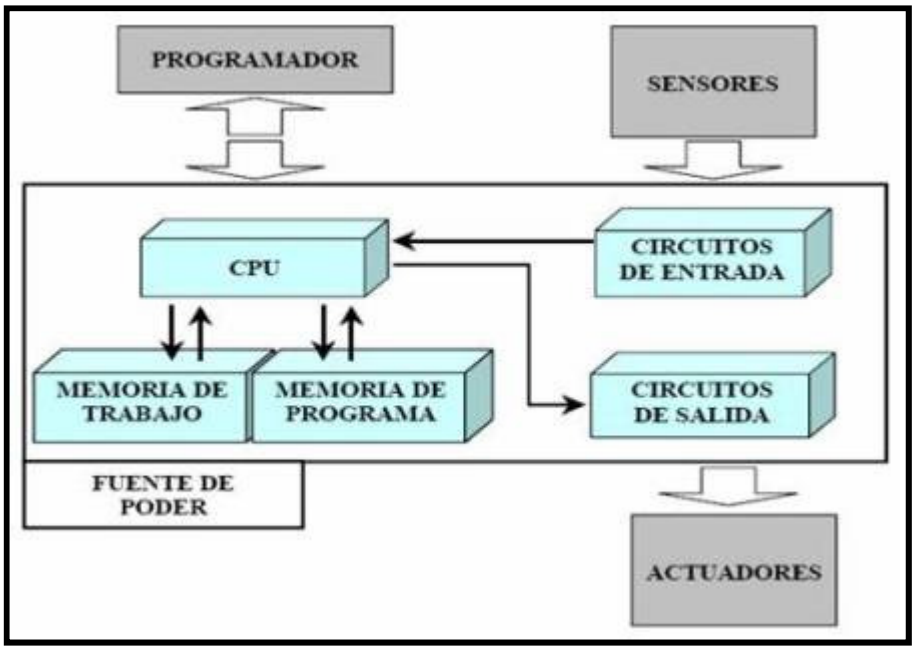


FIGURA 2.16 Arquitectura interna del PLC

[Fuente: http://www.el.bqto.unexpo.edu.ve/lab_plc/paginas/estructura.html]

Hoy en día, los PLCs no sólo controlan la lógica de funcionamiento de máquinas, plantas y procesos industriales, sino que también pueden realizar operaciones aritméticas, manejar señales analógicas, realizar estrategias de control, tales como controladores proporcional integral derivativo (PID).

Los PLC actuales pueden comunicarse con otros controladores y computadoras en redes de área local, y son una parte fundamental de los modernos sistemas de control distribuido. También existe un rango de PLCs fabricados para aplicaciones en automotores, embarcaciones, ambulancias y sistemas móviles para el mercado internacional.

En el presente proyecto se utiliza el autómata Siemens S7-200 con la CPU 222, con el cual se realiza el control y sincronización del entorno industrial a implementarse.

2.3 Sensores.

Un sensor o captador, es un dispositivo diseñado para recibir información de una magnitud del exterior como: temperatura, intensidad lumínica, distancia,

aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc.; y transformarla en otra magnitud, normalmente eléctrica (resistencia eléctrica, capacidad eléctrica, tensión eléctrica o corriente eléctrica) de la que se pueda cuantificar y manipular.[5]

Los sensores permiten la adquisición de la información necesaria para el control de un sistema o proceso.

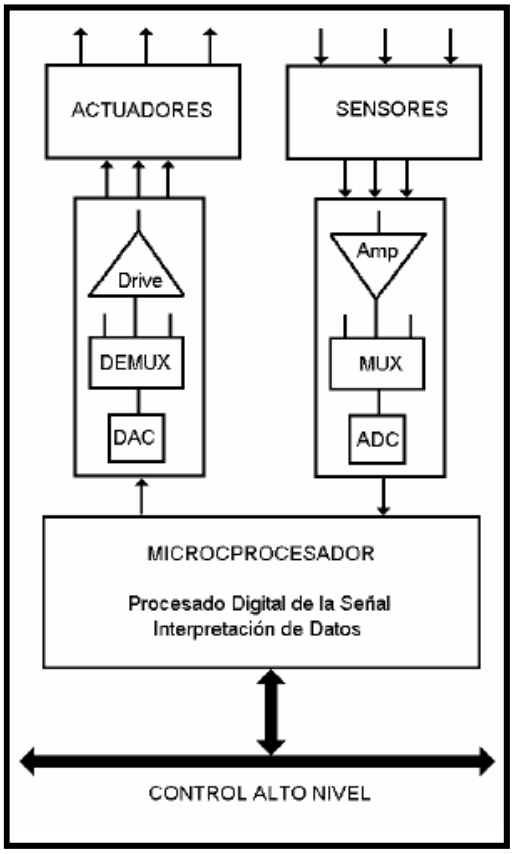


FIGURA 2.17 Diagrama de bloques de los elementos de un sensor

[Fuente: http://todoproductividad.blogspot.com/2010/03/sobre-los-sensores-inteligentes.html]

En la figura 2.17 Se muestra el diagrama de bloques de los elementos de un sistema de medida y control electrónico típico para un sensor. Los sensores proporcionan información analógica al sistema, esta información se acondiciona

[5] Extraído del capítulo 4 de "Introducción a la Robótica" de SAHA, Subir.

antes de pasar al microprocesador. El procesador interpreta la información, realiza las actuaciones necesarias y aplican esas decisiones vía los actuadores.

Hay diferentes términos que se emplean para definir el funcionamiento y caracterización a los sensores, entre los más importantes se tiene:

- Linealidad
- Histéresis
- Rango y margen
- Error
- Exactitud
- Sensibilidad
- Resolución
- Repetitividad
- Estabilidad
- Error por histéresis
- Error por no linealidad
- Banda/tiempo muerto

Clasificación de Sensores

Existen varios tipos de sensores, éstos dependen de su funcionamiento, medida o aplicación. En la Figura 2.18 se detallan los sensores, partiendo de dos grandes grupos: sensores internos y sensores externos, con sus respectivas subdivisiones.

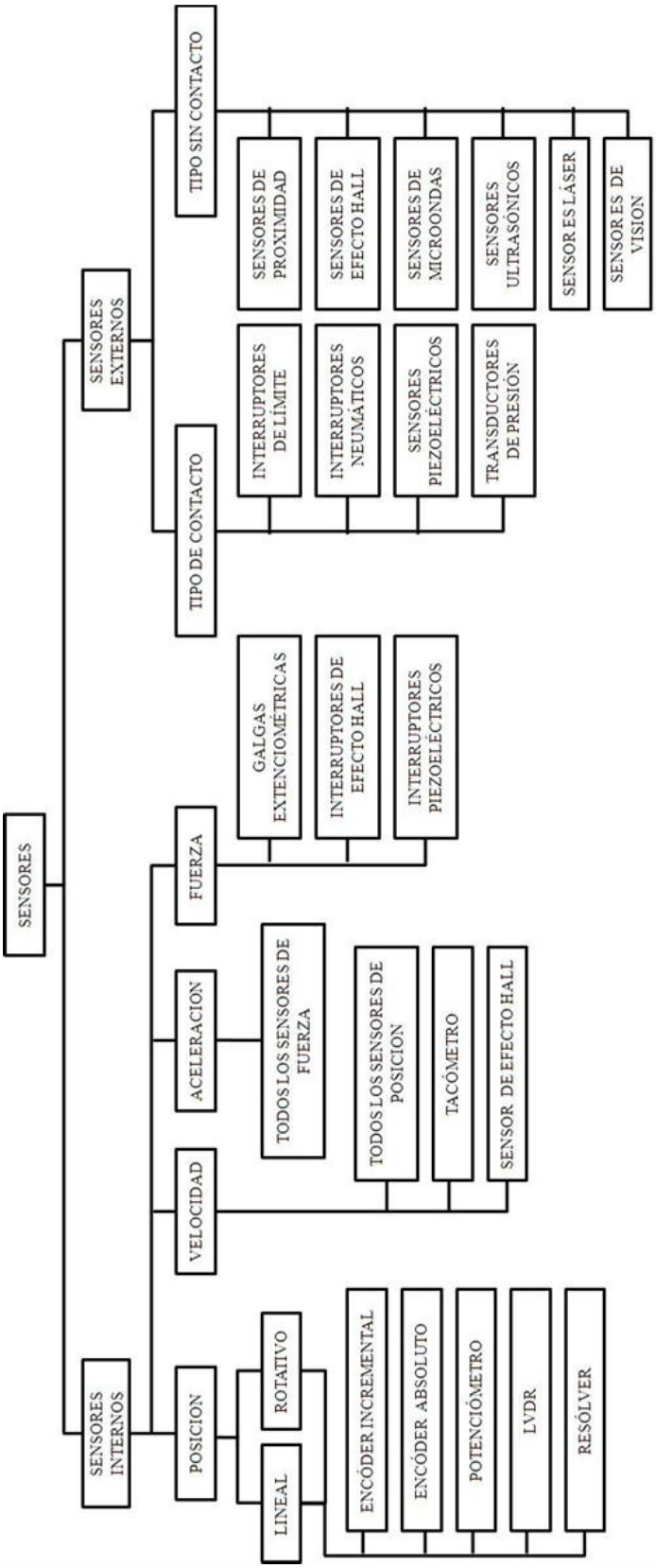


FIGURA 2.18 Clasificación de sensores

[Fuente: "Introducción a la Robótica" de SAHA, Subir.]

Dentro de los tipos de sensores, conviene tratar el concepto que en la literatura en inglés se conoce como "Smart Sensor". Se trata de sensores que no lo producen la magnitud que se pretende medir, sino también incorporan funciones de procesamiento tales como: Comunicación bidireccional, auto calibración, filtrados, fusión multisensorial, etc.

La implementación de estos sensores inteligentes requiere la inclusión de tres partes principales. Estos son: el procesamiento de la señal, el control y manipulación de las señales digitales y la comunicación con el exterior mediante un bus.

Los sensores inteligentes se clasifican según la generación a la cual pertenecen. Los dispositivos de primera generación no tienen electrónica asociada, mientras que los sensores de segunda generación forman parte de sistemas puramente analógicos con un control remoto del sensor. Por otra parte los sensores de tercera generación, que son en los que se basan la mayoría de los sistemas actuales, la etapa de amplificación se realiza en el módulo del sensor obteniendo así una señal analógica de alto nivel, codificada con una variación de tensión o como una señal de frecuencia variable. Esta señal se convierte a digital y posteriormente se procesa mediante un microprocesador. La cuarta generación de sensores se caracteriza por que la mayor parte de la electrónica, tanto analógica como digital, está en un chip, permitiendo así el direccionamiento del sensor y en algunos casos el autotest mediante comunicación entre el sensor y el microcontrolador.

Los sensores de quinta generación están caracterizados por varios atributos: comunicación bidireccional de datos y ordenes, transmisión totalmente digital, procesamiento digital local, testeo propio, algoritmos definidos por usuario y algoritmos de compensación.[6]

[6] J. E. Brignell, "Smart Sensors. A Comprehensive Survey," University of Southampton, UK.ISBN. 1989

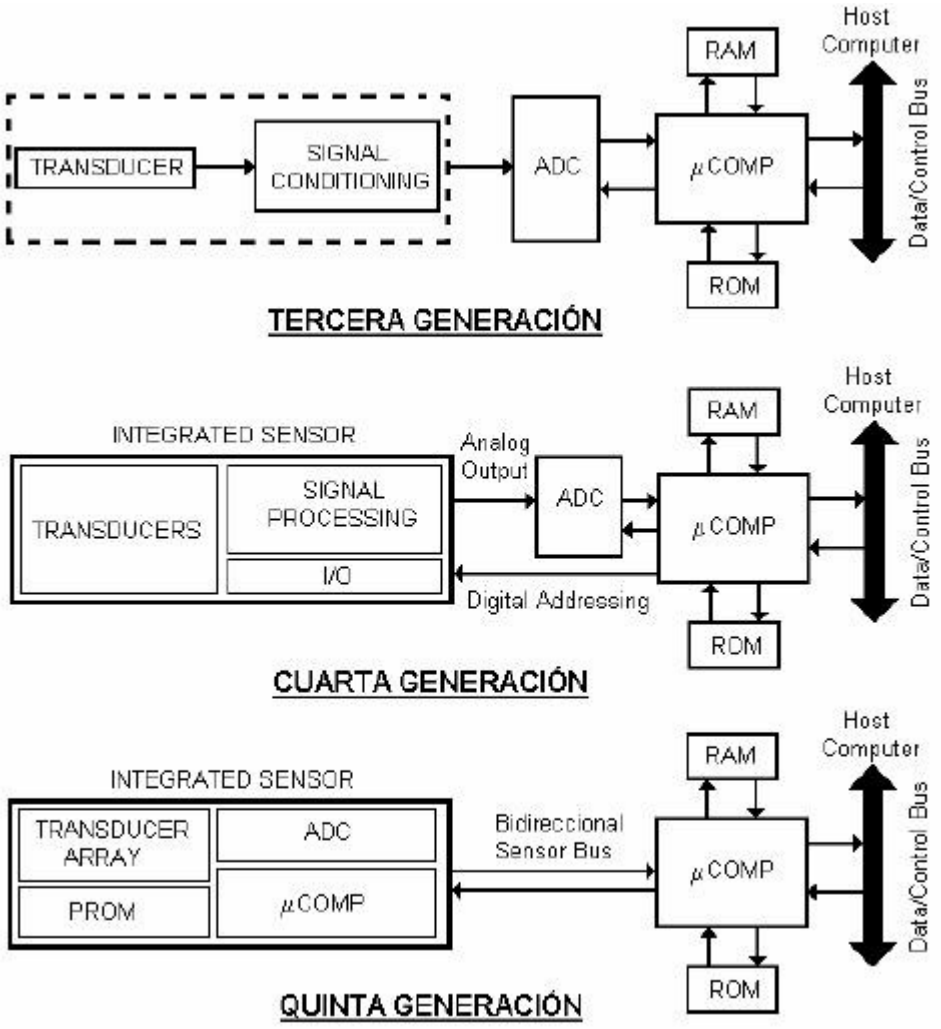


FIGURA 2.19 Generaciones de sensores inteligentes

[Fuente: J. E. Brignell, "Smart Sensors. A Comprehensive Survey," University of Southampton, UK.ISBN. 1989]

2.4 Comunicación OPC.

A través de la historia de las comunicaciones, uno de los problemas que se ha tenido, es que los datos de diferentes sistemas tienen infinitos formatos y protocolos. Esto sucede, por ejemplo, cuando dispositivos son conectados a un sistema PC basado en un sistema SCADA(Control de Supervisión y Adquisición de Datos) o DCS(Sistema de Control Distribuido). Para lo cual, los fabricantes de software crean Drivers únicos para cada tipo diferente de equipo de control.

Es por estas razones, que a finales de 1995 y principios de 1996, cinco empresas Intellution, Opto-22, Fisher-Rosemount, Rockwell Software e Intuitive Software, junto a Microsoft, decidieron trabajar juntas para eliminar estas barreras de comunicación y definieron las interfaces basadas en OLE/COM (ObjectLinking and Embedding/CommonObjectModel) que sentó las bases para OPC (Ole Process Control).

Por lo tanto, OPC es un estándar abierto que permite el intercambio de datos que puede simplificar el desarrollo de Drivers de Dispositivos de entrada y salida y mejorar el rendimiento de los sistemas de interfaz, entre dispositivos de campo y aplicaciones de computadora de diferentes fabricantes.

Con OPC el usuario decide libremente que componente de hardware es el mejor para una determinada aplicación, dado que no habrá discusiones sobre el desarrollo de Drivers.

Como se muestra en la Figura 2.21con OPC, la integración de sistemas en un entorno heterogéneo se vuelve simple.

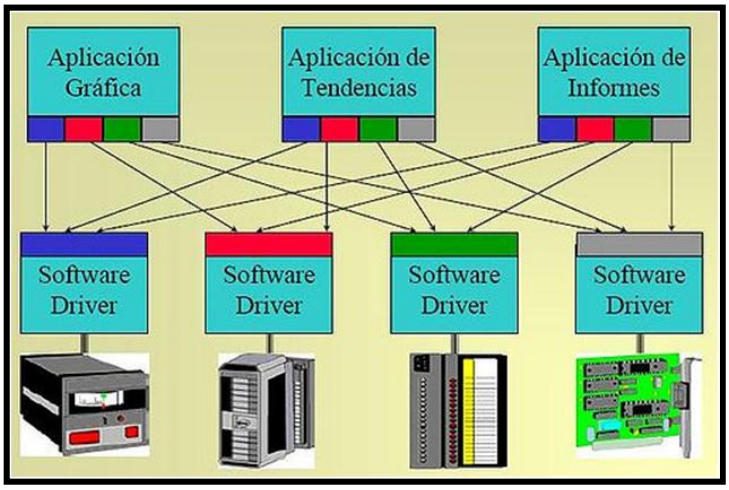


FIGURA 2.20 Problema de comunicación sin tecnología OPC.

[Fuente: http://es.wikipedia.org/wiki/OPC]

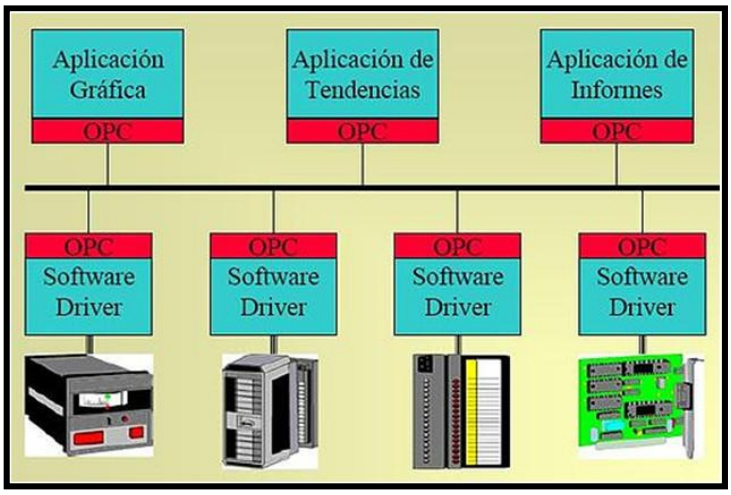


FIGURA 2.21Solución al problema al contar con tecnología OPC.

[Fuente: http://es.wikipedia.org/wiki/OPC]

Arquitectura, OPC Cliente/Servidor

Una Arquitectura OPC se refiere a la infraestructura de comunicaciones que incluye uno o varios Clientes OPC y Servidores OPC comunicándose entre sí.

Donde los servidores son los proveedores de recursos o servicios, y los demandantes, son llamados clientes; por lo tanto un cliente realiza peticiones y el servidor es quien le da respuesta.

Una aplicación cliente OPC se comunica a un servidor OPC específico por medio de una interfaz de automatización. El servidor OPC lleva a cabo la interfaz cliente, y opcionalmente lleva a cabo la interfaz de automatización

Un cliente OPC se puede conectar a servidores OPC proporcionados por más de un "proveedor".

La comunicación OPC en el presente proyecto esta empleada entre el Robot móvil Robotino de Festo y el PLC S7-200, teniendo de intermedio a Easy Port (puerto fácil). El software RobotinoView2.3 se comporta como cliente, mientras que la aplicación EZOpc de Easy Port, es el servidor. De esta manera se consigue interconectar entradas y salidas del PLC S7-200 con las entradas y salidas de Robotino.

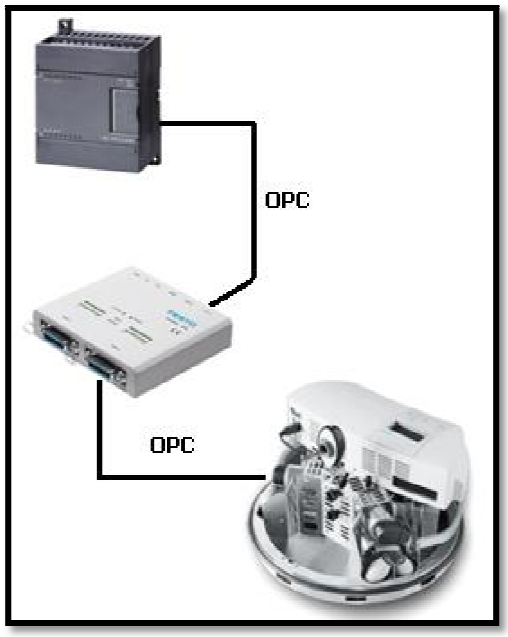


FIGURA 2.22 Comunicación OPC

[Fuente: Autoras Tesis]

2.5 Interfaz HMI.

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y de accionar.

La creación de las interfaces de usuario ha sido un área del desarrollo de software que ha evolucionado de forma continua, este campo está muy ligado al diseño de equipos electrónicos, pero comienza a tenerse en cuenta solo con la interacción de objetos de oficina y en particular con el uso de computadores.

El *Xerox Alto*, desarrollado en el Xerox PARC en 1973, fue el primer ordenador personal, así como el primero que utilizó la metáfora de escritorio y una interfaz gráfica de usuario.[7]

[7] http://es.wikipedia.org/wiki/Xerox_Alto

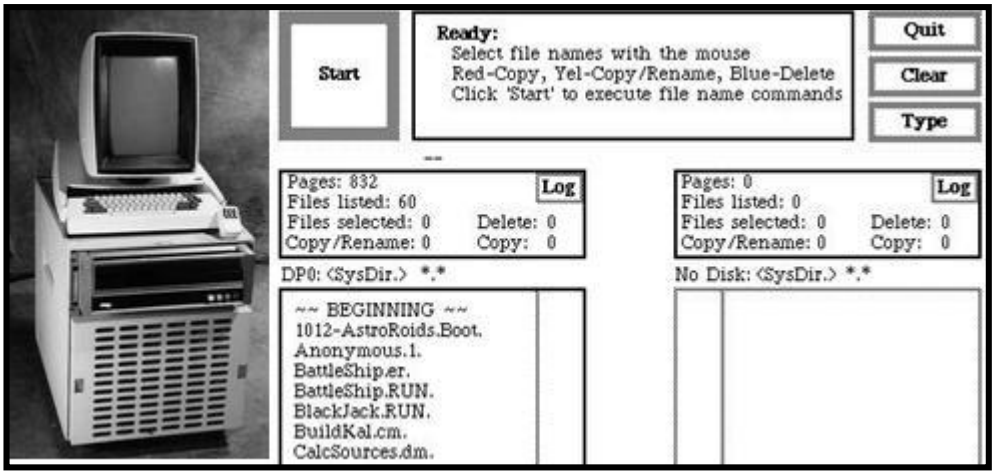


FIGURA 2.23Xerox Alto y su interfaz de usuario

[Fuente: http://www.textualmenteactivo.com/archives/4228]

En la actualidad las interfaces de usuario no se encuentran solamente en los computadores, sino también en equipos electrónicos de última tecnología, los cuales facilitan la interacción y la obtención de datos en cualquier lugar. Como Palm, Controles Remotos, Celulares, PDAs, GPS, o Tablets.

FIGURA 2.24 Diversos equipos electrónicos con interfaces de usuario.

[Fuente: http://complementosinformaticos.blogspot.com/2011/07/consejos-para-elegir-una-tablet.html]

Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, teclado, ratón y algunos sonidos que la máquina hace, y en

general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la máquina. La mejor interacción humano-máquina través de una interfaz (Interfaz de Usuario), es aquella que brinde tanto comodidad, como eficiencia.

Hoy en día las interfaces de operación directa (también llamadas interfaces gráficas de usuario, GUI por sus siglas en inglés) son prácticamente universales.

Las principales funciones de una interfaz HMI son las siguientes:

- Puesta en marcha y apagado.
- Control de las funciones manipulables del equipo.
- Manipulación de archivos y directorios.
- Herramientas de desarrollo de aplicaciones.
- Comunicación con otros sistemas.
- Información de estado.
- Configuración de la propia interfaz y entorno.
- Intercambio de datos entre aplicaciones.
- Control de acceso.
- Alarmas.
- Históricos.

- Supervisión y monitoreo

- Sistema de ayuda interactivo.

Existen muchos tipos de software para la creación de interfaces de usuario. El sistema de ventanas permite la división de la pantalla en diferentes regiones rectangulares, llamadas ``ventanas''. El sistema de ventanas provee de procedimientos que permiten a la aplicación el dibujar figuras en la pantalla y sirve como medio de entrada de las acciones del usuario.

INTOUCH

INTOUCH es la principal Interfaz Hombre Maquina del ámbito industrial, permite a ingenieros, supervisores, administradores y operadores visualizar e interactuar con el desarrollo de toda una operación a través de representaciones graficas de procesos de producción. Permite visualizar datos en tiempo real desde una PC ubicada en cualquier lugar de la red.

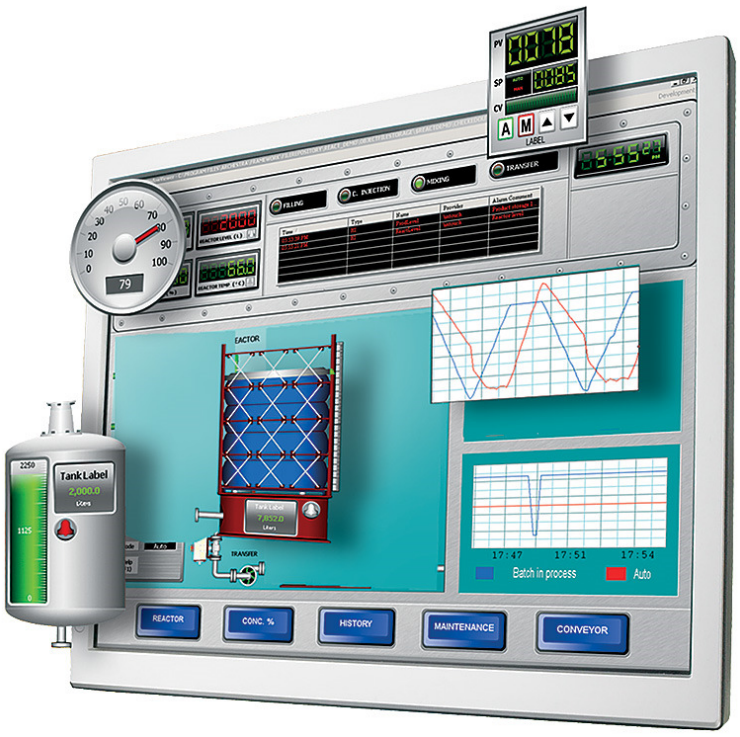


FIGURA 2.25Interfaz HMI del software IN TOUCH

[Fuente: http://procesoslagos.blogspot.com/2010/05/chile-en-el-mundial_05.html]

IN TOUCH es el software con el cual es posible ejecutar interfaces HMI, aplicables a sistemas de automatización de procesos. Por medio de esta interfaz se tendrá conocimiento de las operaciones que realiza un PLC dentro del proceso que se está llevando a cabo.

Entre las principales aplicaciones de INTOUCH se encuentran una variedad de mercados, incluyendo la elaboración de alimentos, los semiconductores, el petróleo y el gas, la industria automotriz, química, farmacéutica y papel, servicios públicos y más.

Principales ventajas

- Facilidad de uso, permite de forma rápida y sencilla a los desarrolladores y operadores a ser más productivos.

- Integración de dispositivos y conectividad a prácticamente todos los dispositivos y sistemas.

- Impresionante representación gráfica visual y la interacción con su operación trae la información correcta a las personas adecuadas en el momento adecuado

- Historia de la ruta de migración sin interrupciones de versión de software, lo que significa que la inversión en aplicaciones HMI está protegida

CAPITULO 3

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

3.1 DISEÑO

3.1.1 Diseño previo y diagrama de bloques del sistema.

El prototipo de entorno industrial a ser implementado, hace referencia a la utilización de dos brazos robóticos RV-2AJ Mitsubishi y el robot móvil Robotino de Festo; cuyos procesos independientes serán sincronizados a través del PLC S7-200 de Siemens.

El sistema consta de dos estaciones fijas y una móvil, distribuidas de la siguiente manera: Estación de Ensamblaje (Brazo Robótico Nº1), Estación de Clasificación y Almacenamiento (Brazo Robótico Nº2) y finalmente de Transporte (Robot móvil Robotino).

Diagrama de Bloques:

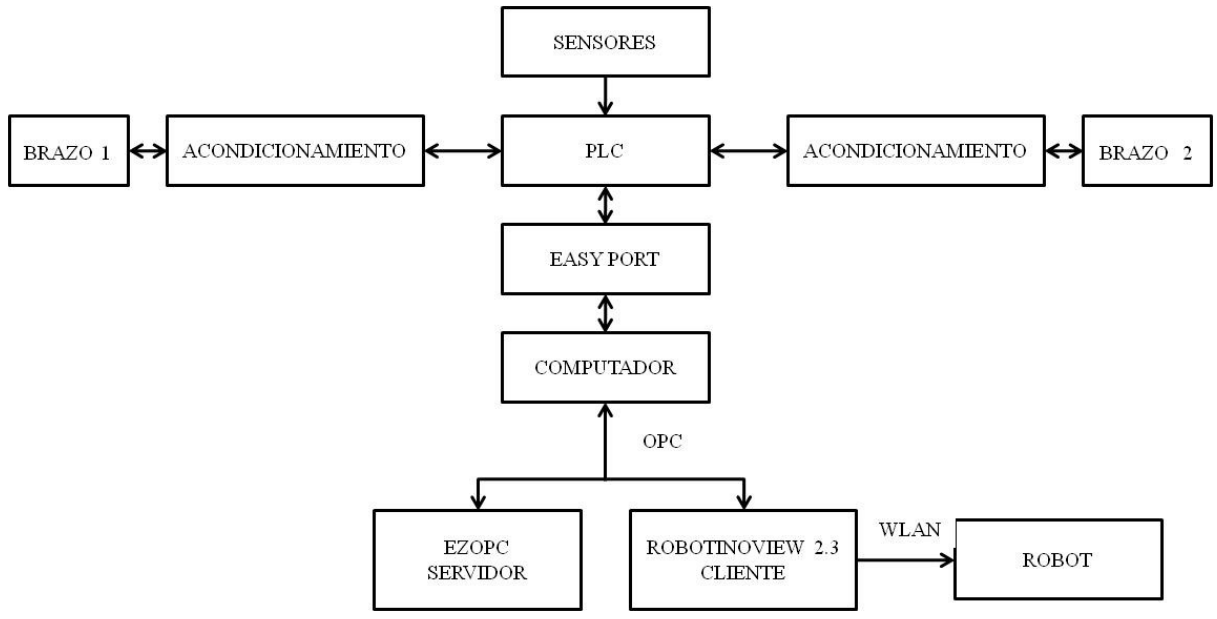


FIGURA 3.1 Diagrama de bloques del sistema

[Fuente: Autoras Tesis]

El PLC S7-200 funciona como controlador de todo el sistema, a éste ingresan señales para operar los diferentes brazos robóticos mediante un

acondicionamiento para su conexión, además se integran las señales de los sensores que se encuentran en el entorno de trabajo. Y para el acoplamiento del robot móvil se utiliza una interfaz EasyPort, la cual envía el estado del PLC hacia el computador. Se realiza la comunicación OPC dentro del computador entre el servidor EzOPC y el cliente RobotinoView2.6, y finalmente a través de una comunicación inalámbrica se envían dichos datos al robot móvil.

3.1.2 Entorno de trabajo

Ensamblaje de piezas Transporte Clasificación y almacenamiento

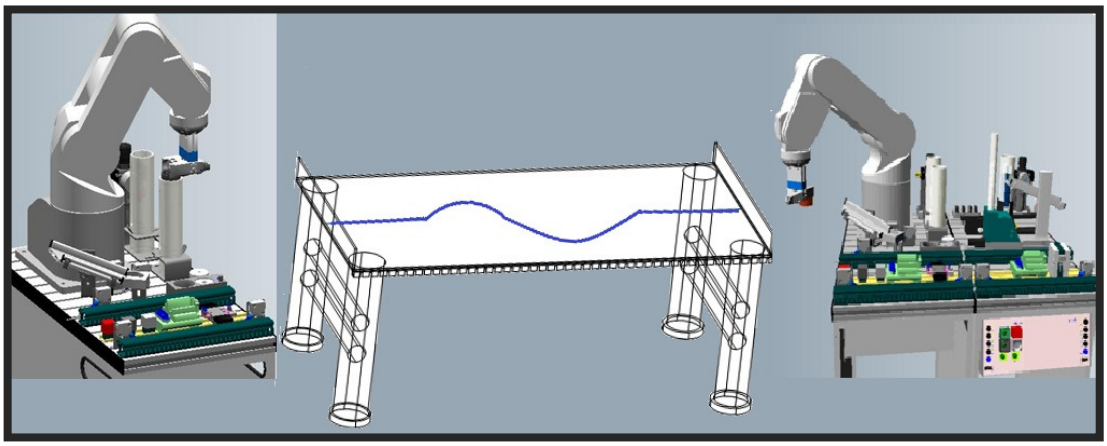


FIGURA 3.2 Diseño del entorno del sistema.

[Fuente: Autoras tesis]

Estación de Ensamblaje:

FIGURA 3.3 Estación de ensamblaje.

[Fuente: Autoras tesis]

Está compuesta por 2 estaciones complementarias, Robot manipulador y estación de montaje. Donde el manipulador ensambla de componentes (émbolo, muelle y tapa) en la pieza procesada (camisade cilindro), mientras que la estación de montaje, proporciona los elementos de acoplamiento del cilindro a la estación del robot. Obteniendo así, cilindros de simple efecto.

La función de la estación de ensamblaje es:

- Determinar las características del material de una pieza (cilindro),

- Depositar las piezas en el retenedor de montaje

- Armar un cilindro de simple efecto, colocando de forma correcta el émbolo, muelle y tapa; tomando en cuenta el tipo de cilindro como se muestra en la figura 3.3.

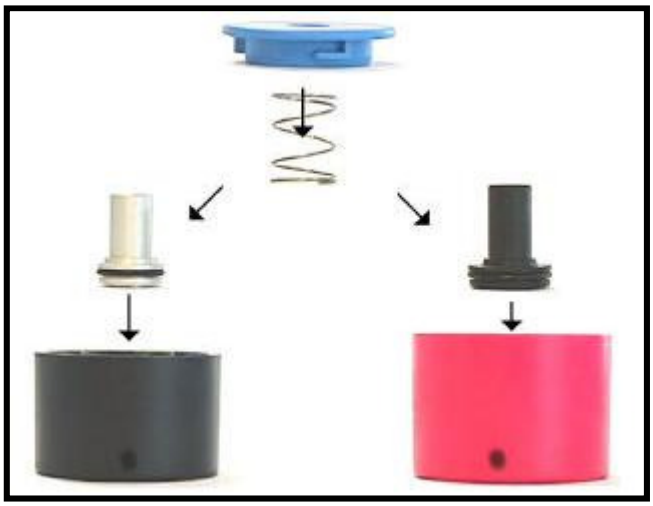


FIGURA 3.4 Ubicación de émbolo, muelle y tapa en el cilindro.

[Fuente: Célula de Fabricación Flexible, MPS-C Estación 6: Robot Festo Didactic-2003]

- Pasar las piezas a la estación siguiente.

Transporte:

FIGURA 3.5 Robot móvil

[Fuente: http://www.festo.com/net/startpage/robotino]

El robot móvil Robotino, es el encargado de trasportar las piezas, desde la estación de ensamblaje hasta la estación de clasificación y almacenamiento. Para lo cual, Robotino tiene incorporado un pallet en la parte posterior de su chasis.

Robotino se traslada entre estaciones, como seguidor de línea por medio de una cámara web incorporada. Y se detiene y posiciona en cada estación por medio de sus sensores infrarrojos de distancia.

El pallet contiene 2 sensores digitales de tipo LDR para la detección de existencia de piezas; y 2 motores DC para expulsar las piezas del pallet hacia la estación de clasificación.

Estación de clasificación y almacenamiento:

FIGURA 3.6 Estación de clasificación.

[Fuente: Autoras tesis]

La función de la estación de clasificación y almacenamiento es:

- Retirar piezas de la rampa de la estación.
- Determinar las características del material de una pieza,
- Depositar las piezas en el retenedor de montaje

- Depositar las piezas en el almacén de "rojas" o en el de "negras" según corresponda.

3.1.3 Diseño Electrónico

Acondicionamiento de sensores y motores para pallet de Robotino

Para el reconocimiento de la presencia de piezas en el pallet se utiliza dos sensores LDR, los cuales requieren de un circuito electrónico como acondicionamiento, para obtener una señal de salida digital entre 24V y 0V. Esta señal ingresará directamente a las entradas digitales de la bornera I/O del robot móvil Robotino.

Para el depósito de piezas armadas en la estación se clasificación se utilizarán 2 motores, cuyo circuito de control está diseñado en conjunto con el acoplamiento de los sensores.

Los elementos electrónicos usados para dicho fin son:

- 2 Sensores LDR
- 2 Potenciómetros de precisión 5KΩ
- 7 Resistencias de 10 KΩ
- 7 Resistencias de 4.7 KΩ
- 2 Transistores 2N3904
- 1 Circuito integrado L293D
- 8 Diodos 1N4007

- Bornera de 10 pines

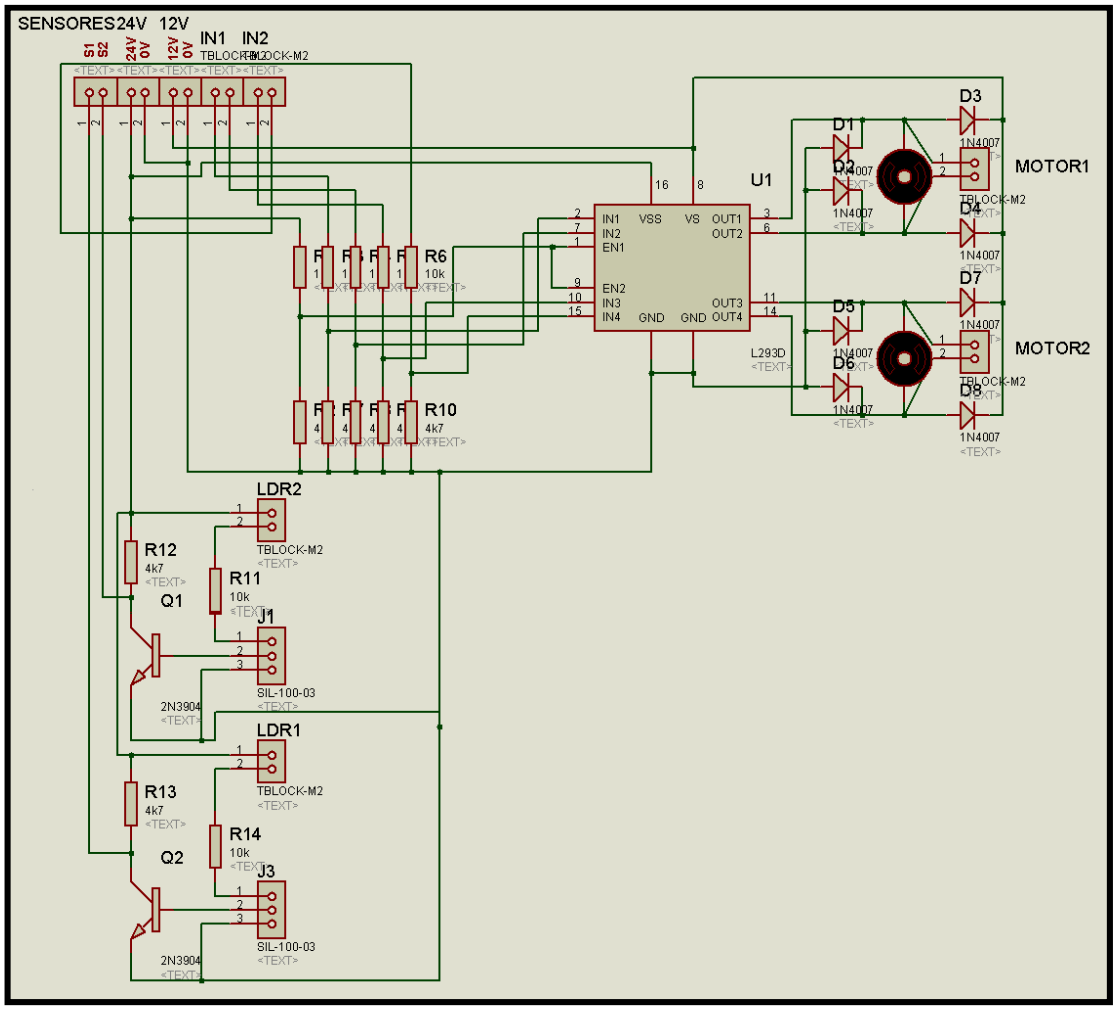


FIGURA 3.7 Diseño de acoplamiento de sensores y motores realizada en Proteus 7 Professional

[Fuente: autoras tesis]

El circuito diseñado en la figura anterior, por una parte permite tomar los datos de 2 LDR que indicarán la existencia de piezas en el pallet, por lo que se ha hecho una configuración para que la señal sea de tipo digital, para esto se utilizan los transistores 2N3904 que conmutan dependiendo de la intensidad de luz que llegue a las LDRs, además se puede calibrar la sensibilidad mediante J1 y J3 que son potenciómetros de precisión de 5KΩ.

Por otro lado, se tiene el circuito de control de giro de 2 motores DC, dicho control se realiza con un circuito integrado L293D que funciona con la configuración de Puente H, con lo cual se puede accionar los motores en un

sentido horario y anti horario, los diodos utilizados son para limitar la corriente y activarlos según la dirección de giro que se requiera. Las señales de entrada del L293D soportan hasta 7Vdc, por lo que se ha limitado el voltaje de 24Vdc de la entrada del circuito a través de las resistencias de 10KΩ y 4.7KΩ. Se utiliza una fuente externa de 12Vdc para la alimentación exclusiva de los motores.

3.1.4 Circuitos de control

El controlador de todo el sistema es el PLC S7-200, a este se ingresan todas las señales del entorno industrial a controlar y estas son; las señales de los sensores del entorno, las señales del robot móvil Robotino mediante el EasyPort y las señales de los brazos robóticos.

Gracias al controlador lógico programable se puede obtener una sincronía y monitoreo en el proceso industrial a controlar.

El diagrama de conexión se muestra a continuación:

Nomenclatura; 1=Entrada, 0=Salida

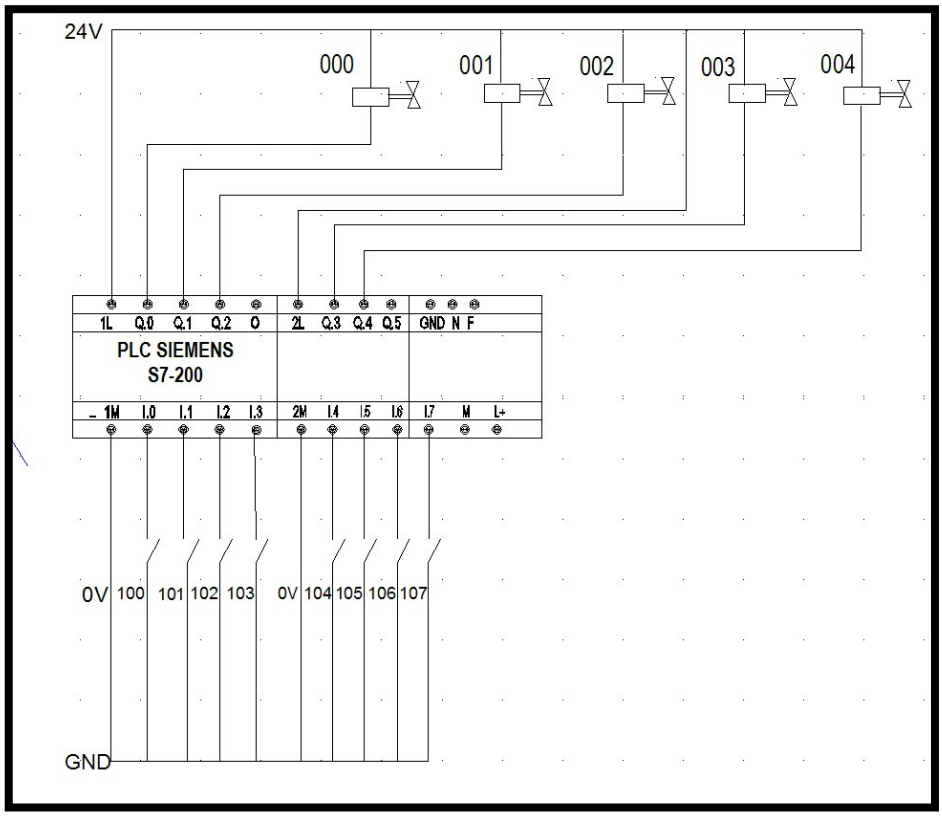


FIGURA 3.8 Diseño de las conexiones para PLC.

[Fuente: Autoras tesis]

DESCRIPCION	I/PLC	ETIQUETA	O/PLC	ETIQUETA
STOP	I0.0	100	Q0.0	000
START	I0.1	101	Q0.1	001
LLEGA ROBOTINO BR1	I0.2	102	Q0.2	002
LLEGA ROBOTINO BR2	I0.4	104	Q0.3	003
INICIA CONTADOR C1	I0.3	103		
SENSA COLOR BR1	I0.5	105		
INICIA CONTADOR 1 BR2 C3	I0.6	106		
SENSA COLOR BR2	I0.7	107		
BR1 ARMA 2 PIEZAS			Q0.4	004

TABLA 3.1 Entradas y salidas de PLC.

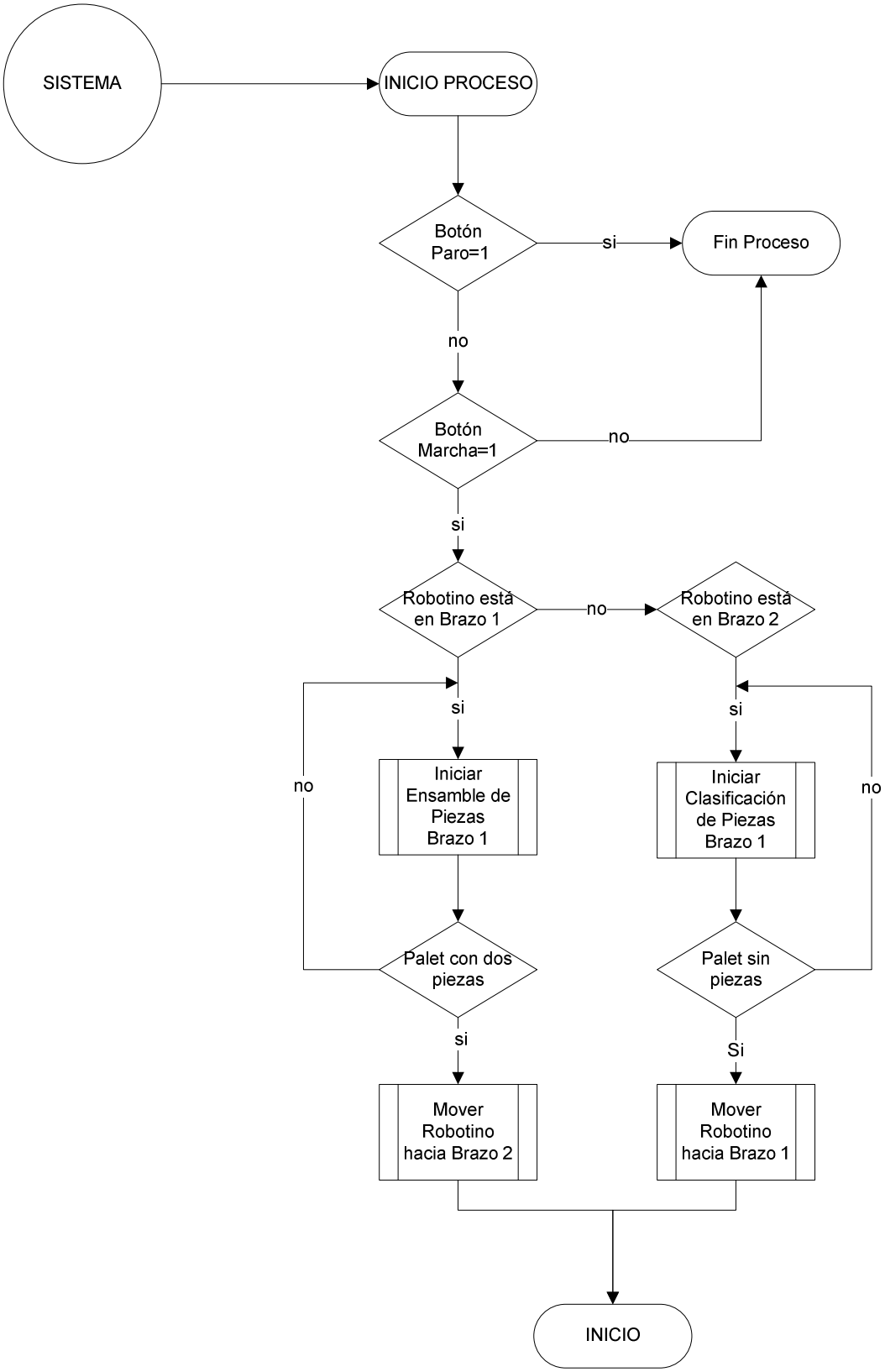
[Fuente: Autoras Tesis]

3.1.5 Diseño del Software

3.1.5.1 Algoritmo del sistema

A continuación se muestra el algoritmo de funcionamiento de todo el sistema a ser implementado, donde se incluyen los brazos robóticos RV-2AJ, el robot móvil Robotino de Festo y el controlador lógico programable PLC S7200.

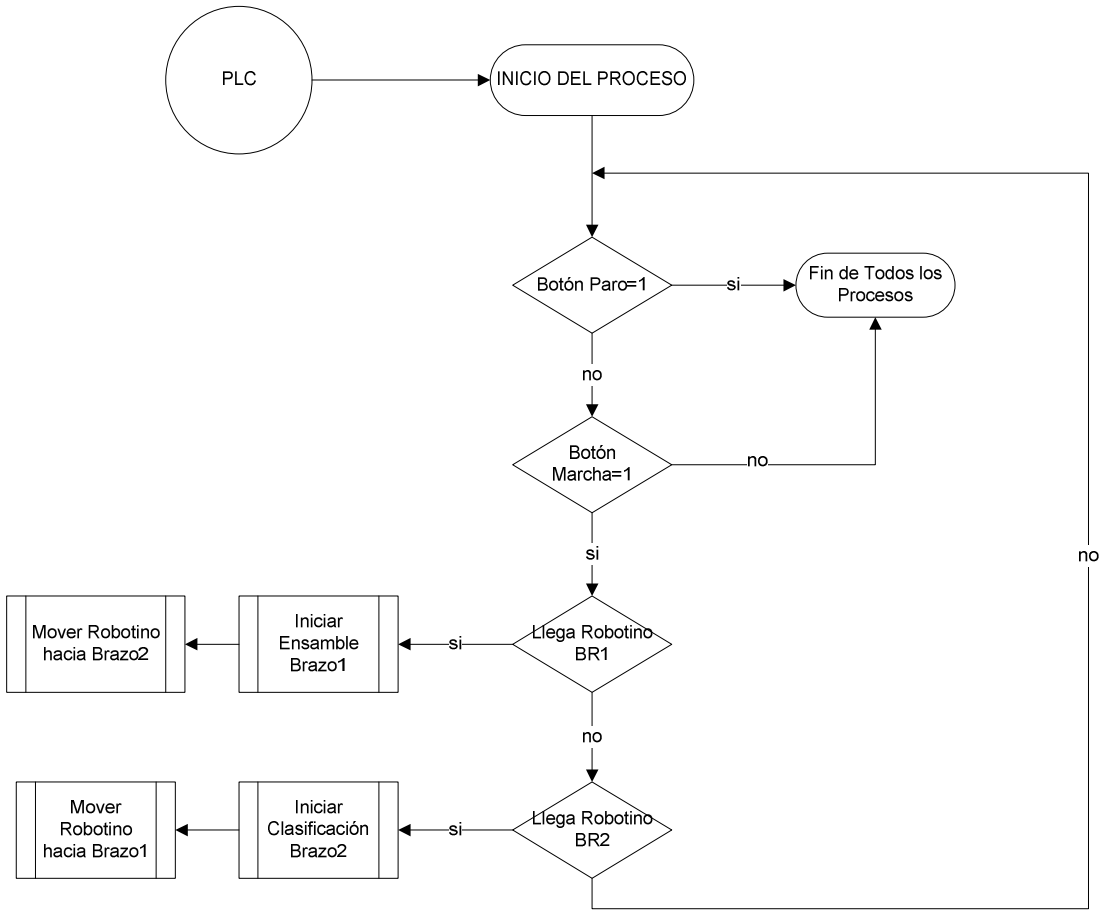
El sistema entra a trabajar cuando la señal de Inicio 101 esta activada. El brazo 1 de la estación de ensamblaje empieza armar las piezas con la señal de Inicio y con la señal de que Robotino está en dicha estación. Una vez armadas 2 piezas y colocadas sobre el Pallet de Robotino, este último se dirige hacia la estación de clasificación mediante seguidor de línea con cámara. Cuando llega a la estación de clasificación, expulsa las piezas y se activa una señal para que el brazo 2 recoja las piezas de la rampa y las clasifique según corresponda en los contenedores 1 y2.


SISTEMA
INICIO PROCESO
Botón Paro=1
si
Fin Proceso
no
Botón Marcha=1
no
si
Robotino está en Brazo 1
no
Robotino está en Brazo 2
si
si
no
Iniciar Ensamble de Piezas Brazo 1
Iniciar Clasificación de Piezas Brazo 1
no
Palet con dos piezas
Palet sin piezas
si
Si
Mover Robotino hacia Brazo 2
Mover Robotino hacia Brazo 1
INICIO

3.1.5.2 Algoritmo del PLC

El siguiente diagrama de flujo especifica el diseño de programación para el Controlador Lógico Programable, el cual está a cargo del control del entorno industrial. Por lo tanto esta directamente conectado al panel de control de la estación de ensamblaje, donde se puede accionar los botones de Marcha/Paro, con la señal de marcha todos los sistemas están listos para iniciar, y con la señal de Paro todos los sistemas se detienen, puesto que se trata de un paro de emergencia.

El PLC tiene a cargo las siguientes señales: Inicio de todos los procesos, Paro de todos los procesos, señal de llegada de Robotino a las estaciones y señal que se ensamblaron 2 piezas; algunas de estas señales son bidireccionales, llegan y salen desde PLC.



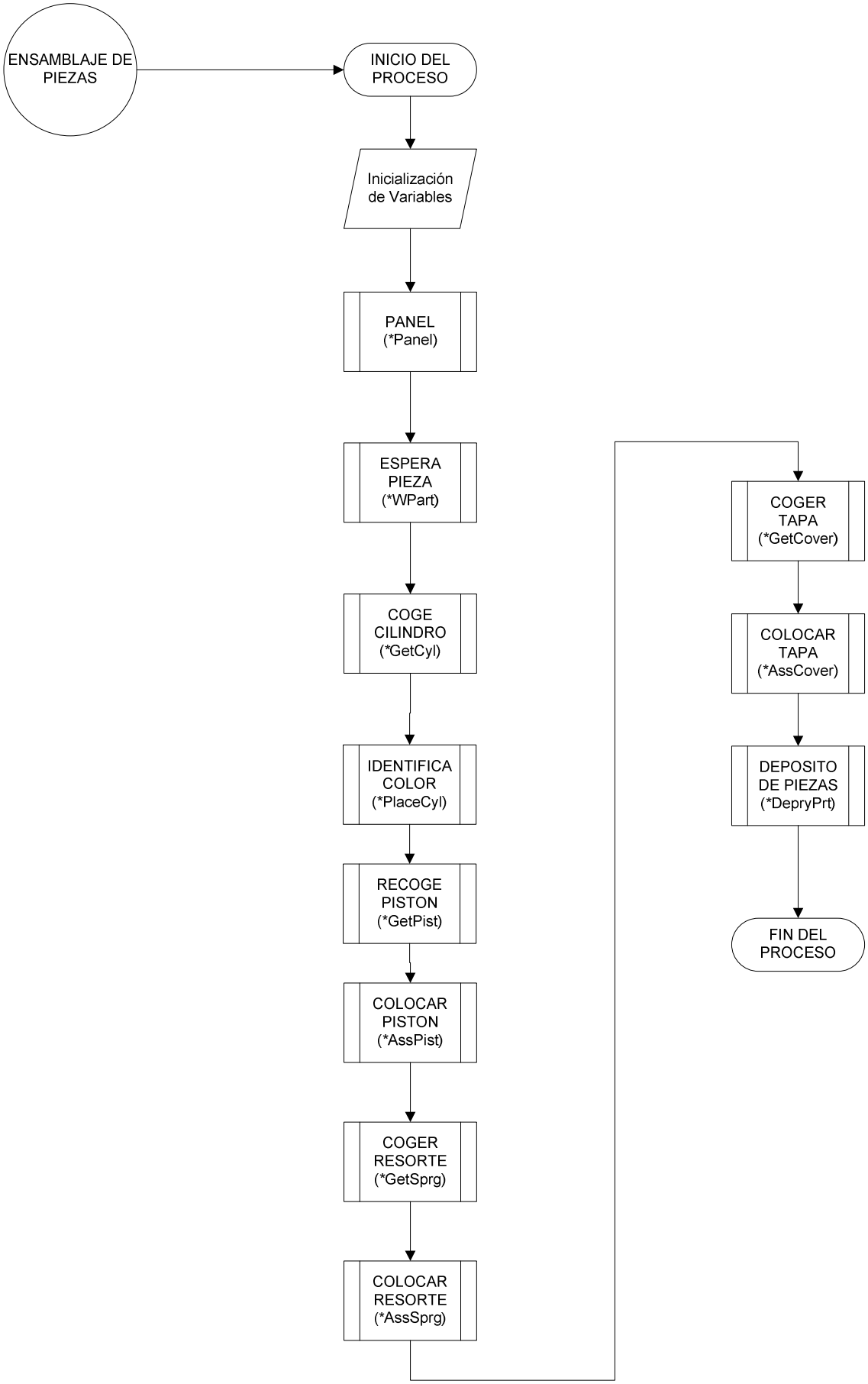
3.1.5.3 Algoritmo del Brazo Robótico (estación de ensamblaje)

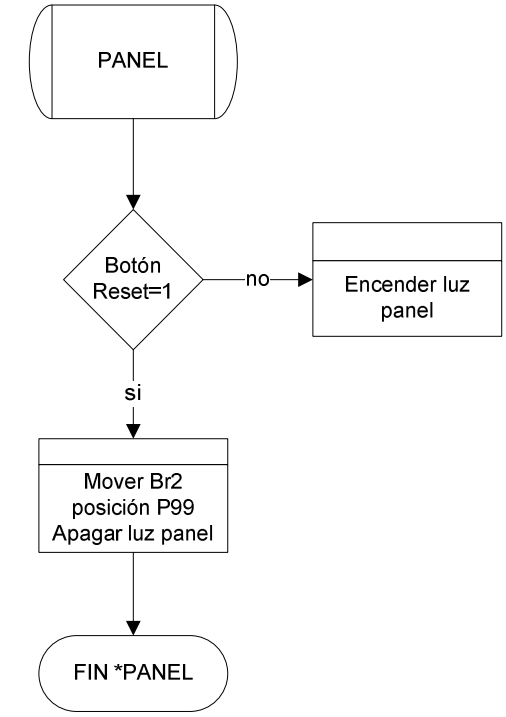
El algoritmo de programación para el brazo robótico de la estación de ensamblaje se detalla en el siguiente diagrama de flujo, donde se puntualiza cada subrutina del programa principal de dicha estación.

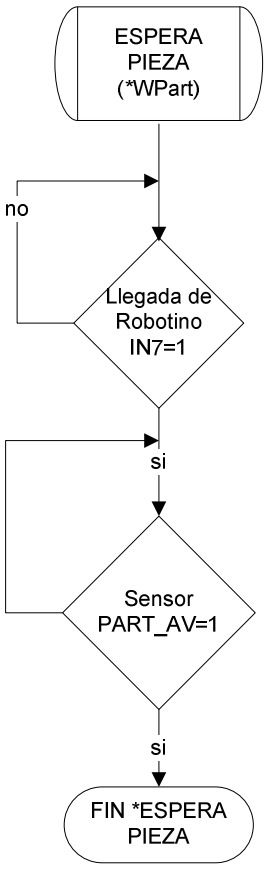
La puesta en marcha del proceso se da mediante el pulsador “Inicio” del panel principal, con el cual, el brazo 1 se coloca en la posición inicial P99. Para iniciar el ensamblaje de las piezas se requiere la señal de entrada del PLC, el cual nos indica que el Robotino se encuentra en la posición adecuada, además se requiere verificar que existan cilindros colocados en la rampa del entorno para dar inicio a la estación. Si estas señales están activas el brazo robótico recoge el cilindro y lo coloca adecuadamente en la estación de ensamblaje. Para la identificación del color el brazo se coloca en la posición P3. Una vez identificado el color este escoge el correspondiente pistón y lo coloca dentro del cilindro. A continuación se activa la salida del pistón para expulsar el resorte, el brazo robótico detecta

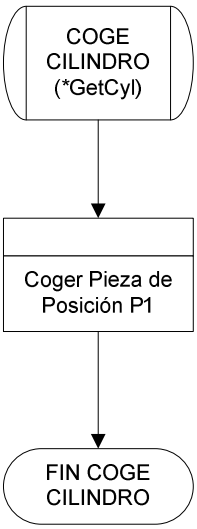
mediante sensores ópticos que exista resorte y lo recoge de la posición P4 y lo coloca dentro del cilindro. De la misma forma se activa el pistón para expulsar la tapa, el brazo robótico verifica mediante sensores ópticos que exista tapa y la recoge de la posición P9, la transporta a la posición P12 para colocar la tapa en la posición adecuada y finalmente la coloca y ajusta sobre el cilindro.

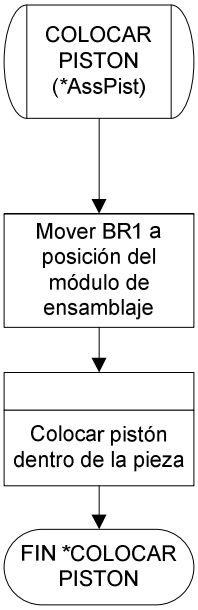
Una vez terminado el ensamblaje de la pieza el brazo robótico la coloca sobre el pallet que lleva el Robotino en la posición P13 y P14 respectivamente. Después de armar dos piezas el brazo robótico se detiene hasta iniciar un nuevo proceso.

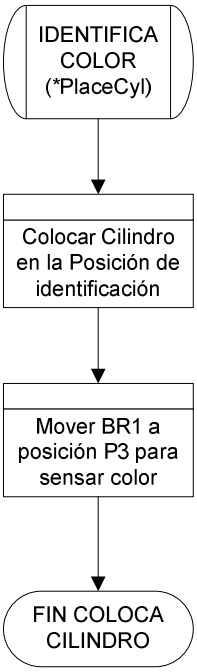

ENSAMBLAJE DE PIEZAS
INICIO DEL PROCESO
Inicialización de Variables
PANEL (*Panel)
ESPERA PIEZA (*WPart)
COGE CILINDRO (*GetCyl)
IDENTIFICA COLOR (*PlaceCyl)
RECOGE PISTON (*GetPist)
COLOCAR PISTON (*AssPist)
COGER RESORTE (*GetSprg)
COLOCAR RESORTE (*AssSprg)
COGER TAPA (*GetCover)
COLOCAR TAPA (*AssCover)
DEPOSITO DE PIEZAS (*DepryPrt)
FIN DEL PROCESO

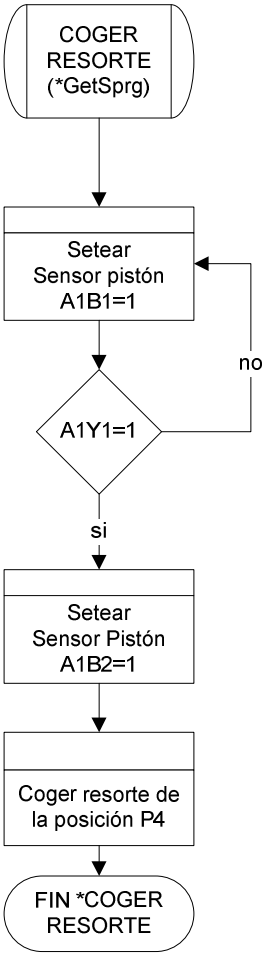

PANEL
Botón
Reset=1
no
Encender luz
panel
si
Mover Br2
posición P99
Apagar luz panel
FIN *PANEL

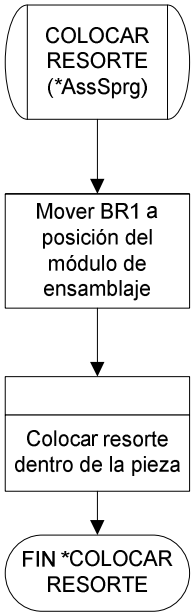

ESPERA
PIEZA
(*WPart)
no
Llegada de
Robotino
IN7=1
si
Sensor
PART_AV=1
si
FIN *ESPERA
PIEZA

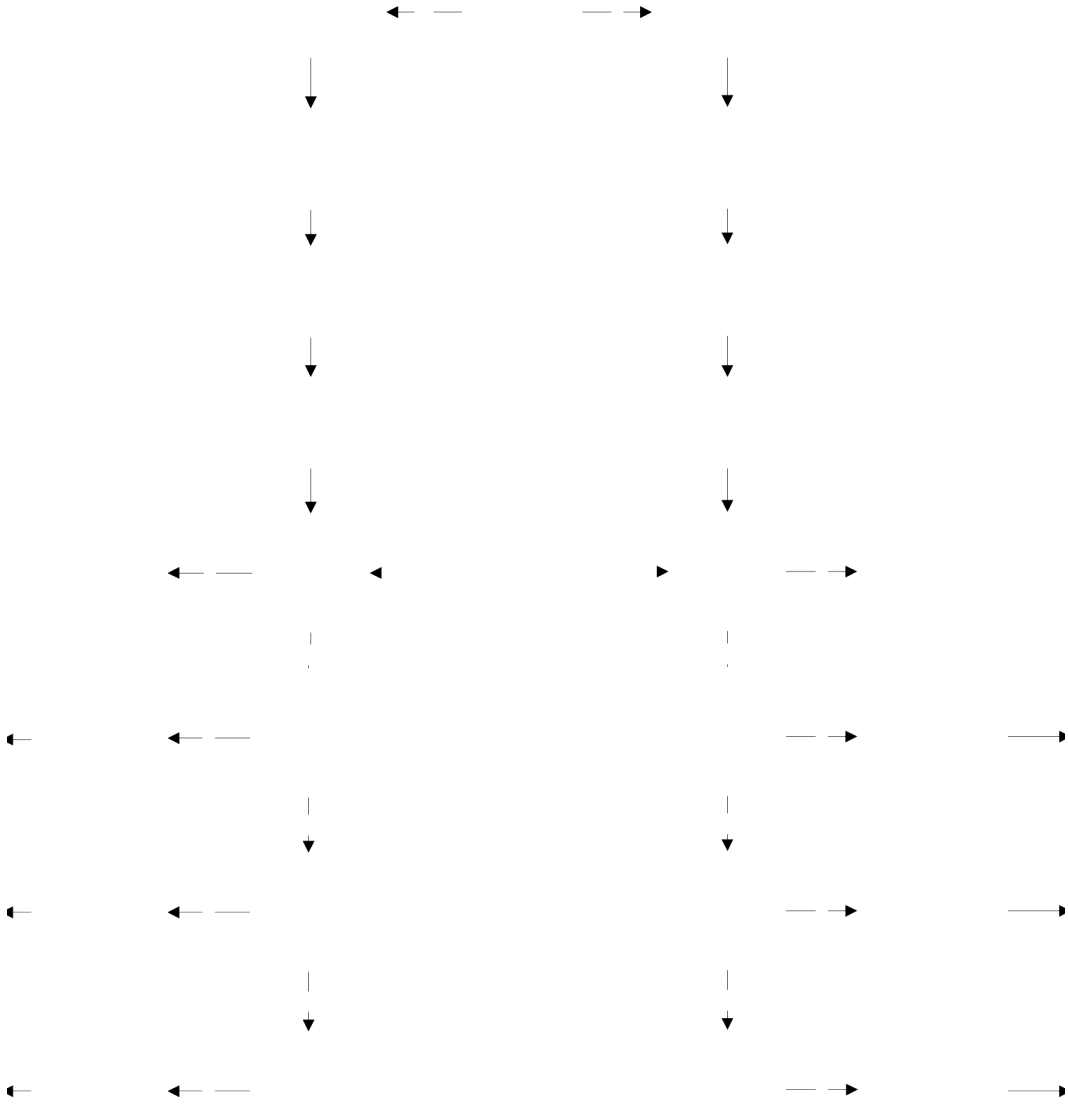

COGE CILINDRO (*GetCyl)
Coger Pieza de Posición P1
FIN COGE CILINDRO

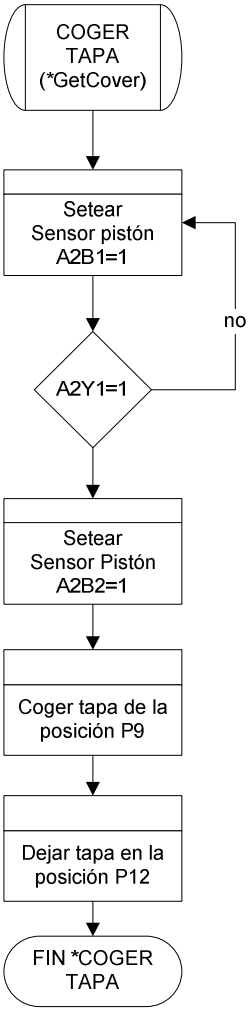

COLOCAR PISTON (*AssPist)
Mover BR1 a posición del módulo de ensamblaje
Colocar pistón dentro de la pieza
FIN *COLOCAR PISTON

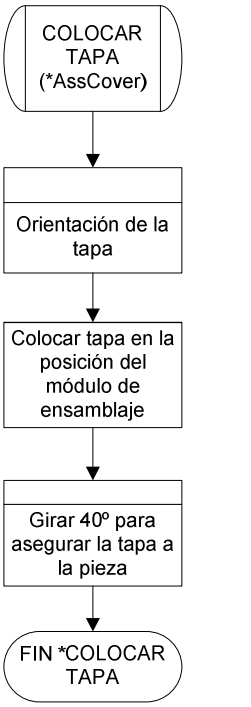

IDENTIFICA COLOR (*PlaceCyl)
Colocar Cilindro en la Posición de identificación
Mover BR1 a posición P3 para sensar color
FIN COLOCA CILINDRO

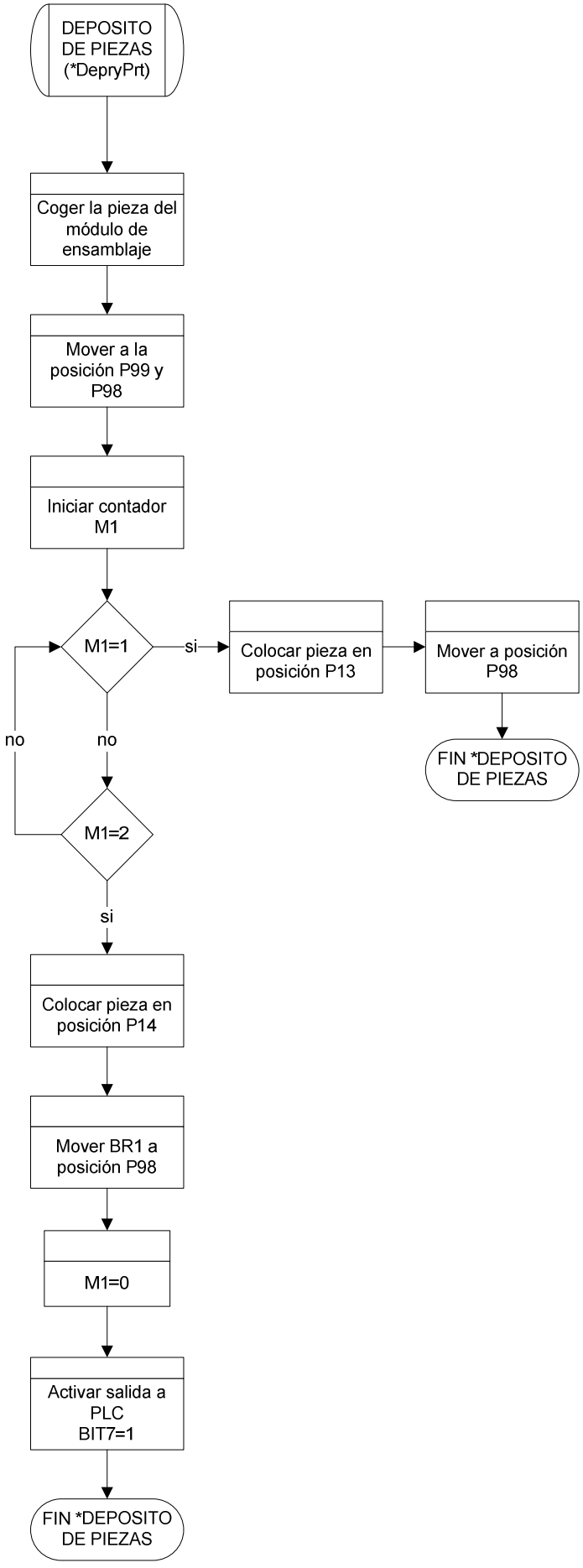

COGER RESORTE (*GetSprg)
Setear Sensor pistón A1B1=1
A1Y1=1
no
si
Setear Sensor Pistón A1B2=1
Coger resorte de la posición P4
FIN *COGER RESORTE


COLOCAR RESORTE (*AssSprg)
Mover BR1 a posición del módulo de ensamblaje
Colocar resorte dentro de la pieza
FIN *COLOCAR RESORTE




COGER TAPA (*GetCover)
Setear Sensor pistón A2B1=1
no
A2Y1=1
Setear Sensor Pistón A2B2=1
Coger tapa de la posición P9
Dejar tapa en la posición P12
FIN *COGER TAPA

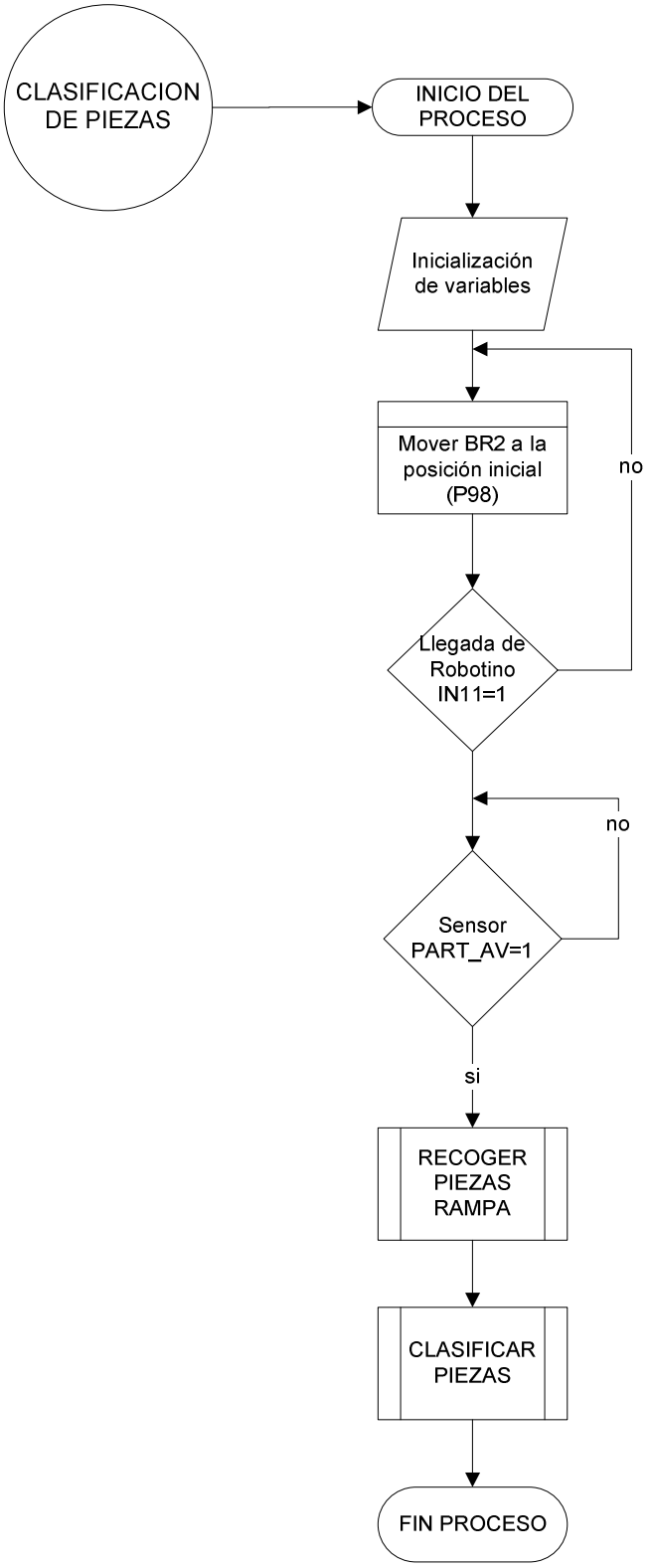

COLOCAR TAPA (*AssCover)
Orientación de la tapa
Colocar tapa en la posición del módulo de ensamblaje
Girar 40º para asegurar la tapa a la pieza
FIN *COLOCAR TAPA

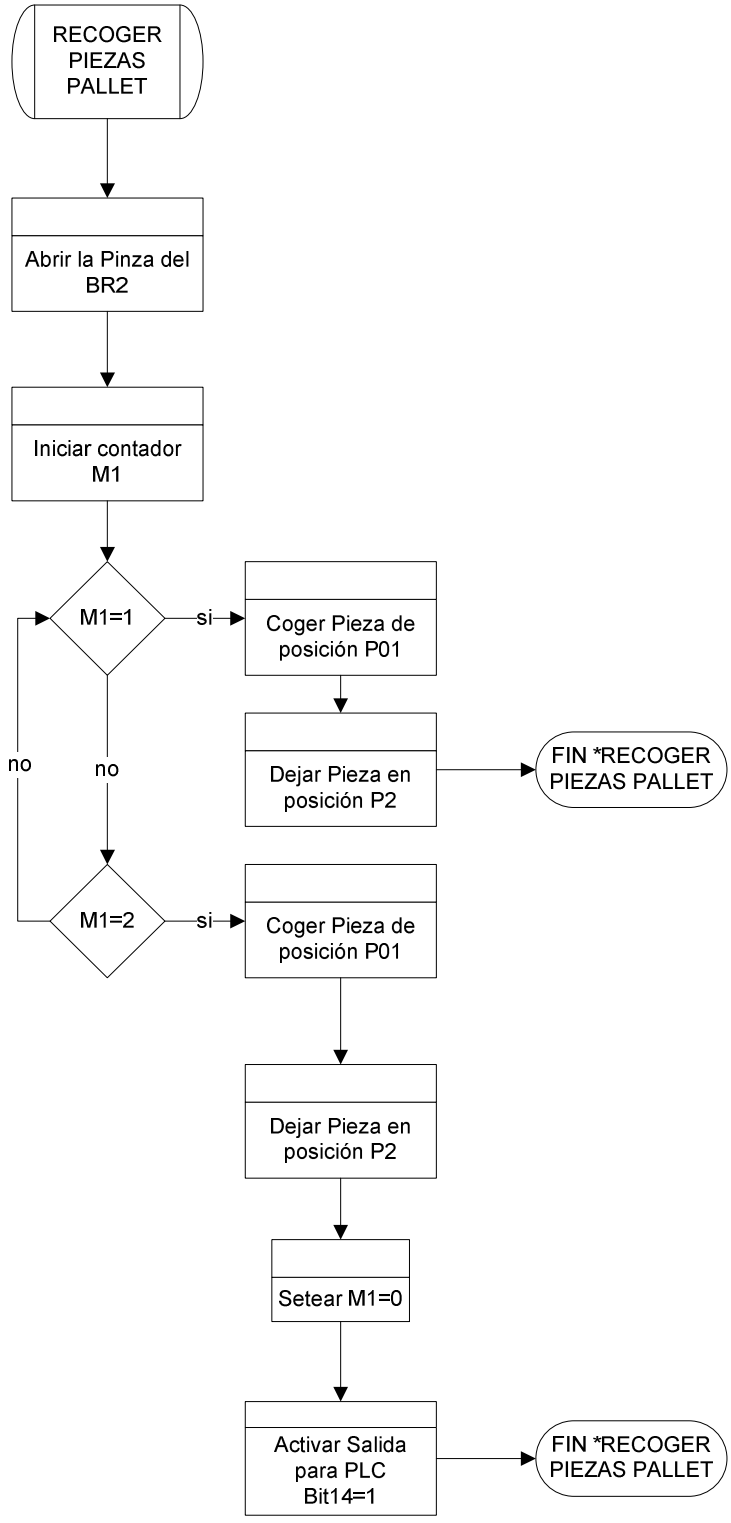

DEPOSITO DE PIEZAS (*DepryPrt)
Coger la pieza del módulo de ensamblaje
Mover a la posición P99 y P98
Iniciar contador M1
M1=1
si
Colocar pieza en posición P13
Mover a posición P98
FIN *DEPOSITO DE PIEZAS
no
no
M1=2
si
Colocar pieza en posición P14
Mover BR1 a posición P98
M1=0
Activar salida a PLC BIT7=1
FIN *DEPOSITO DE PIEZAS

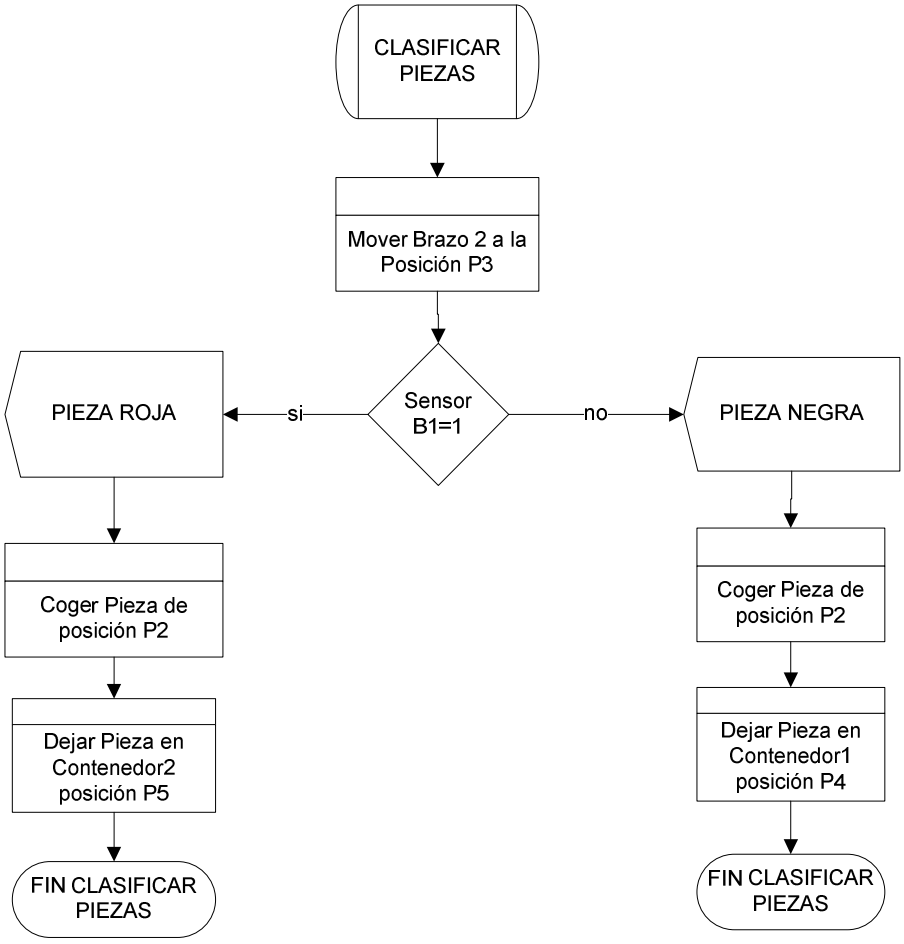
3.1.5.4 Algoritmo del Brazo Robótico (estación de clasificación y almacenamiento)

El algoritmo de programación para el brazo robótico de la estación de clasificación se lo realiza en base al siguiente diagrama de flujo, donde se especifica cada subrutina del programa principal. La condición de inicio para la estación está dada mediante la señal del PLC que indica que ha llegado el Robotino con las piezas a ser clasificadas y además mediante la señal de existencia de piezas en la rampa del entorno. Si estas señales han sido activadas, entonces inicia la subrutina Recoger Piezas, donde, el brazo se ubica en la posición P1 recoge la pieza de la rampa y la coloca en el modulo de montaje posición P2.

Luego el brazo se dirige a la posición P3 donde sensa el color de la pieza, es decir se encuentra en la subrutina de Clasificar piezas, Si el sensor B1 =1, la pieza es roja y se coloca en el contenedor 2 (Posición P5), si el sensor B1=0, la pieza es negra y se debe colocar en el contenedor 1(Posición P4).

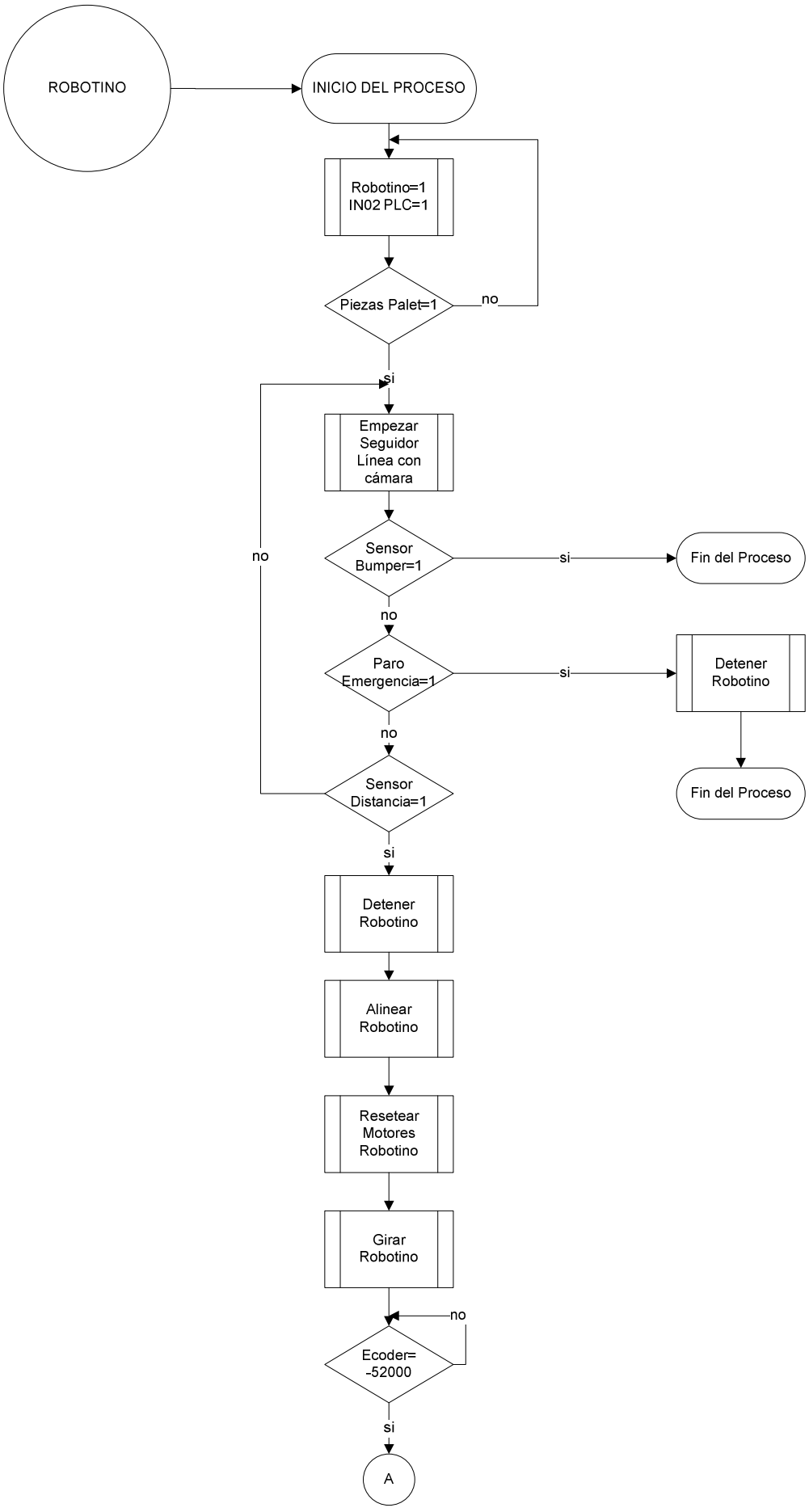

CLASIFICACION DE PIEZAS
INICIO DEL PROCESO
Inicialización de variables
Mover BR2 a la posición inicial (P98)
no
Llegada de Robotino IN11=1
no
Sensor PART_AV=1
si
RECOGER PIEZAS RAMPA
CLASIFICAR PIEZAS
FIN PROCESO

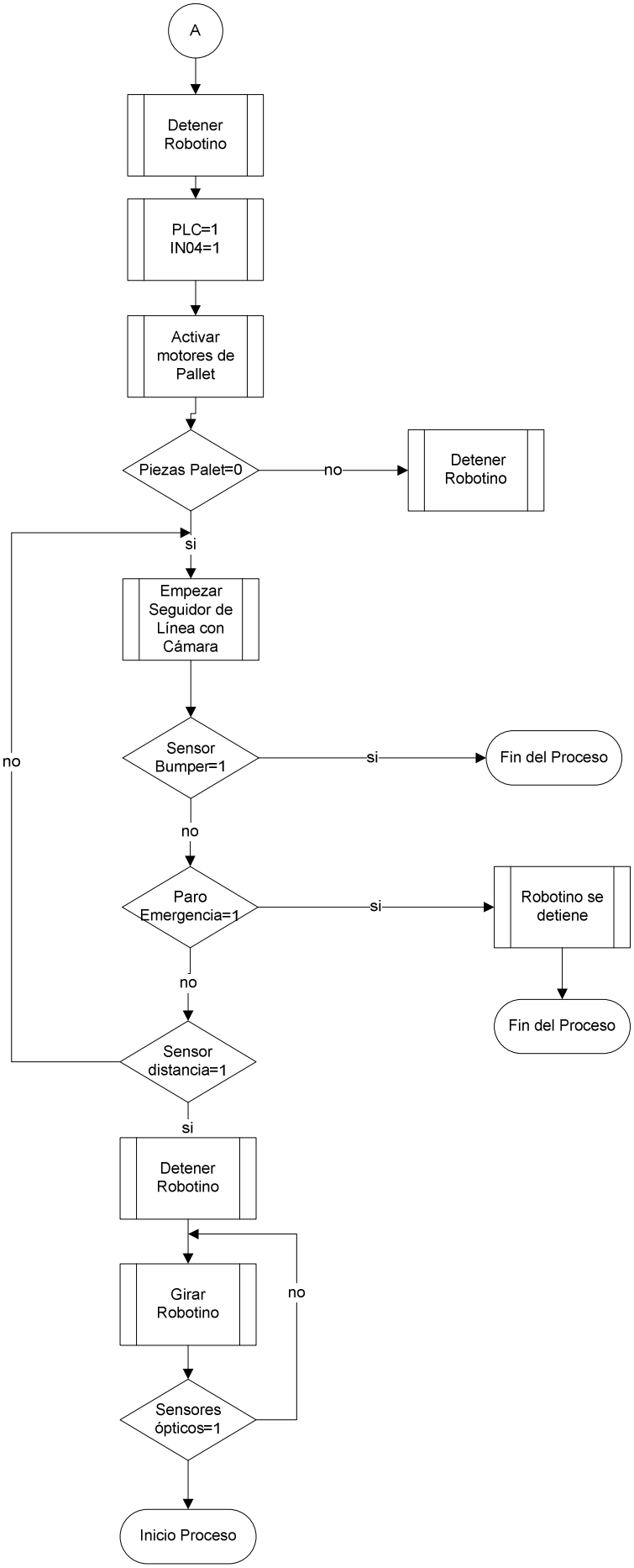

RECOGER PIEZAS PALLET
Abrir la Pinza del BR2
Iniciar contador M1
M1=1
si
Coger Pieza de posición P01
Dejar Pieza en posición P2
FIN *RECOGER PIEZAS PALLET
no
no
M1=2
si
Coger Pieza de posición P01
Dejar Pieza en posición P2
Setear M1=0
Activar Salida para PLC Bit14=1
FIN *RECOGER PIEZAS PALLET



3.1.5.5 Algoritmo del robot móvil Robotino

El algoritmo de programación para el robot móvil Robotino se lo realiza en base al siguiente diagrama de flujo, en el cual se detalla la rutina de funcionamiento. Inicialmente el Robotino se encuentra en la estación de ensamblaje, la condición de inicio para el Robotino está dada por la señal del PLC, la cual indica que se han ensamblado dos piezas, además el Robotino comprueba tener las dos piezas colocadas en el Pallet mediante la señal de dos sensores. Cuando estas dos señales están activas, el Robotino se desplaza por la pista como seguidor de línea usando una cámara web hasta llegar a la estación de clasificación, en la cual expulsa las piezas del pallet hacia la rampa del entorno de la estación. Para regresar a la posición inicial comprueba que las piezas no se encuentren en el pallet (a través de los sensores LDR) y se moviliza hacia la estación de ensamblaje para repetir todo el proceso.


ROBOTINO
INICIO DEL PROCESO
Robotino=1
IN02 PLC=1
Piezas Palet=1
no
si
Empezar
Seguidor
Línea con
cámara
Sensor
Bumper=1
si
Fin del Proceso
no
Paro
Emergencia=1
si
Detener
Robotino
Fin del Proceso
no
Sensor
Distancia=1
no
si
Detener
Robotino
Alinear
Robotino
Resetear
Motores
Robotino
Girar
Robotino
no
Ecoder=
-52000
si
A


A
Detener Robotino
PLC=1
IN04=1
Activar motores de Pallet
Piezas Palet=0
no
Detener Robotino
si
Empezar Seguidor de Línea con Cámara
Sensor Bumper=1
si
Fin del Proceso
no
Paro Emergencia=1
si
Robotino se detiene
Fin del Proceso
no
Sensor distancia=1
no
si
Detener Robotino
Girar Robotino
Sensores ópticos=1
no
Inicio Proceso

3.1.6 Diseño de la comunicación OPC

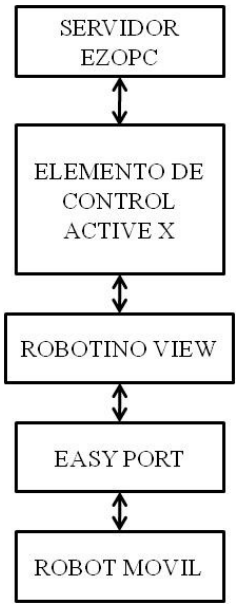


FIGURA 3.8 Diagrama de bloques de comunicación OPC.

[Fuente: autoras tesis]

Servidor EZOPC.- Para conseguir la comunicación OPC a través del EasyPort se utiliza el software de FestoDidactic llamado EzOPC.

Elemento de control ActiveX.- este es un elemento de control que sirve de interfaz de programación para el EasyPort USB. Puede incluirse en el software como objeto COM- Los lenguajes de programación que soporta la interface COM son, Visual Basic, C++, y HTML. Pero también los sistemas de captación de datos como LabVIEW y sistemas de visualización que soporten la interfaz COM.

Robotino View.- es el elemento de programación grafica intuitivo de Festo para el Robotino. Permite crear y ejecutar programas de control para el robot móvil. A este software es posible configurarlo como cliente para obtener la comunicación OPC.

Robot móvil.- Para recibir y enviar los datos al robot móvil por medio de la comunicación OPC, es necesario conectar el Robot móvil al Robotino VIEW por medio de una IP asignada en la red propia del dispositivo.

3.1.7 Diseño de la interfaz de usuario

La interfaz gráfica de usuario GUI, permite la visualización de un entorno de trabajo de manera amigable y fácil de interactuar.

El software InTouch ofrece funciones de visualización gráfica que llevan sus capacidades de gestión de operaciones, control y optimización a un nivel completamente nuevo. Aquello que ahora se conoce en la industria como HMI (Human Machine Interface) comenzó hace más de veinte años con el software InTouch. Ningún otro HMI en el mercado puede compararse al software InTouch en términos de innovación, integridad de arquitectura, conectividad e integración de dispositivos, ruta de migración de versiones de software sin interrupciones y facilidad de uso.

Esto se traduce en sistemas basados en estándares que permiten incrementar al máximo la productividad, optimizar la efectividad del usuario, mejorar la calidad y reducir los costos operacionales, de desarrollo y de mantenimiento.[8]

El diseño de la interfaz HMI consta de dos pantallas una principal o de inicio y otra donde se tendrá el proceso completo.

La pantalla de inicio muestra solamente la portada de la interfaz, desde aquí se podrá acceder a la ventana que contiene el HMI a través del botón "*PROCESO*".

[8] Extraído de: http://global.wonderware.com/LA/Pages/WonderwareInTouchHMI.aspx

FIGURA 3.9 Diseño de pantalla de Inicio de interfaz HMI.

[Fuente: autoras tesis]

Dentro de la ventana *PROCESO*, como muestra la figura 3.8 se visualizarán cada una de las estaciones del entorno en estudio, y la actividad que realice cada una de ellas por medio de la intermitencia de indicadores en el lugar donde suceda un acontecimiento o cambio dentro del proceso. Además se puede visualizar el estado en que se encuentre dicho entorno, es decir, Activado o en Paro de Emergencia. Adicionalmente la interfaz permitirá conocer cuántas piezas Rojas y/o Negras han sido clasificadas.

Para retornar a la ventana principal se debe accionar el botón "*INICIO*".

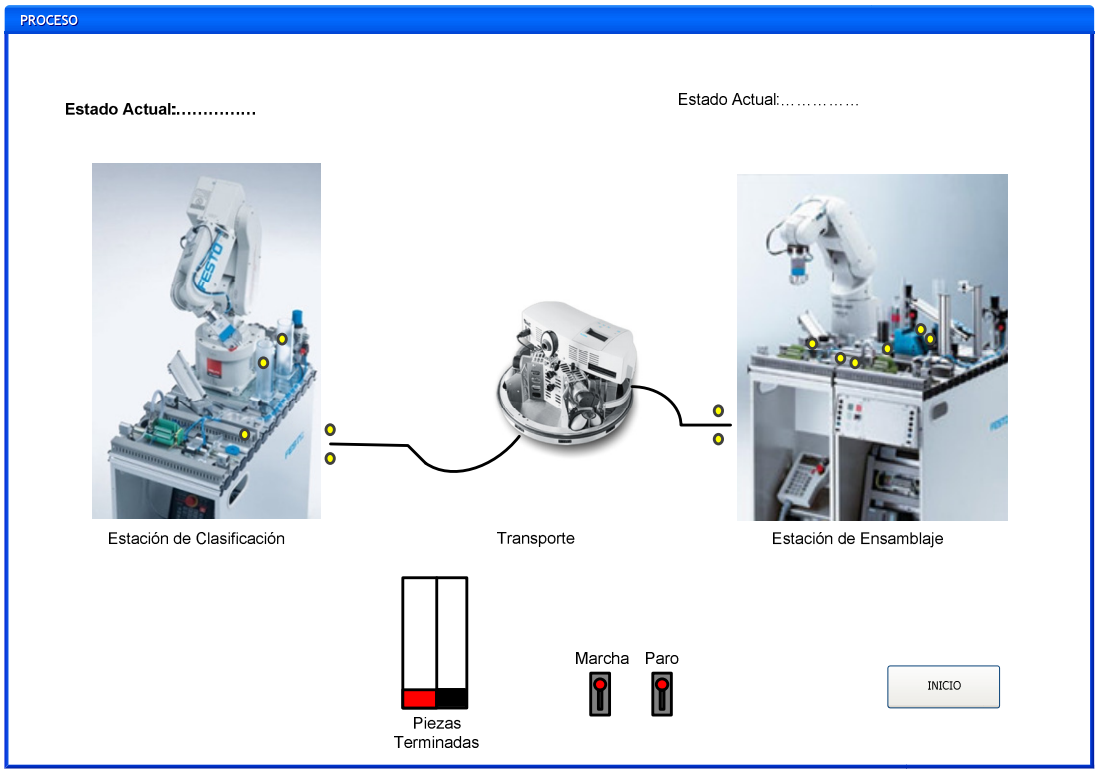


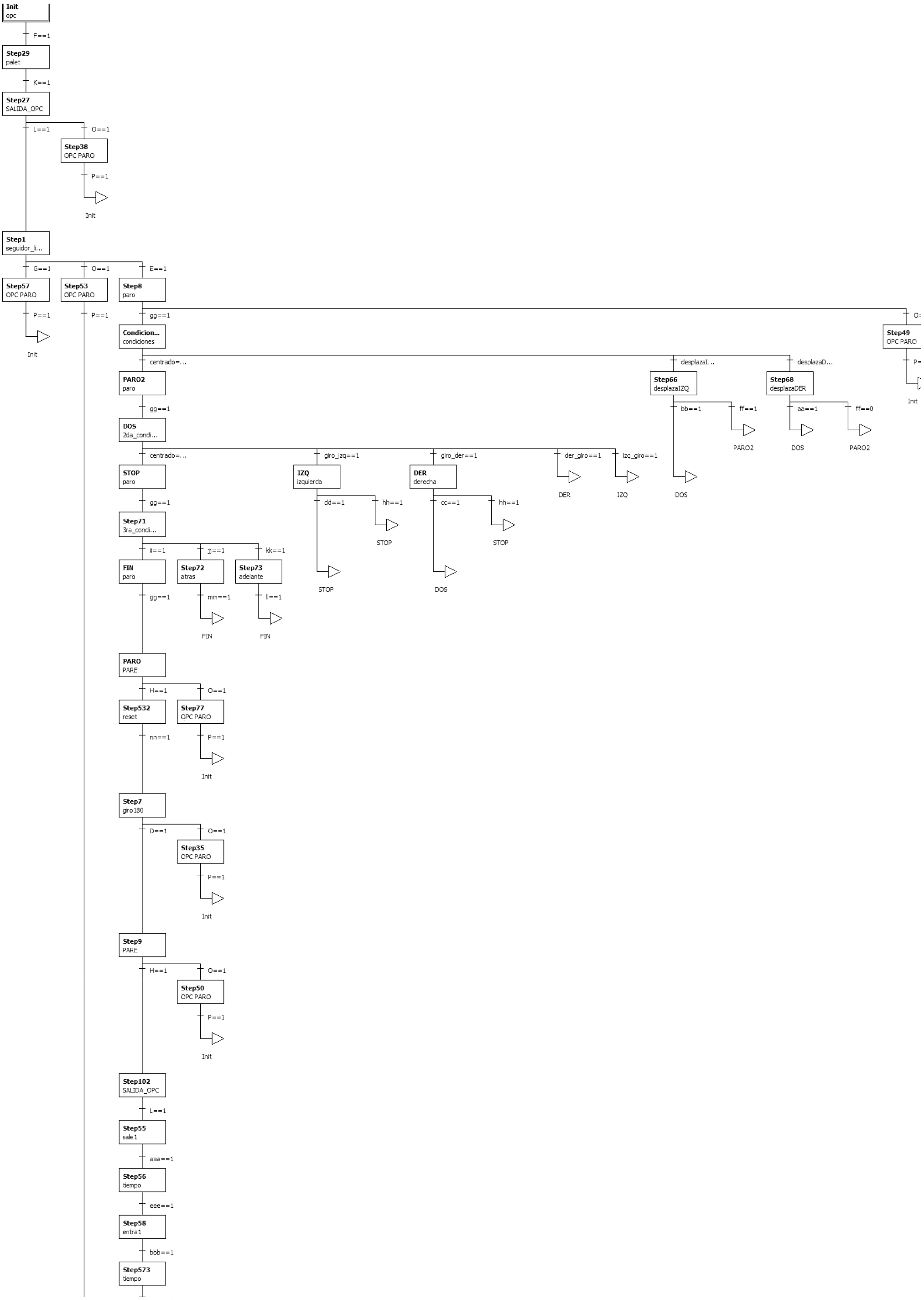
FIGURA 3.10 Diseño de pantalla Proceso de interfaz HMI

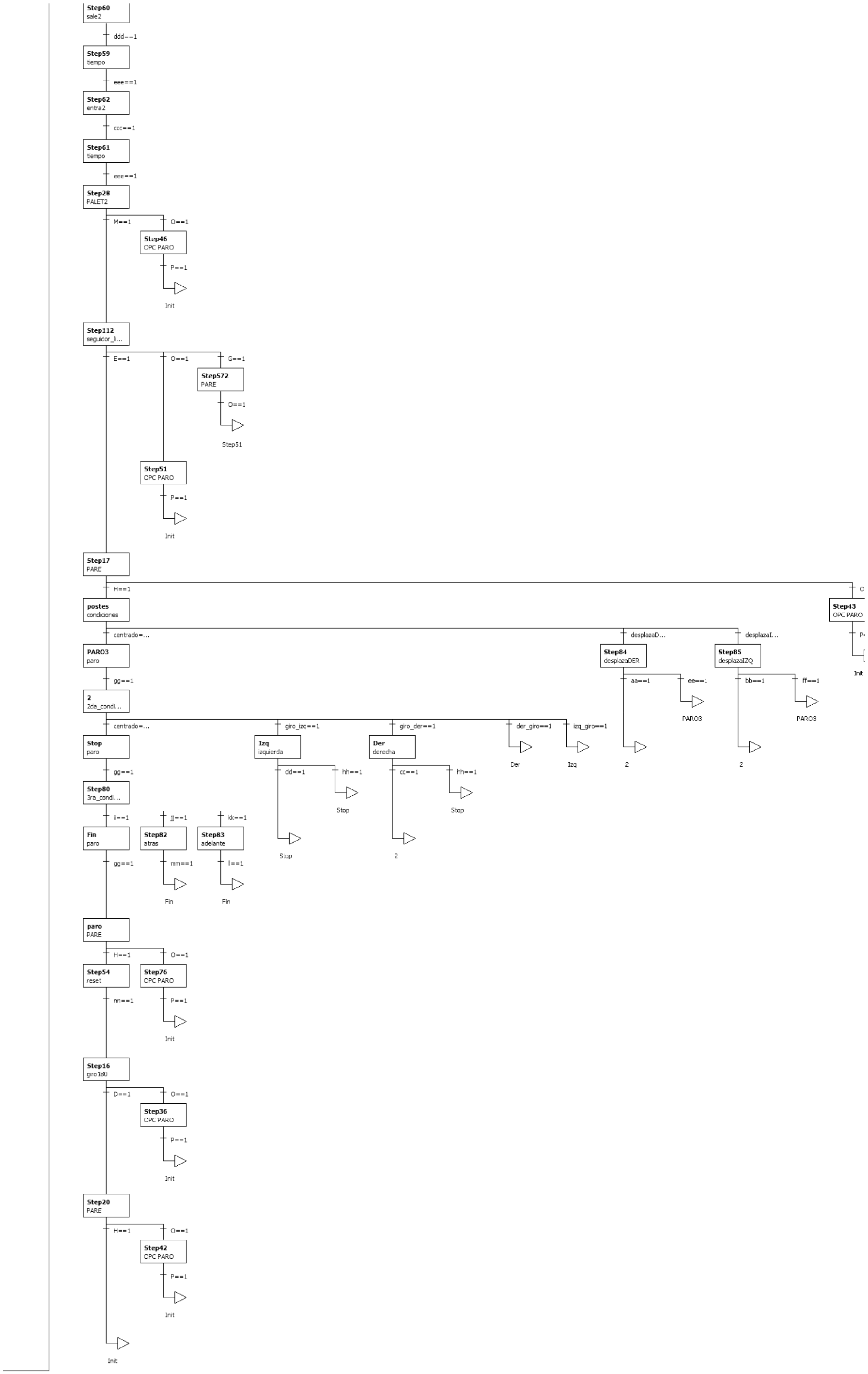
[Fuente: autoras tesis]

3.2 DESARROLLO

3.2.1 Programación del robot móvil Robotino de Festo.

A continuación se muestra el diagrama de flujo que describe el comportamiento del programa principal en Robotino View 2.6.


Init
opc
F==1
Step29
palet
K==1
Step27
SALIDA_OPC
L==1
O==1
Step38
OPC PARO
P==1
Init
Step1
seguidor_li...
G==1
O==1
E==1
Step57
OPC PARO
Step53
OPC PARO
Step8
paro
P==1
P==1
Init
gg==1
Condicion...
condiciones
centrado=...
PARO2
paro
gg==1
DOS
2da_condi...
centrado=...
giro_izq==1
giro_der==1
der_giro==1
izq_giro==1
STOP
paro
IZQ
izquierda
DER
derecha
DER
IZQ
DOS
gg==1
dd==1
hh==1
cc==1
hh==1
Step71
3ra_condi...
STOP
STOP
STOP
DOS
ii==1
jj==1
kk==1
FIN
paro
Step72
atras
Step73
adelante
gg==1
mm==1
ll==1
FIN
FIN
PARO
PARE
H==1
O==1
Step532
reset
Step77
OPC PARO
nn==1
P==1
Init
Step7
giro180
D==1
O==1
Step35
OPC PARO
P==1
Init
Step9
PARE
H==1
O==1
Step50
OPC PARO
P==1
Init
Step102
SALIDA_OPC
L==1
Step55
sale1
aaa==1
Step56
tiempo
eee==1
Step58
entra1
bbb==1
Step573
tiempo
desplazaI...
desplazaD...
Step66
desplazaIZQ
Step68
desplazaDER
bb==1
ff==1
aa==1
ff==0
PARO2
DOS
PARO2
O=
Step49
OPC PARO
P=
Init


Step60
sale2
ddd==1
Step59
tiempo
eee==1
Step62
entra2
ccc==1
Step61
tiempo
eee==1
Step28
PALET2
M==1
O==1
Step46
OPC PARO
P==1
Init
Step112
seguidor_li...
E==1
O==1
G==1
Step572
PARE
O==1
Step51
Step51
OPC PARO
P==1
Init
Step17
PARE
H==1
postes
condiciones
centrado=...
PARO3
paro
gg==1
2
2da_condi...
centrado=...
Stop
paro
gg==1
Step80
3ra_condi...
ii==1
jj==1
kk==1
Fin
paro
Step82
atras
Step83
adelante
gg==1
mm==1
ll==1
Fin
Fin
paro
PARE
H==1
O==1
Step54
reset
Step76
OPC PARO
nn==1
P==1
Init
Step16
giro180
D==1
O==1
Step36
OPC PARO
P==1
Init
Step20
PARE
H==1
O==1
Step42
OPC PARO
P==1
Init
Init
giro_izq==1
Izq
izquierda
dd==1
hh==1
Stop
Stop
giro_der==1
Der
derecha
cc==1
hh==1
Stop
2
der_giro==1
izq_giro==1
Der
Izq
desplazaD...
Step84
desplazaDER
aa==1
ee==1
PARO3
2
desplazaI...
Step85
desplazaIZQ
bb==1
ff==1
PARO3
2
O
Step43
OPC PARO
P
Init

Las subrutinas se detallan a continuación:

OPC-. Espera la señal de inicio del PLC.

PALLET.- Detecta que 2 piezas se coloquen sobre el pallet.

SALIDA_OPC.- Espera señal de Brazo 1 que ha terminado de armar 2 piezas a través del PLC.

OPC PARO.- Señal de paro de todo el sistema, señal enviada por PLC.

SEGUIDOR DE LINEA.- Algoritmo para que Robotino siga una línea negra por medio de la cámara web que posee.

PARO.- Detener a Robotino por cierto tiempo. Motores con velocidad 0.

CONDICIONES.- Aspectos que debe cumplir Robotino para que se ubique de manera centrada entre 3 paredes en cada estación; para lo cual se desplaza hacia la derecha o izquierda (*DesplazaIZQ*y*DesplazaDER*).

2da CONDICION.- Aspectos que debe cumplir Robotino para que se ubique de manera centrada entre 3 paredes en cada estación; para lo cual gira hacia la derecha o izquierda (*DER, IZQ*).

3ra CONDICION.- Aspectos que debe cumplir Robotino para que se ubique de manera centrada entre 3 paredes en cada estación; para lo cual se desplaza hacia la adelante o atrás (*Atrás Adelante*).

RESET.- Reinicia los motores, colocándolos valor 0 en cada encoder.

GIRO 180.- Robotino gira 180º en su propio eje.

SALE 1.-Accionamiento en sentido horario de motor1 de pallet.

TIEMPO.- Activa un reloj de 300ms.

ENTRA 1.-Accionamiento en sentido anti horario de motor1 de pallet.

SALE 2.-Accionamiento en sentido horario de motor2 de pallet.

ENTRA 2.-Accionamiento en sentido anti horario de motor 2 de pallet.

PALLET 2.- Detecta que las 2 piezas del pallet hayan sido expulsadas a la estación de clasificación.

3.2.2 Programación de los brazos robóticos RV-2AJ.

Para configurar los movimientos que va a realizar cada brazo robótico, se debe grabar las posiciones necesarias para lograr el funcionamiento esperado. Para ello se debe realizar los siguientes pasos:

3.2.2.1 Brazo 1 Ensamblaje de Piezas

Para el ensamblaje de piezas del brazo robótico 1 se deben realizar las siguientes tareas:

1. Detectar que el Robot Robotino ha llegado a la estación

2. Detectar que exista pieza en la rampa

3. Recoger pieza de la rampa y depositarla en la posición de identificación

4. Identificar color de la pieza

5. Colocar la pieza en la posición de ensamblaje

6. Recoger pistón negro o plateado según el color de la pieza

7. Colocar Pistón

8. Recoger resorte

9. Colocar resorte

10. Recoger tapa

11. Colocar tapa

12. Depositar la pieza ensamblada en el pallet que lleva el Robotino

Cuando coloca las dos piezas sobre el pallet que lleva el Robotino envía una señal al PLC.

Las posiciones pregrabadas en la controladora de la estación de ensamblaje se muestran a continuación:

[RV-2AJ] F:\21 SEP 2011\no PLC\7.POS

No	Position	Orientation	Comment
P1	253.6,44.1,167.2	-53, 180,R,A	
P2	216.6,197.6,164.8	2, 180,R,A	
P3	185.7,206.2,147.6	0, 180,R,A	
P4	219.9,237.3,148.8	-73, 181,R,A	
P5	-143.1,263.0,146.5	-3, 181,R,A	
P6	134.9,362.4,318.2	87, 181,R,A	
P7	69.8,264.5,147.6	-182, 180,R,A	
P8	193.3,255.8,148.3	-73, 181,R,A	
P9	-6.3,308.9,149.1	-179, 180,R,A	
P99	240.6,7.3,372.2	1,-179,R,A	
P98	14.7,-240.3,372.2	90, 181,R,A	
P12	289.9,197.8,141.0	0, 181,R,A	
P13	65.9,-289.5,229.5	77, 180,R,A	
P14	169.9,-289.5,229.5	77, 180,R,A	

Figura 3.11 Lista de posiciones para la estación de ensamblaje

[FUENTE: Autoras Tesis]

Cada una de las posiciones es importante, ya que son la referencia para direccionar al brazo robótico una trayectoria específica y con esto se consigue ejecutar el programa sin inconvenientes.

P99: Posición inicial y de referencia para el sistema del brazo robótico.

P1: Recogida de pieza.

P2: Posición en módulo de montaje.

P3: Identificación del color de la pieza.

P4: Posición para recoger resorte

P5: Posición para recoger pistón

P9: Posición para recoger tapa

P12: Posición para sensar ubicación correcta de la tapa

P13: Posición 1 depósito de piezas (Pallet)

P14: Posición 2 depósito de piezas (Pallet)

CODIGO

```
'Definición de Posiciones del brazo ensamblador
10 DEF POS VECT30
20 DEF POS VECY50
30 DEF POS VECZ2
50 DEF POS VECZ5
60 DEF POS VECZ30
70 DEF POS VECZ50
80 DEF POS VECZ100
90 DEF POS VECZ200
100 DEF POS VECZINT
110 DEF POS VECGA
120 DEF POS AUXPOS
130 DEF POS VECMINC
150 DEF POS VECCACYL
170 DEF POS VECCACOV
190 DEF POS VECEP1
200 DEF POS VECEP2
210 DEF POS VECCUP
230 DEF POS VECCMPZ
250 DEF POS VECCOZ
270 DEF POS VECCOD
290 DEF POS VECCMPD
310 DEF POS VECCMPPS
330 DEF POS VECCOC
350 DEF POS VECCOM
370 DEF POS VECCTPB
390 DEF POS VECCTPS
410 DEF POS GIRO
420 DEF POS VECZ45
430 VECT30   = (+0.00,+0.00,+0.00,+40.00,+0.00)
440 VECY50   = (+0.00,+50.00,+0.00,+0.00,+0.00)
450 VECZ2    = (+0.00,+0.00,+2.50,+0.00,+0.00)
460 VECZ5    = (+0.00,+0.00,+5.00,+0.00,+0.00)
470 VECZ30   = (+0.00,+0.00,+30.00,+0.00,+0.00)
480 VECZ50   = (+0.00,+0.00,+50.00,+0.00,+0.00)
490 VECZ100  = (+0.00,+0.00,+100.00,+0.00,+0.00)
500 VECZ200  = (+0.00,+0.00,+200.00,+0.00,+0.00)
510 VECCUP   = (+16.25,-30.25,-8.00,+0.00,+0.00)
```

```
530 VECCMPZ  = (+56.25,-76.25,-25.00,+0.00,+0.00)
550 VECCOM   = (+17.00,-76.25,-19.00,+0.00,+0.00)
570 VECCOC   = (-16.25,-76.25,-1.10,+0.00,+0.00)
590 VECCOZ   = (-0.25,+0.00,-3.50,+0.00,+0.00)
610 VECCOD   = (+32.50,+0.00,-21.50,+0.00,+0.00)
630 VECCMPPS = (+56.25,-76.25,-18.00,-180.00,+0.00)
650 VECGA    = (+40.00,+0.00,-2.00,+0.00,+0.00)
670 VECCTPB  = (+55.00,-11.90,-23.00,-180.00,+0.00)
690 VECCTPS  = (-55.00,-12.50,-23.00,+0.00,+0.00)
710 VECZINT  = (+0.00,+0.00,+0.50,+0.00,+0.00)
720 VECMINC  = (+0.00,+0.00,+0.00,+1.00,+0.00)
740 VECCACYL = (+0.00,+0.00,+0.00,-108.00,+0.00)
760 VECCACOV = (+0.00,+0.00,+0.00,-65.00,+0.00)
780 VECEP1   = (+0.00,+100.00,+0.00,+0.00,+0.00)
790 VECEP2   = (+0.00,+100.00,+0.00,+0.00,+0.00)
800 GIRO     = (+0.00,+0.00,+0.00,+12.00,+0.00)
810 VECZ45   = (+0.00,+0.00,-48.00,+0.00,+0.00)



        ´Definición de variables de entrada y salida
830 DEF IO B1_R     = BIT,1      'sensor de color
840 DEF IO B2_R     = BIT,2
850 DEF IO S1_A     = BIT,3      'Botón de inicio panel de control
860 DEF IO S2_A     = BIT,4      'Botón de paro panel de control
870 DEF IO S4_A     = BIT,5      'Botón Reset panel de control
880 DEF IO PART_AV  = BIT,6
890 DEF IO IP_FI    = BIT,15
900 DEF IO A1B1_A   = BIT,8
910 DEF IO A1B2_A   = BIT,9
920 DEF IO A2B1_A   = BIT,10
930 DEF IO A2B2_A   = BIT,11
940 DEF IO A1S1_A   = BIT,12
950 DEF IO B1_A     = BIT,13
960 DEF IO B2_A     = BIT,14
970 DEF IO IP_N_FO  = BIT,15
980 DEF IO H1_A     = BIT,0      'Luz del botón inicio del panel
990 DEF IO H2_A     = BIT,1      'Luz del botón reset del panel
1000 DEF IO H3_A    = BIT,2
1010 DEF IO H4_A    = BIT,3
1020 DEF IO A1Y1_A  = BIT,8
1030 DEF IO A2Y1_A  = BIT,9
1040 DEF IO ROBOTINO = BIT,7     'Bit de entada llegada Robotino
1050 DEF IO PLC_1   = BIT,6          'Bit de salida para PLC
1070 DEF ACT 1,M_IN(4)=0 GOSUB *INTSTOP
1090 DEF INTE CYLTYPE
1100 DEF INTE SLOW
1110 DEF INTE FAST
1120 DEF INTE SUSLOW
1130 DEF INTE B2OLD
1140 DEF INTE B2MESSV
1150 DEF INTE MAXDEGR
1160 DEF INTE COSP
1170 DEF INTE COBP
1180 DEF PLT 2,P5+VECCTPB,P5+VECCTPB+VECEP2,P5+VECCTPB+VECEP2,,5,1,2
1200 DEF PLT 1,P5+VECCTPS,P5+VECCTPS+VECEP1,P5+VECCTPS+VECEP1,,5,1,2
```

```
1210 OVRD 80
1220 SLOW   = 70
1230 SUSLOW = 40
1240 FAST   = 90
1250 COSP = 1
1260 COBP = 1
1270 HOPEN 1
1280 M_OUTW(0) = 0
1290 PLC_1=0
1300 M1=0
1310 ROBOTINO = 0
1320 PLC_1 =0
1330 SPD FAST

        'Programa Principal
1340 *MAINLOOP
1350  GOSUB *PANEL
1360 *MAINL1
1370  GOSUB *WPart
1380  GOSUB *GETCYL
1390  GOSUB *PLACECYL
1400  GOSUB *GETPIST
1410  GOSUB *ASSPIST
1420  GOSUB *GETSPRG
1430  GOSUB *ASSSPRG
1440  GOSUB *GETCOVER
1450  GOSUB *ASSCOVER
1460  GOSUB *DEPRYPRT
1470 GOTO  *MAINL1
1480 END
```

***WPART, subrutina de control para inicialización del programa. Verifica que exista pieza en la rampa para empezar el ensamblaje y además comprueba que la señal de llegada del Robotino este activa, si alguna de estas señales no lo están, el brazo no realiza ninguna acción.**

```
1640 *WPART
1650 IF ROBOTINO = 0 THEN GOTO 1650 ELSE GOTO 1660
1660 DLY 1
1670 IF PART_AV = 0 THEN GOTO 1670
1680 DLY 1
1700 IP_N_FO = 1
1710 RETURN
```

***PANEL, subrutina que controla los botones del panel principal. El botón Reset activa la estación de ensamblaje y además ubica el brazo robótico en la posición inicial P99.**

```
1820 *PANEL
1830 H1_A = 1
1840 *CONT4
1850 IF S1_A =0 then *CONT4
1860 H1_A = 0
1870 H2_A = 1
1880 *CONT3
1890 IF S4_A = 0 THEN *CONT3
1900 H2_A =0
1910 MOV P99, -30
```

```
1920 MOV P99
1930 RETURN
```

***GETCYL, subrutina que recoge el cilindro de la rampa posición P1.**

```
2050 *GETCYL
2060 ACT 1 =1
2070 MOV P1+VECZ50
2080 AUXPOS = P1+VECZ50
2090 MOV AUXPOS
2100 SPD SLOW
2110 MVS AUXPOS - VECZ50
2120 DLY 1.5
2130 HCLOSE 1
2140 PLC_1 =1
2150 DLY 1
2160 MVS AUXPOS
2170 SPD FAST
2180 MOV P1+VECZ50
2190 PLC_1 =0
2200 ACT 1 =0
2210 RETURN
```

***PLACECYL, coloca el cilindro en la posición de identificación y se ubica en la posición P3 para identificar el color de la pieza, además ubica el cilindro en la posición de ensamblaje.**

```
2360 *PLACECYL
2370 ACT 1 =1
2380 AUXPOS = VECCUP+VECGA+VECZ50+P2
2390 MOV AUXPOS + VECZ200
2400 MOV AUXPOS
2410 MVS AUXPOS - VECZ50
2420 DLY 1.5
2430 HOPEN 1
2440 PLC_1 =1
2450 DLY 2.0
2460 PLC_1=0
2470 MVS AUXPOS
2480 AUXPOS = P3 + VECZ50
2490 MOV AUXPOS
2500 MOV P3
2510 DLY 1
2520 CYLTYPE = B1_R
2530 MOV AUXPOS
2540 IF CYLTYPE = 0 THEN goto *BLACK1 ELSE goto *NBLACK1
2550 *BLACK1
2560 AUXPOS = VECZ50-VECZ2+VECCUP+P2
2570 GOTO *CONT1
2580 *NBLACK1
2590 AUXPOS = VECZ50+VECCUP+P2
2600 *CONT1
2610 MOV AUXPOS
2620 SPD SLOW
2630 MVS AUXPOS - VECZ50
2640 DLY 1.5
```

```
2650 HCLOSE 1
2660 DLY 1
2670 MVS AUXPOS
2680 SPD FAST
2690 AUXPOS = AUXPOS - VECCUP + VECCOZ
2700 MOV AUXPOS
2710 SPD SLOW
2720 AUXPOS = AUXPOS - VECZ50
2730 MVS AUXPOS
2740 ACT 1 =0
2750 GOSUB *TESTOR
2760 ACT 1 =1
2770 AUXPOS = AUXPOS+ VECZ50
2780 MVS AUXPOS
2790 AUXPOS= AUXPOS+VECCOM+VECCACYL
2800 MOV AUXPOS
2810 MVS AUXPOS + VECZ45
2820 MVS AUXPOS + VECZ45+GIRO
2830 DLY 1.5
2840 HOPEN 1
2850 PLC_1 =1
2860 MVS AUXPOS
2870 SPD FAST
2880 MOV AUXPOS + VECZ200
2890 PLC_1 =0
2900 ACT 1 =0
2910 RETURN
```

***GETPIST, subrutina que recoge el pistón plateado o negro, según el color de la pieza previamente identificada, tomando como referencia la posición P5.**

```
3040 *GETPIST
3050 ACT 1 = 1
3060 IF CYLTYPE = 0 THEN *BLACK2 ELSE *NBLACK2
3070 *BLACK2
3080 If CoBP< 5 Then *CONT21 else *wpiston
3090 CoBP = 1
3100 *wpiston
3110 H2_A=1
3120 if S4_A =1 then goto 3130 else *wpiston
3130 CoBP=1
3140 goto *cont21
3150 *CONT21
3160 H2_A=0
3170 AUXPOS = PLT 1, CoBP
3180 CoBP = CoBP +1
3190 GOTO *CONT2
3200 *NBLACK2
3210 IF CoSP< 5 Then *CONT22 else *wpiston2
3220 CoSP = 1
3230 *wpiston2
3240 H2_A=1
3250 if S4_A =1 then goto 3260 else *wpiston2
3260 CosP=1
3270 goto *cont22
3280 *CONT22
```

```
3290 H2_A=0
3300 AUXPOS = PLT 2, CoSP
3310 CoSP = CoSP +1
3320 *CONT2
3330 MOV P5 + VecZ100
3340 MVS AuxPos + VecZ50
3350 SPD Slow
3360 MVS AuxPos
3370 DLY 1.5
3380 HCLOSE 1
3390 PLC_1 =1
3400 DLY 1
3410 MVS AuxPos + VecZ50
3420 SPD Fast
3430 MOV AuxPos + VecZ100
3440 MOV P5 + VecZ100
3450 PLC_1 =0
3460 ACT 1 = 0
3470 RETURN
```

***ASSPIST, subrutina que coloca el pistón dentro del cilindro que se encuentra en la posición de ensamblaje P2.**

```
3570 *ASSPIST
3580 ACT 1 =1
3590 SPD FAST
3600 MOV VECZ200+P2
3610 AUXPOS =  VECCMPPS+VECZ50+P2
3620 MOV AUXPOS
3630 IF CYLTYPE = 1 THEN goto *NBLACK3
3640 AUXPOS = AUXPOS-VECZ2
3650 *NBLACK3
3660 MOV AUXPOS
3670 SPD SLOW
3680 MVS AUXPOS - VECZ50
3690 DLY 1.5
3700 HOPEN 1
3710 PLC_1 =1
3720 MVS AUXPOS
3730 SPD FAST
3740 PLC_1 =0
3750 ACT 1 =0
3760 RETURN
```

***GETSPRG, subrutina que activa el pistón neumático del contenedor de resortes, una vez que el brazo identifique que existe dicho resorte lo recoge de la posición P4.**

```
3860 *GETSPRG
3870 ACT 1=1
3880 IF A1S1_A=1 THEN *CONT5
3890 *LOOP1
3900 A1Y1_A=1
3910 *WAIT1
3920 IF A1B1_A=0 then *WAIT1
3930 DLY 1
```

```
3940 A1Y1_A=0
3950 DLY 1
3960 *WAIT2
3970 IF A1B2_A=0 then *WAIT2
3980 IF A1S1_A=1 THEN *CONT5
3990 H3_A = 1
4000 H1_A = 1
4010 *WAIT3
4020 IF S1_A = 0 THEN *WAIT3
4030 H3_A = 0
4040 H1_A = 0
4050 GOTO *LOOP1
4060 *CONT5
4070 AUXPOS = P4 + VECZ50
4080 MOV AUXPOS
4090 SPD SLOW
4100 MVS AUXPOS - VECZ50
4110 DLY 1
4120 HCLOSE 1
4130 PLC_1 =1
4140 DLY 1
4150 MVS AUXPOS
4160 SPD FAST
4170 PLC_1 =0
4180 ACT 1 =0
4190 RETURN
```

***ASSSPRG, subrutina que coloca el resorte dentro del cilindro.**

```
4290 *ASSSPRG
4300 ACT 1 =1
4310 AUXPOS =  VECCMPPS+VECZ50+P2
4320 IF CYLTYPE = 1 THEN goto *NBLACK5
4330 AUXPOS = AUXPOS-VECZ2
4340 *NBLACK5
4350 MOV AUXPOS
4360 SPD SLOW
4370 MVS AUXPOS - VECZ50
4380 DLY 1.5
4390 HOPEN 1
4400 PLC_1 =1
4410 MVS AUXPOS
4420 SPD FAST
4430 AUXPOS = AUXPOS + VECZ100
4440 MOV AUXPOS
4450 PLC_1 =0
4460 ACT 1=0
4470 RETURN
```

***GETCOVER, subrutina que activa el pistón del contenedor de tapas, el brazo robótico lo toma de la posición P9, además se desplaza a la posición P12 para identificar la debida ubicación de la tapa en el cilindro.**

```
4600 *GETCOVER
4610 ACT 1 = 1
```

```
4620 if B1_A =0 then *CONT6
4630 *LOOP2
4640 A2Y1_A = 1
4650 *WAIT4
4660 if A2B1_A=0 then *WAIT4
4670 DLY 1
4680 A2Y1_A = 0
4690 *WAIT5
4700 IF A2B2_A=0 then *WAIT5
4710 IF B1_A = 0 THEN *CONT6
4720 H4_A = 1
4730 H1_A = 1
4740 *WAIT6
4750 IF S1_A = 0 THEN *WAIT6
4760 H4_A = 0
4770 H1_A = 0
4780 GOTO *LOOP2
4790 *CONT6
4800 MOV P9, -200
4810 MOV P9, -50
4820 SPD SLOW
4830 MVS P9
4840 DLY 1.5
4850 HCLOSE 1
4860 PLC_1 =1
4870 DLY 1
4880 MVS P9,-50
4890 SPD FAST
4900 MOV P9 + VECZ200
4910 MOV P9 + VECZ200 - VECY50
4920 MOV P99
4930 MOV P12, -200
4940 MOV P12,-50
4950 SPD SUSLOW
4960 MVS P12
4970 DLY 1.5
4980 HOPEN 1
4990 MVS P12,-50
5000 SPD FAST
5010 ACT 1 =0
5020 RETURN
```

***ASSCOVER, subrutina que coloca la tapa en el cilindro y la asegura dando un giro de 40°.**

```
5160 *ASSCOVER
5170 ACT 1 =1
5180 AUXPOS = VECZ50+VECCOD+P2
5190 MOV AUXPOS
5200 SPD SLOW
5210 AUXPOS =AUXPOS-VECZ50
5220 MVS AUXPOS
5230 DLY 1.5
5240 HCLOSE 1
5250 DLY 1
5260 GOSUB *TESTOR
```

```
5270 PLC_1 =0
5280 MVS AUXPOS+VECZ50
5290 AUXPOS = AUXPOS+VECCOC+VECCACOV+VECZ50
5300 IF CYLTYPE = 1 THEN *NBLACK7
5310 AUXPOS = AUXPOS-VECZ2
5320 *NBLACK7
5330 MOV AUXPOS
5340 SPD SLOW
5350 MVS AUXPOS - VECZ50
5360 SPD SUSLOW
5370 MVS AUXPOS - VECZ50 + VECT30
5380 SPD SLOW
5390 DLY 1.5
5400 HOPEN 1
5410 PLC_1 =1
5420 MVS AUXPOS + VECT30
5430 MVS AUXPOS
5440 SPD FAST
5450 PLC_1 =0
5460 ACT 1 =0
5470 RETURN
```

***DEPRYPRT, subrutina que recoge la pieza terminada de la posición de ensamblaje y la coloca en el pallet que lleva el Robotino, usando como referencia las posiciones P13y P14.**

```
5590 *DEPRYPRT
5600 ACT 1 = 1
5610 *WAIT7
5620 IF IP_FI = 0 THEN *WAIT7
5630 SPD SLOW
5640 AUXPOS = VecCMPZ+VECZ50+P2
5650 MOV AUXPOS
5660 MVS AUXPOS - VECZ50
5670 DLY 1.5
5680 PLC_1 =1
5690 HCLOSE 1
5700 DLY 1
5710 MVS AUXPOS - VECZ50
5720 MVS AUXPOS
5730 SPD FAST
5740 MOV P99
5750 PLC_1 =0
5760 MOV P98
5770 M1=M1+1
5780 SELECT M1
5790  CASE 1
5800 SPD 20
5810 PLC_1 =1
5820 MOV P13,-50
5830 MVS P13
5840 DLY 1.5
5850 HOPEN 1
5860 DLY 1.0
5870 MVS P13,-50
5880 MOV P98
5890 MOV P99
```

```
5900 PLC_1 =0
5910 BREAK
5920  CASE 2
5930 SPD 20
5940 PLC_1=1
5950 MOV P14,-50
5960 MVS P14
5970 DLY 1.5
5980 HOPEN 1
5990 DLY 1.0
6000 MVS P14,-50
6010 MOV P98
6020 MOV P99
6030 PLC_1=0
6040 M1=0
6050  BREAK
6060    END SELECT
6070 IP_N_FO = 0
6080 ACT 1 = 0
6090 RETURN
```

***TESTOR, subrutina que detecta alguna anomalía del programa y activa el paro de emergencia de todo el sistema en caso de existir un error.**

```
6230 *TESTOR
6240 ACT 1 =1
6250 B2OLD = 0
6260 MAXDEGR = 0
6270 DLY 0.5
6280 *NFOUND9
6290 B2MESSV = B2_R
6300 IF B2MESSV = 1 AND MAXDEGR = 0 THEN goto *LABEL9
6310 IF B2MESSV = 1 AND B2OLD = 0 THEN goto *FOUND9
6320 *LABEL9
6330 AUXPOS = AUXPOS + VECMINC
6340 MVS AUXPOS
6350 B2OLD = B2MESSV
6360 IF MAXDEGR > 140 THEN goto *ERRL
6370 MAXDEGR = MAXDEGR +1
6380 GOTO *NFOUND9
6390 *FOUND9
6400 ACT 1 =0
6410 RETURN
6420 *INTSTOP
6430 H1_A =1
6440 *WAIT8
6450 IF S1_A=0 THEN *WAIT8
6460 H1_A=0
6470 RETURN 0
6480 *ErrL
6490 HLT
```

3.2.2.2 Brazo2 Clasificación de Piezas

Para la clasificación de piezas del brazo robótico 2 se debe realizar las siguientes tareas:

1. Detectar que el Robotino ha llegado con las piezas a ser clasificadas

2. Tomar la pieza del Pallet en las posiciones P06 y PA respectivamente.

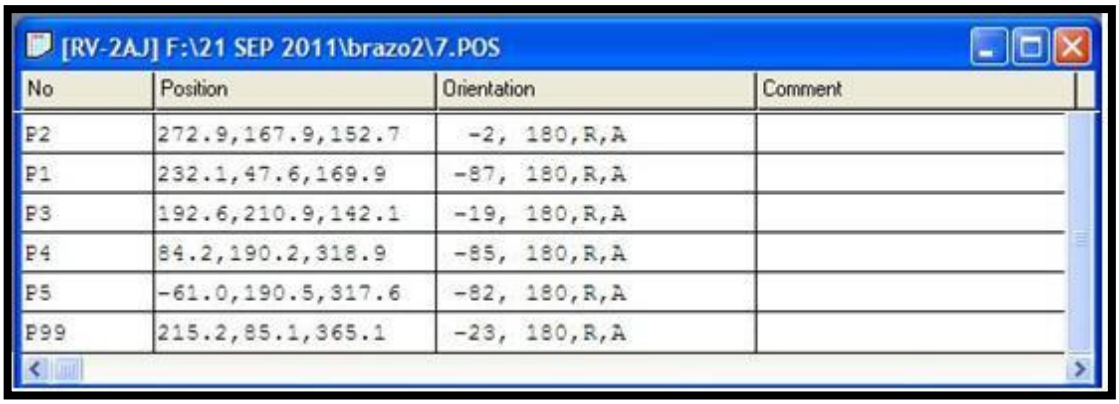
3. Identificar piezas plásticas de piezas cromadas mediante el sensor 9.

4. Las piezas plásticas deben ser colocadas en el contenedor #1.

5. Las piezas cromadas deben colocarse en el contenedor #2.

Cuando los contenedores estén llenos, el robot se detiene y envía una señal al PLC.

Las posiciones pregrabadas en la controladora de la estación de clasificación se muestras a continuación:



No	Position	Orientation	Comment
P2	272.9,167.9,152.7	-2, 180,R,A	
P1	232.1,47.6,169.9	-87, 180,R,A	
P3	192.6,210.9,142.1	-19, 180,R,A	
P4	84.2,190.2,318.9	-85, 180,R,A	
P5	-61.0,190.5,317.6	-82, 180,R,A	
P99	215.2,85.1,365.1	-23, 180,R,A	

FIGURA 3.12 Lista de posiciones para estación de clasificación

[Fuente: Autoras Tesis]

Cada una de las posiciones es importante, debido a que son la referencia para direccionar al brazo robótico una trayectoria específica y consiga ejecutar el programa sin inconvenientes.

P99: Posición inicial y de referencia para el sistema del brazo robótico.

P1: Recogida de pieza.

P2: Posición en módulo de montaje.

P3: Identificación del color de la pieza.

P4: Depósito de piezas rojas.

P5:Depósito de piezas negras.

CÓDIGO:

```
   *DECLARACIÓN DE VARIABLES:
10DEFINTE M1
20DEF IO B1 = BIT,9
30DEF IO ROBOTINO = BIT,11
40DEF IO PLC = BIT,14
50DEF IO INTSTOP = BIT,12
60DEF IO S1_A = BIT,13
70DEF ACT 1,M_IN(12)=0GOSUB *INTSTOP1
80DEF IO PART_AV=BIT,8
90 M1%=0
   *PROGRAMA PRINCIPAL
100 *MAIN
110GOSUB *INICIO
120GOTO *MAIN
130END
 *SUBRUTINA DE INICIO. El brazo empieza su funcionamiento cuando la señal de Marcha este activada, cuando Robotino esté en la posición adecuada y el sensor PART_AV este activado.
140 *INICIO
150IF S1_A =1THENGOTO160ELSEGOTO150
160MOV P99
170IF ROBOTINO=1THENGOTO180ELSEGOTO170
180IF PART_AV = 1THENGOSUB *DETECTA ELSEGOTO180
190RETURN
```

***SUBRUTINA DE DETECCIÓN. El brazo recoge las piezas de la rampa de la estación y las ubica en el módulo de montaje.**

```
200 *DETECTA
210 ACT 1  =1
220MOV P99
230HOPEN1
240 M1%=M1%+1
250 SELECT M1%
260  CASE 1
270MOV P99
280SPD40
290MOV P1,-50
300MVS P1
310DLY1.0
320 PLC=1
330HCLOSE1
340DLY1.0
350 PLC=0
360MVS P1,-50
370MOV P99
380MOV P2,-50
390SPD40
400MVS P2
410DLY1
420 PLC=1
430HOPEN1
440MVS P2,-50
450DLY1
460 PLC=0
470MVS P3, -50
480SPD40
490MOV P3
500DLY1.0
510 ACT 1 =0
520GOSUB *TIPO
530 ACT 1 =1
540 BREAK
550  CASE 2
560MOV P99
570SPD40
580MOV P1,-50
590MVS P1
600DLY1.0
610 PLC=1
620HCLOSE1
630MVS P1,-50
640DLY1.0
650 PLC=0
660MOV P99
670MOV P2,-50
680SPD40
690MVS P2
700DLY1
710 PLC=1
720HOPEN1
730MVS P2,-50
```

```
740DLY1
750 PLC=0
760MVS P3, -50
770SPD40
780MOV P3
790DLY1.0
800 ACT 1 =0
810 M1%=0
820GOSUB *TIPO
830 ACT 1 =1
840 BREAK
850END SELECT
860 ACT 1 =0
870RETURN
   *SUBRUTINA TIPO. Identifica piezas rojas de piezas negras a través del sensor B1.
880 *TIPO
890IF B1=1THEN *ROJO ELSE *NEGRO
900GOSUB *ROJO
910GOSUB *NEGRO
920RETURN
     *SUBRUTINA PIEZAS ROJAS. El robot recoge la pieza del módulo de montaje y la coloca en el contenedor 1.
930 *ROJO ' ROJO O CROMADO
940 ACT 1 =1
950MVS P3,-50
960DLY1.0
970MVS P2, -50
980SPD20
990MVS P2
1000DLY1
1010 PLC=1
1020HCLOSE1
1030MVS P2,-50
1040DLY0.5
1050 PLC=0
1060MOV P99
1070SPD40
1080MOV P5,-30
1090MVS P5
1100 PLC=1
1110 PLC=0
1120HOPEN1
1130DLY1.0
1140MVS P5,-30
1150 PLC=1
1160MOV P99
1170 PLC=0
1180DLY1
1190 ACT 1 =0
1200GOSUB *INICIO
1210RETURN
   *SUBRUTINA PIEZAS NEGRAS. El robot recoge la pieza del módulo de montaje y la coloca en el contenedor 2.
1220 *NEGRO
1230 ACT 1 =1
1240MVS P3,-50
```

```
1250DLY1.0
1260MVS P2, -50
1270SPD40
1280MVS P2
1290DLY1
1300 PLC=1
1310HCLOSE1
1320MVS P2,-50
1330DLY1.0
1340 PLC=0
1350MOV P99
1360SPD40
1370MOV P4,-30
1380MVS P4
1390 PLC=1
1400 PLC=0
1410HOPEN1
1420MVS P4,-30
1430 PLC=1
1440DLY1.0
1450MOV P99
1460 PLC=0
1470DLY1
1480 ACT 1 =0
1490GOSUB *INICIO
1500RETURN
1510 *INTSTOP1
1520DLY5.0
1530GOSUB *INICIO
1540RETURN0
```

3.2.3 Programación del PLC.

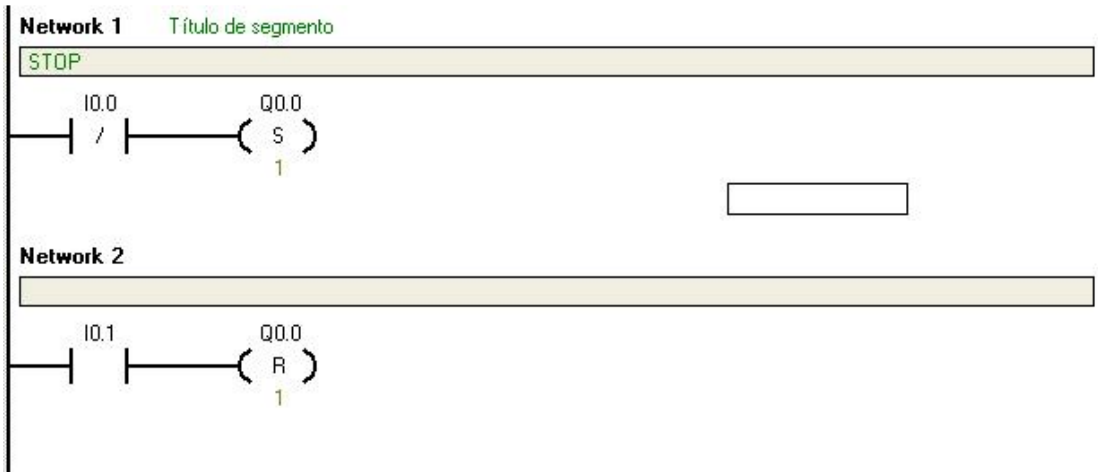
El Controlador Lógico Programable que se utilizó para el control y monitoreo del sistema es S7-200 de Siemens, con CPU 222, AC/DC/Relé; es decir, la alimentación para el dispositivo es de corriente alterna, entradas con alimentación de corriente continua y salidas de tipo relé.

Para la programación se utiliza el software Simatic Step-7 MicroWIN V4.0, el cual permite ingresar el código en forma de escalera (programación lader), allí se especificará la funcionalidad de las entradas y salidas del PLC.

PLC es quien comanda las señales de inicio y de paro de emergencia para cada sistema. Los botones para esta actividad, se encuentran en el panel frontal de la estación de ensamblaje.

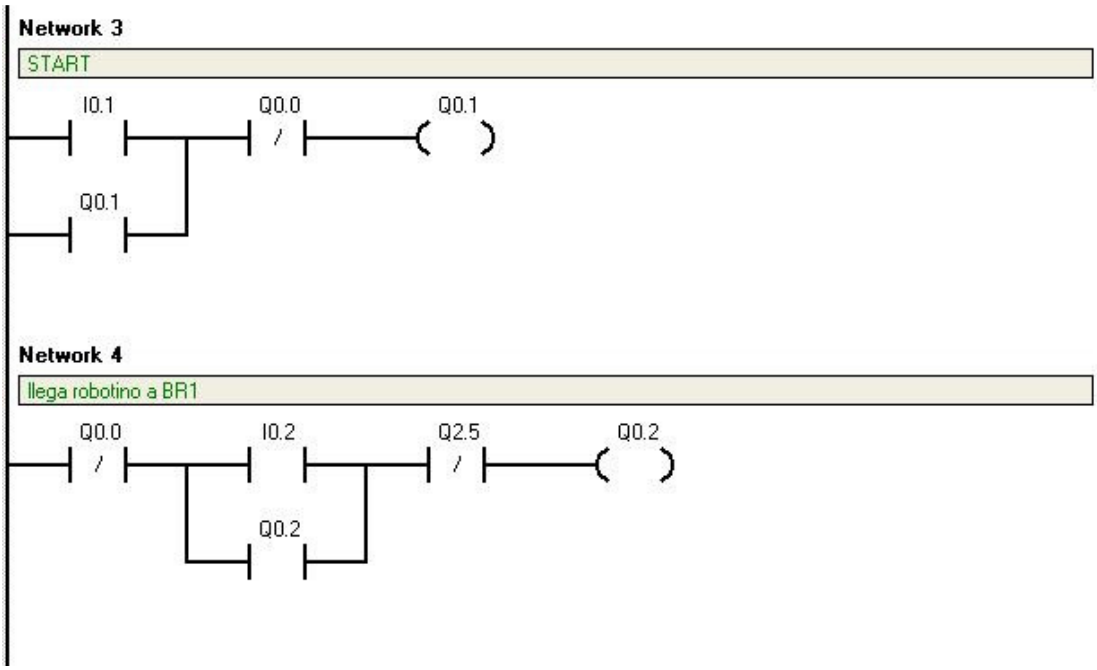
En el Network 1 y 2, se especifica la señal de Paro de emergencia. Físicamente se tiene un pulsador normalmente cerrado (NC), al cual se asigna la entrada I0.0 y activará a la salida Q0.0. La señal de paro actúa de forma dependiente con la señal de Inicio, debido a que, sí el sistema se encuentra en marcha, entonces el paro se encontrara desactivado, y forma contraria cuando el paro se active.

Una vez accionado el paro de emergencia detendrá todos los procesos, sin importar en qué etapa del funcionamiento se encuentre el sistema. Cuando esto suceda, se debe reiniciar a cada uno de los procesos y poner al sistema en estado inicial, luego de lo mencionado será posible poner en marcha nuevamente el sistema.



La señal de Inicio está asignada a la entrada I0.1 y a la salida Q0.1. Todos los procesos deben estar conectados a Q0.0 y a Q0.1, para poner en marcha o detener al sistema respectivamente.

A partir del Network 4, las condiciones dependerán de la señal de paro de emergencia, para de esta manera interrumpir el funcionamiento y resetear el sistema.



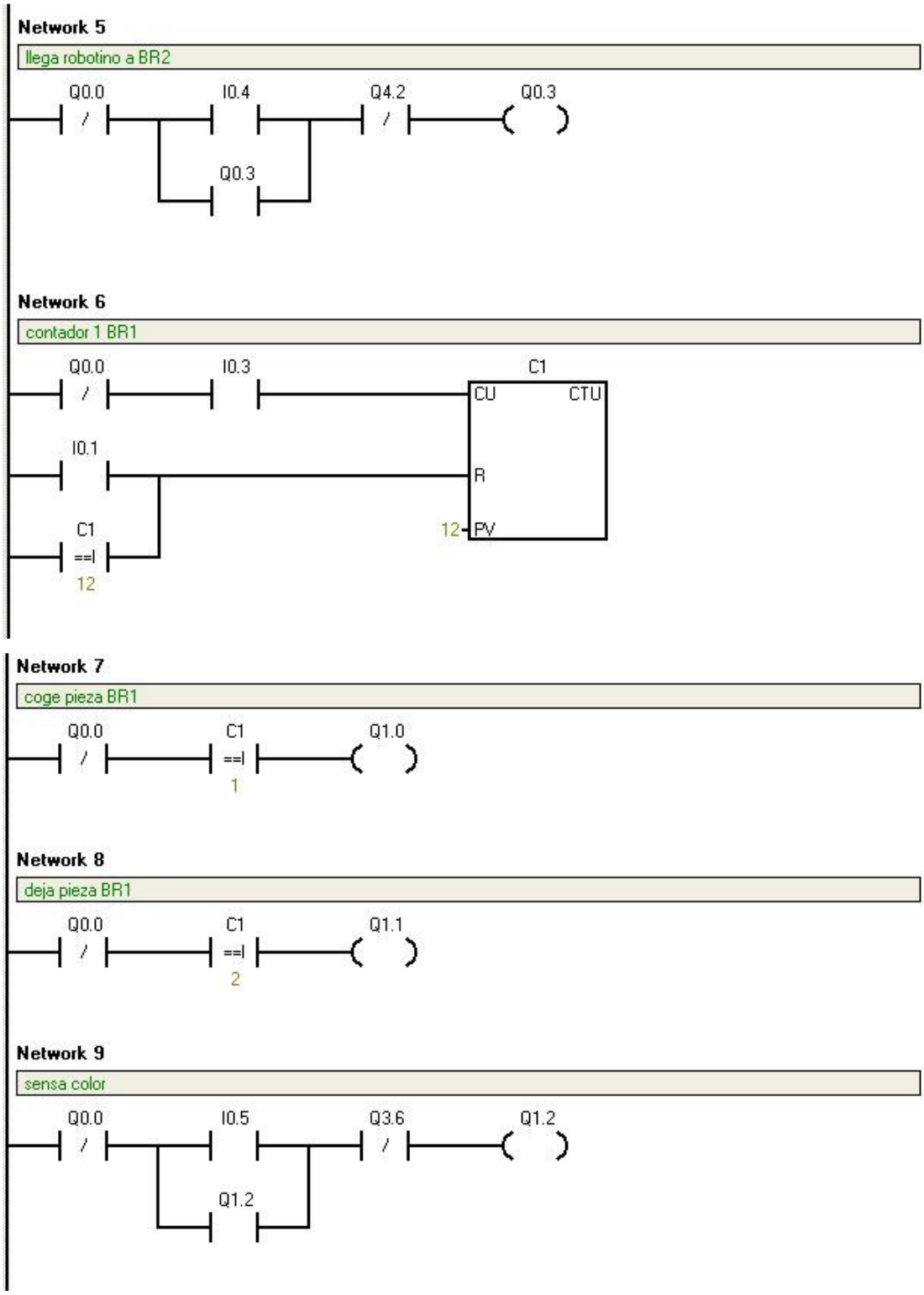
La entrada I0.2 se activa una vez que Robotino se encuentre en la estación de ensamblaje, esta señal es enviada desde RobotinoView 2.6 a través de la comunicación OPC entre Robotino y PLC. De manera similar la entrada I0.4 del Network 5 se enciende cuando Robotino esté en la estación de clasificación.

El contador C1 se activa mediante la entrada I0.3, esta señal es enviada desde la controladora del Brazo 1 (Estación de ensamblaje), la cual se activa cada vez que haya terminado una subrutina. C1 está dimensionado para que cuente hasta 12 activaciones de I0.3; permitiendo así la visualización de las actividades de la Estación de Ensamblaje, en la interfaz HMI, por lo que las salidas asociadas a C1 no son salidas físicas.

De manera similar sucede con los contadores C2, C3, C4, C5 y C6; donde, C2 informa que se han terminado de armar 2 piezas en la primera estación. C3es el contador de las subrutinas de la controladora del Brazo 2 y es activado por I0.6. C4 y C5 son contadores de las piezas rojas y negras que ingresan en los contenedores de la estación de clasificación respectivamente. Por último, C6 indica que se han clasificado 2 piezas.

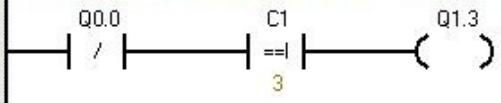
Por otro lado, las entradas I0.5 e I0.7 son las señales del sensor óptico ubicado en la pinza de cada brazo, para determinar el color de la pieza ya sea para

armarla o para clasificarla. Donde, I0.5 corresponde al sensor de la estación de ensamblaje, mientras que I0.7 a la estación de clasificación.



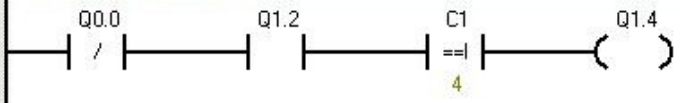
Network 10

COLOCA PIEZA EN POSICION DE ENSAMBLAJE



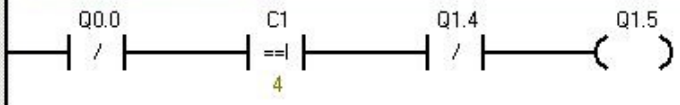
Network 11

coge Piston de pieza roja



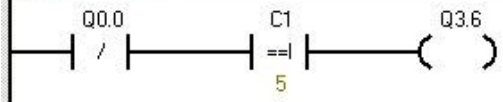
Network 12

COGE piston de pieza negra



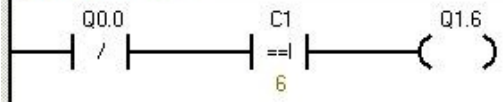
Network 13

DEJA PISTON



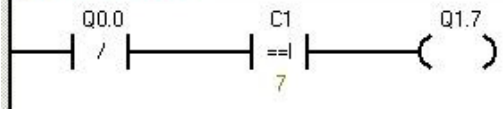
Network 14

COGE RESORTE



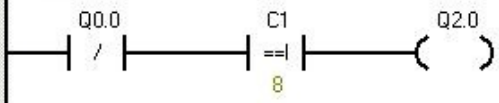
Network 15

COLOCA RESORTE



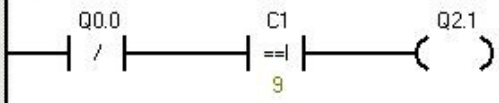
Network 16

COGE TAPA



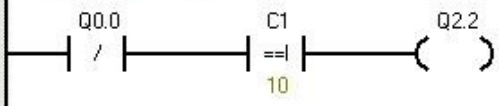
Network 17

DEJA TAPA



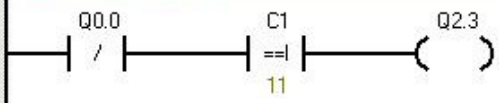
Network 18

COGE PIEZA ARMADA

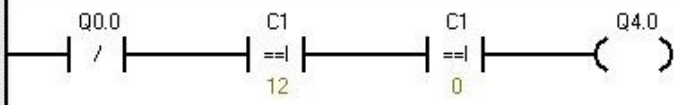


Network 19

COLOCA PIEZA PALET

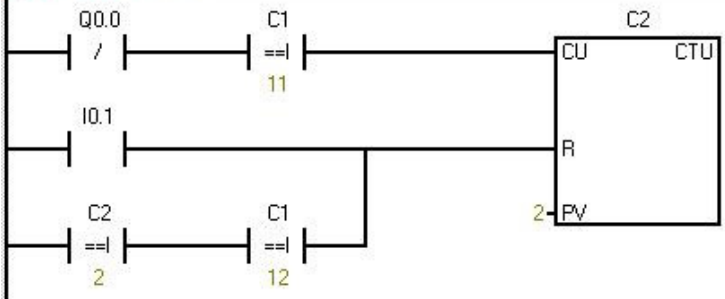


Network 20



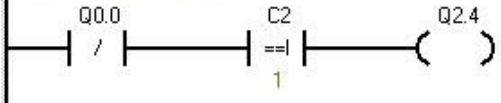
Network 21

INICIA contador 2



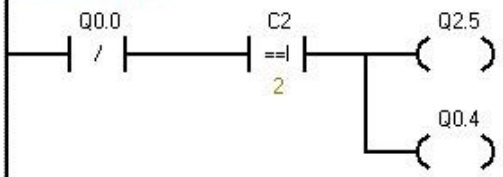
Network 22

TERMINA 1ERA PIEZA



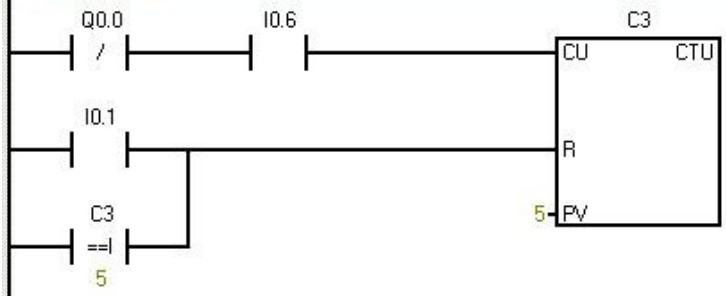
Network 23

termina 2da pieza



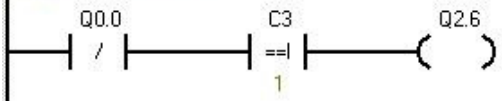
Network 24

contador1 BR2



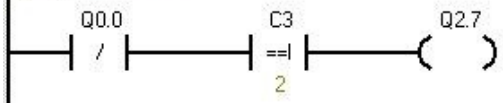
Network 25

coge pieza armada



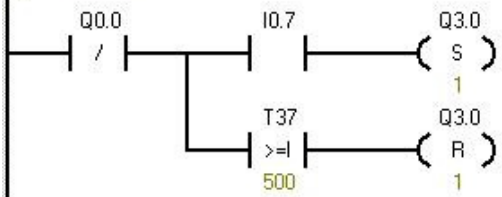
Network 26

deja pieza armada



Network 27

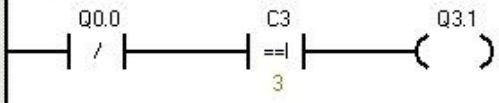
sensa color



Network 28

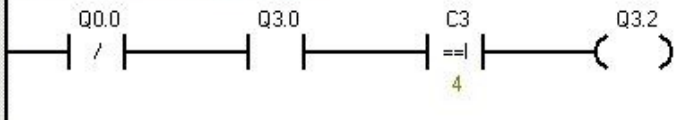
Network 29

COGE PIEZA



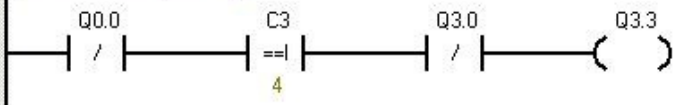
Network 30

DEJA contenedor 1= pieza roja

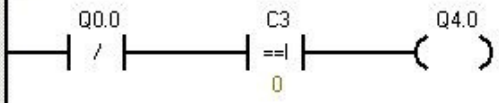


Network 31

contenedor 2= pieza negra

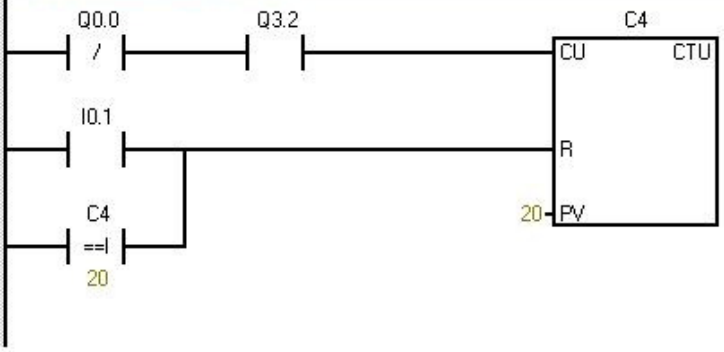


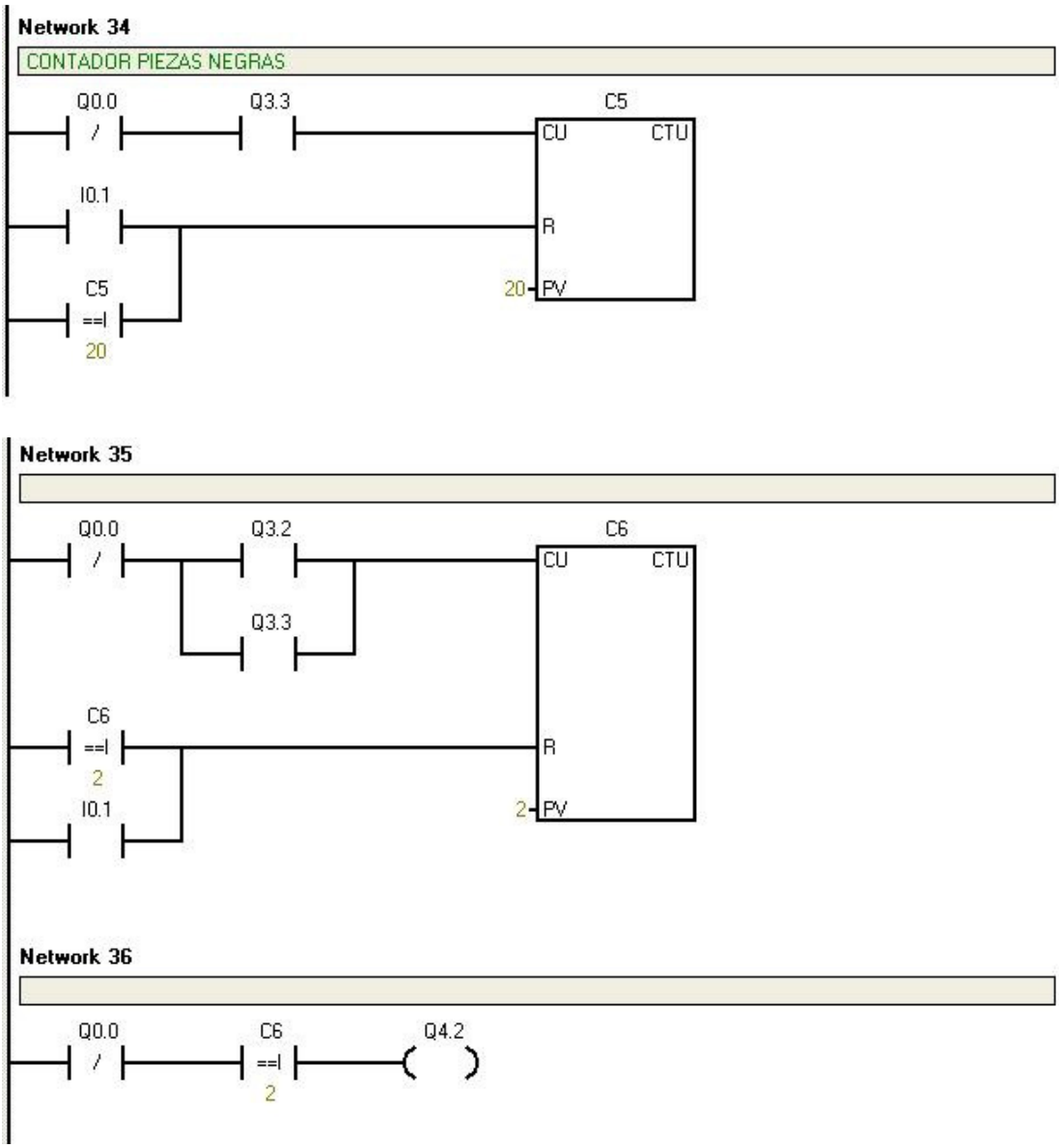
Network 32



Network 33

Inicia contador PIEZAS ROJAS





3.3 IMPLEMENTACIÓN

3.3.1 Implementación de los circuitos electrónicos.

Implementación de circuito de acondicionamiento para sensores y motores.

Una vez diseñado simulado y comprobado el circuito de acondicionamiento, se procede a realizar las pistas para el circuito final a ser implementado.

A continuación se muestra las pistas realizadas con la herramienta Proteus 7 Professional:

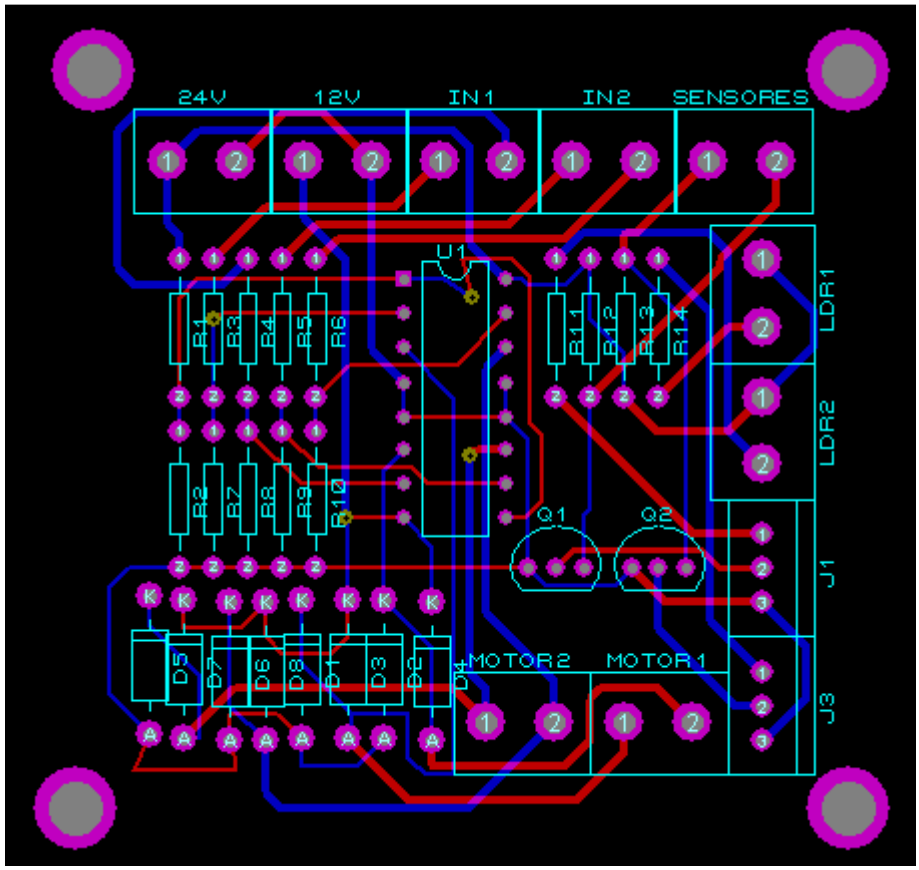


FIGURA 3.13 Ruteado de pistas de circuito de acondicionamiento de sensores y motores en Proteus 7 Professional

[Fuente: Autoras tesis]

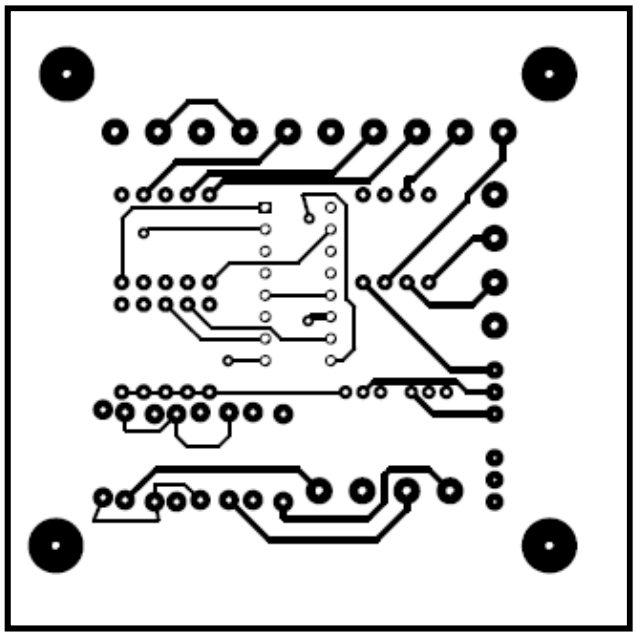


FIGURA 3.14 Ruteado de pistas de circuito de lado 1.

[Fuente: Autoras tesis]

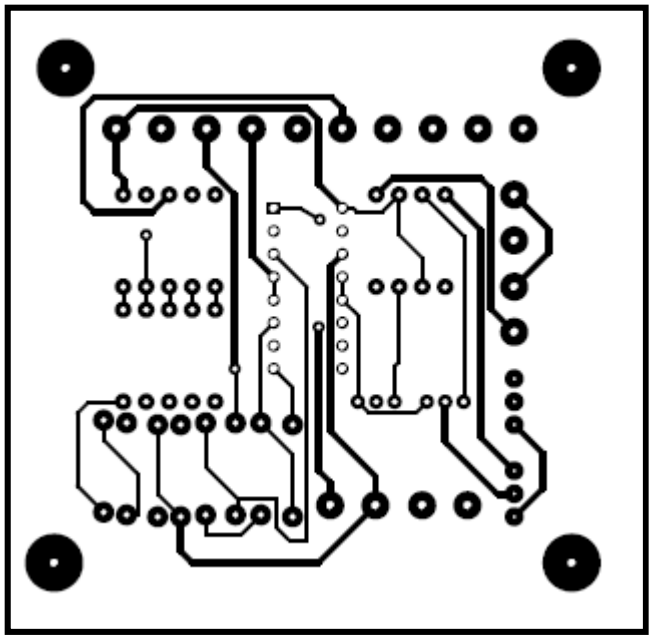


FIGURA 3.15 Ruteado de pistas de circuito de lado 2.

[Fuente: Autoras tesis]

Finalmente el circuito impreso se colocará en el pallet que lleva el robot móvil para transportar las piezas de una estación a otra, lo cual se muestra a continuación:

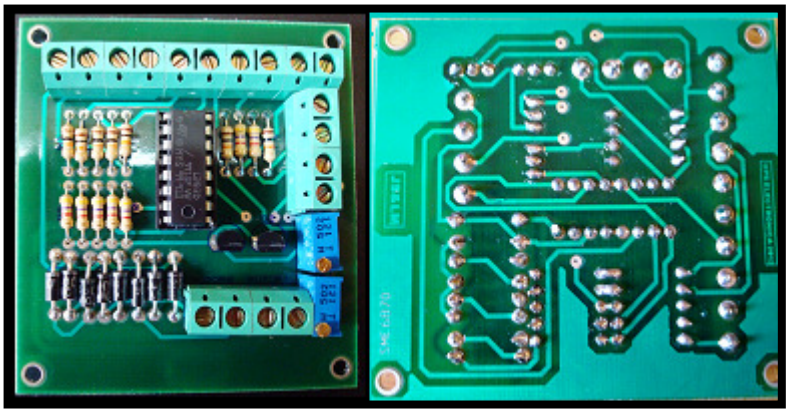


FIGURA 3.16 Pistas impresas y circuito final de acondicionamiento de sensores y motores del Pallet

[Fuente: Autoras Tesis]

3.3.2 Implementación de la comunicación OPC

La comunicación OPC requiere de un servidor y de al menos un cliente para el envío y recepción de datos, para lo cual se tiene al software EzOPC como servidor y a Robotino View como cliente, por otro lado, para la conexión física se utiliza un EasyPort de Festo y un entrenador EduTrainer de entradas y salidas digitales, las cuales serán conectadas directamente al PLC S7200 CPU224 consiguiendo así información entre Robotino y PLC.

La implementación de la comunicación OPC se detalla en los siguientes pasos:

1. Encender el computador y verificar que se disponga de los siguientes programas:

    - EasyPort*DemoAplication* (Elemento de control ActiveX)

    - *EzOPC*V5.3

- *S7-PLCSIM* y

- *Robotino View2*.

2. Conectar físicamente el EasyPort con al entrenador de entradas y salidas a través de uno de los puertos de EasyPort por medio de un cable de conexión adecuado.

3. EnchufaraEasyPort a 24Vdc.

4. Utilizando un cable USB realizar la conexión entre EasyPort y el computador.

5. Abrir el programa *EasyPort Demo Aplication* que se encuentra dentro de la carpeta FestoDidactic.

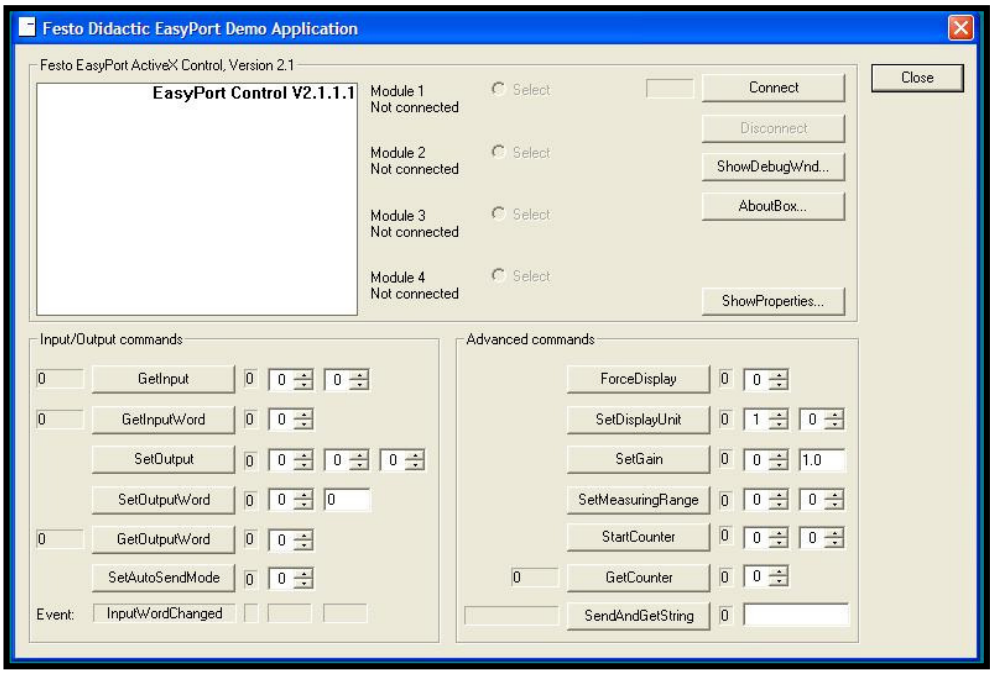


FIGURA 3.17 Ventana de programa EasyPort Demo Aplication

[Fuente: Autoras Tesis]

6. Este software permite reconocer al EasyPort como dispositivo dentro del computador. Se debe hacer clic en *Connect* para que el computador le designe un puerto. Para este caso ha sido otorgado el puerto COM 4 del módulo EasyPort 3. Además de conseguir la conectividad es posible enviar y escribir datos en el EasyPort, pero sin un protocolo OPC; por lo que para esta aplicación se debe cerrar esta ventana.

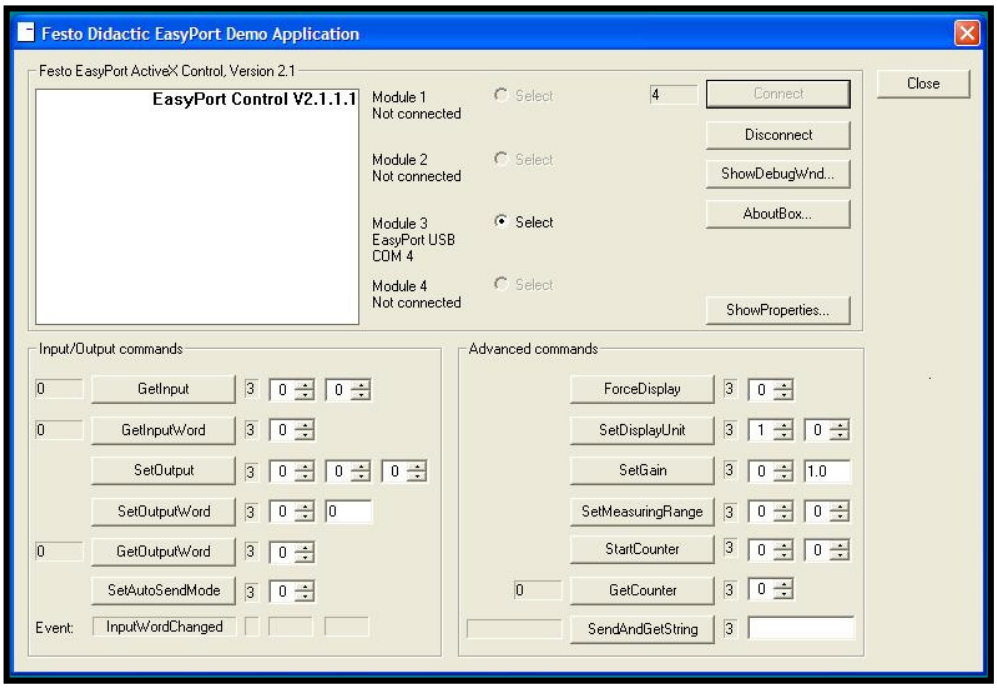


FIGURA 3.18 EasyPort conectado y enlazado al Módulo 3 del COM4

[Fuente: Autoras Tesis]

7. Ahora es necesario abrir EzOPC v5.3. Este es el software que funcionará como servidor en la comunicación. En la pestaña *Overview* se elige el modo en que se va a realizar el enlace OPC. Para el caso se elige: *ProcessviaEasyPort, Virtual Controller*y *S7-PLCSIM controller*. En esta ventana se puede apreciar el estado de la comunicación, se tienen: 0 clientes conectados, un EasyPort conectado, S7-PLCSIM conectando y el controlador virtual está inicializando, se encuentra en este modo debido a que S7-PLCSIM aún no está conectado, para lo cual se realiza el siguiente paso.

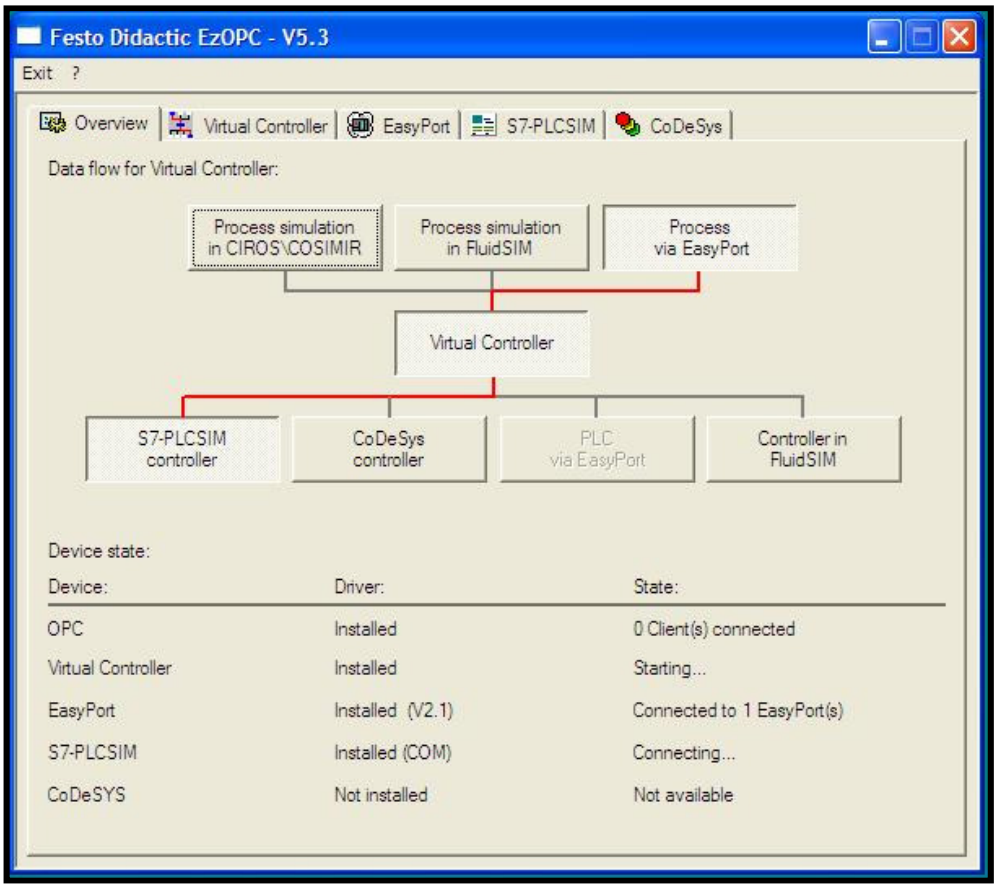


FIGURA 3.19 Ventana de EzOPC v5.3

[Fuente: Autoras Tesis]

8. Abrir el simulador de módulos S7-PLCSIM que se encuentra en: *Simatic, Step 7, S7-PLCSIM*.

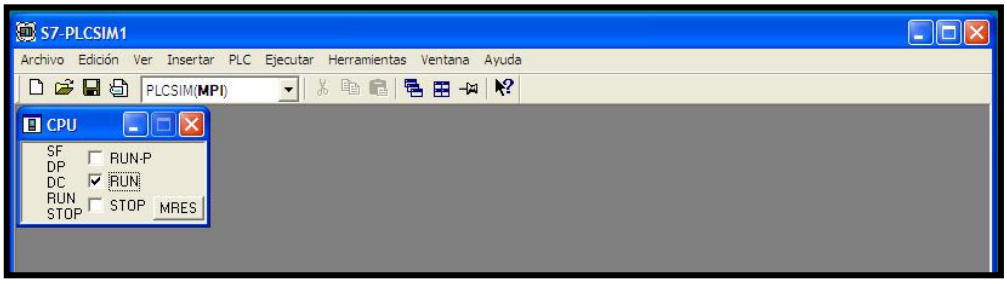


FIGURA 3.20 Ventana del simulador S7-PLCSIM

[Fuente: Autoras Tesis]

9. Ahora se ha logrado la comunicación total en el controlador virtual o servidor.

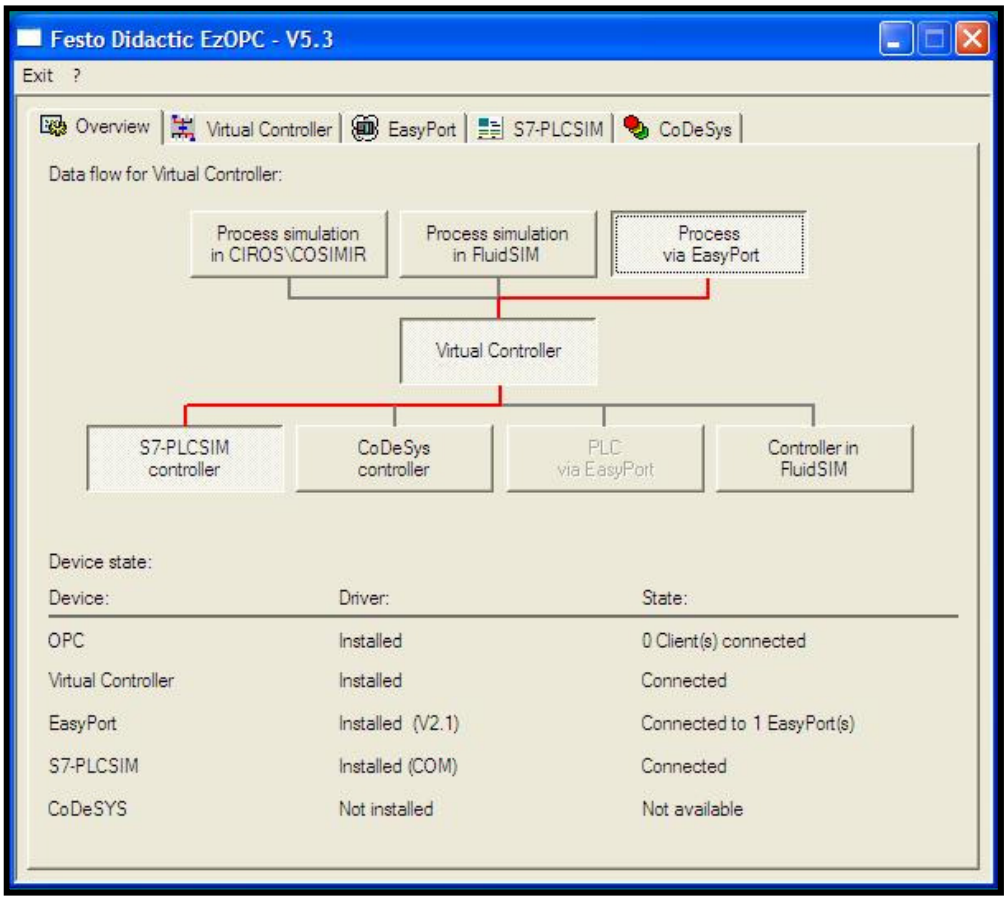


FIGURA 3.21 Virtual Controller, EasyPort y S7-PLCSIM conectados.

[Fuente: Autoras Tesis]

10. Se puede observar que el EasyPort ha sido reconocido por el servidor y está listo para inicializar la comunicación con un cliente.

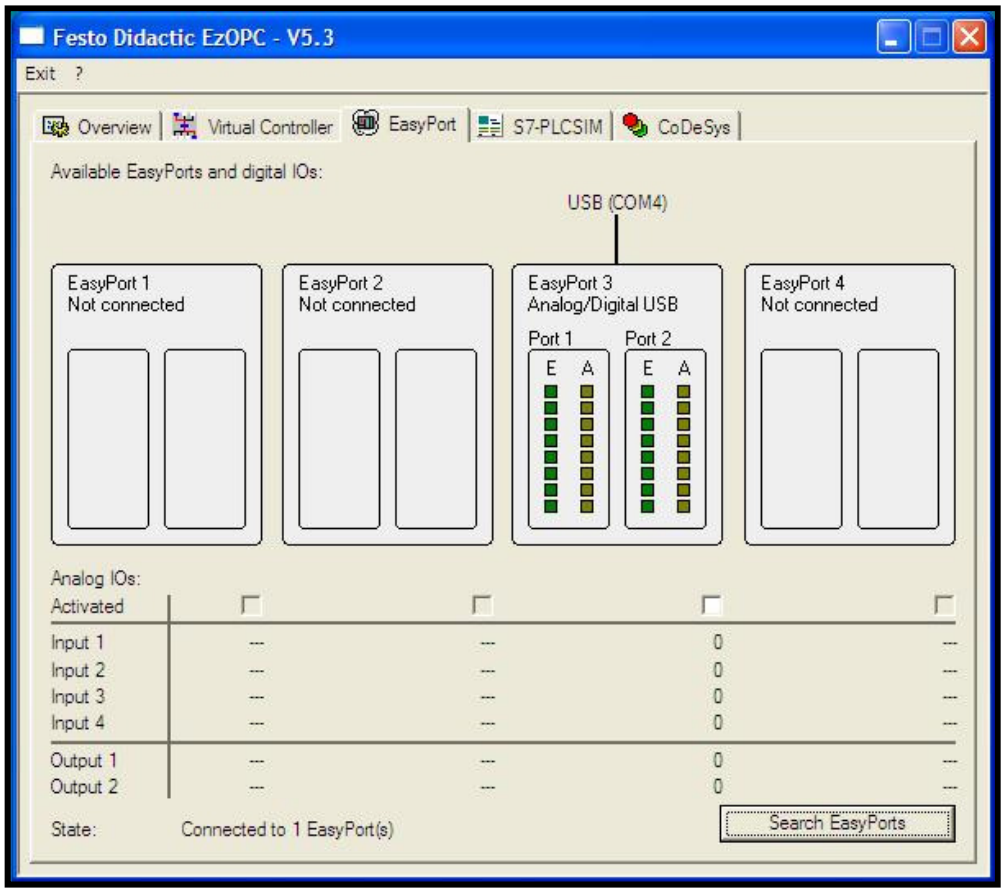


FIGURA 3.22 EasyPortenlazadocon el servidor

[Fuente: Autoras Tesis]

11. Abrir el software Robotino View2 quien será el cliente dentro de la comunicación. Ubicarse en la Librería de bloques de Función y hacer doble clic sobre *OPC Client*.

FIGURA 3.23 Menú de RobotinoView para librería OPC Cliente.

[Fuente: Autoras Tesis]

12. Aquí aparecerá una ventana para configurar al cliente. Dentro de la columna *Tag* se encuentra el EasyPort, es importante tomar en cuenta el puerto que ha sido asignado, para el caso: EasyPort3. En la columna *Valor* se podrá observar el estado de las entradas y salidas conectadas a la comunicación OPC. En la parte superior de la ventana se tienen 2 botones, *Escaneo* que permite la búsqueda del servidor y *Conectar* el cual sirve para agregar el cliente a la comunicación OPC.

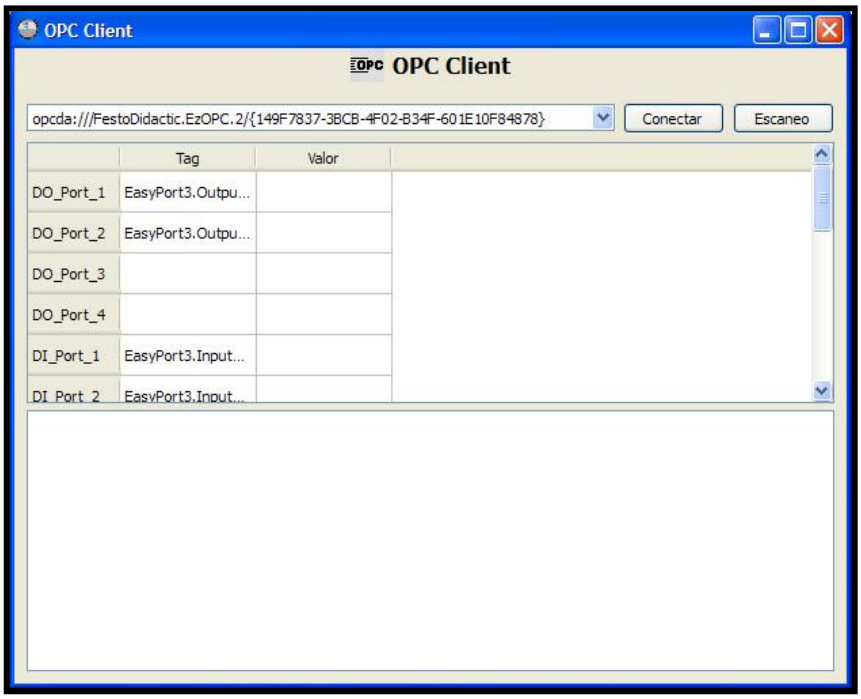


FIGURA 3.24 Ventana de OPC Cliente de RobotinoView.

[Fuente: Autoras Tesis]

13. Hacer clic en *Conectar.*

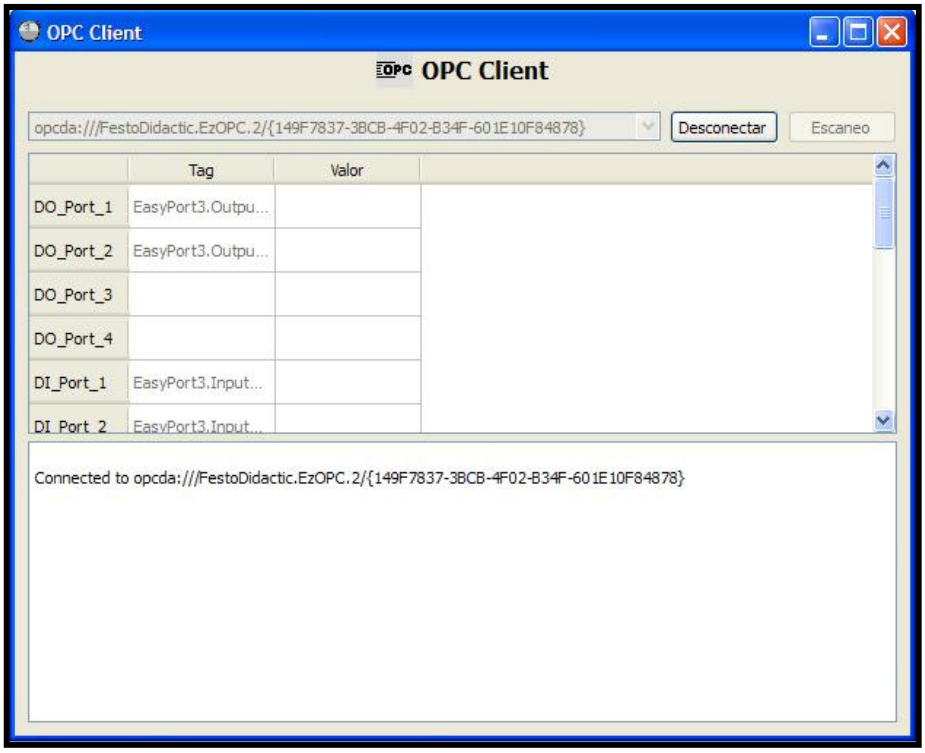


FIGURA 3.25 OPC Cliente conectado

[Fuente: Autoras Tesis]

14. Ahora se enviará una señal a la salida del EasyPort por medio de RobotinoView2. Se puede apreciar que en la columna *Valor* el estado de la salida digita DO_Port_1 es 1.

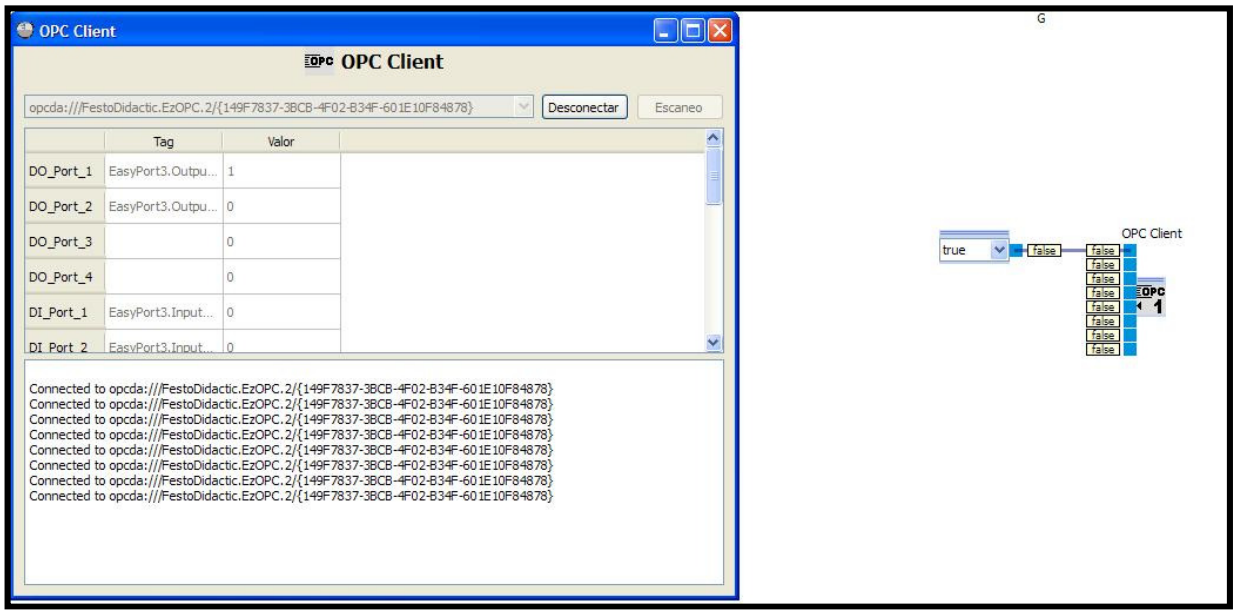


FIGURA 3.26 Verificación de conectividad en RobotinoView.

[Fuente: Autoras Tesis]

15. En EzOPC se también se muestra que la salida digital Nº 1 del puerto 1 está encendida.

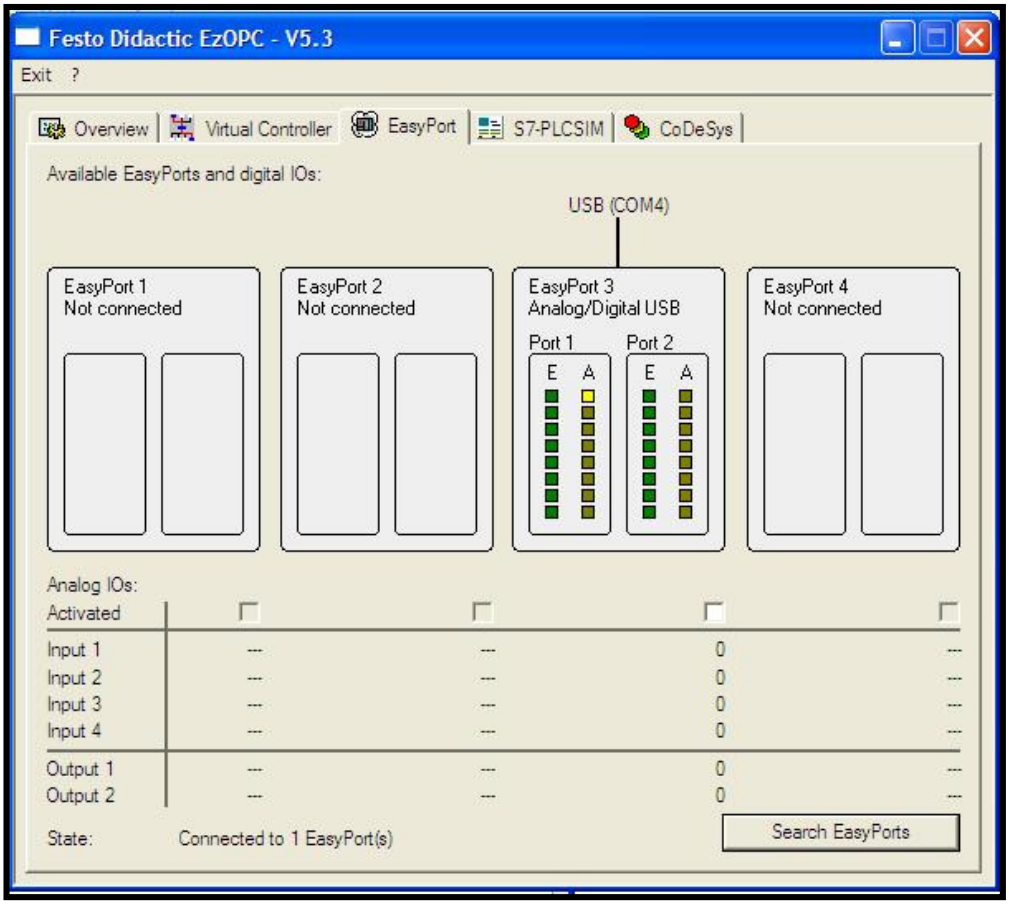


FIGURA 3.27 Verificación de conectividad en EzOPC.

[Fuente: Autoras Tesis]

16. Para completar la comunicación bidireccional se escribe en la entrada 1 del entrenador digital un 1L, el cual se aprecia en RobotinoView2 que la entrada DI_Port_1 está encendida.

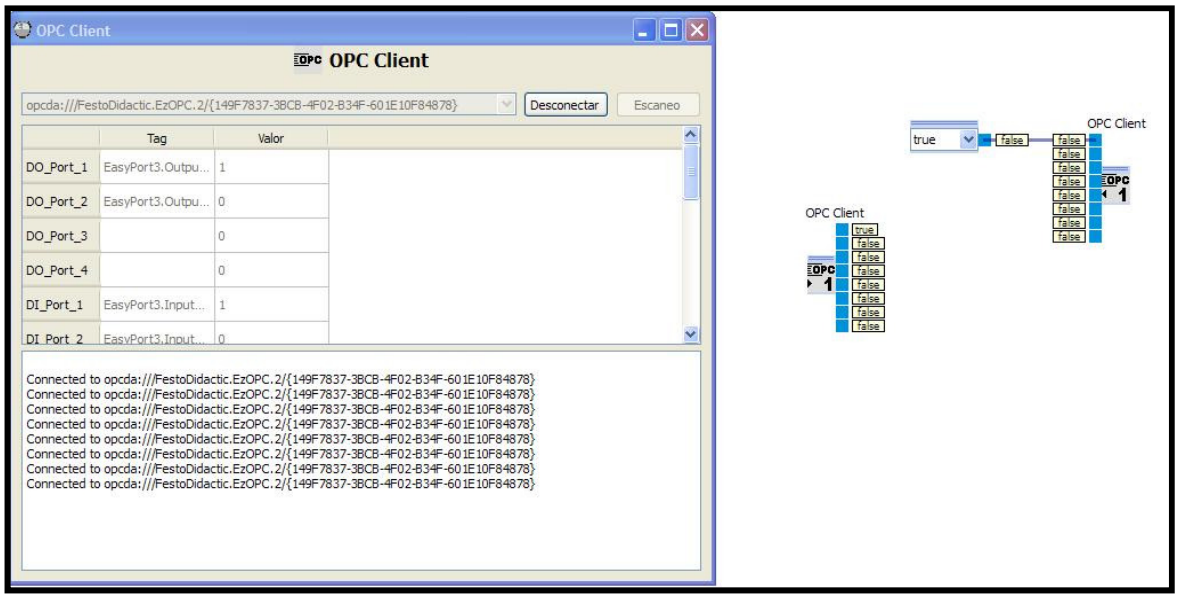


FIGURA 3.28 Verificación de envío y recepción de datos en RobotinoView.

[Fuente: Autoras Tesis]

17. Finalmente se consigue visualizar la comunicación en el servidor con un cliente donde se escriben y leen los datos que sean enviados.

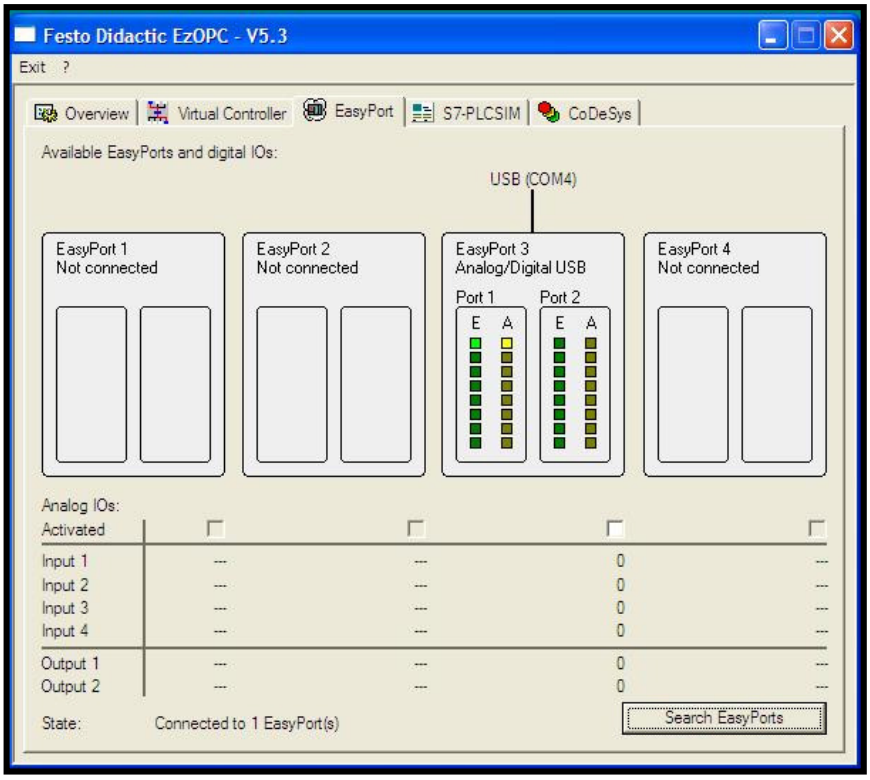


FIGURA 3.29 Verificación de envío y recepción de datos en EzOPC.

[Fuente: Autoras Tesis]

En las figuras que se muestran a continuación se muestran las conexiones físicas entre Edutrainer y EasyPort, con su respectivo funcionamiento.

FIGURA 3.30 Conexiones físicas para comunicación OPC

[Fuente: Autoras Tesis]

FIGURA 3.31 EasyPort enviando y recibiendo datos a través de la comunicación OPC.

[Fuente: Autoras Tesis]

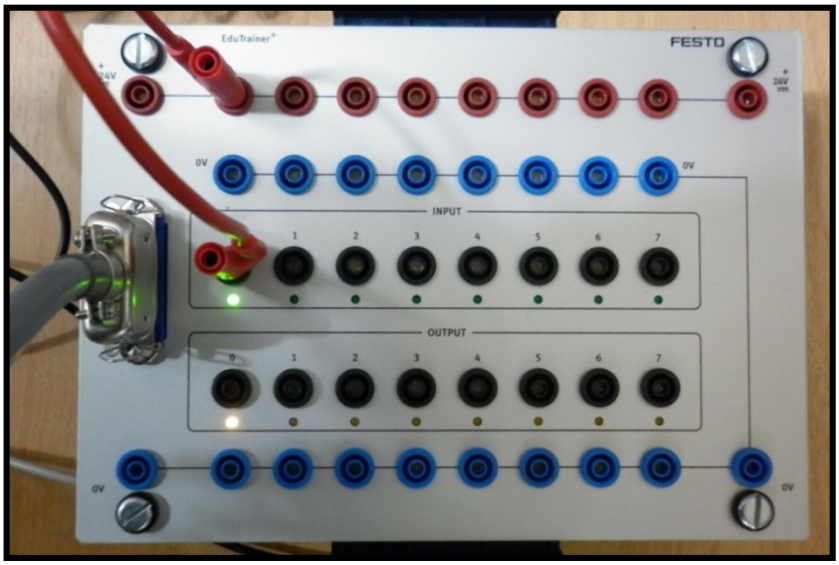


FIGURA 3.32 Edutrainer en funcionamiento.

[Fuente: Autoras Tesis]

3.3.3 Implementación de la interfaz HMI.

La interfaz HMI a ser implementada permitirá el monitoreo y supervisión del sistema.

Se requiere abrir InTouch de Wonderware y crear un nuevo proyecto. Para este caso se tendrán 2 ventanas (como se había diseñado), una ventana de Inicio y otra de Proceso.

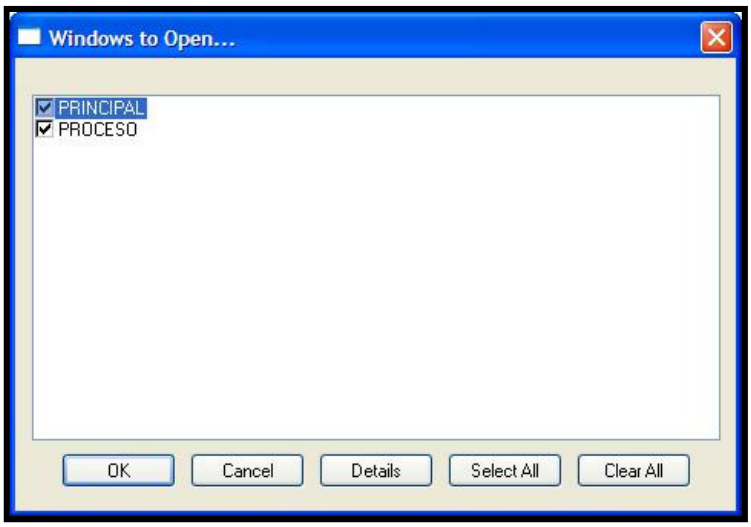


FIGURA 3.33 Ventanas creadas en InTouch.

[Fuente: Autoras tesis]

FIGURA 3.34 Ventana Principal

[Fuente: Autoras tesis]

La ventana PRINCIPAL es la presentación del sistema y posee un botón *INICIO* el cual permite el acceso a la ventana de Proceso.

El botón se lo programa haciendo doble clic sobre el mismo y escogiendo la opción *Show Window*, además se debe seleccionar la ventana a la cual va a dirección el botón una vez activado.

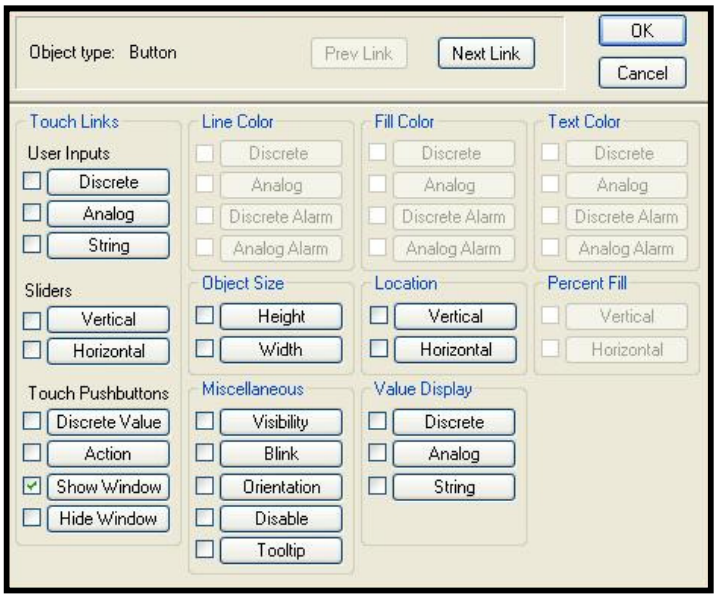


FIGURA 3.35 Ventana de opciones para animación.

[Fuente: Autoras tesis]

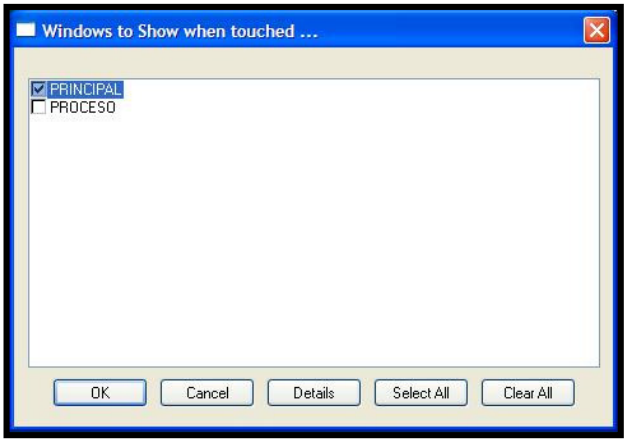


FIGURA 3.36 Ventana para mostrar y direccionar a otras ventanas.

[Fuente: Autoras tesis]

La ventana PROCESO contiene todo el entorno industrial implementado, aquí se puede monitorear el estado decada una de las estaciones del sistema y su funcionamiento.

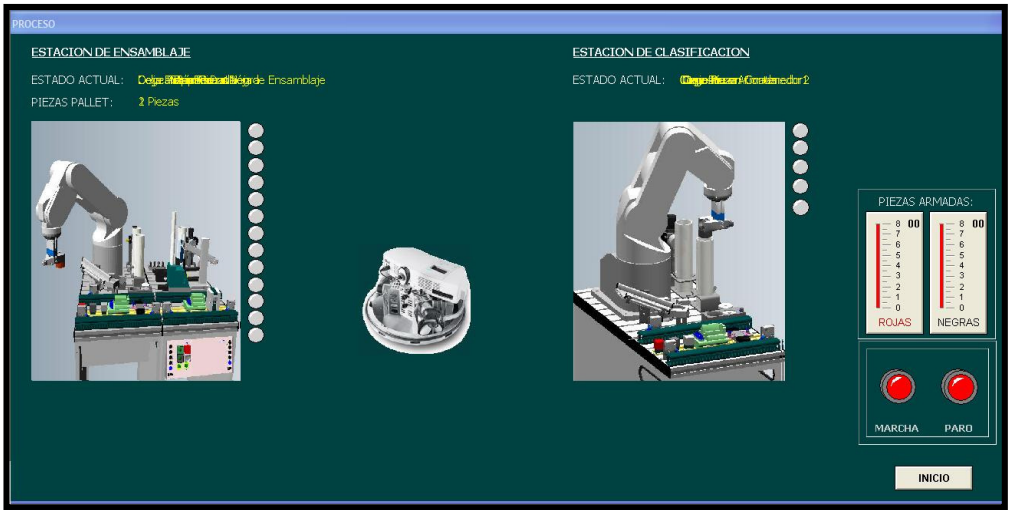


FIGURA 3.37 Ventana de Proceso

[Fuente: Autoras tesis]

Para cual es necesario entablar una comunicación entre InTouch y el PLC S7200 quien es el controlador de todo el sistema. Esta comunicación se realiza a través del cable de programación PPI (Point-to-Point-Interface) del PLC y drivers para InTouch llamados IOServers los cuales permitirán obtener información de los registros que han sido programados en el PLC y que se requieran visualizar en la interfaz HMI.

Wonderware, como fabricante de software, tiene la imperiosa necesidad de comunicarse con cualquier tipo de PLC o dispositivo de campo. Es decir debe desarrollar:

- Drivers que lean protocolos serie (profibus, modbus, devicenet…)

- Protocolos basados en TCP/IP (Industrial Ethernet, Modbus TCP/IP, Ethernet IP)

- Comunicarse vía OPC (Ole forProcess Control)

Para ello, Wonderware cuenta con:

- Drivers Desarrollados directamente por Wonderware.

- Drivers Desarrollados por Third-Party (empresas que han desarrollado drivers específicos para protocolos o equipos y que han sido testeados por Wonderware).

Cuando el driver que se vaya a utilizar haya sido Desarrollado directamente por Wonderware, se adquiere InTouch con IO Server.

Cuando el driver pertenezca a una Third-Party utilizaremos InTouch sin IO Server.

Para este caso se consiguió el driver s7200ppi desarrollado por Worderware, para comunicar con todos los PLC Siemens de la familia del 200. El cual debe ser instalado en el disco C. Una desventaja de dicho driver es que no posee licencia, por lo que solo puede ser utilizado durante 30 días.

Ahora hay que programar las animaciones de la ventana PROCESO, a cada una de las animaciones hay q asignarle un Tag y una dirección específica del PLC. Para esto se crea un Access Name dentro de InTouch para acceder al PLC.

Abrir el diccionario de Tags y hacer clic en *Access Name*:

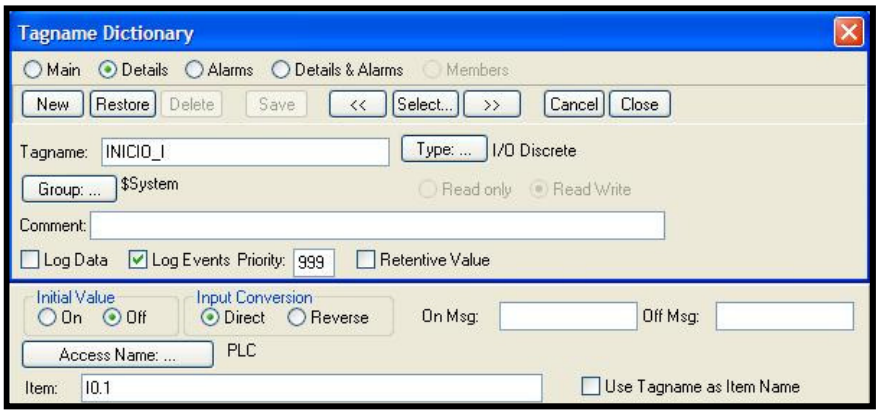


FIGURA 3.38 Ventana del diccionario de Tagnames.

[Fuente: Autoras tesis]

Hacer clic en *Add* para crear un nuevo acceso:

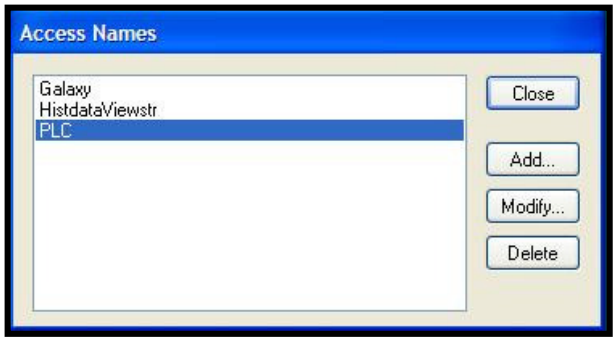


FIGURA 3.39 Ventana para crear Access Name

[Fuente: autoras tesis]

Ingresar los datos para el acceso, hay que tener en cuenta el nombre que se le asigne, puesto que esto servirá para enlazar la comunicación. En *ApplicationName* se coloca el nombre del IO Server del PLC y en este caso el tipo de protocolo a usar es DDE (Dynamic Data Exchange)[9]; hacer clic en *OK* y el acceso ha sido creado.

[9] DDE (Dynamic Data Exchange), http://es.wikipedia.org/wiki/Dynamic_Data_Exchange.

FIGURA 3.40 Ventana para configuración de Access name.

[Fuente: autoras tesis]

Ahora a cada animación se le concede un *Tagname* y una dirección física o lógica del PLC en *Item*, como se muestra en la siguiente figura:

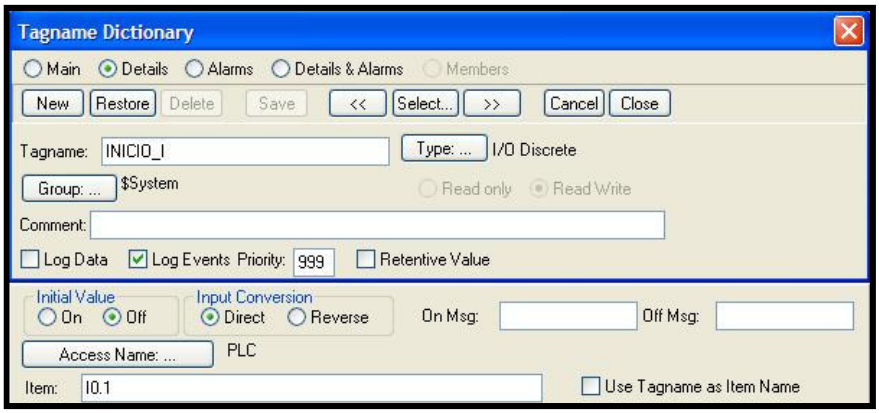


FIGURA 3.41 Ventana del diccionario de Tagnames.

[Fuente: Autoras tesis]

La Tabla 3.1 indica las direcciones utilizadas en el PLC, los *Tagnames* que se han asignado en InTouch y los bits de las controladoras de los brazos robóticos, para de esta manera tener un mapa de las direcciones que se han utilizado y las que se desean visualizar.

DESCRIPCION	I/PLC	O/PLC	INTOUCH	I/BR1	O/BR1	I/BR2	O/BR2
STOP	I0.0	Q0.0	PARO_I/ PARO_O				
START	I0.1	Q0.1	INICIO_I/INICIO/O				
LLEGA ROBOTINO BR1	I0.2	Q0.2	INICIA_O	BIT 7			
				BIT			
LLEGA ROBOTINO BR2	I0.4	Q0.3	INICIO_I	11			
INICIA CONTADOR C1	I0.3				BIT 6		
COGE PIEZA BR1 C1=1	C1=1	Q1.0	CP_BR1		BIT 6		
DEJA PIEZA BR1	C1=2	Q1.1	DP_BR1		BIT 6		
SENSA COLOR BR1	I0.5	Q1.2	SC_BR1	BIT 1			
DEJA PIEZA POSICION							
ENSAMBLAJE	C1=3	Q1.3	POS_ENSB		BIT 6		
PISTON PIEZA ROJA	C1=4	Q1.4	PP_ROJA		BIT 6		
PISTON PIEZA NEGRA	C1=4	Q1.5	PP_NEGRA		BIT 6		
DEJA PISTON	C1=5	Q3,6	D_PISTON		BIT 6		
COGE RESORTE	C1=6	Q1.6	C_RESORTE		BIT 6		
COLOCA RESORTE	C1=7	Q1.7	D_RESORT		BIT 6		
COGE TAPA	C1=8	Q2.0	C_TAPA		BIT 6		
COLOCA TAPA	C1=9	Q2.1	D_TAPA		BIT 6		
COGE PIEZA ARMADA	C1=10	Q2.2	P_ARMADA		BIT 6		
COLOCA PIEZA PALET	C1=11	Q2.3	P_PALET		BIT 6		
TERMINA 1ERA PIEZA	C1=12	Q2.4	FIN_P1		BIT 6		
INICIA CONTADOR C2	C1=12				BIT 6		
TERMINA 2DA PIEZA	C2=2	Q2.5	FIN_P2				
INICIA CONTADOR 1							
BR2 (C3)	I0.6						BIT 8
COGE PIEZA BR2	C3=1	Q2,6	CPP_BR2				BIT 8
DEJA PIEZA BR2	C3=2	Q2,7	DP_BR2				BIT 8
SENSA COLOR BR2	I0.7	Q3.0	SC_BR2			BIT 9	
COGE PIEZA	C3=3	Q3.1	CP_BR2				BIT 8
DEJA CONTENEDOR 1	C3=4	Q3.2	CONT1				BIT 8
DEJA CONTENEDOR 2	C3=4	Q3.3	CONT2				BIT 8
CLASIFICA 1 PIEZA	C3=4	Q3,4	CLAS1				BIT 8

INICIA CONTADOR 2 BR2 (C4)	C3=4						BIT 8
CLASIFICA 2 PIEZAS	C3=2	Q3.5	CLAS2				BIT 8

TABLA 3.2 Mapa de las direcciones de PLC, InTouch y Brazos Robóticos

[Fuente: Autoras tesis]

Una vez asignados los tags a las animaciones, se puede visualizar en el diccionario de Tags todos los nombres que se han ingresado, se debe tener en cuenta que todos estén dentro del Access Name: PLC, puesto que el acceso que se creó para la comunicación entre PLC e InTouch.

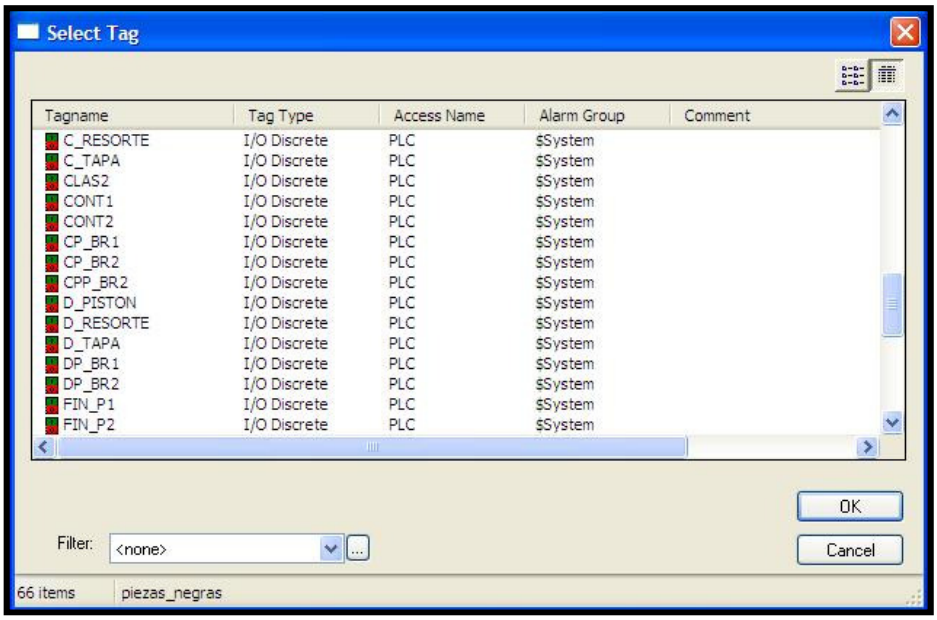


FIGURA 3.42 Lista de Tagnames.

[Fuente: Autoras tesis]

La mayoría de animaciones solo son del tipo visibilidad, es decir aparecen o desaparecen según el caso. Para esto se debe hacer doble clic sobre el elemento que se desea animar y escoger la opción: *Visibility*.

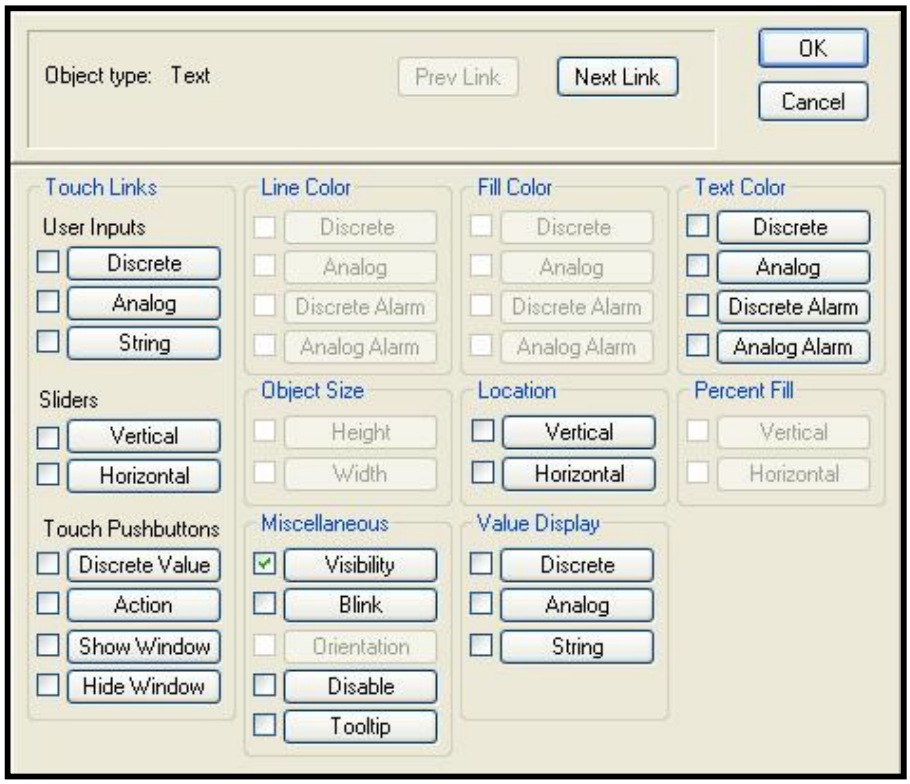


FIGURA 3.43 Ventana de opciones para animación.

[Fuente: Autoras tesis]

Dentro de dicha opción, hay que colocar la condición para que la animación funcione según lo que se desea.

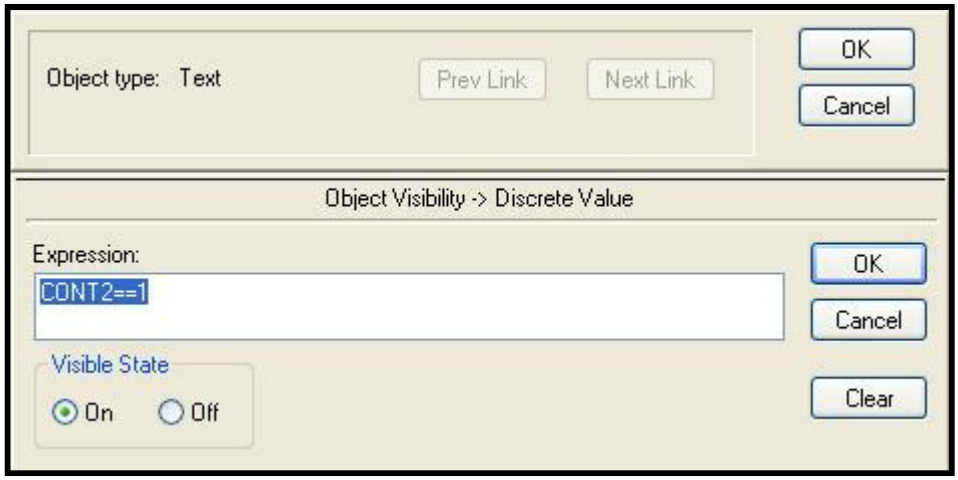


FIGURA 3.44 Ventana para configurar condiciones de Visibilidad.

[Fuente: Autoras tesis]

Otra animación que se realiza en la ventana Proceso es contar el número de piezas armadas, para esto se coloca un icono de tipo *Meters* y se le asigna la

dirección de un contador que se ejecuta dentro del PLC. Cabe recalcar que en el computador solo se debe visualizar y todo el control se realiza en el PLC (Controlador Lógico Programable).

FIGURA 3.45 Contador de piezas armadas

[Fuente: Autoras tesis]

La ventana PROCESO también cuenta con un botón, el cual permite regresar a la ventana de inicio, dicho botón debe ser programado de la misma manera que el botón de la ventana PRINCIPAL.

3.3.4 Puesta en Marcha

Una vez realizadas las pantallas de la interfaz, de debe conectar el PLC al computador a través de un cable PPI. Para este caso el cable debe ser del tipo RS485.

Como se ha nombrado anteriormente, se requiere un driver (IO server) para lograr la comunicación entre el PLC y el software InTouch, por lo tanto, el primer paso es abrir el IO server S7200PPI, que es la herramienta para el PLC que se utiliza en este proyecto.

FIGURA 3.46 IO Server para comunicar PLC S7200 con InTouch

[Fuente: Autoras tesis]

A continuación se deben configurar algunos aspectos que se encuentran en el menú de configuración de S7200PPI.

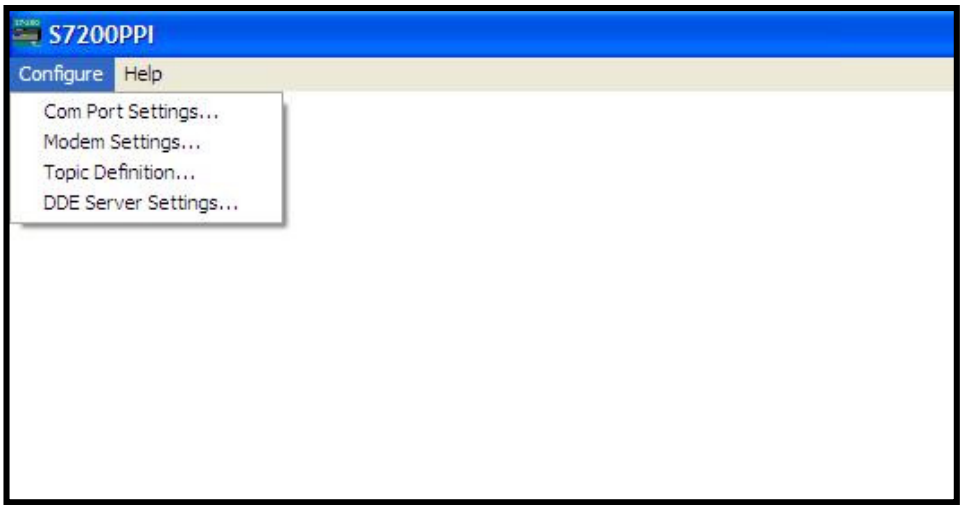


FIGURA 3.47 Menú de configuración de S7200PPI

[Fuente: Autoras Tesis]

La configuración del puerto de conexión del cable PPI es lo primero que se debe tomar en cuenta y especificar.

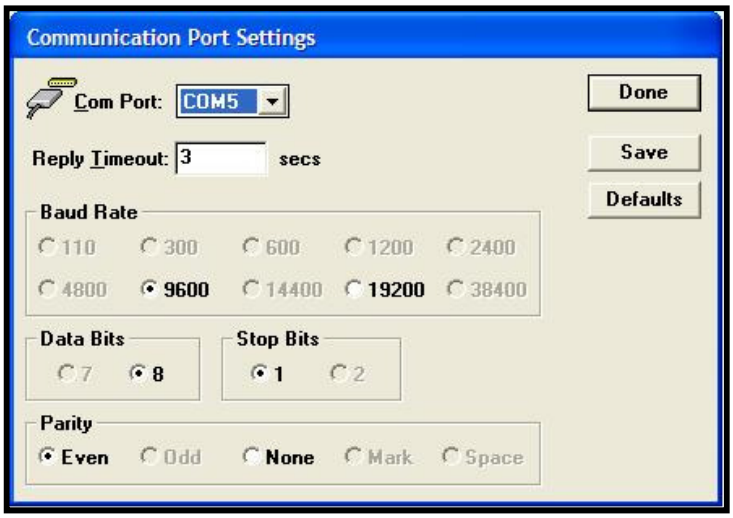


FIGURA 3.48 Ventana de configuración del puerto de comunicación de IO Server

[Fuente: Autoras Tesis]

Para la ventana Modem Setup solo se define el puerto y las demás opciones, quedan por defecto.

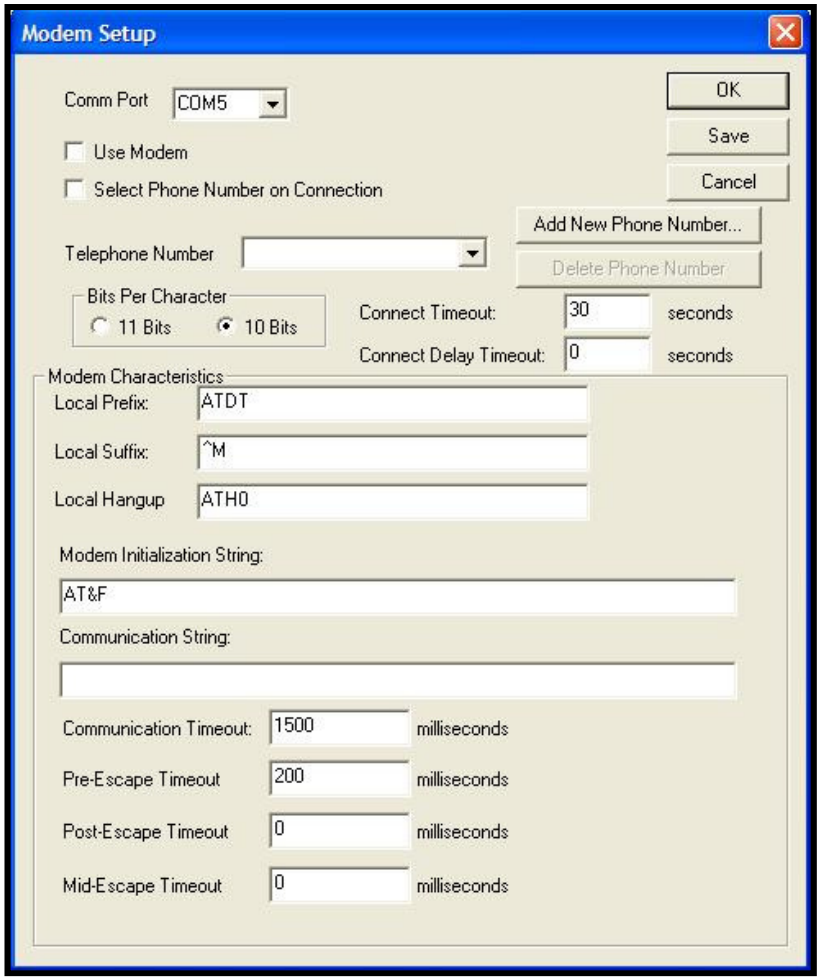


FIGURA 3.49 Ventana de configuración Modem Setup de IO Server

[Fuente: Autoras Tesis]

Ahora se tiene que crear el Topic o acceso, el cual debe ser el mismo que se creó en InTouch para definir los Tagnames que tienen relación con el PLC. Para esto, se hace clic en *New* de la ventana Topicdefinition.

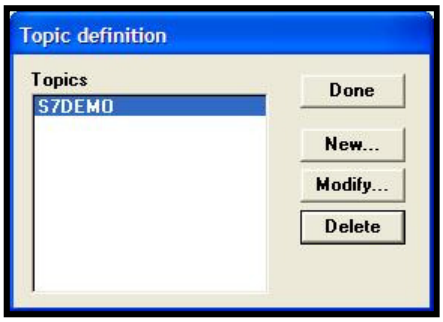


FIGURA 3.50 Ventana de TopicDefinition de IO Server

[Fuente: Autoras Tesis]

A continuación, se coloca en *TopicName* el mismo que el anteriormente creado, es decir, PLC; la dirección del PLC es la número 2, y se debe especificar el puerto.

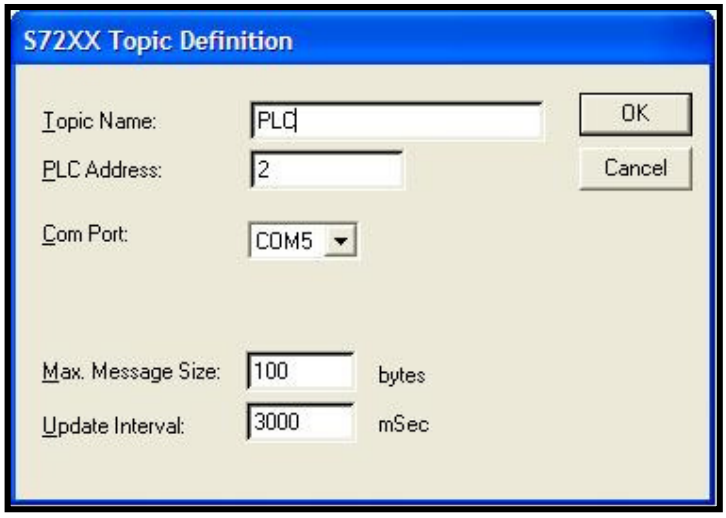


FIGURA 3.51 Ventana de configuración del acceso de comunicación de IO Server

[Fuente: Autoras Tesis]

Finalmente, en la ventana DDE SeverSettings, se coloca el tiempo de pulsos para este protocolo de comunicación, 50msec.y el directorio donde se encuentra instalado el IO Server.

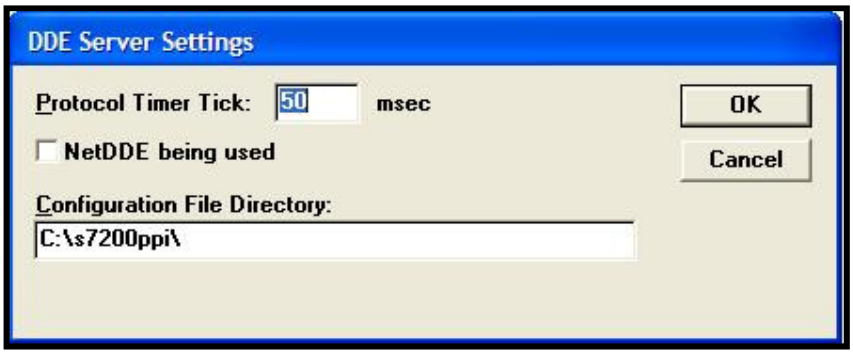


FIGURA 3.52 Ventana de configuración para DDE server de IO Server

[Fuente: Autoras Tesis]

Una vez configurado S7200PPI se requiere abrir el Cliente, quien permitirá inicializar la comunicación, puesto que se está implementando una comunicación de tipo Cliente/Servidor.

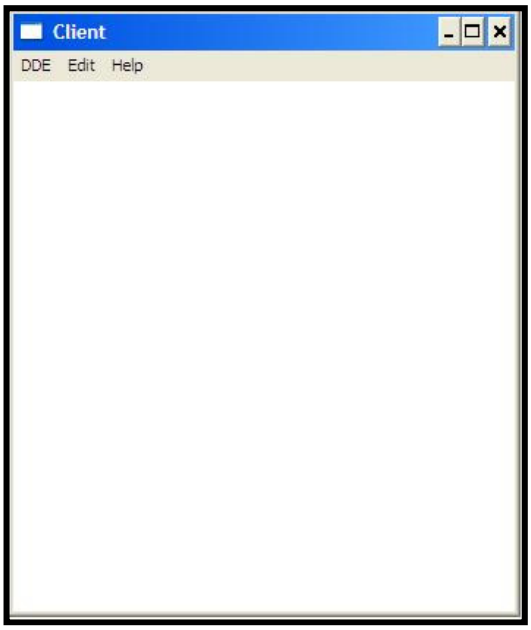


FIGURA 3.53 Cliente de IO Server para comunicar PLC S7200 con InTouch

[Fuente: Autoras Tesis]

Como muestra la figura, el cliente tiene un menú de opciones para la comunicación, se elegirá *Initiate* con el cual se abre una nueva ventana.

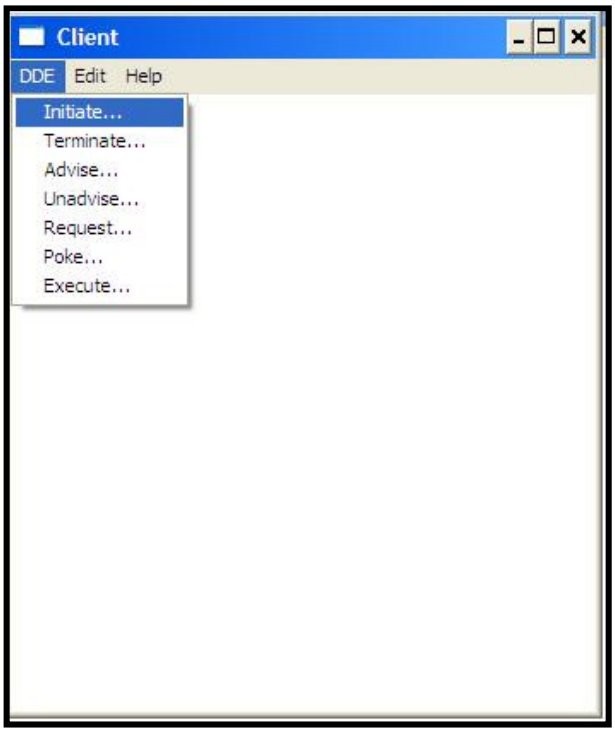


FIGURA 3.54 Menú para comunicación entre Cliente y servidor S7200PPI

[Fuente: Autoras Tesis]

Aquí se debe especificar la aplicación y el acceso, que para el caso son: S7200PPI y PLC, respectivamente.

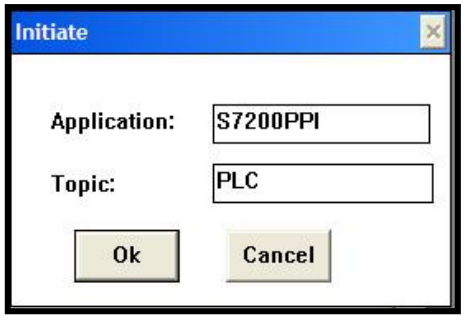


FIGURA 3.55 Ventana de configuración para iniciar comunicación.

[Fuente: Autoras Tesis]

De esta manera se iniciará la conexión, como se puede apreciar en la siguiente figura:

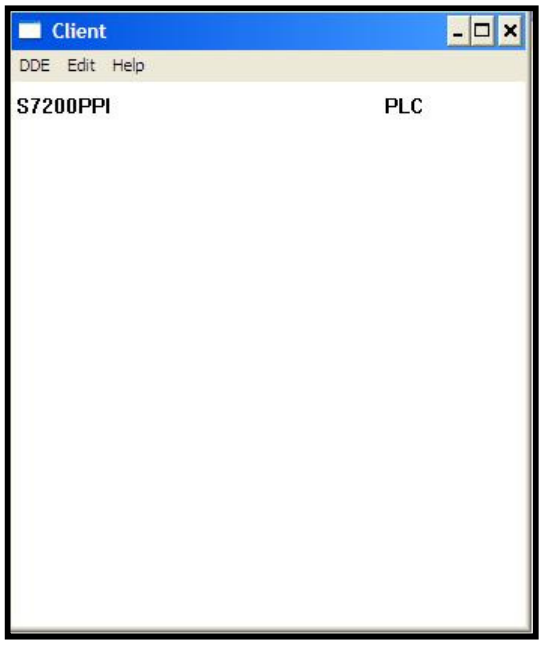


FIGURA 3.56 Comunicación Inicializada.

[Fuente: Autoras Tesis]

Dentro del menú DDE se tiene la opción *Request* que permitirá verificar la comunicación con el PLC, para esto se coloca la dirección una entrada o salida del PLC para leer su estado.

En la figura se ha leído la salida Q0.1 que es la señal de Marcha del sistema, el cual ha enviado el valor 1, es decir que su estado es activo.

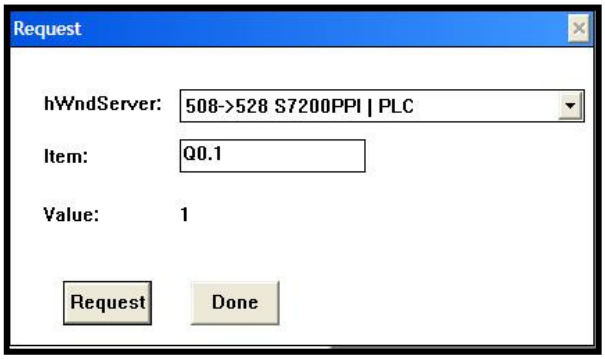


FIGURA 3.57 Ventana de Request para comprobar la comunicación.

[Fuente: Autoras Tesis]

En S7200PPI también es posible verificar la comunicación, ya que presenta la acción que se ha realizado.

FIGURA 3.58 Lectura de puertos físicos de PLC S7200.

[Fuente: Autoras Tesis]

Una vez comprobada la comunicación entre el PLC y los IO Servers se puede poner en marcha InTouch haciendo clic en el botón *Runtime* que se encuentra en la parte superior derecha de la pantalla, de esta manera la interfaz correrá en tiempo real.

Para el funcionamiento adecuado del proceso implementado, se debe tomar en cuenta que todos los elementos estén encendidos y en modo *RUN* para poder accionar el botón *Start* del panel de control de la estación de ensamblaje y seguidamente se debe activar el botón *Reset* del mismo panel.

En caso producirse un Paro de emergencia al presionar el botón *Stop*, se debe reiniciar completamente cada estación de trabajo y activar el botón *Start*.

Una vez armadas cuatro piezas rojas o negras en la estación de ensamblaje, los pistones se agotarán, por lo que el brazo robótico se detendrá, hasta que manualmente se coloquen dichos pistones en las posiciones apropiadas y se presione el botón *Reset* del panel de control.

En la figura se puede apreciar que el indicador de Marcha esta encendido, como se había verificado anteriormente.

FIGURA 3.59 Comprobación de comunicación en InTouch.

[Fuente: Autoras Tesis]

3.3.1 Pruebas y Ajustes

La Tabla 3.2 muestra las modificaciones que se han tenido que realizar durante la implementación y sincronía de los sistemas que se controlan en el entorno de proceso industrial en estudio.

Nº	ACTIVIDAD	OBSERVACIONES	APORTE
1	Programación Br1 y Br2	Colocar señales de salida en cada subrutina hacia el PLC. para sincronización del sistema	Válido
2	Acoplamiento del Br1 al sistema	Conexión adecuada de cables internos de la concentradora	Válido
3	Acoplamiento del Br2 al sistema	Se requiere la adición de una concentradora para obtener I/O del entorno del brazo hacia la controladora	Válido
4	Programación Inicio y paro de Emergencia	Se acopla el panel de control del Br1 hacia todo el sistema a través del PLC, para lo cual se requiere la programación de la señal de inicio y paro de emergencia.	Válido
5	Programación PLC	Fallas en la sincronización con la interfaz HMI para lo cual se utiliza contadores y temporizadores internos en el PLC	Válido
6	Programación Robotino (Prueba1)	Programación de sensor inductivo para detección de llegada a estaciones (Prueba fallida)	Fallido
7	Programación Robotino (Prueba2)	Programación de sensor infrarrojo de distancia para detección de llegada a estaciones	Válido

8	Implementación HMI	Sincronización entre el PLC y la PC a través de los IO server	Válido
9	Montaje de estaciones en el lugar de trabajo y pruebas del Robotino en pista	Existe mucho reflejo de luz en la pista, cambiar filtros de imagen en el Robotino	Válido
10	Cableado de las estaciones y pruebas generalesde los brazos	Cambiar posiciones para posicionamiento del pallet	Válido
11	Montaje Pallet en Robotino	La placa necesita de un aislante para su acople en el chasis del robot	Válido
12	Programación Robotino (Prueba 3)	Giro de 180ª impreciso, se adiciona sensores ópticos.	Fallido
13	Acoplamiento de EzOPC al sistema	Se requiere unir la señal común de tierra a todo sistema	Válido
14	Tomar posiciones Br1 y Br2	Tomar posiciones para dejar o recoger las piezas del pallet que lleva el Robotino	Válido
15	Programación Br2	Se requiere cambiar la posición P3 de identificación de color de pieza para su debido acoplamiento con el entorno de trabajo	Válido
16	Programación Robotino (Prueba4)	Reemplazo de sensores ópticos por sensores de distancia infrarrojos para corregir posicionamiento en cada estación. Pruebas con cubos de 10cm y 3cm a 240º y 120º del Robotino y con Postes en los mismos ángulos	Fallido
17	Programación Robotino (Prueba5)	Utilización de 7 sensores ópticos para el posicionamiento con respecto a 3 paredes dentro del entorno	Válido
18	Modificación de diseño Pallet	Adición de 2 motores en pallet, para el desplazamiento de las piezas hacia la estación de clasificación.	Válido
19	Programación Robotino (Prueba6)	Programación del accionamiento de los motores en Pallet	Válido
20	Bandeja en estación de clasificación	Diseño y elaboración de una bandeja para el desplazamiento de las piezas hacia la rampa del entorno del Brazo2.	Válido

TABLA 3.3 Pruebas y ajustes para sincronía del sistema.

[Fuente: Autoras Tesis]

3.3.2 Integración total del sistema

Para la implementación del prototipo para el control y monitoreo de un proceso industrial mediante el acoplamiento sincronizado del robot móvil Robotino de Festo y los brazos robóticos articulados RV-2AJ de Mitsubishi, utilizando una interfaz HMI, es necesario unir a cada elemento de manera que se tenga una comunicación adecuada para cada uno, siendo el PLC S7200 de CPU 222 el controlador principal del sistema, por lo tanto este debe conectarse a las estaciones de ensamblaje, transporte y clasificación.

De la misma manera que se había diseñado, se tiene que: la Estación de Ensamblaje arma un cilindro de simple efecto, colocando de forma correcta el émbolo, muelle y tapa; tomando en cuenta el tipo de cilindro rojo o negro. El transporte traslada las piezas armadas hacia la estación de clasificación. La estación de clasificación separa piezas rojas de negras y las ubica en contenedores diferentes.

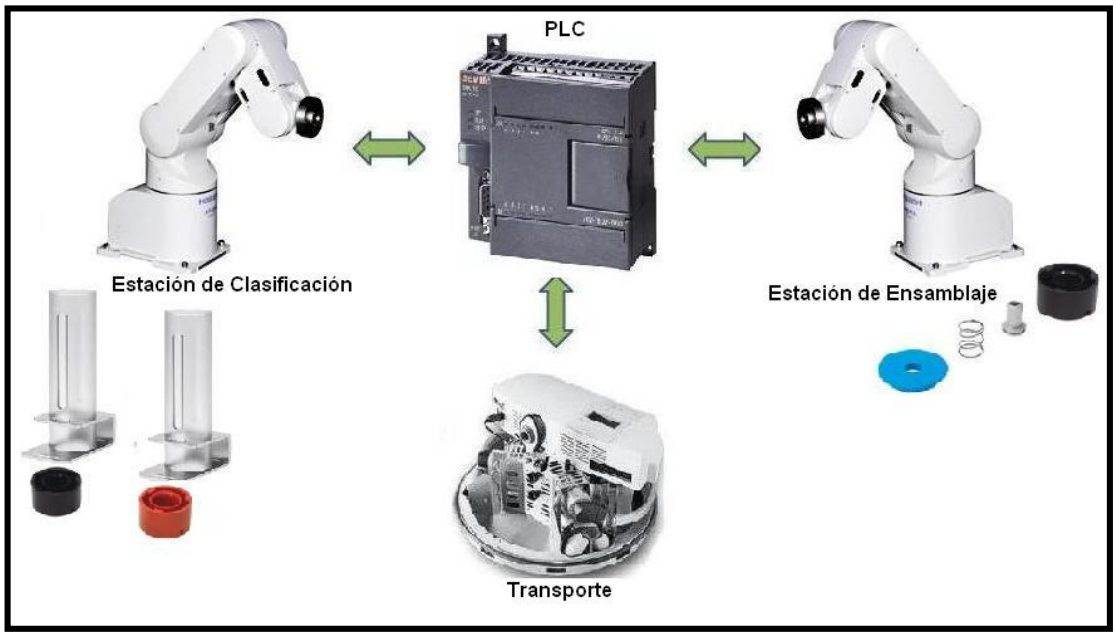


FIGURA 3.60 Diagrama de la integración del proceso.

[Fuente: Autoras Tesis]

En la siguiente figura se muestra las conexiones del PLC S7200, CPU 222, del cual salen todas las señales de control a las diferentes estaciones del proceso. Se ha codificado a las Entradas con 1 y a las Salidas con 0.

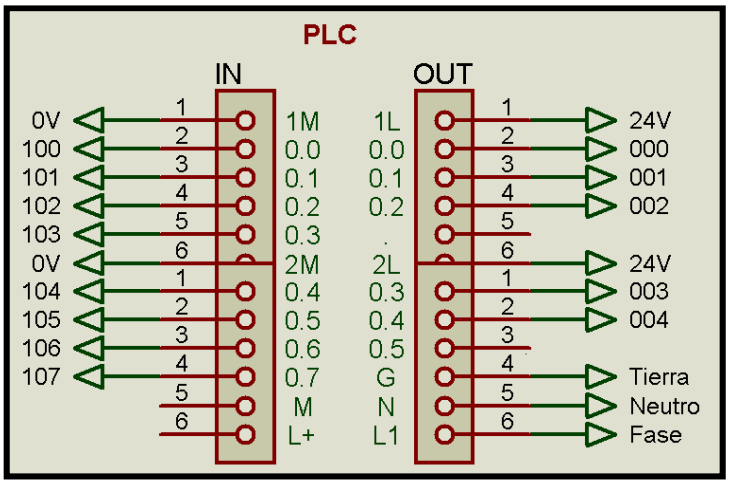


FIGURA 3.61 Conexiones para PLC S7200

[Fuente: Autoras Tesis]

Las señales de Marcha/Paro son tomadas desde el Panel de Control de la estación de ensamblaje, el tiene los botones de Start y Stop.

FIGURA 3.62 Panel de control

[Fuente: www.festo-didactic.com/int-en/learning-system]

En la figura se puede apreciar el acoplamiento necesario para la conexión y comunicación entre el PLC y los brazos robóticos de las estaciones.

Tanto la estación de ensamblaje, como la de clasificación poseen en su entorno de trabajo una bornera de Entradas y Salidas que se comunican directamente a la controladora de los brazos RV-2AJ, pero para la conexión con el PLC fue necesario lo siguiente:

- Tomar las señales desde el PLC y conectarlas a la bornera del entorno de los brazos.

- Llevar las señales desde la bornera hacia la concentradora *Robot interfaz unit* mediante un conector de 24 pines (IEEE 488 (SysLink)).

- Reconectar las señales internas de la concentradora según lo que se requiere para cada estación.

- Enlazar la concentradora hacia la controladora de los Brazos RV-2AJ a través de un conector de 50 pines.

- Finalmente la controladora se comunica con el brazo robótico.

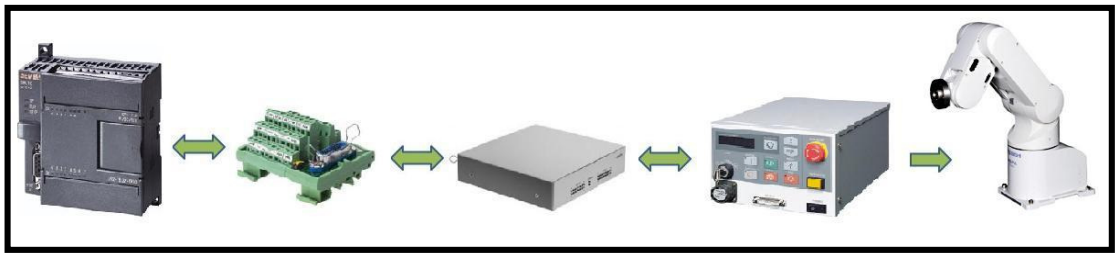


FIGURA 3.63 Diagrama de la integración de los brazos robóticos al proceso.

[Fuente: Autoras Tesis]

A continuación se detallan las conexiones en las borneras y concentradoras de los Brazos de cada estación en las figuras: 3.64, 3.65, 3.67 y 3.68 respectivamente.

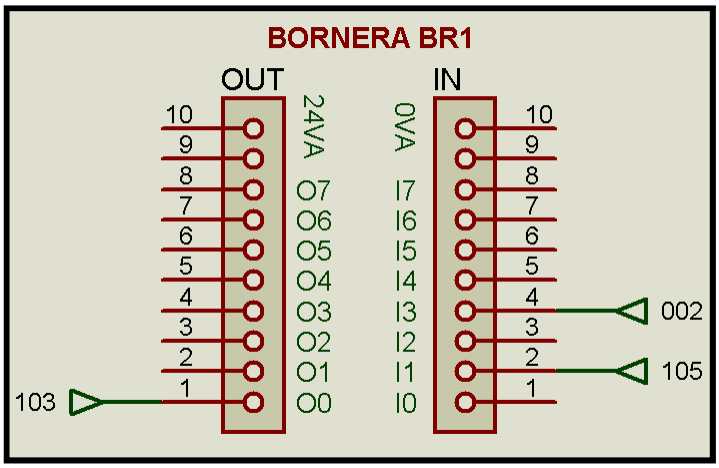


FIGURA 3.64 Conexiones para bornera de Brazo 1, estación de ensamblaje.

[Fuente: Autoras Tesis]

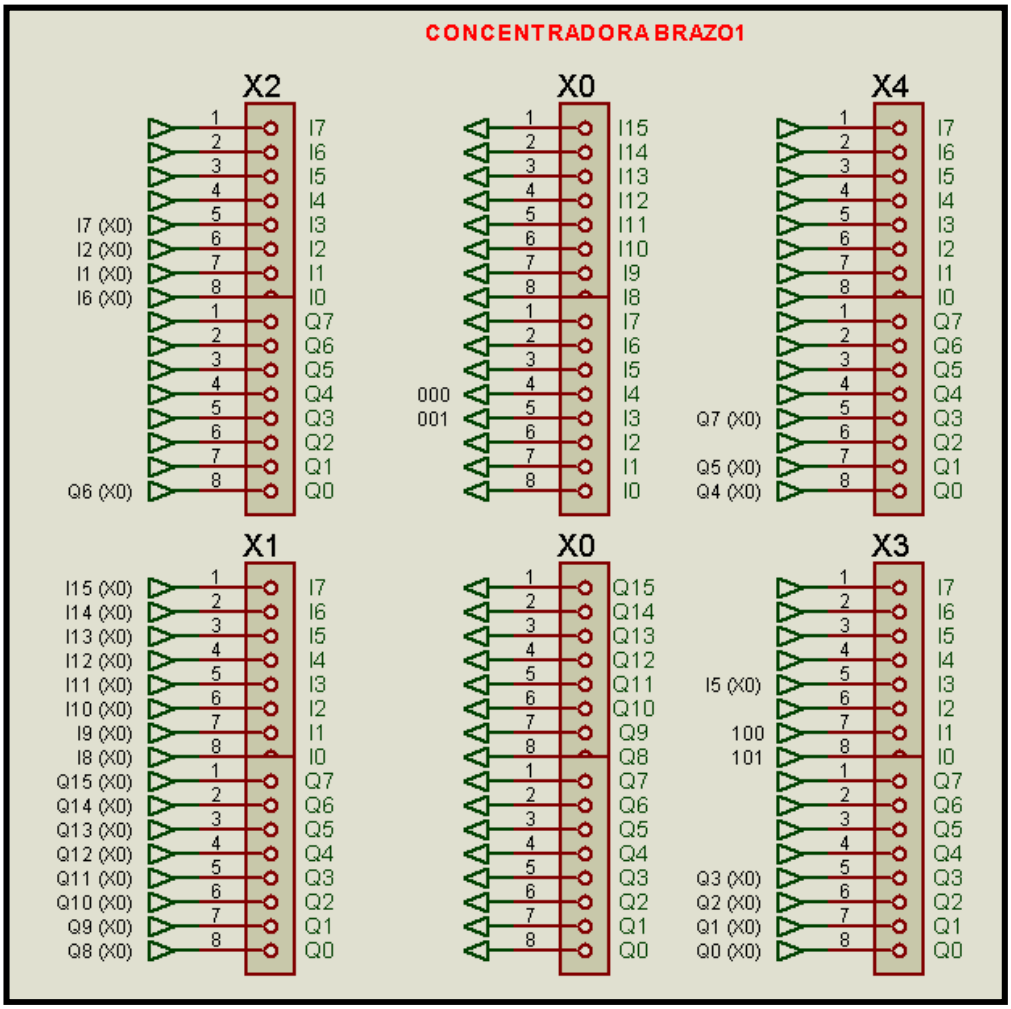


FIGURA 3.65 Conexiones para concentradora de brazo 1, estación de ensamblaje.

[Fuente: Autoras Tesis]

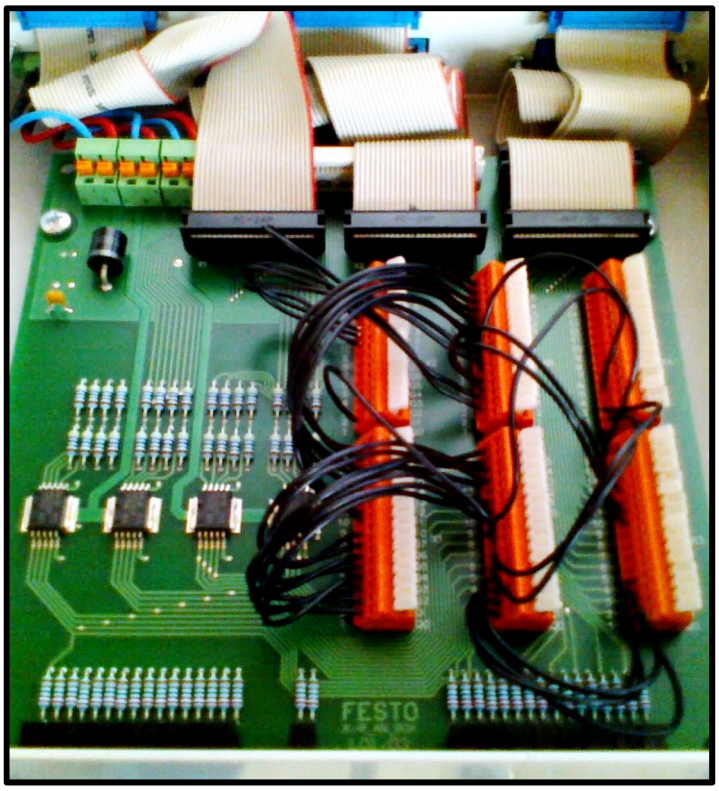


FIGURA 3.66 Conexiones físicas para la concentradora brazo 1. Estación de ensamblaje

[Fuente: Autoras Tesis]

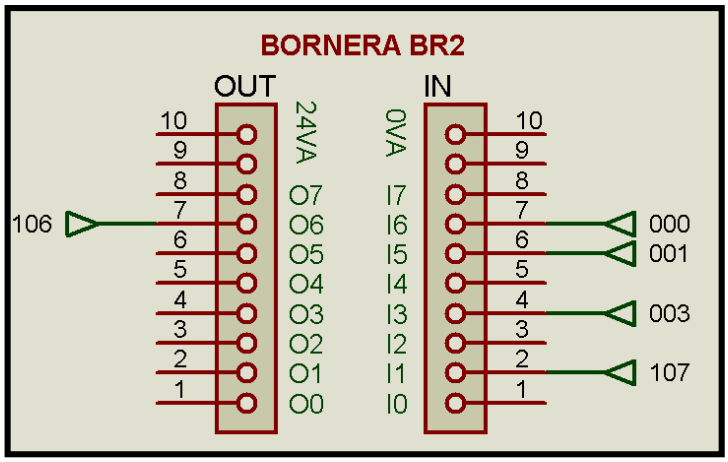


FIGURA 3.67 Conexiones para bornera de brazo 2, estación de clasificación.

[Fuente: Autoras Tesis]

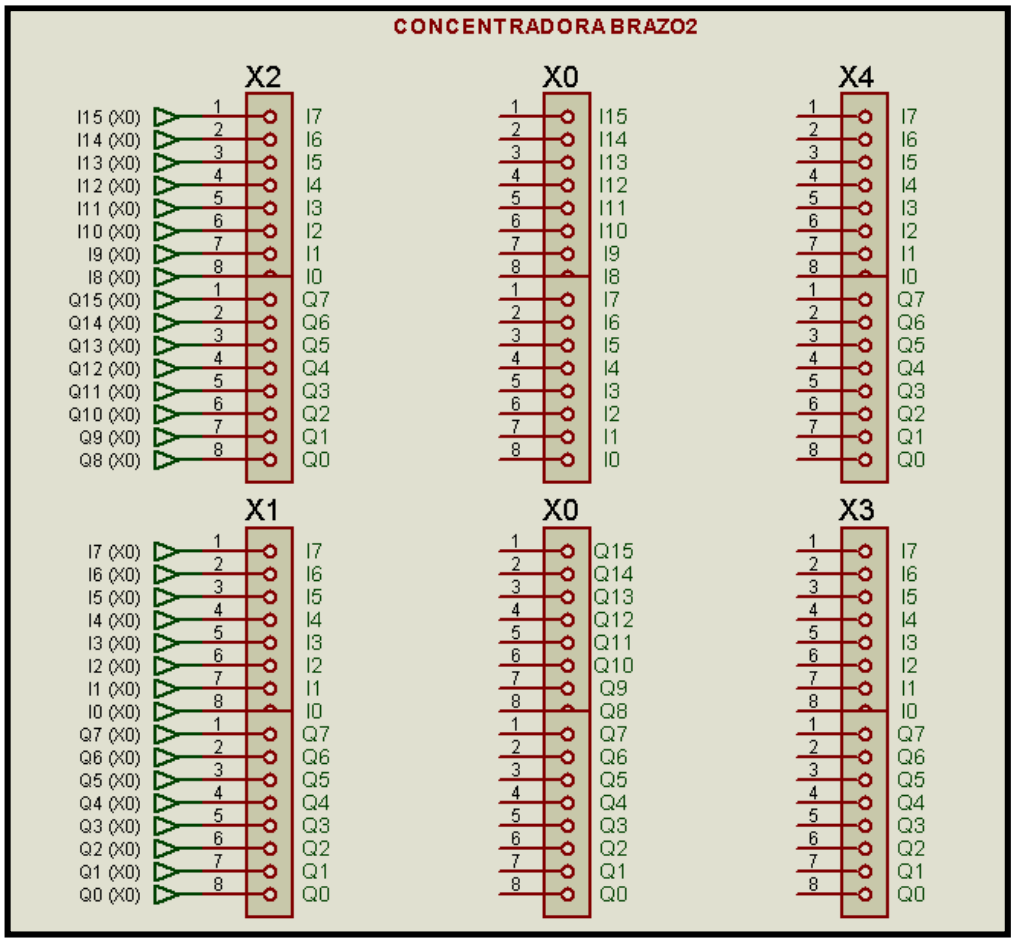


FIGURA 3.68 Conexiones para concentradora de brazo 2, estación de clasificación.

[Fuente: Autoras Tesis]

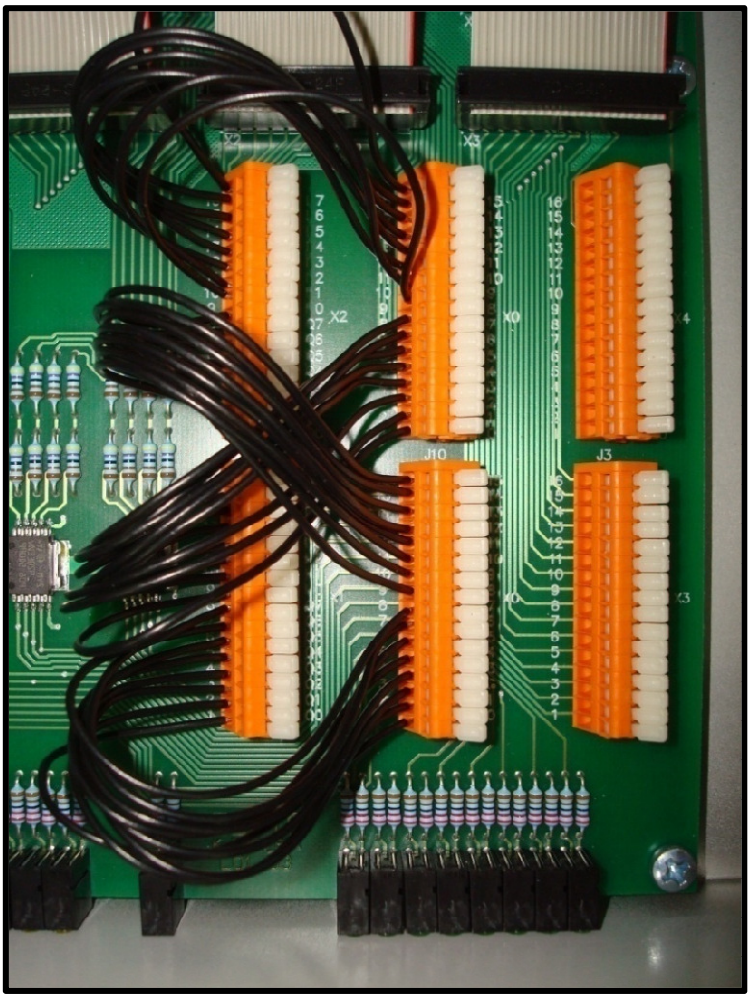


FIGURA 3.69 Conexiones físicas para concentradora de brazo 2, estación de clasificación.

[Fuente: Autoras Tesis]

La siguiente figura 3.70 detalla el acoplamiento necesario para comunicar el PLC con Robotino. Para tener control y sincronización sobre el mismo se requiere:

- Conectar las señales del PLC a la unidad de conexión universal, mediante el acoplamiento de E/S a través de conectores de laboratorio de 4 mm.

- La unidad de conexión universal se enlaza al EasyPort a través de un conector de 24 pines (IEEE 488 (SysLink)).

- Las señales del EasyPort se envían al computador mediante un cable USB y a través de la comunicación OPC entre EZOpc y RoborinoView se acoplan las señales hacia el Robotino.

- El computador se comunica con Robotino inalámbricamente por medio del Access Point que contiene Robotino.

FIGURA 3.70 Diagrama de la integración de Robotino al proceso.

[Fuente: Autoras Tesis]

En la figura se tienen las señales que ingresan del PLC hacia la unidad de conexión universal (EduTrainer I/O digitales).

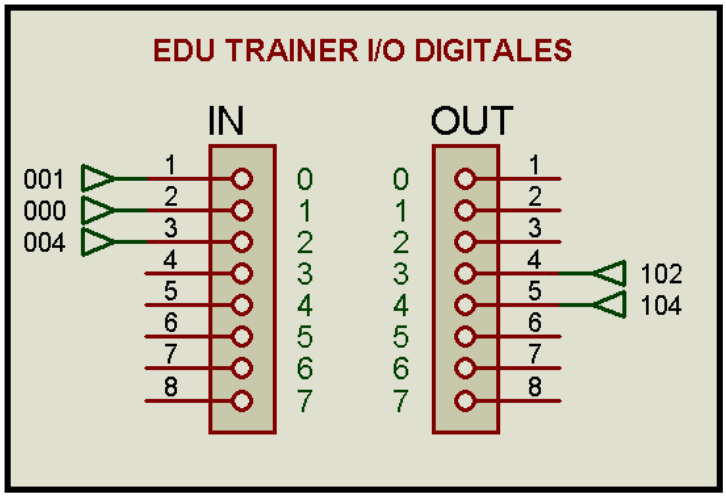


FIGURA 3.71 Conexiones para I/O digitales de EduTrainer, estación de transporte.

[Fuente: Autoras Tesis]

ACOPLAMIENTO MECÁNICO:

El acoplamiento mecánico usado para la integración total del sistema consta de:

- Acoplamiento estación de ensamblaje vs entorno

  Para dicho acoplamiento se usa una base metálica que sirve de sujeción para unir y alinear las dos estaciones. La sujeción se la realiza desde la base de la mesa del entorno y la base de la estación, unidas por tornillos a los extremos como se muestra en la Fig. 3.72.

FIGURA 3.72 Acoplamiento estación de ensamblaje vs entorno

[Fuente: Autoras Tesis]

- Acoplamiento estación de clasificación vs entorno

  Para dicho acoplamiento se usan dos estructuras; la primera es una bandeja que une la mesa del entorno con la rampa de la estación de clasificación por donde se desplazan las piezas a ser clasificadas, como se muestra en la Fig. 3.73, y segundo, una base metálica que sirve de sujeción para unir y alinear las dos estaciones. La sujeción se la realiza desde la base de la mesa del entorno y la base de la estación, unidas por tornillos a los extremos como se muestra en la Fig. 3.74.

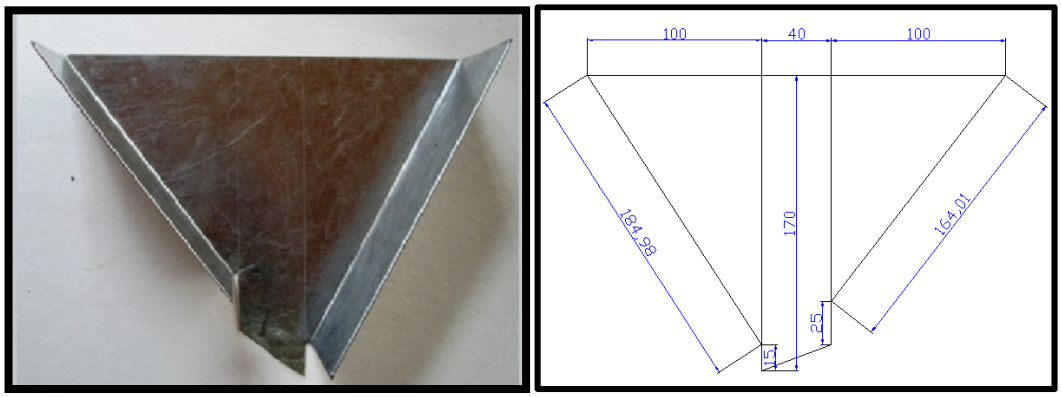


FIGURA 3.73 Bandeja estación de clasificación

[Fuente: Autoras Tesis]

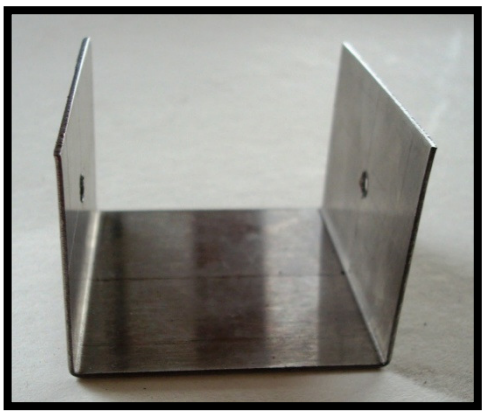


FIGURA 3.74 Acoplamiento estación de clasificación vs entorno

[Fuente: Autoras Tesis]

- Acoplamiento Robot móvil vs entorno

Para el acoplamiento del robot móvil Robotino al entorno, se le incorpora un pallet hecho de acrílico, ver figura 3.75, sujetado mediante una estructura metálica en el chasis del robot, ver figura 3.76. Este acoplamiento se lo realiza para que la estación de ensamblaje pueda depositar las piezas armadas en el pallet, este a su vez transportara las piezas y finalmente accionará un par de motores para desplazar las piezas por la bandeja que lleva a la estación de clasificación. Ver figura 3.77.

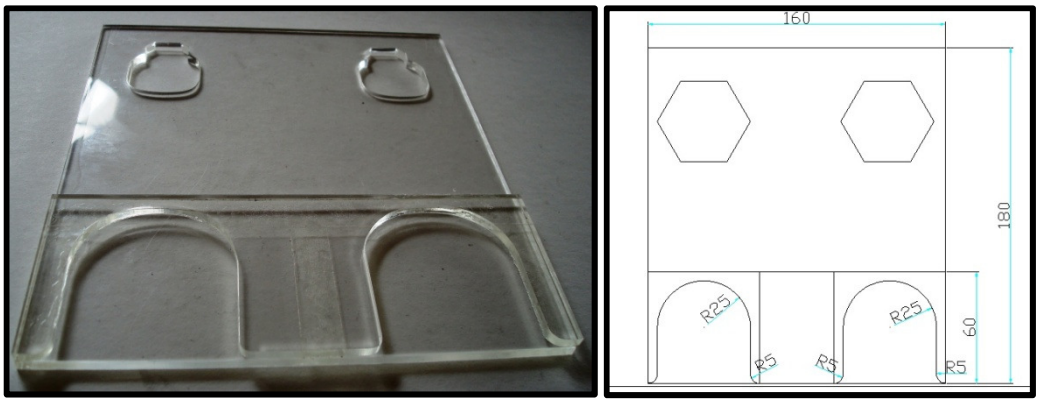


FIGURA 3.75 Pallet

[Fuente: Autoras Tesis]

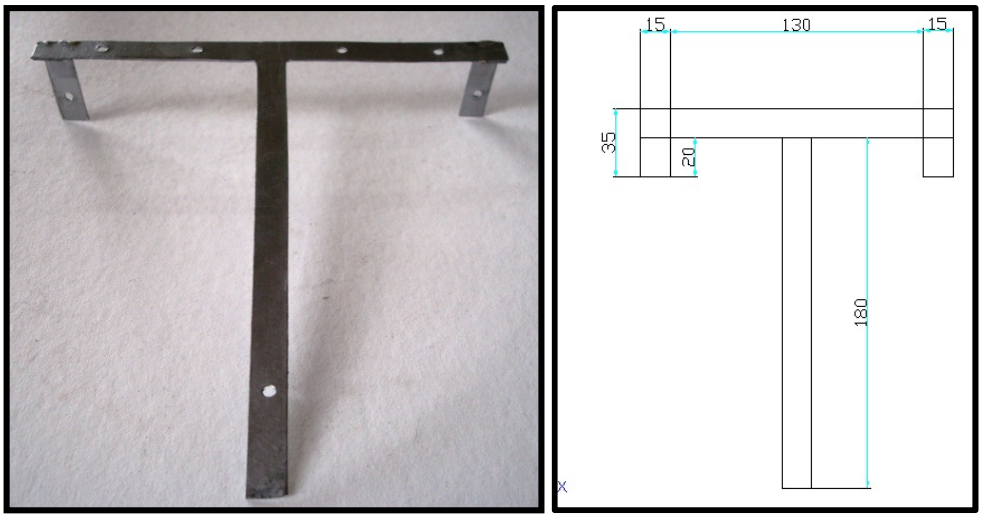


FIGURA 3.76 Soporte Pallet

[Fuente: Autoras Tesis]

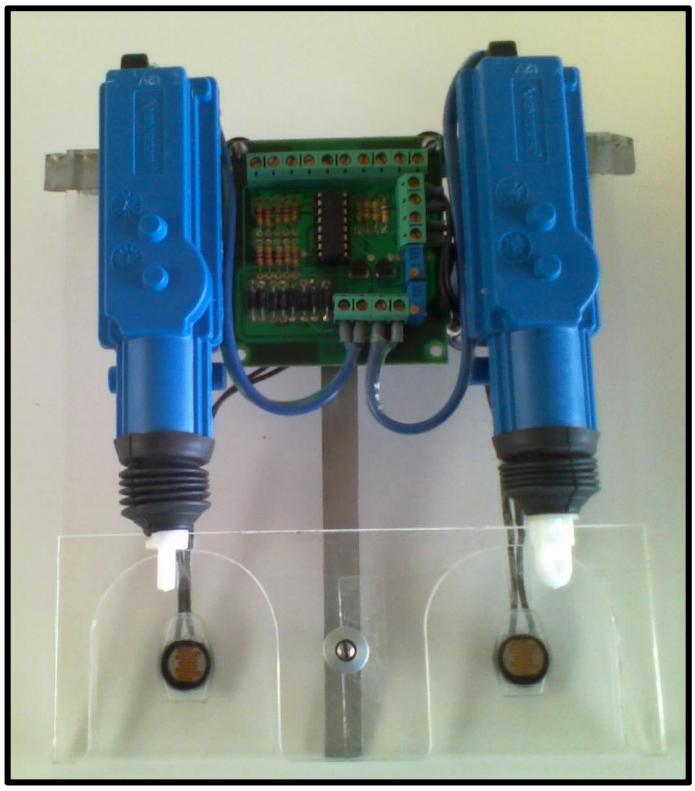


FIGURA 3.77 Estructura para chasis del Robotino

[Fuente: Autoras Tesis]

FIGURA 3.78 Pallet implementado en Robotino.

[Fuente: Autoras tesis]

CAPITULO 4

4. ANALISIS DE RESULTADOS

4.1 Identificación de parámetros para cotejar resultados.

Para la medición de resultados del sistema implementado se han tomado en cuenta ciertos parámetros de las diferentes estaciones de trabajo, para de esta manera tener un análisis de la precisión y comportamiento de los elementos importantes que conforman el proceso industrial en cuestión.

Se debe tomar en cuenta que los siguientes datos se los tomo al 50% del rendimiento de las estaciones.

Parámetros de la Estación de Ensamblaje:

- Tiempo

- Repetibilidad de:

  Coger pieza

  Colocar pieza

  Coger pistón

  Colocar pistón

  Coger resorte

  Colocar resorte

  Coger tapa

Colocar tapa

Colocar pieza en pallet

Parámetros del Transporte:

- Tiempo
- Desviación estándar seguidor de línea
- Repetibilidad activación de motores del pallet

Parámetros de la Estación de Clasificación:

- Tiempo
- Repetibilidad de:

Clasificar piezas rojas

Clasificar piezas negras

4.2 Presentación de resultados.

Estación de Ensamblaje Repetibilidad:

Nota: Si=1; No=0.

Nº	COGER PIEZA	COLOCAR PIEZA	COGER PISTON	COLOCAR PISTON	COGER RESORTE	COLOCAR RESORTE	COGER TAPA	COLOCAR TAPA	COLOCAR PIEZA EN PALLET
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1
7	1	1	1	1	0	0	1	1	1
8	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	0	1

TABLA 4.1 Datos tomados de la Estación de Ensamblaje.

[Fuente: Autoras tesis]

Repetibilidad Transporte:

Nota: Si=1; No=0.

Nº PRUEBA	ACTIVACIÓN DE MOTORES PALLET
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

TABLA 4.2 Datos tomados de los motores del pallet.

[Fuente: Autoras tesis]

ROBOTINO: Desviación estándar seguidor de línea con cámara web.

Los datos se tomaron a partir de los valores en tiempo real que genera el ícono *Detector de líneas* de RobotinoView2, la tabla muestra 30 datos tomados en 30s, es decir 1dato/segundo.

PRUEBA 1		PRUEBA 2		PRUEBA 3		PRUEBA 4		PRUEBA 5	
Nº	x	Nº	x	Nº	x	Nº	x	Nº	x
1	185	1	150	1	163	1	177	1	127
2	163	2	173	2	179	2	180	2	151
3	149	3	172	3	197	3	184	3	168
4	136	4	164	4	193	4	181	4	185
5	117	5	157	5	189	5	178	5	197
6	101	6	142	6	179	6	166	6	189
7	118	7	127	7	165	7	157	7	185
8	134	8	106	8	157	8	145	8	171
9	149	9	114	9	155	9	131	9	160
10	160	10	128	10	138	10	112	10	155
11	171	11	139	11	124	11	101	11	147
12	180	12	146	12	107	12	116	12	130
13	184	13	151	13	108	13	133	13	106
14	189	14	166	14	105	14	143	14	88
15	196	15	182	15	123	15	160	15	114
16	209	16	180	16	142	16	167	16	130
17	210	17	183	17	157	17	176	17	148
18	203	18	209	18	171	18	181	18	167
19	189	19	212	19	174	19	182	19	176
20	169	20	207	20	182	20	195	20	185
21	149	21	188	21	186	21	212	21	186
22	148	22	165	22	193	22	210	22	192
23	147	23	152	23	193	23	189	23	192
24	146	24	149	24	184	24	170	24	188
25	147	25	146	25	171	25	156	25	174
26	148	26	148	26	164	26	149	26	170
27	149	27	147	27	160	27	146	27	160
28	153	28	147	28	159	28	146	28	158
29	155	29	150	29	159	29	148	29	158
30	158	30	149	30	157	30	150	30	159

TABLA 4.3 Datos tomados del seguidor de línea de Robotino.

[Fuente: Autoras tesis]

PRUEBA 6		PRUEBA 7		PRUEBA 8		PRUEBA 9		PRUEBA 10	
Nº	x	Nº	x	Nº	x	Nº	x	Nº	x
1	184	1	185	1	169	1	173	1	177
2	180	2	189	2	182	2	193	2	176
3	172	3	193	3	171	3	195	3	179
4	176	4	195	4	157	4	194	4	175
5	162	5	196	5	141	5	176	5	168
6	156	6	194	6	118	6	158	6	162
7	151	7	191	7	104	7	152	7	157
8	141	8	186	8	114	8	129	8	156
9	120	9	182	9	126	9	106	9	140
10	115	10	175	10	147	10	113	10	115
11	102	11	171	11	151	11	91	11	96
12	116	12	167	12	166	12	127	12	120
13	120	13	161	13	183	13	133	13	136
14	129	14	156	14	192	14	142	14	143
15	136	15	147	15	204	15	150	15	148
16	145	16	127	16	210	16	157	16	164
17	149	17	120	17	208	17	170	17	170
18	164	18	118	18	202	18	176	18	178
19	170	19	111	19	198	19	184	19	179
20	178	20	108	20	158	20	179	20	183
21	180	21	115	21	150	21	195	21	203
22	193	22	98	22	148	22	193	22	207
23	209	23	109	23	146	23	182	23	203
24	207	24	119	24	146	24	170	24	188
25	203	25	123	25	147	25	158	25	173
26	191	26	133	26	149	26	160	26	155
27	174	27	149	27	148	27	159	27	145
28	165	28	151	28	154	28	158	28	147
29	150	29	158	29	157	29	159	29	148
30	149	30	167	30	156	30	157	30	149

TABLA 4.4 Datos tomados del seguidor de línea de Robotino.

[Fuente: Autoras tesis]

Estación de Clasificación Repetibilidad:

Nº PRUEBA	CLASIFICAR PIEZAS ROJAS	CLASIFICARPIEZAS NEGRAS
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1

TABLA 4.5 Datos tomados de la estación de clasificación.

[Fuente: Autoras tesis]

Tiempo total de producción y clasificación de dos piezas:

Nº PRUEBA	TIEMPO						
	E. ENSAMBLAJE		TRANSPORTE		E. CLASIFICACION		TOTAL
	1era	2da	ida	vuelta	1era	2da	
1	2,28	2,35	1,18	1,26	0,45	0,5	7,52
2	2,26	3,17	1,17	1,28	0,45	0,5	6,59
3	2,39	2,36	1,11	1,28	0,46	0,51	5,86
4	2,32	2,46	1,14	1,26	0,45	0,51	8,14
5	2,34	2,33	1,15	1,25	0,46	0,49	8,02
6	2,34	2,34	1,12	1,26	0,45	0,5	8,01
7	2,37	2,36	1,13	1,25	0,43	0,51	8,05
8	2,47	2,5	1,11	1,26	0,43	0,51	8,28
9	2,35	2,34	1,15	1,27	0,45	0,5	8,06
10	2,34	2,35	1,17	1,26	0,45	0,51	8,08

TABLA 4.6 Tiempo.

[Fuente: Autoras tesis]

4.3 Análisis e Interpretación de resultados obtenidos.

A continuación se analizara la repetibilidad en las estaciones que conforman el proceso.

La repetibilidad es sinónimo de precisión. A mayor repetibilidad menor dispersión de los valores de salida para un valor de la señal de entrada del proceso y, por lo tanto, mayor precisión.[10]

La fórmula empleada para calcular la repetibilidad es:

$$\sqrt{\frac{\sum(x_i - x)^2}{N}}$$

Estación de Ensamblaje Repetibilidad

PROMEDIO	COGER PIEZA	COLOCAR PIEZA	COGER PISTON	COLOCAR PISTON	COGER RESORTE	COLOCAR RESORTE	COGER TAPA	COLOCAR TAPA	COLOCAR PIEZA EN PALLET
	1	1	1	1	0,8	0,8	1	0,9	1

TABLA 4.7 Resultados de la estación de ensamblaje.

[Fuente: Autoras tesis]

Una vez obtenidos los valores en las mismas condiciones de trabajo se emplea la fórmula de repetibilidad, obteniendo el siguiente resultado:

$$\sqrt{\frac{0.09}{9}} = 0.1\%$$

Por lo tanto el valor obtenido de 0.1% representa una gran precisión de la estación en estudio.

[10]CREUS, Antonio, "Instrumentación Industrial", 7ma edición, pp34-37

Repetibilidad Transporte

PROMEDIO	ACTIVACIÓN DE MOTORES PALLET
	1

TABLA 4.8 Resultados de los motores del pallet.

[Fuente: Autoras tesis]

Empleando la fórmula de repetibilidad obtenemos el siguiente resultado

$$\sqrt{\frac{(1-1)^2}{1}} = 0\%$$

Por lo tanto el 0% nos indica que se ha obtenido una precisión absoluta.

ROBOTINO: Desviación estándar seguidor de línea con cámara web.

PROMEDIO	PRUEBA 1	PRUEBA 2	PRUEBA 3	PRUEBA 4	PRUEBA 5
	160.4	158.3	161.13	161.37	160.53

PROMEDIO	PRUEBA 6	PRUEBA 7	PRUEBA 8	PRUEBA 9	PRUEBA 10
	159.57	153.13	160.07	159.63	161.3

TABLA 4.9 Resultados seguidor de línea de Robotino.

[Fuente: Autoras tesis]

La desviación estándar representa la dispersión de los datos con respecto al punto central. En el caso del estudio del robot móvil tenemos que el punto central es 160, debido a que la cámara web dentro del software (Robotino View) visualiza un rango de 320 a 240 pixeles.

La desviación estándar se la calcula con la siguiente fórmula:

$$s = \sqrt{\frac{\sum_i^n (x_i - \bar{x})^2}{n}}$$

Obteniendo el siguiente resultado:

$$s = \sqrt{\frac{55.74}{10}} = 2.36$$

Con el valor obtenido se concluye que el seguidor de línea con cámara tiene una dispersión de 2.36 con respecto a la media de 160, con lo cual podemos decir que el sistema es bastante preciso.

Estación de Clasificación Repetibilidad:

	CLASIFICAR PIEZAS ROJAS	CLASIFICARPIEZAS NEGRAS
Promedio	1	1

TABLA 4.10 Resultados de la estación de clasificación.

[Fuente: Autoras tesis]

Empleando la formula de repetibilidad obtenemos el siguiente resultado

$$\sqrt{\frac{(1-1)^2}{1}} = 0\%$$

Por lo tanto el 0% nos indica que se ha obtenido una precisión absoluta.

Tiempo Total de Producción y Clasificación de Dos Piezas:

PROMEDIO	TIEMPO						
	E. ENSAMBLAJE		TRANSPORTE		E. CLASIFICACION		TOTAL
	1era	2da	ida	vuelta	1era	2da	
	2,346	2,456	1,142	1,262	0,448	0,504	8,158

TABLA 4.11 Tiempo de producción.

[Fuente: Autoras tesis]

Con el promedio de los datos obtenidos se concluye que el tiempo total de producción y clasificación de dos piezas es aproximadamente 8 minutos y 16 segundos. Por lo cual el tiempo de producción en una hora seria de 15 piezas.

CAPITULO 5

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se implementó un prototipo de control sincronizado del robot móvil Robotino de Festo y los brazos robóticos articulados RV-2AJ de Mitsubishi a través del controlador lógico programable S7200 CPU 222

- Se realizó el monitoreo del prototipo del sistema industrial mediante el software InTouch que permite la visualización del proceso en tiempo real, desde el controlador lógico programable.

- Para el acoplamiento del robot móvil Robotino de Festo y los brazos robóticos articulados RV-2AJ de Mitsubishi en un entorno adecuado fue necesario implementar soluciones mecánicas.

- Para acoplar el proceso industrial realizado se requirió de un entorno muy controlado, debido a que los sistemas usados obtenían diferentes grados de precisión.

- Para lograr el acoplamiento del robot móvil Robotino al sistema completo, fue necesario implementar una comunicación OPC cliente-servidor entre el software Robotino View2 y el PLC a través del EasyPort.

5.2 Recomendaciones

- Para que la librería de OPC funcione en el Robotino se requiere que éste se encuentre actualizado, es decir, se necesita RobotinoView 2.x y la versión 2.0 Ubuntu / Linux para el sistema operativo. Ver Anexo 2.

- Si el sistema implementado va a ser utilizado de forma continua, se recomienda adquirir e instalar las licencias para InTouch debido a que solo permite lectura de un limitado grupo de datos, además es necesario dichas licencias para los IO Servers correspondientes.

- Se recomienda que para realizar trabajos conjuntos entre la robótica fija con la robótica móvil, no solo se requiere del controlador lógico programable, sino también del acoplamiento electrónico y de ajustes mecánicos.

- Si se va a usar de referencia al presente prototipo en una implementación industrial, se recomienda la utilización de equipos robustos en un entorno controlado, además será necesario realizar una comunicación industrial entre los dispositivos de control.

BIBLIOGRAFIA

OLLERO BATURONE Aníbal ROBOTICA Manipuladores y robots móviles, Marcombo,2001

RUSSELL J. Stuart y NORVING Peter INTELIGENCIA ARTIFICIAL UN ENFOQUE MODERNO, segunda edición, Pearson, 2004

BARRIENTOS, Antonio y otros FUNDAMENTOS DE ROBOTICA, Mc Graw Hill 2da edición 2007

Barrientos, A, Peñín, Automatización de la fabricación. Transductores, actuadores y autómatas programables, Sección de Publicaciones ETSI Industriales de Madrid, Universidad Politécnica de Madrid, 1995.

Manual RV_2AJ

Manual Robotino® de Festo

Manual S7-200

PAGINAS WEB:

http://en.wikipedia.org/wiki/Humanoid_robot

http://www.ifr.org/uploads/media/2010_Executive_Summary_rev.pdf

http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm

www.esi2.us.es/~vivas/ayr2iaei/LOC_MOV.pdf

http://sorento4x4ima.iespana.es/datostec/Navegacion.pdf

http://www.definicionabc.com/tecnologia/locomocion.php

http://www.manualvuelo.com/INS/INS29.html

http://www.cienciapopular.com/n/Experimentos/Giroscopios/Giroscopios.php

http://www.sc.ehu.es/sbweb/fisica_/solido/rotacion/giroscopo/giroscopo.xhtml

http://sensoresdeproximidad.galeon.com/

http://www.efdeportes.com/efd9/gps.htm

http://www.alegsa.com.ar/Dic/cmos.php

http://www.alegsa.com.ar/Dic/ccd.php

http://www.digitalfotored.com/imagendigital/sensorcmos.htm

http://www.digitalfotored.com/imagendigital/sensorccd.htm

ANEXOS

Anexo 1

DATOS TÉCNICOS

Robot móvil Robotino de Festo

Parámetro	Valor
Alimentación de tensión	24 V DC, 4.5 A
Entradas digitales	8
Salidas digitales	8
Entradas analógicas	8 (0 – 10 V)
Salidas por relé	2

TABLA A.1 Datos técnicos de Robotino

Motor DC (GR 42x25)	Unidad de medida	
Tensión nominal	V DC	24
Velocidad nominal	RPM	3600
Par nominal	Ncm	3,8
Intensidad nominal	A	0,9
Par de arranque	Ncm	20
Intensidad de arranque	A	4
Velocidad sin carga	RPM	4200
Intensidad sin carga	A	0,17
Intensidad de desmagnetización	A	6,5
Momento de inercia de la masa	gcm^2	71
Peso del motor	gr.	390

TABLA A.2 Datos técnicos motor de Robotino

Reductor planetario (PLG 42 S)	
De una sola etapa, Nm:	3,5
De una sola etapa, i:	4 :1 – 8 :1
2-etapas, Nm:	6
2-etapas, i:	16 :1 – 64 :1
3-etapas, Nm:	14
3-etapas, i:	100 :1 – 512 :1

TABLA A.3 Datos técnicos reductor de Robotino

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

TABLA A.4 Datos técnicos rueda omnidireccional de Robotino

Brazo Robótico RV-2AJ:

Item		Unit	Specifications			
Type			RV-1A	RV-1AC-SB	RV-2AJ	RV-2AJC-SB
Degree of freedom			6		5	
Installation posture			On floor, hanging	On floor	On floor, hanging	On floor
Structure			Vertical, multiple-joint type			
Drive system			ACservo motor (J1 toJ3:50W with brake, J4,J6:15W no brake, J5:15Wwith brake)			
Position detection method			Absolute encoder			
Arm length	Shoulder shift	mm	0			
	Upper arm		250			
	Fore arm		160			
	Elbow shift		90		0	
	Wrist length		72			
Operating range	J1	Degree	300(−150 to +150)			
	J2		180(−60 to +120)			
	J3		95(+60 to +155)		230(−110 to +120)	
	J4		320(−160 to +160)		—	
	J5		180(−90 to +90)			
	J6		400(−200 to +200)			
Speed of motion	J1	Degree/s	180			
	J2		90			
	J3		135			
	J4		180		—	
	J5		180			
	J6		210			
Maximum resultant velocity		mm/s	Approx. 2200		Approx. 2100	
Load	Maximum[Note1]	kg	1.5		2	
	Rating		1		1.5	
Pose repeatability[Note2]		mm	± 0.02			
Ambient temperature		℃	0 to 40			
Mass		kg	Approx. 19		Approx. 17	

Allowable moment load	J4	N・m	1.44		—	
	J5		1.44		2.16	
	J6		0.73		1.10	
Allowable inertia	J4	$kg \cdot m^2$	2.16×10^{-2}		—	
	J5		2.16×10^{-2}		3.24×10^{-2}	
	J6		5.62×10^{-3}		8.43×10^{-3}	
Arm reachable radius (front p-axis center point)		mm	418		410	
Tool wiring[Note3]			Four input signals (Hand section), Four output signals (Base section), Motorized hand output (Hand section)			
Tool pneumatic pipes			Φ4x4 (Base to hand section)	Φ4x3 (Base to hand section)	Φ4x4 (Base to hand section)	Φ4x3 (Base to hand section)
Supply pressure		MPa	0.5 ± 10%			
Protection specification[Note4]			IP30	—	IP30	—
Degree of cleanliness[Note5]			—	100(0.3 μm)	—	100(0.3 μm)
Paint color			Light gray (Equivalent to Munsell: 7.65Y7.6/0.73)			

TABLA A.5 Datos técnicos brazo robótico RV-2AJ

PLC S7-200 CPU 222:

	CPU 221	CPU 222	CPU 224	CPU 224XP CPU 224XPsi	CPU 226
Memoria					
Tamaño del programa de usuario con edición en runtime sin edición en runtime	4096 bytes 4096 bytes		8192 bytes 12288 bytes	12288 bytes 16384 bytes	16384 bytes 24576 bytes
Datos de usuario	2048 bytes		8192 bytes	10240 bytes	10240 bytes
Respaldo (condensador de alto rendimiento) (pila opcional)	Típ. 50 h (mín. 8 h a 40°C) Típ. 200 días		Típ. 100 h (mín. 70 h a 40°C) Típ. 200 días	Típ. 100 h (mín. 70 h a 40°C) Típ. 200 días	
E/S					
E/S digitales	6 E/4 S	8 E/6 S	14 E/10 S	14 E/10 S	24 E/16 S
E/S analógicas	Ninguna			2 E/1 S	Ninguna
Tamaño de la imagen de E/S digitales	256 (128 E/128 S)				
Tamaño de la imagen de E/S analógicas	Ninguno	32 (16 E/16 S)	64 (32 E/32 S)		
Nº máx. de módulos de ampliación	Ninguno	2 módulos[1]	7 módulos[1]		
Nº máx. de módulos inteligentes	Ninguno	2 módulos[1]	7 módulos[1]		
Entradas de captura de impulsos	6	8	14		24
Contadores rápidos Fase simple Dos fases	4 contadores en total 4 a 30 kHz 2 a 20 kHz		6 contadores en total 6 a 30 kHz 4 a 20 kHz	6 contadores en total 4 a 30 kHz 2 a 200 kHz 3 a 20 kHz 1 a 100 kHz	6 contadores en total 6 a 30 kHz 4 a 20 kHz
Salidas de impulsos	2 a 20 kHz (sólo en salidas DC)			2 a 100 kHz (sólo en salidas DC)	2 a 20 kHz (sólo en salidas DC)
Datos generales					
Temporizadores	256 temporizadores en total: 4 temporizadores de 1 ms, 16 temporizadores de 10 ms y 236 temporizadores de 100 ms				
Contadores	256 (respaldo por condensador de alto rendimiento o pila)				
Marcas internas almacenadas al desconectar la CPU	256 (respaldo por condensador de alto rendimiento o pila) 112 (almacenamiento en EEPROM)				
Interrupciones temporizadas	2 con resolución de 1 ms				
Interrupciones de flanco	4 flancos positivos y/o 4 flancos negativos				
Potenciómetros analógicos	1 con resolución de 8 bits		2 con resolución de 8 bits		
Velocidad de ejecución booleana	0.22 µs por operación				
Reloj de tiempo real	Cartucho opcional		Incorporado		
Cartuchos opcionales	Memoria, pila y reloj de tiempo real		Memoria y pila		
Comunicación integrada					
Puertos (potencia limitada)	1 puerto RS-485			2 puertos RS-485	
Velocidades de transferencia PPI, MPI (esclavo)	9,6, 19,2 y 187,5 kbit/s				
Velocidades de transferencia Freeport	1.2 kbit/s a 115.2 kbit/s				
Longitud máx. del cable por segmento	Con repetidor aislado: 1000 m hasta 187,5 kbit/s, 1200 m hasta 38,4 kbit/s Sin repetidor aislado: 50 m				
Nº máximo de estaciones	32 por segmento, 126 por red				
Nº máximo de maestros	32				
Punto a punto (modo maestro PPI)	Sí (NETR/NETW)				
Enlaces MPI	4 en total, 2 reservados (1 para una PG y 1 para un OP)				

TABLA A.6 Datos técnicos generales de la CPU del PLC S7-200

Tabla A-5 Datos de alimentación de las CPUs

	DC		AC	
Potencia de entrada				
Tensión de entrada	20,4 a 28,8 VDC		85 V a 264 VAC (47 a 63 Hz)	
Intensidad de entrada CPU 221 CPU 222 CPU 224 CPU 224XP CPU 224XPsi CPU 226	CPU sólo a 24 VDC 80 mA 85 mA 110 mA 120 mA 120 mA 150 mA	Carga máx. a 24 VDC 450 mA 500 mA 700 mA 900 mA 900 mA 1050 mA	sólo CPU 30/15 mA a 120/240 VAC 40/20 mA a 120/240 VAC 60/30 mA a 120/240 VAC 70/35 mA a 120/240 VAC - 80/40 mA a 120/240 VAC	Carga máx. 120/60 mA a 120/240 VAC 140/70 mA a 120/240 VAC 200/100 mA a 120/240 VAC 220/100 mA a 120/240 VAC - 320/160 mA a 120/240 VAC
Corriente de irrupción	12 A a 28,8 VDC		20 A a 264 VAC	
Aislamiento (campo a circuito lógico)	Sin aislamiento		1500 VAC	
Tiempo de retardo (desde la pérdida de corriente)	10 ms a 24 VDC		20/80 ms a 120/240 VAC	
Fusible (no reemplazable)	3 A, 250 V, de acción lenta		2 A, 250 V, de acción lenta	
Alimentación de sensores 24 VDC				
Tensión de sensores (potencia limitada)	L+ menos 5 V		20,4 a 28,8 VDC	
Intensidad límite	1,5 A pico, límite térmico no destructivo (v. tabla A-3, carga nominal)			
Rizado/corriente parásita	Derivado de potencia de entrada		Menos de 1 V pico a pico	
Aislamiento (sensor a circuito lógico)	Sin aislamiento			

TABLA A.7 Datos de alimentación de las CPU

EasyPort USB

Interface de procesos EasyPort USB D16A	
Tensión de funcionamiento	24 VDC +/–10%
Consumo	3 VA
Cantidad de salidas	16 digitales de 24 VDC, 2 analógicas de 0...10 VDC ó –10...+10 VDC, resolución de 12 bit
Carga admisible	0,7 A por salida digital 10 mA por salida analógica
Protección contra cortocircuitos	Sí
Cantidad de entradas	16 digitales de 24 VDC, 4 analógicas de 0...10 VDC ó –10...+10 VDC, resolución de 12 bit
Umbral de conmutación de las entradas digitales	12 VDC
Histéresis de las entradas digitales	3 V
Filtro	5 ms
Cantidad de contadores	2 entradas utilizables como contadores rápidos, $f_{máx}$ = 20 kHz/U_{in} = 5...24 VDC
Interfaces de comunicación	RS232, con separación galvánica USB2.0, con separación galvánica
Protocolo	ASCII, 115,2 kBaud,8,N,1
Clase de protección	IP20
Símbolo CE	Según directiva UE/CEM
Temperatura ambiente admisible para funcionamiento/almace-namiento	0...55 °C/0...70 °C
Medidas en mm (largo x ancho x alto)	135 x 167 x 37
Peso en kg	0,65

TABLA A.8 Datos técnicos de EasyPort USB

Anexo 2

ACTUALIZACION Robotino

El controlador de Robotino® consiste en un PC embebido con una tarjeta compact flash, en la cual se han instalado varias aplicaciones de demostración y el sistema operativo (Linux). Las aplicaciones de demostración pueden ejecutarse directamente desde el panel de control del Robotino®.

Robotino® puede programarse con el software Robotino®View en un PC a través del LAN inalámbrico. Robotino®View es capaz de transmitir señales al controlador del motor, así como visualizar, cambiar y evaluar valores de los sensores. Robotino® puede programarse con Robotino®View incluso durante el funcionamiento real.

Para realizar las actualizaciones del sistema operativo es necesario describir a “dd” puesto que este término será utilizado para la ejecución de esta tarea.

ddes un programa común de Unix cuya finalidad principal es el bajo nivel de la copia y conversión de datos en bruto. dd es una aplicación que va a "convertir y copiar un archivo" . "DD" es sinónimo de descripción de datos, se utiliza para copiar un número especificado de bytes o bloques, realizando sobre la marcha de las conversiones de orden de bytes, dd también se puede utilizar para copiar las regiones de los archivos de dispositivos básicos, por ejemplo, copia de seguridad del sector de arranque de un disco duro, o para leer cantidades fijas de datos de archivos especiales como / dev / zero o / dev / random.

Instalación de la versión 2.0 para el sistema operativo del Robotino. Ubuntu / Linux

1. Extraer la tarjeta de memoria del Robotino (memoria de 1GB)

2. Verificar la versión actual instalada en la memoria.

3. Descargar el software para la versión 2.0 para lo cual se debe realizar lo siguiente:

4. Descargar "dd para Windows" dehttp://www.chrysocome.net/dd

5. Cambiar el nombre dedd.exeadd-removable.exe. Esto asegurará que no haya un daño en el disco duro. El ejecutable dd-removable.exe sólo se debe escribir en dispositivos USB.

6. Leer detenidamente las instrucciones para la actualización del sistema operativo del robot.

7. Conectar la tarjeta de memoria del robot, al conector adecuado del computador.

8. Verificar que no exista otro disco extraíble dentro del computador.

9. Tomar en cuenta que el software descargado es compatible con Windows XP.

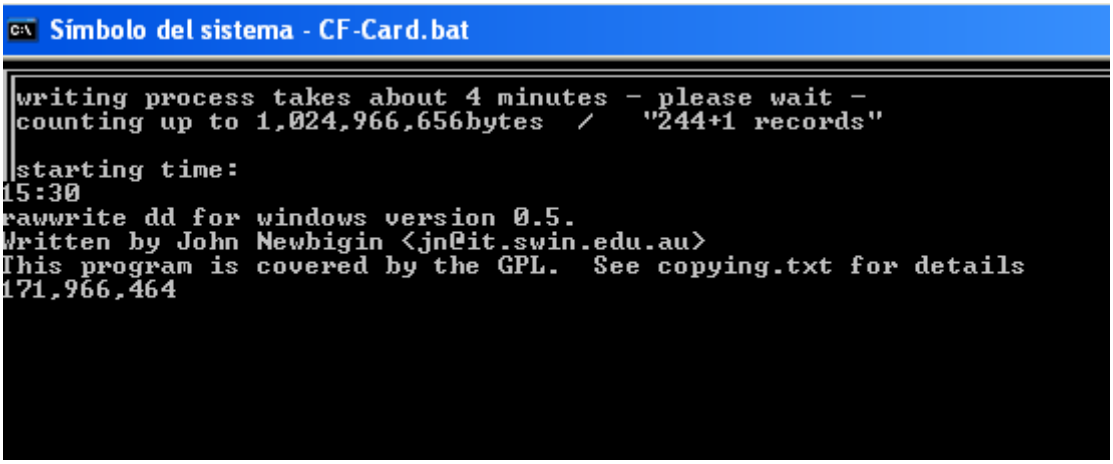
10. Ingresar al sistema del computador a través de DOS.

11. Explorar la carpeta contenedora del software de actualización.

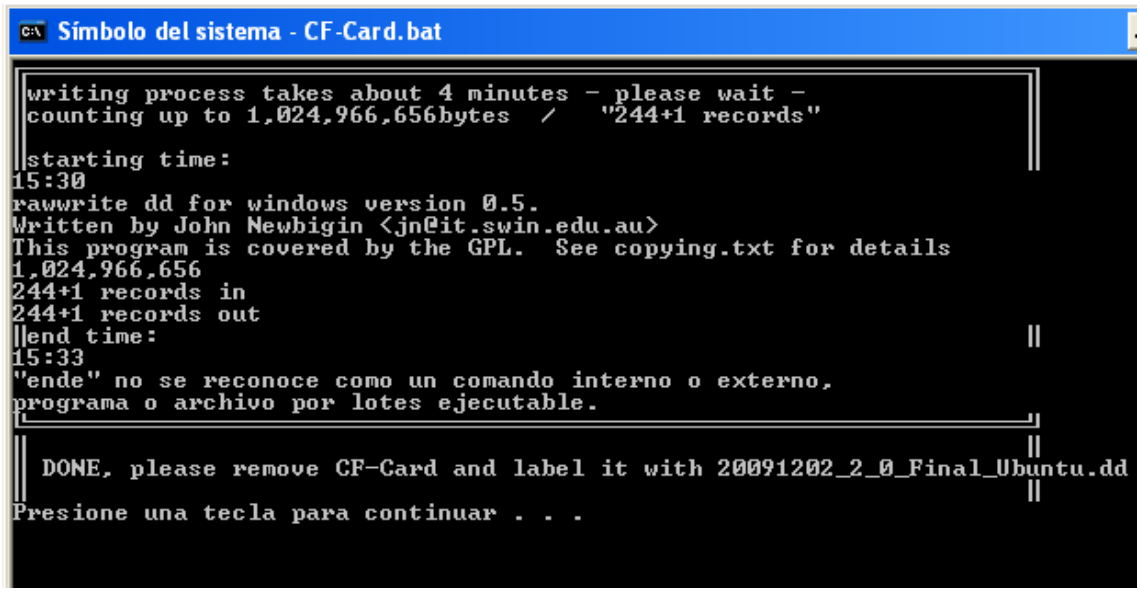
12. Modificar la línea número 3 del programa. Esta línea hace referencia al número asignado para el disco extraíble. Para obtener este dato fue necesario ingresar al administrador de hardware y reconocer el número del dispositivo.

13. Una vez cambiada la línea antes mencionada, se procede a ejecutar el programa. El cual se demora 4 minutos en instalar la versión actualizada del sistema operativo del Robotino. Mientras sucede la instalación. Se

puede observar las tareas que se están realizando para completar la actualización.



14. Al completarse la instalación, aparece un mensaje de finalización y de retirar la tarjeta de memoria del computador.



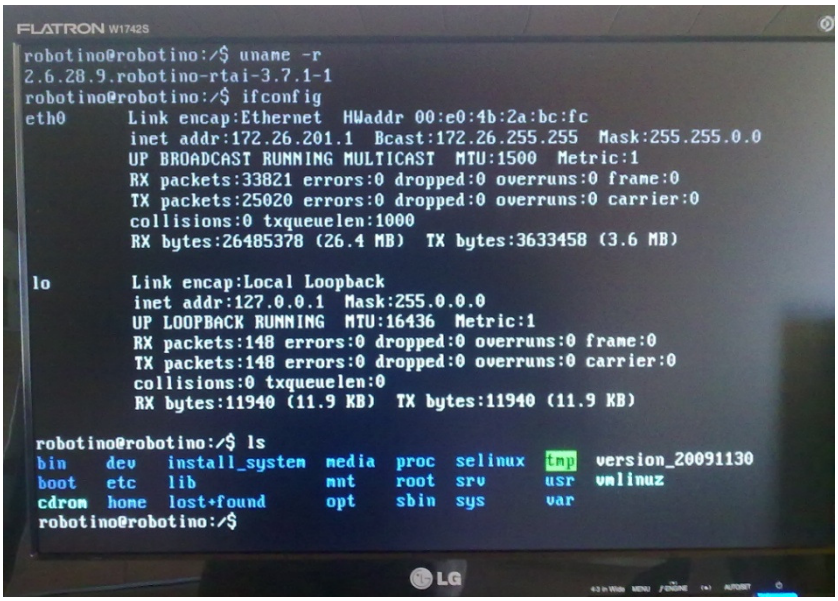
15. Ahora la tarjeta de memora ya tiene la versión actualizada 2.0, por lo tanto se procede a conectar la tarjeta en el Robotino para comprobar que el proceso realizado se cumplió exitosamente.

16. El siguiente paso es encender al Robotino, aquí tardará unos segundos hasta mostrar en la pantalla los datos, esto se da debido a que el Robot necesita un tiempo para asimilar las actualizaciones realizadas en su Sistema Operativo.

17. Seguidamente en la pantalla del robot se muestran los datos del Robotino y aparece la versión actual instalada 2.0.

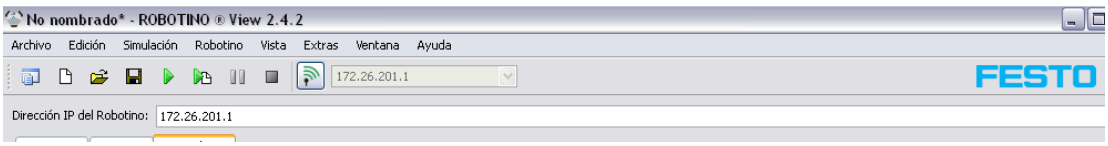
18. Al revisar el menú del Robotino se logró verificar ciertos cambios entre la versión anterior y la actual.

19. Además se conectó al robot un teclado, mouse y monitor de un computador para observar de forma directa desde la tarjeta de memoria lo que el sistema operativo posee y sus características.

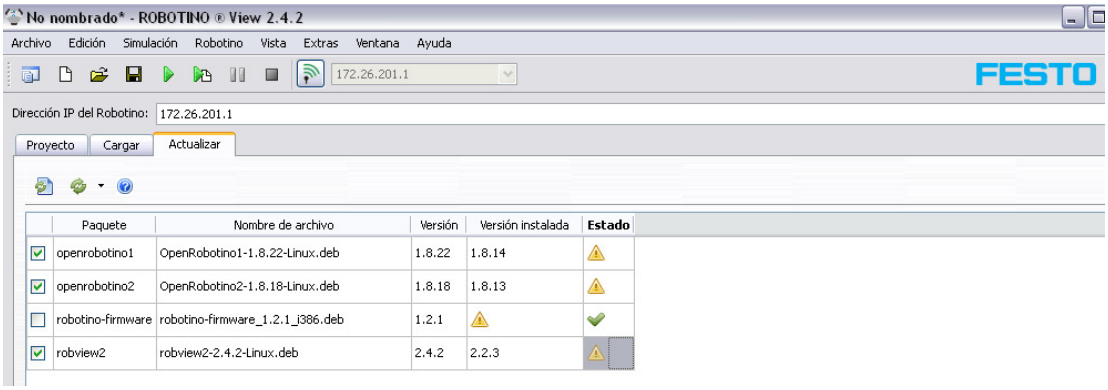


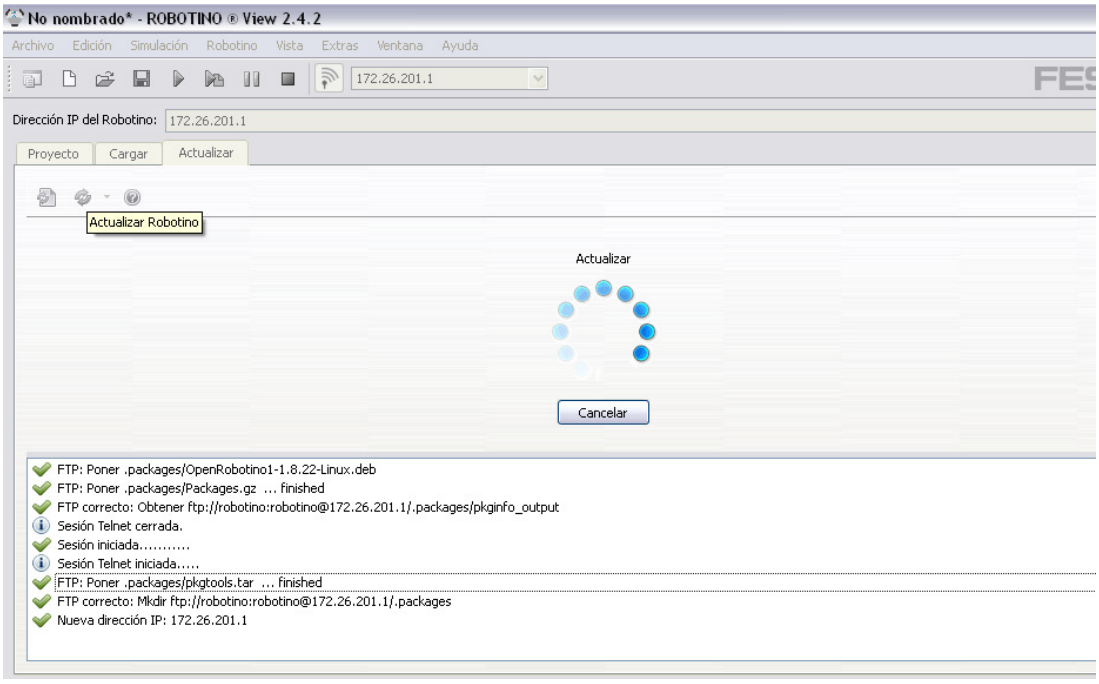
Para verificar el correcto funcionamiento se realizó la comunicación inalámbrica entre una computadora y el Robotino, a través del software RobotinoView 2.

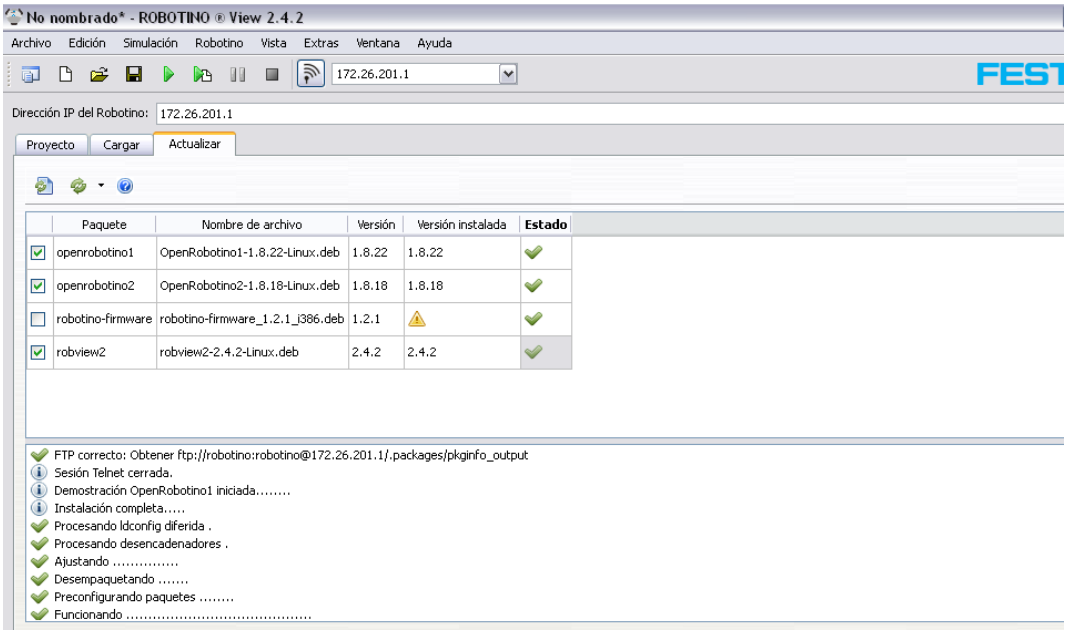
La comunicación es exitosa.



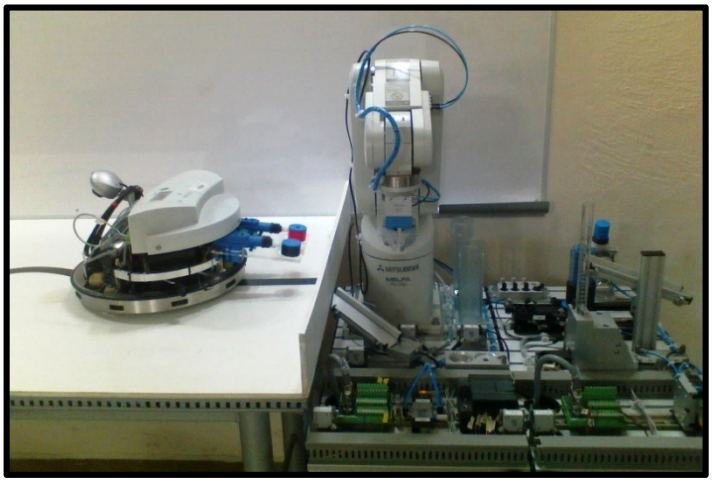
Además de las actualizaciones en el Sistema operativo de la tarjeta del robot, se pueden apreciar actualizaciones dentro del software que controla el robot, una de ellas, es la posibilidad de reajustar el software mencionado y otras aplicaciones propias del programa, para lo cual se deben seleccionar las aplicaciones que necesiten la actualización y hacer clic en "Actualizar". En pocos minutos se configura y se obtiene un software completo con acceso a todas las aplicaciones que posee el Robotino.

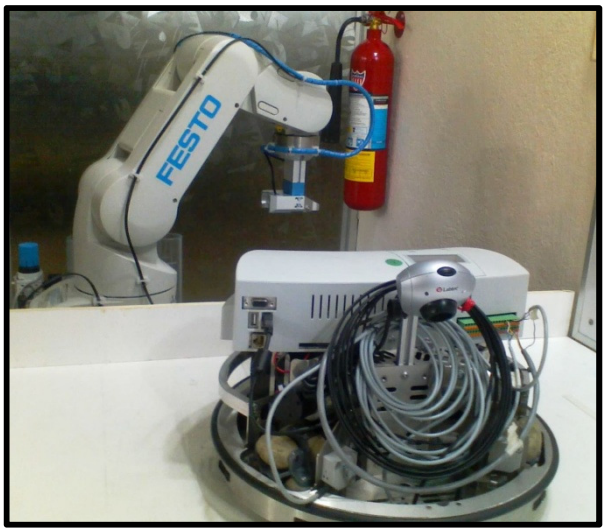

No nombrado* - ROBOTINO ® View 2.4.2
Archivo Edición Simulación Robotino Vista Extras Ventana Ayuda
172.26.201.1
FESTO
Dirección IP del Robotino: 172.26.201.1
Proyecto Cargar Actualizar
Paquete Nombre de archivo Versión Versión instalada Estado
openrobotino1 OpenRobotino1-1.8.22-Linux.deb 1.8.22 1.8.14
openrobotino2 OpenRobotino2-1.8.18-Linux.deb 1.8.18 1.8.13
robotino-firmware robotino-firmware_1.2.1_i386.deb 1.2.1
robview2 robview2-2.4.2-Linux.deb 2.4.2 2.2.3


No nombrado* - ROBOTINO ® View 2.4.2
Archivo Edición Simulación Robotino Vista Extras Ventana Ayuda
172.26.201.1
Dirección IP del Robotino: 172.26.201.1
Proyecto Cargar Actualizar
Actualizar Robotino
Actualizar
Cancelar
FTP: Poner .packages/OpenRobotino1-1.8.22-Linux.deb
FTP: Poner .packages/Packages.gz ... finished
FTP correcto: Obtener ftp://robotino:robotino@172.26.201.1/.packages/pkginfo_output
Sesión Telnet cerrada.
Sesión iniciada.....
Sesión Telnet iniciada.....
FTP: Poner .packages/pkgtools.tar ... finished
FTP correcto: Mkdir ftp://robotino:robotino@172.26.201.1/.packages
Nueva dirección IP: 172.26.201.1


No nombrado* - ROBOTINO ® View 2.4.2
Archivo Edición Simulación Robotino Vista Extras Ventana Ayuda
172.26.201.1
Dirección IP del Robotino: 172.26.201.1
Proyecto Cargar Actualizar
Paquete Nombre de archivo Versión Versión instalada Estado
openrobotino1 OpenRobotino1-1.8.22-Linux.deb 1.8.22 1.8.22
openrobotino2 OpenRobotino2-1.8.18-Linux.deb 1.8.18 1.8.18
robotino-firmware robotino-firmware_1.2.1_i386.deb 1.2.1
robview2 robview2-2.4.2-Linux.deb 2.4.2 2.4.2
FTP correcto: Obtener ftp://robotino:robotino@172.26.201.1/.packages/pkginfo_output
Sesión Telnet cerrada.
Demostración OpenRobotino1 iniciada.....
Instalación completa.....
Procesando ldconfig diferida .
Procesando desencadenadores .
Ajustando .....
Desempaquetando .....
Preconfigurando paquetes .....
Funcionando .....

Anexo 3





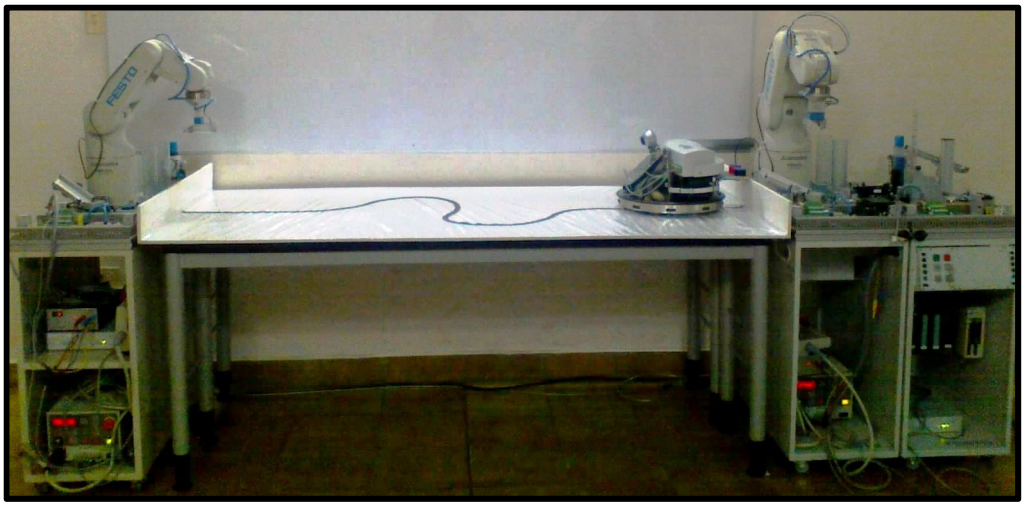


FIGURA A3.1 Implementación del sistema industrial

[Fuente: Autoras Tesis]