



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERIA ELECTRÍCA
DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA



AUTOMATIZACIÓN DE UNA LÍNEA DE IMPRESIÓN
Y EL PROCESO DE EMBALAJE DE UNA EMPRESA PAPELERA
MEDIANTE UN DISPOSITIVO CONTROLADOR Y UN ROBOT INDUSTRIAL

AUTORES:
DANIEL VALENTINER
VICTOR AGUIAR

TUTOR:
WILMER SANZ

VALENCIA, NOVIEMBRE 2012



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERIA ELECTRÍCA
DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA



AUTOMATIZACIÓN DE UNA LÍNEA DE IMPRESIÓN
Y EL PROCESO DE EMBALAJE DE UNA EMPRESA PAPELERA
MEDIANTE UN DISPOSITIVO CONTROLADOR Y UN ROBOT INDUSTRIAL

TRABAJO ESPECIAL DE GRADO PARA OPTAR POR EL TÍTULO DE INGENIERO
ELECTRICISTA

AUTORES:
DANIEL VALENTINER
VICTOR AGUIAR

TUTOR:
WILMER SANZ

VALENCIA, NOVIEMBRE 2012



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERIA ELECTRÍCA
DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA



CERTIFICADO DE APROBACIÓN

Los abajo firmantes, miembros del jurado designado para estudiar el Proyecto Especial de Grado titulado: “AUTOMATIZACIÓN DE UNA LÍNEA DE IMPRESIÓN Y EL PROCESO DE EMBALAJE DE UNA EMPRESA PAPELERA MEDIANTE UN DISPOSITIVO CONTROLADOR Y UN ROBOT INDUSTRIAL”, realizado por los bachilleres: Daniel Alexander Valentiner Márquez, portador de la cédula de identidad V-19.219.223 y Víctor Ulises Aguiar Tirado, portador de la cédula de identidad V-19.505.146, hacemos constar que hemos revisado y aprobado dicho Proyecto Especial.

Prof. Wilmer Sanz

Prof. Aida Pérez

Prof. Whendy García

Valencia, Noviembre de 2012



Dedicatoria

A mi padre Tulio, quien es mi ejemplo, mi modelo a seguir.

A mi madre Sandra, que me ha brindado su amor y apoyo siempre.

A mi hermana Irene, quien siempre me apoyo y espero que este logro sea un ejemplo de que las metas se alcanzan con perseverancia, apoyo y oración

A María Graciela mi compañera, mi pareja, que con su apoyo me acompañó en tiempos difíciles

A la Virgen de Coromoto, a quien encomendé mis estudios desde el primer día que comenzaron.

A mi abuela que partió antes de verme graduado, pero espero que desde el cielo se alegre de este logro alcanzado por su primer nieto.

A mi abuela Odila por sus constantes oraciones.

Daniel Valentiner



Agradecimientos

A Dios por darme la vida, por bendecirme con muchas personas que me aman y me apoyan. Por ser mi fuerza en momentos difíciles y mi alegría en los momentos de gozo.

A mi padre Tulio, su ayuda y apoyo fue invaluable para el desarrollo de este trabajo, a demás de ser mi ejemplo de perseverancia y mi modelo a seguir.

A mi madre Sandra y a mi hermana Irene que siempre me brindaron amor y mucho apoyo en tiempos difíciles.

A mis familiares y buenos amigos que siempre me brindaron su apoyo.

Al profesor Wilmer Sanz por su guía y su ayuda, además de ser un ejemplo como profesional y persona.

A Víctor mi compañero de tesis por la dedicación y esfuerzo en la elaboración de este trabajo.

A Lourdes Fortoul por su colaboración y soporte.

A todo el personal de FESTO por su ayuda. Especialmente a Melanie Loaiza que me apoyo y ayudo de manera incondicional siempre con una sonrisa.

A Enrique Basauri por su continuo apoyo.

A todo el personal de DAFOINCA especialmente a Héctor Gutiérrez y a Miguel Alcalde

A Luis Guada por su ayuda.

Daniel Valentiner



Dedicatoria

Le dedico este trabajo a todos los seres que amo profundamente, y son fuente de inspiración para mí, entre ellos destaco:

Primeramente a Dios mi Señor, a quien dedico y ofrezco cada logro de mi vida, siendo este uno de ellos. ¡A ti te debo el honor y la Gloria!

A mis Padres, personas de infinita dedicación, constancia, y que me han brindado un afecto incondicional.

A todos mis familiares, específicamente a mis queridos hermanos Henry, Johan; a Oscar, que aunque tuvo que partir temprano, sé que desde el cielo mira con alegría la culminación de este trabajo. También a mi segunda madre, Liby, consejera siempre desde el aspecto profesional hasta el personal.

A Saraíz, la persona que El Señor dispuso para conocerlo a Él, y que se ha convertido en mi amiga, confidente, y pareja.

A mis compañeros de la carrera y amigos, como una muestra de que con fe, y mucho esfuerzo, es posible alcanzar las metas propuestas.

Víctor Aguiar



Agradecimientos

A Dios Padre, quien dispuso desde antes de que me creara, que fuera por medio de esta carrera que lo llegase a conocer, y que preparó cada camino que he recorrido y recorreré para darme cuenta de cuánto me ama, y se desvive por mí, y de todo lo bueno que me tiene preparado. Todo lo que ha dispuesto en mi vida ha sido para bien (Rm 8:28). En este caso, este Proyecto de Grado, que me ha brindado un importante crecimiento como profesional.

A Jesucristo, quien me ha dado la fuerza para llevar a cabo cada tarea en mi vida, y que nunca se ha alejado de mí, siempre dándome su aliento y amor. Aprovecho para confesarte por aquí que, aún con mis debilidades, ¡Te amo Señor!

Al Espíritu Santo, quien me ha brindado entendimiento y sabiduría en cada reto del día a día, especialmente durante este trabajo. Le agradezco la guía que me ha dado en este trabajo, pero por sobre todas las cosas, el haberme hecho conocer a Jesús sacramentado en la Santa Eucaristía.

A mi compañero Daniel, por su esfuerzo y disciplina, que fueron claves para poder llevar a cabo juntos este trabajo, y su amistad sincera, que hicieron de este un proyecto ameno de hacer.

Al profesor Wilmer, quien brindó el soporte en cada aspecto del proyecto, con la mejor disposición, siempre con un espíritu humilde y servicial.

A mis padres, Henry y Meiler, quienes con su providencia me permitieron dedicarme plenamente a este trabajo, y con su apoyo me motivaron a esforzarme en este proyecto.

Al Señor Tulio, quien nos brindó una valiosa ayuda durante la elaboración de todo este trabajo.

A la Universidad de Carabobo, específicamente a la Escuela de Ingeniería Eléctrica, conformada por profesores dedicados a formar a los futuros profesionales.

Víctor Aguiar



Índice

	Pág.
ÍNDICE DE FIGURAS.....	xv
ÍNDICE DE TABLAS.....	xxiv
RESUMEN.....	xxv
INTRODUCCIÓN.....	27
CAPÍTULO I.....	29
1.1 Planteamiento del problema.....	29
1.2 Justificación	31
1.3 Objetivos	32
1.3.1 Objetivo general.....	32
1.3.2 Objetivos específicos.....	32
1.4 Alcance.....	33
CAPÍTULO II.	35
2.1 Antecedentes.....	35
2.2 Bases teóricas.....	37
2.2.1 Microprocesador.....	37
2.2.2 Microcontrolador.....	39



	Pág.
2.2.3 PIC.....	46
2.2.4 Microcontrolador PICmicro 6F877.....	48
2.2.5 Programación del PIC.....	66
2.2.6 Neumática.....	68
2.2.7 Señales.....	78
2.2.8 Robots	82
CAPÍTULO III.....	127
3.1 Metodología.....	127
3.1.1 Procedimiento para la recolección de información.....	128
3.1.2 Técnicas de procesamiento y análisis de información.....	130
3.1.3 Fases.....	130
CAPÍTULO IV.....	133
4.1 Estudio del proceso de impresión y embalaje	133
4.1.1 Recopilación de información sobre las características de operación de la máquina rotativa de impresión.....	133
4.1.2 Recopilación de información del proceso de impresión.....	138
4.1.3 Recopilación de información del proceso de embalaje.....	140
4.1.4 Identificación de las normas de seguridad que se deben seguir en el departamento de embalaje	141



	Pág.
4.2 Análisis del control de los procesos de impresión y embalaje.....	142
4.2.1. Identificación de los sistemas de control u operación manual en los procesos de impresión y embalaje	143
4.2.2 Determinación de si los procesos o controles identificados perjudican la producción y/o productividad	144
4.3 Diseño y prueba de un dispositivo controlador-registrador para la automatización la máquina de impresión	146
4.3.1 Selección de un micro-controlador y periféricos adecuados para su aplicación en la automatización en la máquina de impresión	148
4.3.2 Desarrollo de un programa que permita el funcionamiento del dispositivo.....	151
4.3.3 Desarrollo de una aplicación para computadora que permita extraer la información registrada por el dispositivo.....	157
4.3.4 Simulación por software del dispositivo en operación.....	163
4.3.5 Construcción de un prototipo del dispositivo para probar el correcto funcionamiento del mismo.....	177
4.4 Desarrollo de un sistema de control automatizado para el proceso de embalaje, mediante un Robot Industrial.....	187
4.4.1 Selección de un robot adecuado para la automatización del proceso de embalaje.....	187
4.4.2 Diseño de una herramienta de agarre del robot que se adecue a las características de los paquetes.....	189
4.4.3 Desarrollo del programa para la ejecución automatizada del proceso de embalaje a utilizar en el robot.....	201



	Pág.
4.4.4 Prueba de funcionamiento mediante la utilización de un Robot Industrial a escala fuera del proceso.....	204
CAPÍTULO V.....	221
5.1 Conclusiones.....	221
5.2 Recomendaciones.....	223
BIBLIOGRAFÍA Y REFERENCIAS	225
PRESENTACIÓN DE APÉNDICES.....	231
Apéndice A. Programa PIC.....	233
Apéndice B. Diagrama de flujo para el programa del PIC.....	245
Apéndice C. Programa de la aplicación para computadora. Interfaz PIC-PC	247
Apéndice D. Diagrama de flujo para el programa del Robot RV2AJ.....	255
Apéndice E. Flujo para subrutina de interrupción del programa del Robot...	259
Apéndice F. Programa de embalado de robot. Formato 9 ½ x 11.....	261
Apéndice G. Programa de embalado paquetes 9 ½ x 5 ½.....	265
Apéndice H. Código del Subprograma del robot.....	271



Índice de Figuras

Figura		Pág.
2.1	Conexión de un Microprocesador.....	38
2.2	Arquitectura Von-Neumann.....	40
2.3	Arquitectura Harvard.....	41
2.4	Temporizadores.....	42
2.5	Esquema de Convertidor A/D.....	43
2.6	Función del Convertidor A/D.....	44
2.7	Arquitectura de un PIC.....	47
2.8	Estructura interna del PICmicro 18F4520.....	49
2.9	Diagrama de patas (Pin-out) del PIC 18F4520.....	50
2.10	Esquema de memoria EEPROM.....	51
2.11	Estructura interna del PIC 18F4520.....	52
2.12	Funcionamiento del PIC 18F4520.....	53
2.13	Pulsador modelo SW762.....	53
2.14	Conexión Activa baja con resistencia de Pull Up.....	54
2.15	Conexión Activa alta con resistencia de Pull Up.....	54
2.16	Pantalla LCD 2x16.....	56



Figura	Pág.
2.17 Conexión de Pantalla LCD.....	56
2.18 Polaridad y Conexión del LED.....	57
2.19 Modelos de Relés.....	58
2.20 Conexión segura del Relé.....	59
2.21 Conexión RS232 con PIC.....	63
2.22 Cables de USB.....	64
2.23 Circuito de operación del DS1307.....	65
2.24 Configuración de los pines DS1307.....	66
2.25 Entorno del Proton Basic	67
2.26 Entorno Proteus.....	68
2.27 Esquema de válvula de escape rápido	71
2.28 Diagrama de válvula de escape rápido	71
2.29 Válvula 3/2 Vías.....	72
2.30 Cilindro de simple efecto	73
2.31 Cilindro de doble efecto	74
2.32 Generador de Vacío	75
2.33 Ventosa.....	76



Figura	Pág.
2.34 Sistema de FRL.....	78
2.35 Simbología de la unidad de mantenimiento.....	78
2.36 Disco de encoder	81
2.37 Motor de CC con encoder en ranuras.....	82
2.38 Morfología del Robot.....	83
2.39 Estructura del manipulador	84
2.40 Conexión de computador con actuadores y teach pendant.....	85
2.41 Teach Pendant.....	86
2.42 Ejemplo de robot Cartesiano	88
2.43 Ejemplo de robot Cilíndrico	89
2.44 Estructura de robot Esférico	90
2.45 Robot de brazo articulado	91
2.46 Robot SCARA.....	93
2.47 Parámetros a y α	94
2.48 Robot Antropomórfico de 5 ejes RV-2AJ Mitsubishi-FESTO.....	99
2.49 Sistema tractor-remolque.....	107



Figura		Pág.
2.50	Maniobras secuenciales para la reorientación del remolque.....	110
2.51	Maniobra única de reorientación.....	111
2.52	Maniobra de reorientación con trayectorias canónicas.....	113
2.53	Maniobra restringida para reorientación.....	115
2.54	Sistema Planificador.....	116
2.55	Reorientación basada en modelo encadenado con distinto a_1	118
2.56	Reorientación basada en trayectoria canónica diferente punto intermedio	119
2.57	Reorientación mediante maniobra restringida.....	120
2.58	Robot paletizador.....	122
Figuras Capítulo IV		
4.1	Cuerpo de alimentación.....	135
4.2	Módulos de impresión.....	135
4.3	Módulo de Impresión offset.....	136
4.4	Módulo de Control.....	137
4.5	Instalación de bobina de papel en cuerpo de alimentación.....	139
4.6	Operador manipulando controles de arranque y parada de máquinas ...	139
4.7	Transferencia de rollos a la máquina empacadora.....	140



Figura	Pág.
4.8 Embalaje manual.....	141
4.9 Normas de seguridad e higiene ambiental.....	142
4.10 Detención de máquina empacadora por sobresaturación de bandeja...	144
4.11 Diagrama de bloques controlador-registrador.....	147
4.12 PIC18F4520.....	149
4.13 Pantalla LCD.....	150
4.14 Chip calendario.....	150
4.15 Pantalla de inicio Proton.....	151
4.16 Librería.....	152
4.17 Oscilador PIC.....	152
4.18 Configuración comunicación serial.....	153
4.19 LCD PIC.....	153
4.20 Puerto PIC.....	154
4.21 Variable 1 PIC.....	154
4.22 Variable 2 PIC.....	155
4.23 Variable 3 PIC.....	155
4.24 Constantes PIC.....	156



Fig.		Pág.
4.25	Selección del lenguaje Basic.....	157
4.26	Cuadro de herramientas	158
4.27	Características del objeto.....	159
4.28	Código de recibir	160
4.29	Variables globales.....	161
4.30	Controles especiales.....	162
4.31	Interfaz PIC-PC.....	162
4.32	Diagrama de bloques controlador-registrador con componentes.....	164
4.33	Búsqueda de dispositivos.....	165
4.34	Circuito para simulación	166
4.35	Circuito identificado controlador-registrador interfaz.....	167
4.36	Circuito identificado controlador-registrador comunicación.....	168
4.37	Pantalla principal controlador-registrador.....	168
4.38	Indicación de metros.....	169
4.39	Pantalla de metros.....	169
4.40	Confirmación de metros.....	170
4.41	Pantalla de impresión.....	170



Fig.		Pág.
4.42	Impresión lista.....	171
4.43	Pantalla de conexión.....	172
4.44	Conexión Transferencia.....	173
4.45	Indicador de conexión transferencia.....	173
4.46	Guardado de archivo	174
4.47	Visualización de archivo	175
4.48	Visualización de archivo (memoria borrada).....	175
4.49	Visualización de archivo LCD1.....	176
4.50	Esquemático final.....	177
4.51	Protoboard.....	179
4.52	Tarjeta del prototipo.....	180
4.53	Placa sin soldar.....	181
4.54	Prototipo terminado.....	182
4.55	Sincronización con dispositivo.....	183
4.56	Prototipo de impresión.....	184
4.57	Datos extraídos del prototipo.....	185
4.58	Borrado de memoria del prototipo.....	186



Fig.		Pág.
4.59	RV-12SD.....	187
4.60	Paquetes del formato 9 ½” x 11”	189
4.61	Programa de vacío paquete.....	191
4.62	Programa de vacío distribución.....	192
4.63	Programa de vacío ventosa.....	193
4.64	Programa de vacío tubería.....	193
4.65	Programa de vacío generador.....	194
4.66	Programa de vacío brazo.....	195
4.67	Programa de vacío simulación.....	195
4.68	Programa de vacío resultado.....	196
4.69	Vista AutoCAD.....	197
4.70	Herramienta isometría identificación de partes.....	198
4.71	Herramienta ubicación tubos neumáticos.....	199
4.72	RV-12SV neumática interna.....	200
4.73	Vistas de herramienta.....	201
4.74	RV-2AJ.....	204
4.75	Sensor de paquetes.....	206



Fig.		Pág.
4.76	Sensor de cajas.....	206
4.77	Cilindro neumático.....	207
4.78	Válvula electro-neumática.....	208
4.79	Panel de pulsadores.....	209
4.80	Válvula de escape rápido.....	210
4.81	Herramienta RV-2AJ.....	211
4.82	Piezas de agarre.....	212
4.83	Paquete de prueba.....	212
4.84	Dos paquetes de prueba.....	213
4.85	Caja de prueba.....	214
4.86	Guías metálicas.....	215
4.87	Prueba inicial.....	216
4.88	Módulo de entradas y salidas.....	216
4.89	Prueba de sujeción de paquete.....	217
4.90	Prueba soltar paquete.....	218
4.91	Prueba caja llena.....	219



Índice de Tablas

Tabla.		Pág.
2.1	Diferencias entre un microprocesador y un microcontrolador.....	45
2.2	Funciones de los pines del conector RS 232 DB9.....	62
2.3	Parámetros DH del robot antropomórfico de 5 ejes.....	99
4.1	Datos técnicos Impresora Thimsom TRS 2000.....	133
4.2	Especificaciones técnicas de Robot RV-12Sd.....	188
4.3	Dimensiones físicas de paquetes.....	190
4.4	Variables de entrada y salida del Robot.....	202
4.5	Datos técnicos del robot RV-2AJ.....	205



AUTOMATIZACIÓN DE UNA LÍNEA DE IMPRESIÓN
Y EL PROCESO DE EMBALAJE DE UNA EMPRESA PAPELERA
MEDIANTE UN DISPOSITIVO CONTROLADOR Y UN ROBOT INDUSTRIAL

RESUMEN

El análisis del proceso de impresión de la empresa papelerera DAFOINCA, permitió determinar debilidades importantes en el área de impresión, entre las cuales, la operación manual de detención y control de producción de la máquina rotativa origina desperdicios de papel, pérdidas de producción y ausencia de datos para el pago de incentivos de producción. También en el proceso de embalaje existe una diferencia importante en el tiempo en que la maquinaria de empaque produce, con la capacidad de embalaje de los operarios; originando paradas de máquina, horas de sobretiempo y accidentes laborales. Como solución a esta situación se propone para el proceso de impresión, el diseño de un sistema automatizado que permita registrar la cantidad de metros procesados, detener la máquina en caso de completar los límites de producción y almacenar los datos para su posterior visualización. Para el proceso de embalaje se plantea utilizar un Robot Industrial como sistema de embalaje automatizado, el cual realice la tarea del embalado de una manera más rápida y segura. El objetivo principal de la propuesta es entonces la automatización de una línea de impresión y el proceso de embalaje de una empresa papelerera, mediante un dispositivo controlador y un Robot Industrial. La metodología está sustentada en la de un proyecto factible que consiste en una propuesta de acción para solucionar un problema de una organización. Se espera que esta propuesta de automatización impacte positivamente la producción, productividad, seguridad laboral y aspecto ecológico para la empresa y sus trabajadores.

Palabras Clave: Automatización, Robot Industrial, Impresión, Máquina rotativa, Embalaje, Seguridad laboral, Ecológico, Controlador.



Resumen





Introducción

La industria de manufactura es el eslabón de la economía encargado del procesamiento de materias primas, por medio del uso de personal, energía, maquinaria y equipos, para la obtención de un producto final con valor agregado. Con el fin de aumentar la rentabilidad deben hacer las operaciones de forma rápida, confiable y segura. Bajo esta necesidad nace la automatización de los procesos industriales. Esta es una herramienta que se basa en el uso de sensores, actuadores y uno o más elementos controladores para la ejecución de los procesos de forma completa o parcialmente automática, de manera que la supervisión requerida por parte del operador hacia el proceso sea menor.

Dentro de los sistemas de automatización se encuentran como soluciones de interés para el presente trabajo la implementación de micro-controladores a través de tarjetas electrónicas, con sus debidos periféricos que le permitan realizar la supervisión y control de determinado proceso. Este tipo de solución es ideal para la recolección y procesamiento de datos, y el envío de señales que ejecuten acciones en máquinas o sistemas pertenecientes al proceso.

En aquellos procesos en los que se requiere de la manipulación de un determinado producto, con un movimiento repetitivo, se ha desarrollado en la industria el Robot Industrial. Esta herramienta hace las veces del operador convencional, la cual siendo una vez programado, realiza su operación de forma rápida, precisa y confiable.

El presente trabajo de grado tiene como finalidad el diseño de un dispositivo controlador-registrador en base de un microcontrolador PIC para la automatización del proceso de impresión de la empresa DAFOINCA. De igual forma pretende diseñar un sistema para la automatización del proceso de embalaje de dicha empresa, por medio de la programación de un robot FESTO – Mitsubishi y el diseño de una herramienta de agarre adaptable a dicho proceso.



Este trabajo de investigación está constituido por cinco capítulos organizados de tal forma que permiten alcanzar los objetivos planteados al culminar el desarrollo de dicha investigación.

Inicialmente en el Capítulo I se plantea el problema existente, se describe la situación actual de la empresa DAFOINCA, detallándose cada una de sus fases de proceso. Luego de esto se realiza, la justificación de la investigación, la definición de los objetivos propuestos y se establece el alcance.

Como continuación, en el Capítulo II, se describe el marco teórico que incluye los antecedentes y las bases teóricas relacionadas con micro-controladores, integrados PIC y sus periféricos, protocolos de comunicación, robots, y toda su teoría de cinemática relacionada.

En el Capítulo III, se señalan el tipo, diseño y metodología de investigación utilizados, resaltando las técnicas de investigación empleadas y las etapas metodológicas a seguir para el desarrollo del proyecto.

Se procede luego con el Capítulo IV, donde se detallan los resultados obtenidos en las etapas metodológicas planteadas, las cuales son: Estudio del proceso de embalaje, análisis del control de los procesos de impresión y embalaje, diseño y prueba de un dispositivo controlador-registrador para la automatización la máquina de impresión, desarrollo de un sistema de control automatizado para el proceso de embalaje, mediante un Robot Industrial.

Para finalizar, en el Capítulo V, se exponen las conclusiones y recomendaciones finales de acuerdo a los objetivos del proyecto de grado.



Capítulo I

1.1. Planteamiento del problema

En la zona industrial de Valencia en el estado Carabobo se encuentra ubicada la empresa Data Formas Industriales C. A. (DAFOINCA). Su actividad primordial es la impresión de facturas, recibos, hojas membretadas, entre otras. Para realizar esta tarea, la empresa sigue cinco procesos, identificados como: diseño, fotograbado, impresión, empaque y embalaje.

El proceso de diseño consiste en realizar el modelo en digital del arte a imprimir que pasa a un proceso de fotograbado, donde el diseño es transferido a una foto en negativo. Esta foto es copiada en una plantilla metálica o plancha de impresión.

El proceso de impresión consiste en tomar la plantilla del proceso antes mencionado y se instala en la máquina rotativa de impresión que utiliza rollos de papel como materia prima donde un rodillo con la plantilla instalada hace contacto con la tinta y por rotación impregna al papel para realizar la impresión.

Una vez realizada la impresión el papel se vuelve a enrollar en carretes con dimensiones determinadas y pasan al proceso de empaque donde los carretes que contienen los formatos impresos se pasan por la máquina llamada “empacadora” que agrupa las copias, dobla o corta el formato de acuerdo a especificaciones y separa en paquetes.

El último proceso es el de embalaje, en el cual el operario toma los paquetes y los coloca en cajas de diferentes tamaños de acuerdo a las características de los paquetes. Posteriormente estas cajas se envían al departamento de transporte para su distribución.

Durante el proceso de impresión, el control de la cantidad de metros de papel que ingresan y salen de la máquina, es manual. El operador tiene que estar pendiente en todo momento de la cantidad de metros de papel procesados, un descuido del operador provoca



que se exceda la cantidad predefinida de metros de papel lo que se traduce en desperdicio de dinero al no poder ser reprocesada la materia prima, un consumo extra de energía eléctrica por el tiempo adicional de proceso y un desperdicio ecológico.

Esta situación está siendo considerada por la empresa DAFOINCA como crítica porque además de la pérdida monetaria y daño ecológico, se ha implementado un programa de incentivos a producción y al no poseer medios para medir de manera precisa la cantidad de materia procesada o la cantidad de desperdicios reales; no se puede evaluar la mejoría en la producción, dificultando el pago de un incentivo por producción a los trabajadores.

Otro aspecto que trae inconvenientes a la empresa es el embalaje. Cuando los pedidos de los clientes son numerosos, se retrasa la producción por el tiempo que les lleva a los operarios embalar cada pedido, provocando incluso que se dé la orden de detener las máquinas de empaque, ya que estas producen los pedidos más rápido de lo que los operarios pueden embalarlos. Esta situación hace que los operarios tengan que quedarse horas extras para cumplir los plazos de entrega. También la prisa por embalar provoca que los trabajadores sean propensos a lesiones. Estos atrasos en producción y el congestionado sistema de embalaje se traducen en pérdida de dinero y tiempo para la empresa, además de un riesgo a la salud de los trabajadores.

Una vez conocida la situación presentada en el proceso de impresión, se propone el diseño de un sistema automatizado que permita registrar la cantidad de metros procesados, avisar al operador cuando se acerque el límite de producción, detener la máquina en caso de que se alcance el límite de producción y almacenar los datos de la cantidad procesada para su posterior visualización. Con esta propuesta se puede lograr un control adecuado para el pago de productividad y una mejora en el proceso que reduzca en forma significativa el desperdicio y el consumo de energía eléctrica, lo que constituye un beneficio para la empresa, el trabajador y un impacto positivo en el medio ambiente.



Para solucionar la situación presentada en el proceso de embalaje se plantea utilizar un Robot Industrial como sistema de embalaje automatizado, el cual realice la tarea de colocar los paquetes en sus respectivas cajas rápidamente y con seguridad. Esta propuesta permite una mejor sincronización entre el proceso de empaque y embalaje, minimiza el sobretiempo y asegura menores condiciones de riesgo ocupacional para los trabajadores y la empresa.

1.2. Justificación

Este trabajo es un aporte a la investigación y/o desarrollo tecnológico en el área de conocimiento de sistemas y automática, y la línea de investigación “Robótica y Visión Industrial”, ya que propone la automatización de un proceso de impresión y embalaje en una empresa papelera. En tal sentido se plantea:

La inclusión de un dispositivo para la automatización de una parte del proceso de impresión que reduciría la dependencia del proceso con los operarios, la cantidad de desperdicio, el tiempo de producción y permitiría determinar indicadores que favorezcan el cálculo de productividad para el pago de incentivos laborales. Esto es de particular importancia ya que el personal de la empresa tiene entre sus formas de pago, un incentivo de productividad, por lo que la instalación del dispositivo generaría un beneficio económico a los operarios al aumentar la producción. Adicionalmente la disminución del desperdicio constituye un beneficio ecológico por la disminución del impacto ambiental que favorece la utilización de una menor cantidad de papel.

La automatización del proceso de embalaje que acortaría los tiempos de embalado, solucionando el cuello de botella que se origina en esta fase, permitiendo procesar más pedidos en el mismo periodo de tiempo proporcionando a la empresa ventajas competitivas.



Un aspecto muy importante que mejora la automatización del embalaje es la disminución de accidentes laborales. El empaqueo dejaría de depender de trabajadores que levantan cajas pesadas durante horas, lo que causa cansancio, mermando sus habilidades y entorpeciendo el proceso. Además se evitarían las lesiones provocadas por levantar paquetes, transportarlos o dejarlos caer. La empresa ahorraría tiempo al evitar accidentes que paren la producción y pagos de indemnizaciones por accidentes.

La empresa DAFOINCA posee poco personal para desarrollar sus actividades diarias por lo que la automatización de estos procesos permitirá que se reubiquen a operarios a otras áreas de la empresa en las cuales hay un déficit de personal.

1.3. Objetivos

1.3.1. Objetivo General

Automatizar las líneas de impresión y el proceso de embalaje de una empresa papelera, mediante un dispositivo controlador y un Robot Industrial.

1.3.2. Objetivos Específicos

- 1.3.2.1 Diseñar un dispositivo controlador-registrador para la automatización de una impresora rotativa industrial.
- 1.3.2.2 Diseñar un programa que permita la ejecución de las actividades que el dispositivo realizará.
- 1.3.2.3 Desarrollar un programa para un Robot Industrial que permita la automatización del proceso de embalaje.
- 1.3.2.4 Diseñar la herramienta de agarre del robot (mano del robot) que permita el manejo de los paquetes.
- 1.3.2.5 Simular el proceso de embalaje con un Robot Industrial a escala.



1.4. Alcance

Este trabajo contempla el estudio de las problemáticas existentes en la sección de la máquina rotativa del proceso de impresión y en la sección de embalaje de la empresa DAFOINCA. La propuesta consiste en dos aspectos:

El primero es la construcción de un prototipo del dispositivo, el cual se les simularán entradas como las del proceso estudiado sin instalarlo directamente en el proceso.

El segundo es la selección de un Robot Industrial que pueda ser utilizado en la línea de embalaje. Se diseñarán los programas que robot utilizará y se verificará el funcionamiento de los mismos mediante una simulación del proceso con un Robot Industrial comercial real a escala.

En tal sentido el alcance será la construcción y prueba de un prototipo para la automatización de la máquina de impresión, y la programación de un Robot Industrial que se verificará a través de una simulación con un robot real. En este proyecto no se contempla la implementación de estos elementos, solo diseño, simulación y prueba con un Robot fuera del proceso.



Capítulo II





Capítulo II

2.1. Antecedentes

El trabajo de investigación propuesto tiene como antecedentes los siguientes trabajos especiales de grado realizados en la Universidad de Carabobo a través de los cuales se han estudiado las distintas herramientas en el área de diseño digital y de elementos de robótica industrial, los cuales se nombran a continuación:

Simone, 2011. “Desarrollo de una estación robotizada para el cardado de calzados, usando el lenguaje Rapid para un robot ABB IRB 6400. Caso de estudio: Fábrica de Calzados INDUSIM C. A.” [1]

En el desarrollo de este proyecto se desarrolló una aplicación de un robot antropomórfico comercial de la Marca ABB de propiedades similares al robot propuesto en el presente trabajo, como lo son: del tipo antropomórfico, y uso industrial. Debido a la semejanza entre el software de programación del robot ABB, y el software del robot usado en el presente trabajo, Mitsubishi-FESTO, el desarrollo del primero se usó como referencia para la implementación del segundo. De igual forma, el marco teórico desarrollado en el mismo, referente a los robots, sus configuraciones, y lenguajes de programación valió para el marco teórico en el actual trabajo.

Estévez y Escalona, 2010. “Diseño y construcción de un prototipo de unidad de control para un sistema de transferencia de desconexión y reposición automática a la red de potencia basado en microcontroladores PIC”. [2]

Este proyecto de grado se basa fundamentalmente en el control de un sistema de potencia en el cual se realiza una conexión automática a la planta de emergencia en caso de haber fallo de suministro eléctrico. Las herramientas de ejecución de un proyecto por medio del



microcontrolador PIC, tales como el montaje del controlador sobre PCB, los elementos de conexión al PIC, y el método de programación del mismo, son utilizadas en el trabajo, al ser semejantes los principios teóricos de los equipos, protocolos, conexiones y medios a utilizar.

Guape, 2010. “Desarrollar un prototipo de telemetría para el proyecto “Baja” de la organización SAEUC Venezuela donde se transmitan y registren los datos de velocidad, revoluciones por minuto del motor y nivel de gasolina del vehículo”. [3]

Este proyecto de grado se basa fundamentalmente en la adquisición de datos en tiempo real usando microcontroladores PIC, además de desarrollar una interfaz gráfica con el usuario usando el software Delphi®.

La utilización de una interfaz gráfica para que el usuario se comunique y extraiga datos del PIC resulta de utilidad por que se tiene planteado realizar una interfaz grafica que se comunique con el dispositivo PIC en este proyecto.

Patiño y Simanca, 2009. “Reingeniería de la línea de envasado del laboratorio de automatización Industrial II con inclusión de dispositivos de bus de campo”. [4]

En este trabajo se describe el proceso de rediseño de un banco de automatización en el Laboratorio de Automatización Industrial II en el cual se realizaron diferentes conexiones entre controladores y actuadores neumáticos.

Esta manera de conexiones puede similarmente ser aplicada a las conexiones a realizar del controlador del Robot Industrial con los elementos neumáticos que lo conforman, como lo es la herramienta de agarre.

Janampa, 2008. “Desarrollo de un banco de control de procesos portátil para el laboratorio de Automatización Industrial I mediante un micro PLC Simatic S7-200”. [5]



En el desarrollo de este trabajo se tocan temas como el uso de sensores de proximidad ópticos, transmisión de señales eléctricas, usos de relés y conexión a controladores.

La necesidad de conexiones a controladores en el proyecto hace importante este trabajo ya que se busca la interconexión de sensores ópticos para el desempeño del Robot Industrial.

Hernández y Montoya, 2007. “Diseño de un sistema emulando el movimiento articulado de una mano, brazo y antebrazo”. [6]

En este trabajo se desarrolla paso a paso la resolución del problema cinemático de un robot antropomórfico, describiendo la solución del problema cinemático directo y del cinemático inverso, haciendo la definición de los parámetros Denavit-Hartenberg, y la obtención de la matriz homogénea. Este desarrollo se usó como patrón para la solución del problema cinemático del robot propuesto del presente trabajo.

2.2 Bases Teóricas

2.2.1 Microprocesador

Un microprocesador es un circuito integrado (“*Integrated Circuit*” en inglés, abreviado *IC*) que ejecuta operaciones aritméticas y lógicas por medio de una secuencia de instrucciones. Constituye la Unidad Central de Procesamiento (“*Central Processing Unit*” en inglés, abreviado *CPU*), de un sistema de cómputo electrónico, como lo son las computadoras personales y otros dispositivos electrónicos, y es el componente que proporciona gran parte de su funcionalidad. Esta unidad opera por medio de la ejecución de una serie de instrucciones, de acuerdo a los pasos definidos por el programa que tenga el *IC*.



El microprocesador se conecta externamente a una memoria y a controladores de dispositivos periféricos de entrada y salida. La memoria contiene los datos de las operaciones (operandos y resultados), así como también el programa a ejecutar. Los periféricos permiten conectar la *CPU* a dispositivos de entrada de datos externos y a dispositivos de salida. La conexión del *CPU* con la memoria y los periféricos se efectúa mediante un sistema de buses. [7]

En la figura 2.1 se observa un diagrama de conexión de un microprocesador, donde las líneas de los buses de datos, direcciones y control se extraen de los pines del circuito integrado del microprocesador.

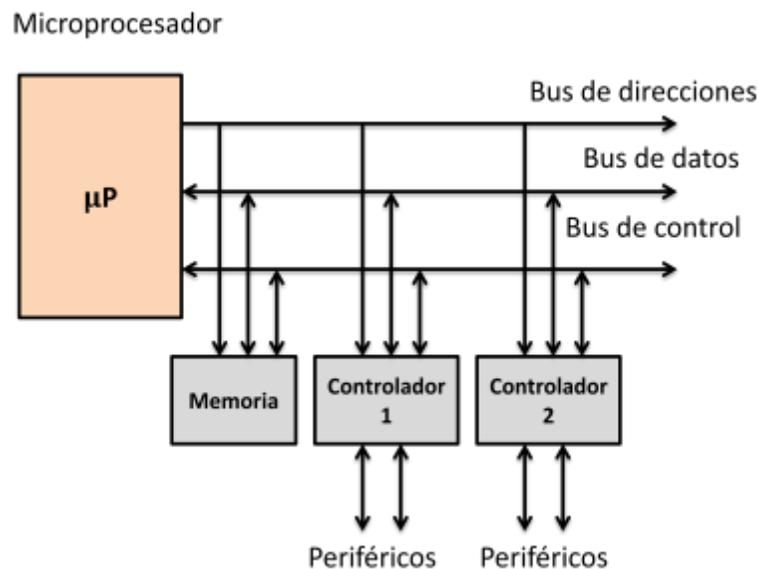


Figura 2.1. Conexión de un Microprocesador



2.2.2 Microcontrolador

a. Aspectos Básicos

Un microcontrolador (abreviado μC) es un circuito integrado programable capaz de ejecutar las órdenes grabadas en su memoria. El mismo lo integran varios bloques funcionales, que realizan una determinada labor. Un μC se divide estructuralmente en tres partes, las cuales son: unidad central de procesamiento, memoria y periféricos de entrada y salida. [7]

El *CPU* es el encargado de controlar al microcontrolador. La conexión interna del microcontrolador se establece entre la *CPU* y los otros subsistemas a través de buses de direcciones y datos. Se encuentran dos tipos de conexión entre las partes del microcontrolador. El primero basado en el modelo de arquitectura Harvard, y el segundo en base al modelo de arquitectura Von-Neumann. La diferencia de estos dos modelos es en el modo de intercambio de datos entre el *CPU* y la memoria. A continuación se describen tales arquitecturas:

Arquitectura Von-Neumann

Esta arquitectura dispone de un solo bloque de memoria conectado a la *CPU* a través de un bus de datos de 8 bits. En el bloque de la memoria se encuentra almacenado tanto los datos volátiles de la memoria *RAM* (“*Random Access Memory*” en Inglés) como los datos del programa, en la memoria *ROM* (“*Read Only Memory*” en Inglés). Este tipo de conexión genera una comunicación lenta e ineficaz. La *CPU* puede leer una instrucción o leer/escribir datos de/en la memoria. Los dos procesos no pueden ocurrir a la vez puesto que las instrucciones y los datos utilizan el mismo bus. En la figura 2.2 se observa un esquema de la arquitectura Von-Neumann. [8]

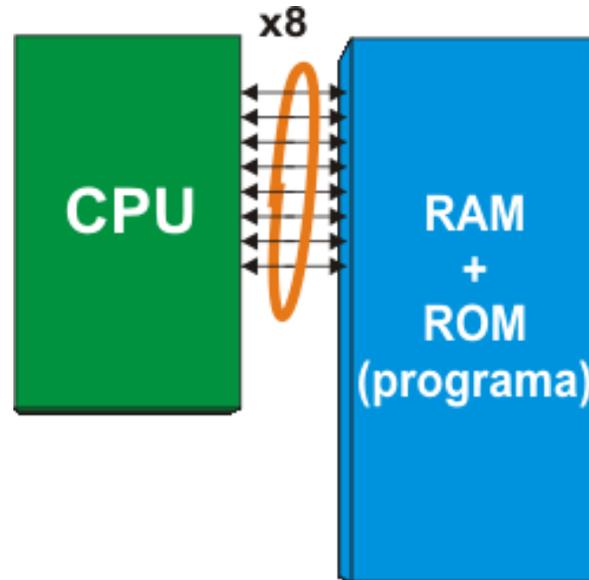


Figura 2.2. Arquitectura Von-Neumann

Fuente: MikroElektronika (2009), documento en línea [8]

Arquitectura Harvard

En esta arquitectura se disponen de dos buses de datos diferentes. El primer bus, de 8 bits de ancho, conecta la *CPU* con la memoria de acceso aleatorio *RAM*. El segundo, se compone de varias líneas (12, 14, o 16) y establece la conexión *CPU* - memoria *ROM*. Esta estructura permite al *CPU* realizar de forma paralela la lectura de instrucciones y el acceso a memoria de datos. Todos los programas escritos para estos microcontroladores serán almacenados en la memoria *ROM* interna del microcontrolador después de haber sido compilados a código máquina. No obstante, estas localidades de memoria *ROM* no tienen 8, sino 12, 14 o 16 bits. 4, 6 o 8 *bits* adicionales representan una instrucción que especifica a la *CPU* qué hacer con los datos de 8 bits. En la figura 2.3 se observa un esquema de la arquitectura Harvard. [8]

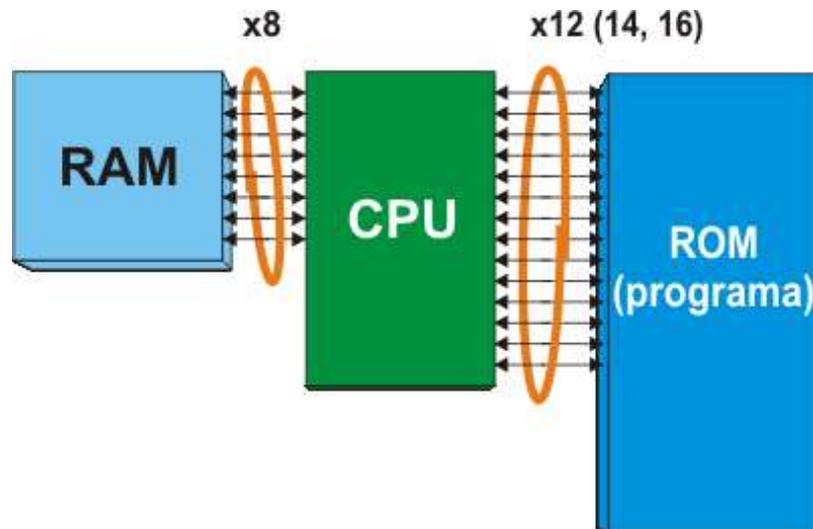


Figura 2.3. Arquitectura Harvard

Fuente: MikroElektronika (2009), documento en línea [8]

Las ventajas de este diseño son las siguientes:

- Los datos del programa tienen un ancho de un *byte* (8 bits). La lectura de la instrucción y del dato puede ser de forma simultánea. Puesto que el bus de datos tiene entre doce (12) y dieciséis (16) líneas. Es por esta razón que todas las instrucciones se ejecutan en un ciclo salvo las instrucciones de salto que son de dos ciclos.
- Puesto que el programa del IC y los datos temporales están separados, la CPU puede ejecutar dos instrucciones de forma simultánea.
- En esta arquitectura, el bus de la palabra de programa es más ancho que un *byte*, lo que permite que cada palabra de programa esté compuesto por una instrucción y un dato. De forma que para cada localidad de memoria se tiene una instrucción de programa. [8]



b. Funciones resaltantes

Temporizadores

Para generar la herramienta denominada temporizadores, el *IC* utiliza su cristal de cuarzo. Esto se realiza fijando los pulsos generados por el cristal, por medio de la configuración de unos valores de *Prescaler*, *Posscaler*, según sea el modelo del *IC* los cuales dan una frecuencia de operación determinada. En la figura 2.4 se observa el esquema de un módulo temporizador. [8]

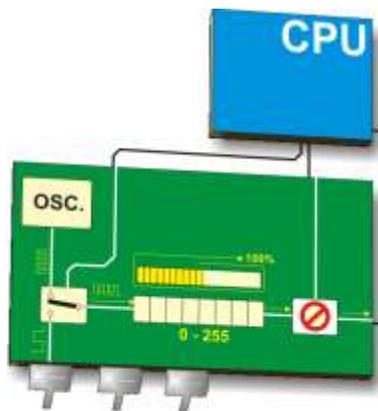


Figura 2.4. Temporizadores

Fuente: MikroElektronika (2009), documento en línea [8]

Contadores

La herramienta de contador se basa en el mismo concepto del temporizador, con la diferencia que la transición de estado viene de una señal externa. Tal señal se recibe por un pin del *IC*. De esta forma el registro aumentará su valor por cada flanco observado en el terminal del integrado. [8]



Temporizador perro guardián (*watchdog timer*)

El perro guardián es un temporizador especial independiente que se encuentra conectado a un oscilador RC dentro del microcontrolador. Y es utilizado para resetear el *IC* en caso de que se quede atrapado en un bucle infinito, o bien se quede “colgado” en una parte del programa. [8]

Convertidor A/D

Todo *IC* está concebido para operar con señales de voltaje continuo, cuyos valores se interpretan como unos o ceros. Sin embargo en la electrónica de campo se manejan señales de naturaleza analógica, las cuales tienen un rango de variación con posibles valores infinitos. De manera que para que el *IC* interprete estas señales, éstas deben ser convertidas en señales *DC*. Un convertidor analógico-digital es un circuito electrónico encargado de convertir las señales continuas en números digitales discretos. En otras palabras, este circuito convierte un número real en un número binario y se lo envía a la *CPU* para ser procesado. Este módulo se utiliza para medir el voltaje en el pin de entrada.

En la figura 2.5 se observa el esquema del módulo de conversión A/D. [8]

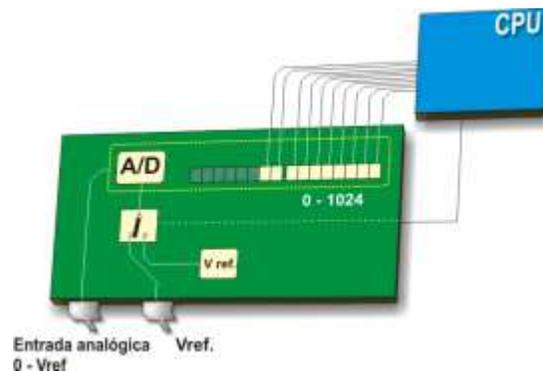


Figura 2.5. Esquema de Convertidor A/D

Fuente: MikroElektronika (2009), documento en línea [8]



El resultado de esta medición es un número, de valor digital, utilizado y procesado más tarde en el programa. La conversión que genera el convertidor A/D en función del valor analógico de entrada se observa en la gráfica de la figura 2.6. [8]

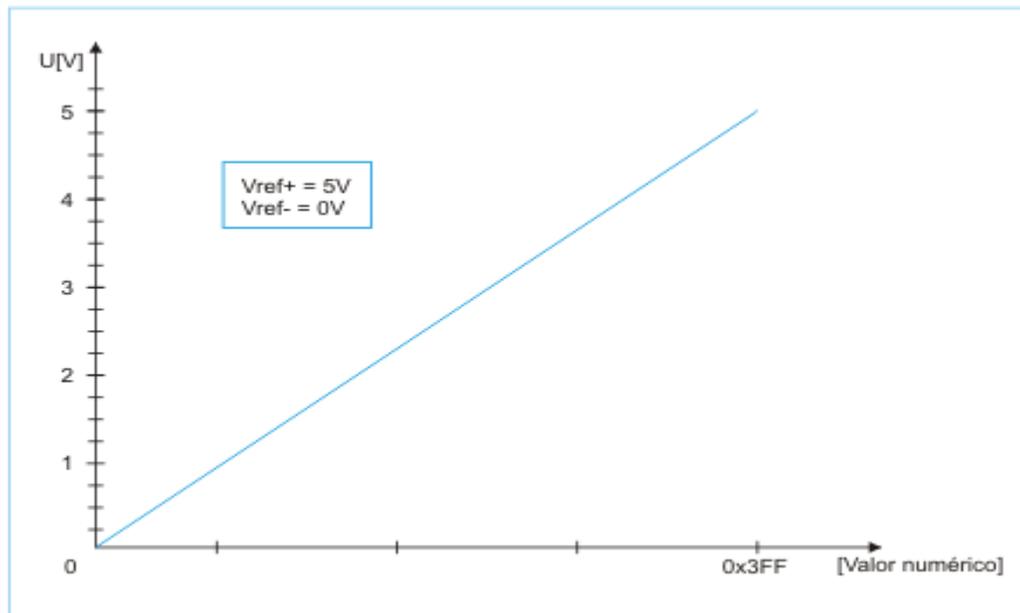


Figura 2.6. Función del Convertidor A/D

Fuente: MikroElektronika (2009), documento en línea [8]

c. Diferencias entre un microprocesador y un microcontrolador

En la tabla 2.1 se presenta de una manera comparativa las diferencias entre un microprocesador y un microcontrolador.



Tabla 2.1 Diferencias entre un microprocesador y un microcontrolador

Microprocesador	Microcontrolador
Sistema “abierto” que puede conectarse a distintas memorias y periféricos.	Sistema cerrado que ya contiene una memoria de tamaño fijo y sus periféricos.
Puede usarse para diversas aplicaciones de acuerdo a los demás elementos a los cuales se conecta.	Generalmente un modelo se usa para una tarea específica dado su carácter limitado en recursos.
Requiere espacio adicional y conexiones en un circuito impreso para agregar demás elementos. El circuito de conexión de buses con periféricos y memoria tiende a ser bastante complejo.	No requiere de espacio o conexión adicional dado que todos los recursos están en el mismo chip (ahorrando espacio, tiempo y dinero en su utilización.)
Para cada arquitectura de microprocesador, es común que el fabricante ofrezca un solo modelo (o pocas subfamilias), dado que el diseñador puede utilizarlo para diversas aplicaciones de acuerdo a la memoria y periféricos que utilice.	Para cada arquitectura de microcontrolador, es común que el fabricante ofrezca numerosos modelos o subfamilias, con distintos tamaños de memoria y distintos periféricos, para que el diseñador escoja el más conveniente para una aplicación específica.

Fuente: Silva R. (2007), documento en línea [9]



2.2.3 PIC

a. Los PICs de Microchip

Comprende una familia de microcontroladores de arquitectura Harvard, patentada y fabricada por Microchip Technology. Desde 1997 son llamados PICmicro para distinguirlos de otros dispositivos (*Peripheral Interface Controller* de los PCs.) Se deriva del PIC1650 creado en 1975 por la compañía *General Instruments* el cual fue diseñado para un propósito distinto. Cuando a este modelo le fue añadida una memoria *EEPROM*, en 1985, el circuito integrado pasó a ser el microcontrolador PIC. Su uso se ha extendido de gran forma por su bajo costo, reducido consumo, pequeño tamaño, facilidad de programación y gran cantidad de herramientas de soporte entre los desarrolladores industriales y aficionados a la electrónica. [10]

b. Arquitectura básica del PIC

- Procesador: Sigue la arquitectura Harvard, con procesador segmentado e instrucciones *RISC*.
- Memoria de Programa: La memoria es no volátil. Pudiendo almacenar todo el código del programa de aplicación. Esta es de un tamaño muy reducido (512 *Bytes* - 16 KB) y es ajustada al tamaño de las instrucciones.
- Memoria de Datos: La memoria de datos es memoria *RAM*: volátil de tamaño reducido, para variables de programa (36 *Bytes*) Sin embargo puede hacer uso de la *EEPROM* para datos importantes que no deben perderse en caso de cortes de corriente (64 *Bytes*)



- Líneas de entrada/salida: Tiene disponible sus pines para la conexión con los periféricos. Éstos manejan la información en paralelo y se agrupan en puertas, pero cada señal se puede configurar independientemente.
- Recursos auxiliares: Posee un circuito de reloj, encargado de la sincronización del sistema. Un Temporizador para controlar los tiempos. Un temporizador perro guardián (*watchdog timer*) provoca una reinicialización cuando el programa se queda colgado. Así mismo un Conversor AD y DA para recibir y enviar señales analógicas. Además posee un Comparador analógico para comprobar el valor de una señal analógica, un sistema de protección ante fallos de alimentación, y del Estado de reposo (*Stand By*), para minimizar el consumo de energía. [10]

En la figura 2.7 se observa un diagrama de la arquitectura interna del PIC, en este caso de un 16F84X

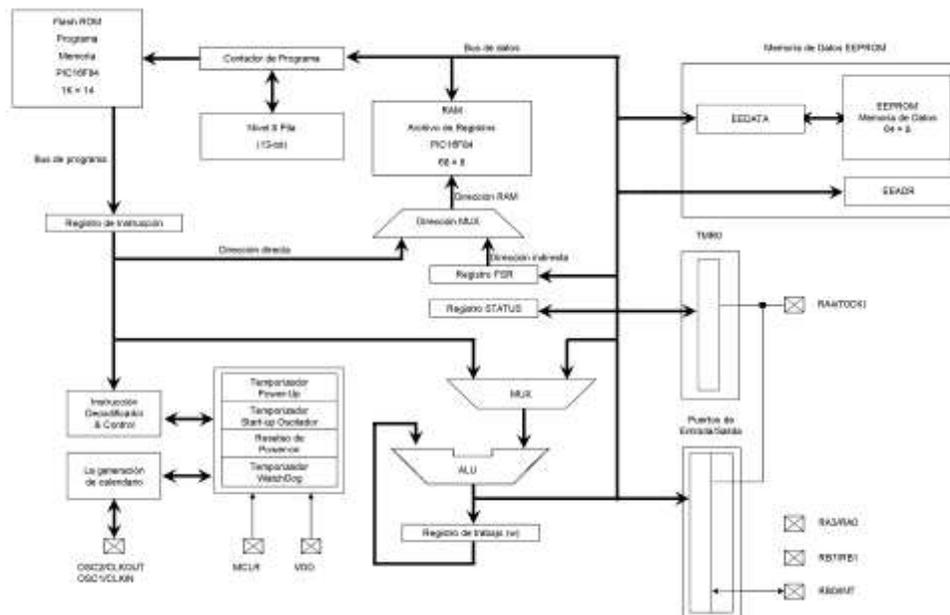


Figura 2.7 Arquitectura de un PIC.

Fuente: Galeano, L. (2008), documento en línea [11]



2.2.4 Microcontrolador PICmicro 18F4520

2.2.4.1 Aspectos básicos

Este microcontrolador pertenece a la gama alta de la familia de los PIC. Posee todas las características de los microcontroladores de las familias de anteriores, incluye una serie de mejoras de rendimiento, mejoras de compatibilidad con periféricos, mayor eficiencia en el consumo de potencia, set de instrucciones extendido, una estructura sensible para el manejo de osciladores tanto internos como externos. Cuenta con gran cantidad de módulos periféricos como módulos de captura mejorados o ECCP por sus siglas en ingles, modulo de comunicación serial mejorado (EUSART) y bus de comunicación I2C. [12]

2.2.4.2 Características Básicas

- Arquitectura *RISC*: El microcontrolador cuenta un set de 77 instrucciones.
- Frecuencia de operación hasta 40 MHz
- Oscilador interno de alta precisión (31kHz a 8MHz seleccionado por software)
- Voltaje de la fuente de alimentación de 2.0V a 5.5V
- Ahorro de energía en el *Modo de suspensión*
- *Brown-out Reset* (BOR) con opción para controlar por software
- Bus de datos de 8 bits
- 35 pines de entrada/salida
- Memoria de programa de 32K con tecnología FLASH
- Opción de programación serial en el circuito "*In circuit serial Programming*"
- 256 Bytes de memoria *EEPROM*
- 1536 Bytes de memoria *RAM*



- Convertidor A/D (13 canales, resolución de 10 bits)
- Hasta 3 interrupciones externas programables
- Temporizador perro guardián
- Módulo comparador analógico (Dos comparadores analógicos)
- Módulo PWM incorporado
- Módulo USART mejorado
- Puerto Serie Síncrono Maestro (*MSSP*): Soporta los modos *SPI* e *I2C* [10]

Estructura

En la figura 2.8 se observa un la estructura interna del PIC

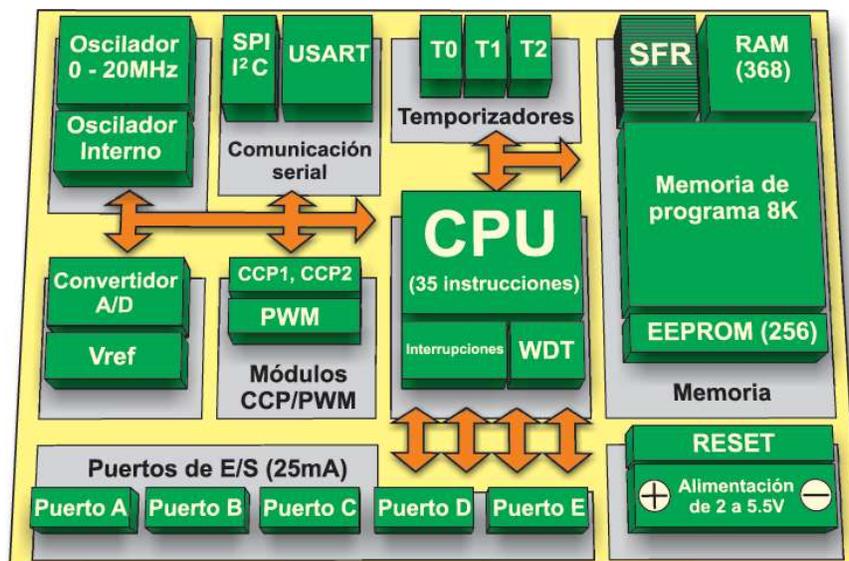


Figura 2.8. Estructura interna del PICmicro 18F4520

Fuente: MikroElektronika (2009), documento en línea [8]



Pines del PIC

En la figura 2.9 se muestra la distribución de los pines del PIC 18F4520. Con el cual se puede determinar, según los requerimientos, en que pin o patita conectar el PIC con el elemento deseado.

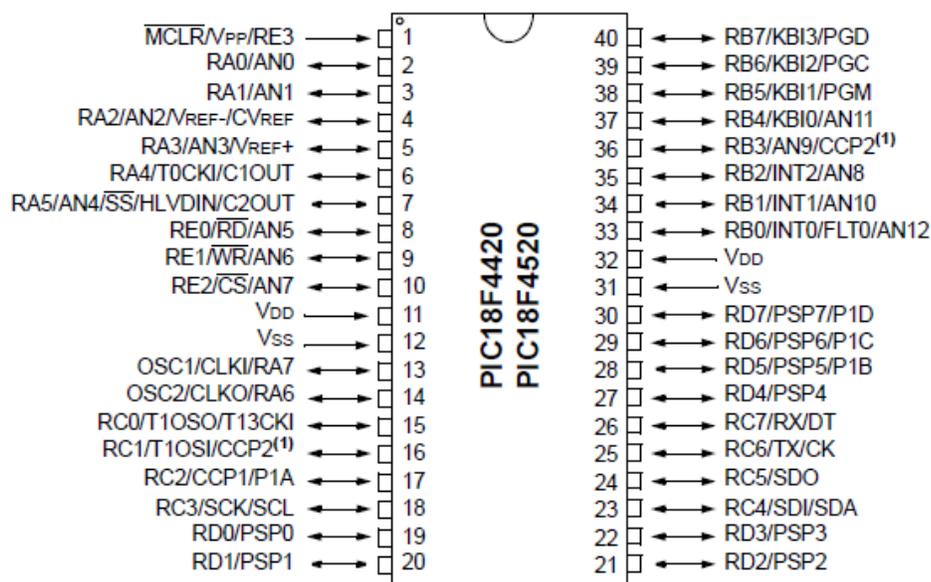


Figura 2.9. Diagrama de patas (Pin-out) del PIC 18F4520

Fuente: Microchip (2008), documento en línea [12]

Memoria

El PIC18F4520 tiene memorias: Flash, RAM y EEPROM. Como cada una tiene las funciones, características y organización específicas, vamos a presentarlas por separado. [12]



Memoria *Flash*

El PIC18F4520 tiene 32Kb de memoria *Flash* de programa. Ya que la memoria de programa está fabricada con tecnología FLASH, su contenido se puede cambiar al proporcionarle un voltaje de programación especial (13V). [12]

Memoria *EEPROM*

El contenido de esta está permanentemente guardado al apagar la fuente de alimentación. Pero a diferencia de la memoria de programa, el contenido puede ser cambiado durante el funcionamiento del microcontrolador. Es ideal para guardar permanentemente algunos resultados creados y utilizados durante la ejecución del programa y posee 256 bytes. [12]

En la figura 2.10 se observa el esquema de memoria de la *EEPROM*.

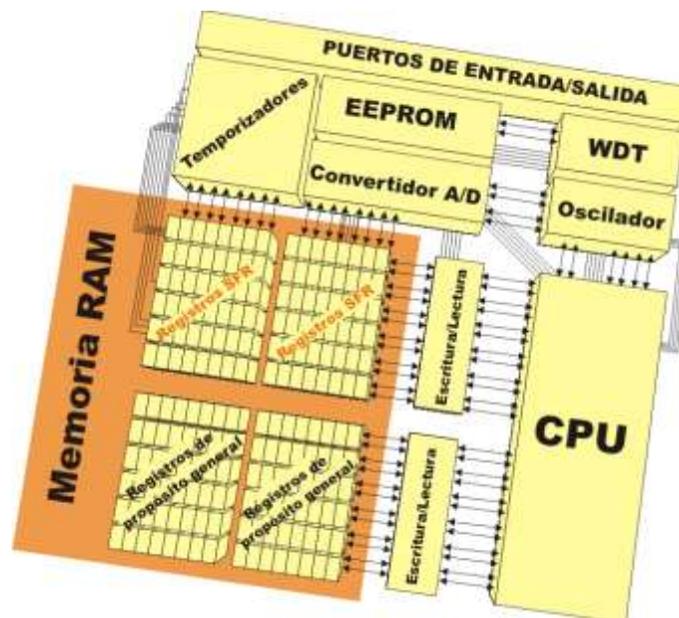


Figura 2.10. Esquema de memoria *EEPROM*

Fuente: MikroElektronika (2009), documento en línea [8]



En la figura 2.11 se observa un diagrama de la arquitectura interna del PIC 18F4520

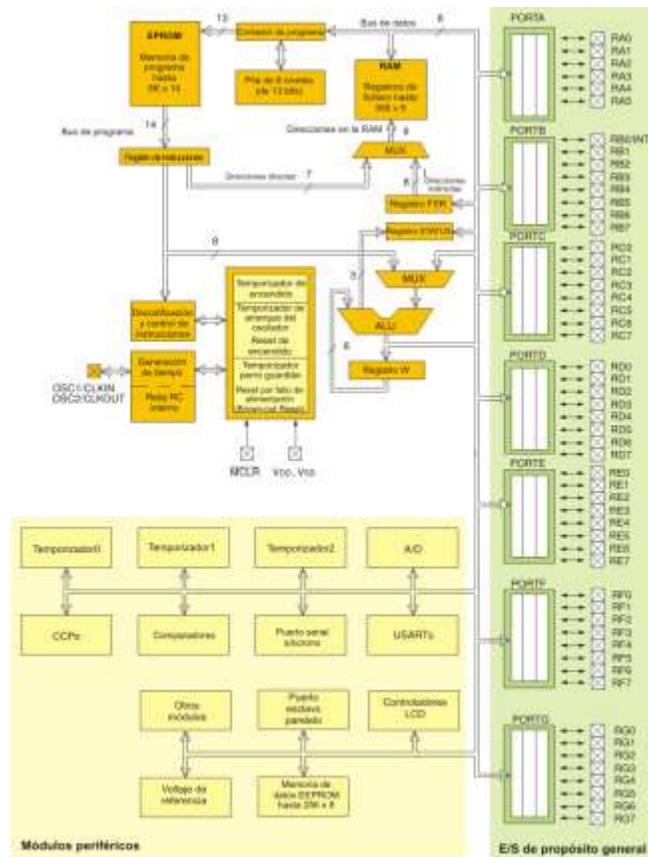


Figura 2.11. Estructura interna del PIC 18F4520

Fuente: MikroElektronika (2009), documento en línea [8]

Para entender mejor el principio estructural del funcionamiento del PIC, la figura 2.12 muestra un diagrama de la conexión entre su unidad central de procesamiento (*CPU*) y sus elementos periféricos, internos del controlador. Así mismo se muestra el sentido en el que se realiza el flujo de datos entre el procesador, y los distintos elementos y viceversa.

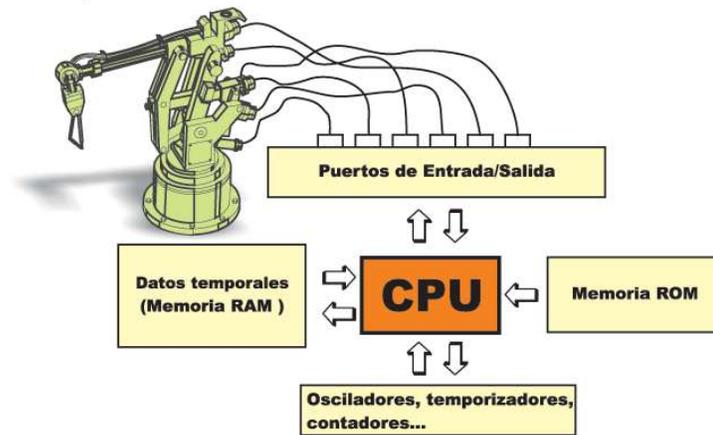


Figura 2.12. Funcionamiento del PIC 18F4520

Fuente: MikroElektronika (2009), documento en línea [8]

2.2.4.3 Elementos Periféricos

a. Pulsadores

Los pulsadores son periféricos de entrada populares que cierran o abren el circuito al cual están conectados, según su configuración. Sus contactos pueden ser NA, normalmente abiertos, o NC, normalmente cerrados. Estos elementos permiten la interacción del con el microcontrolador. En la figura 2.13 se muestra el modelo de pulsador, el SW762. [13]



Figura 2.13. Pulsador modelo SW762

Fuente: Ditel C.A. (2012), sitio web [14]



En las figuras 2.14 y 2.15 se pueden observar el esquema de funcionamiento de los pulsadores según sean Activo alto o Activo bajo respectivamente. El pulsador Activo alto se encuentra alimentado a V_{cc} , proporcionando 5 Vcd en su terminal en condiciones normales, en caso de que se accione el pulsador, la tensión baja a 0 Vcd. El pulsador activo bajo opera de manera inversa. [13]

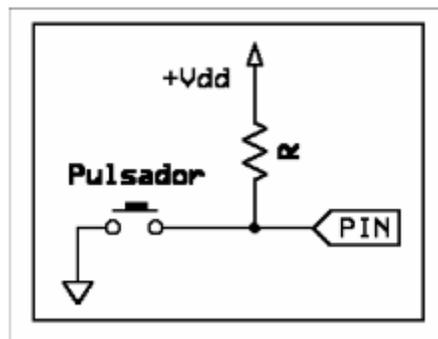


Figura 2.14. Conexión Activa baja con resistencia de Pull Up
Fuente: Villegas y Rodríguez (2008) [13]

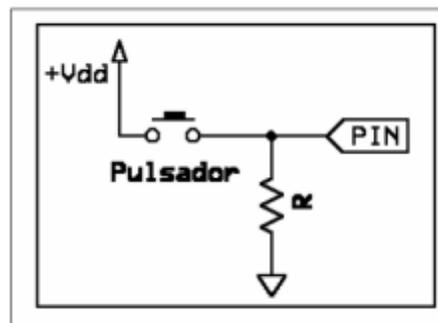


Figura 2.15. Conexión Activa alta con resistencia de Pull Up
Fuente: Villegas y Rodríguez (2008) [13]



Todos los dispositivos electromecánicos originan un fenómeno denominado rebote en el momento en que cambian de estado. El rebote produce una serie de oscilaciones con una duración típica de 10 a 20 ms. La aparición de estas falsas transiciones se debe a que, siendo en última instancia un pulsador (o interruptor) un par de láminas de material conductor que se unen para formar un contacto eléctrico, el contacto entre las láminas no se produce de manera instantánea, sino que, las rugosidades existentes en los bordes del material colisionan repetidas veces hasta alcanzar una posición estable que permite definir si se efectuó o no el contacto. Al ser los sistemas digitales muy veloces, el tiempo que toma la señal en estabilizarse es suficiente para interpretar muchas transiciones en la línea. El tratamiento de este tipo de señales, para una correcta toma de los valores de entrada al PIC, se desarrollan en el programador, agregando un sistema anti-rebote que toma la lectura del pin, luego de pasado cierta cantidad de milisegundos. [13]

b. Pantalla de Cristal Líquido (*LCD*)

Este periférico está diseñado para ser utilizado con los microcontroladores. Es utilizado para observar diversos mensajes en un visualizador de cristal líquido miniatura. (7cm largo x 2cm ancho aprox). Puede mostrar todas las letras de alfabeto, letras de alfabeto griego, signos de puntuación, símbolos matemáticos etc. También es posible visualizar símbolos creados por el usuario. Entre otras características útiles es el desplazamiento automático de mensajes (a la izquierda y a la derecha), aparición del cursor, y retroiluminación *LED*. [8]

En la figura 2.16 se observa un ejemplo de Pantalla *LCD* de dimensiones 2 Filas x 16 Columnas, con caracteres de 5x8 píxeles.



Figura 2.16 Pantalla *LCD* 2x16

Fuente: MikroElektronika (2009), documento en línea [8]

Las pantallas *LCD*, en su formato más comercial, poseen dos líneas con 16 caracteres c/u. Cada carácter lo constituyen 5x8 o 5x11 píxeles. El contraste del visualizador depende del voltaje de alimentación y de si los mensajes se visualizan en una o dos líneas. Por esta razón, el voltaje variable 0-V_{dd} se aplica al pin marcado como V_{ee}. Para tal fin se usa un potenciómetro. Algunos de los visualizadores *LCD* tienen retroiluminación incorporada (diodos *LED* azules o verdes). Al utilizarlo durante el funcionamiento, se debe de conectar una resistencia en serie a uno de los pines para limitar la corriente (similar a diodos *LED*). [8]

En la figura 2.17 se observa un ejemplo del conexionado de Pantalla *LCD*.

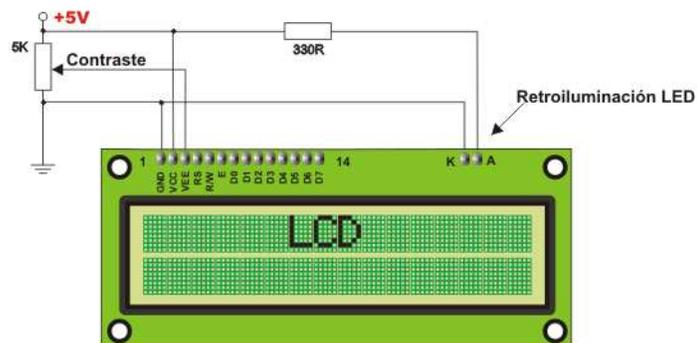


Figura 2.17 Conexión de Pantalla *LCD*

Fuente: MikroElektronika (2009), documento en línea [8]



c. **LED**

El *LED* es un diodo emisor de luz (“*Light Emissor Diode*” en inglés), fabricado a base de un semiconductor, que emite luz cuando se polariza en directa y es atravesado por una corriente eléctrica. Este componente es ampliamente utilizado en el campo de la electrónica como medio de interfaz con el usuario, para dar aviso de que una señal se encuentra activa o inactiva.

El *LED* entre sus características intrínsecas, tiene que sólo puede manejar corrientes de baja intensidad, en el orden de los mA (miliamperios), por esta razón se debe realizar un arreglo circuitual en la conexión para limitar la corriente que circula por el mismo. En la figura 2.18 se muestra la polaridad del *LED*, así como también el esquema de conexión apropiado para el mismo. Es necesario conectar al ánodo del diodo la alimentación de voltaje positiva (Vcd) y el cátodo en serie con una resistencia. [15]

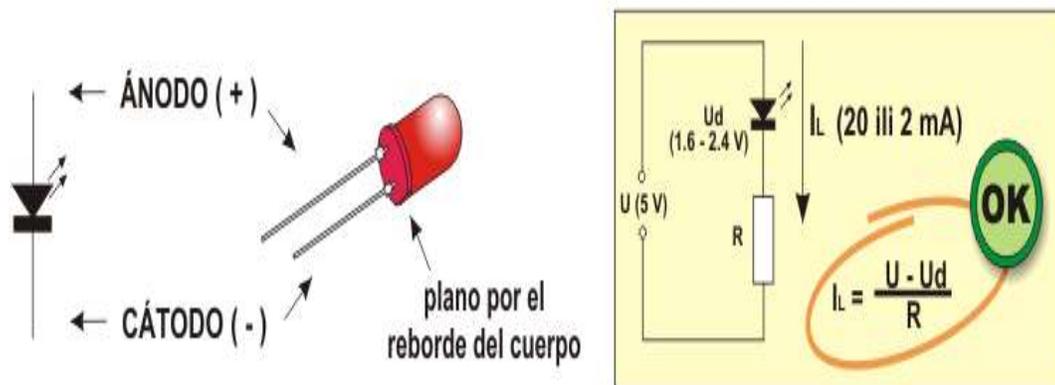


Figura 2.18 Polaridad y Conexión del *LED*

Fuente: MikroElektronika (2009), documento en línea [8]



2.2.4.4 Elementos de conexión

a. Relé

El relé es un interruptor eléctrico que abre y cierra la conexión de dos de sus terminales por medio del control de otro circuito electrónico, conectado al mismo por medio de otros dos terminales. Este componente es útil para controlar un circuito que maneje alta potencia, por medio de otro circuito donde se trabaje a baja potencia, en el caso nuestro, una conexión al PIC. Se puede utilizar para encender/apagar dispositivos de alto consumo, tales como: motores, transformadores, calefactores, bombillas etc. Estos componentes se ubican generalmente a una larga distancia de los componentes sensibles de la placa.

Aunque el principio de operación es el mismo, se encuentran diversos tipos de relés. Al fluir la corriente por la bobina, el relé funciona por medio de un electromagneto, abriendo y cerrando uno o más conjunto de contactos. Similar a los optoacopladores no hay conexión galvánica (contacto eléctrico) entre los circuitos de entrada y salida. Los relés requieren con frecuencia tanto un voltaje más alto y una corriente más alta para empezar a funcionar. También hay relés miniatura que se pueden poner en marcha por una corriente baja obtenida directamente de un pin del microcontrolador. [16] En la figura 2.19 se muestran varios modelos de relés.



Figura 2.19 Modelos de Relés

Fuente: MikroElektronika (2009), documento en línea [8].



La figura 2.20 muestra la solución utilizada con más frecuencia para la conexión de los relés a las patas de los microcontroladores.

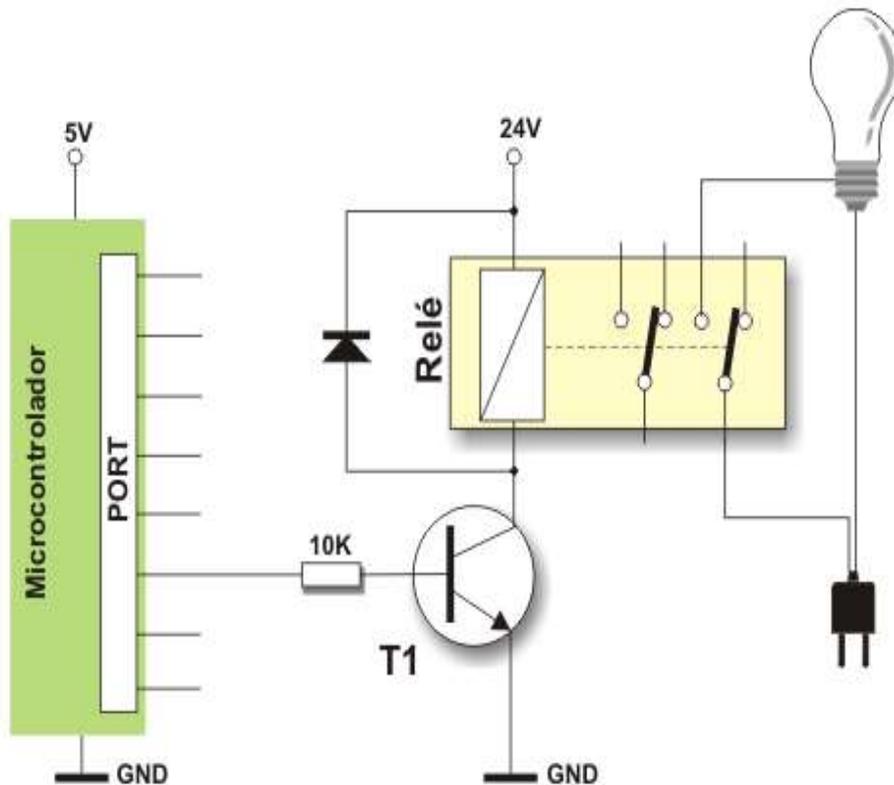


Figura 2.20 Conexión segura del Relé

Fuente: MikroElektronika (2009), documento en línea [8].

Para prevenir la aparición de un alto voltaje de autoinducción, causada por una parada repentina del flujo de corriente por la bobina, un diodo polarizado invertido se conecta en paralelo con la bobina. El propósito de este diodo es de suprimir este pico de voltaje. [8]



2.2.4.5 Protocolos de comunicación

a. Comunicación RS 232

Comunicación Serial

Una solución adecuada para cortas distancias, menor a 10 metros, es la conexión paralela entre el microcontrolador y los periféricos a través de los puertos de entrada/salida. Sin embargo, cuando se requiere establecer comunicación entre dos dispositivos a largas distancias no es posible utilizar la conexión paralela. Para tales condiciones se implementa la conexión serial.

Actualmente, existe una tendencia de dotar a los microcontroladores de diversos sistemas para la comunicación en serie, como un equipo estándar. La decisión de cuál de estos sistemas se utilizará en un caso concreto, depende de varios factores, entre los que se deben considerar: 1) El número de dispositivos con el cual el microcontrolador tiene que intercambiar los datos, 2) la velocidad del intercambio de datos, 3) la distancia entre los dispositivos, 4) la necesidad de transmitir y recibir los datos simultáneamente.

Una de los aspectos más importantes en cuanto a la comunicación en serie es el Protocolo que debe ser estrictamente observado. El protocolo es un conjunto de reglas que se aplican obligatoriamente para que los dispositivos puedan interpretar correctamente los datos que intercambian mutuamente. En este caso, los microcontroladores se encargan de realizar el cumplimiento del protocolo a nivel de firmware. Es por ello que el trabajo de programador/usuario se limita a la configuración de la comunicación a usar, por medio de un comando de software, y la programación correspondiente a la escritura y lectura de datos. [8]



Comunicación RS 232

La comunicación RS232 del PIC con el ordenador es de gran importancia y para esto se ha incorporado un módulo con las características apropiadas para el intercambio de información. Este módulo es conocido como *USART* (“*Universal Synchronous Asynchronous Receiver Transmitter*” en inglés). El módulo USART opera de acuerdo al protocolo RS232, ampliamente utilizado en las computadoras personales.

Este es uno de los dos módulos serie E/S, también llamado Interfaz de Comunicación Serie (*SCI*). El módulo USART puede configurarse como un sistema asincrónico full-dúplex que puede comunicarse con dispositivos periféricos, tales como terminales *CRT* y computadoras personales, o puede configurarse como un sistema sincrónico half-dúplex que puede comunicarse con otros dispositivos tales como conversores A/D y D/A, *EEPROM* s serie, etc. Los modos de operación disponibles son: Asincrónico (full-dúplex), Sincrónico-Maestro (half-dúplex), Sincrónico-Esclavo (half-dúplex). [16]

Protocolo RS232

Este es un protocolo de comunicación serie ampliamente difundido en las computadoras personales y empleado por los puertos COM del ordenador. El acceso físico a estos puertos es a través de un conector DB-25 o DB9, machos y hembras. La norma RS232 se estableció para comunicar un ordenador con un modem, por lo que en los conectores DB-25 aparecen muchos pines que en otras aplicaciones no se utilizan, y en las que es más común emplear el conector DB9. Cada uno de los terminales del conector RS232 tiene una función especificada por la norma. Unos pines transmiten y reciben datos, mientras que otros permiten el control de la comunicación. [16] En la tabla 2.2 se observan estos detalles (RS232 Pin-out).



Tabla 2.2 Funciones de los pines del conector RS 232 DB9

PIN	Señal
1	Data Carrier Detect (DCD)
2	Received Data (RxD)
3	Transmitted Data (TxD)
4	Data terminal Ready (DTR)
5	Signal Ground (SG)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

Fuente: ProgramarPICenC (2012), documento en línea [16]

Un factor importante en este punto es la velocidad de transmisión, la cual se define como la cantidad de información enviada por la línea de transmisión en la unidad de tiempo. Esta se mide en Baudios, la cual indica el número de símbolos transmitidos por segundo, el cual es proporcional a los Bits/segundo (bps). Las velocidades de transmisión normalizadas para los puertos COM son: 75, 150, 300, 600, 1200, 2400, 4800, 9600 Baudios.

Otra cuestión fundamental se refiere a los niveles de voltaje de la norma:

- Los datos se transmiten con lógica negativa, es decir, un voltaje positivo representa 0, mientras que un voltaje negativo representa 1.



- El 0L se encuentra entre +3 y +15V.
- El 1L se encuentra entre -3 y -15V.
- Los voltajes más usados son +12V para el 0L y -12V para el 1L.
- Cuando un puerto no está transmitiendo mantiene el terminal de transmisión en 1L (-12V). [16]

En la figura 2.21 se observa la conexión del PIC con RS232

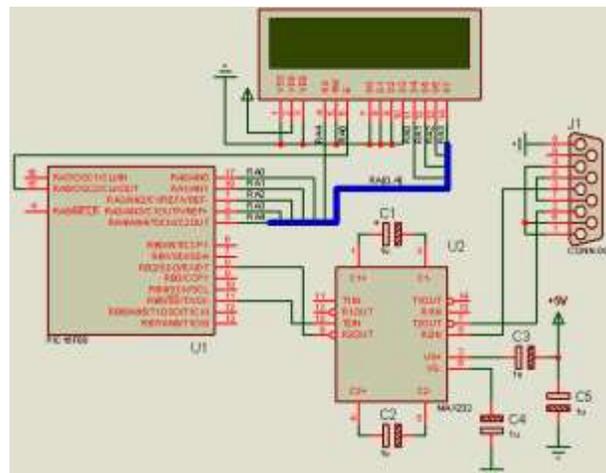


Figura 2.21 Conexión RS232 con PIC

Fuente: ProgramarPICenC (2012), documento en línea [16].

b. Comunicación USB

La comunicación USB (“*Universal Serial Bus*” en inglés) es una interfaz *plug & play* entre la computadora y periféricos, tales como: teclados, mouses, scanners, impresoras, módems, placas de sonido, cámaras, etc. ampliamente usado en el mundo de la informática debido a su fácil uso, y economía. Una característica importante es que permite trabajar a los dispositivos en promedio a unos 12 Mbps, alrededor de 5 veces más rápido que un dispositivo de puerto paralelo y 40 veces más rápido que un dispositivo de puerto serial. [16]



Funcionamiento

Trabaja como interfaz para transmisión de datos y distribución de energía. La conexión consta de 4 hilos: 2 de alimentación (+Vcc de 5V y tierra) y 2 de datos. Este transfiere señales y energía a los periféricos por medio de un conector apantallado o no, si la velocidad de transmisión de datos es de 12 Mbps ó 1,5 Mbps respectivamente. [16]

En la figura 2.22 se muestra un esquema del cable, con dos conductores para alimentación y los otros dos para señal, debiendo estos últimos ser trenzados o no según la velocidad de transmisión.

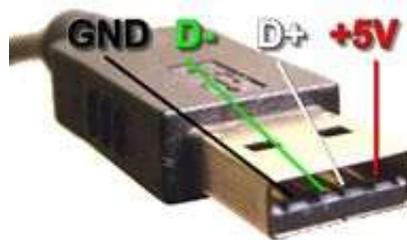


Figura 2.22 Cables de USB

Fuente: Franco E. (2010), documento en línea [17]

c. Convertidor RS 232 - USB

Debido a que la comunicación que soportan los microcontroladores PIC, es comunicación serial con protocolo RS 232, y que en las computadoras personales más recientes este tipo de conexión está en desuso, se hace necesario la adaptación a un protocolo de conexión disponible para las mismas. Ya que el protocolo de conexión más usado en las computadoras actualmente es el puerto USB.[16]



Circuito integrado DS1307

El integrado DS1307 es un reloj-calendario de tiempo real, de bajo consumo de potencia, codificación binaria para los datos decimales de tiempo y fecha, además una memoria de 56 bytes. Las direcciones y datos de este integrado son transferidas a través de un bus bidireccional I2C. El reloj/calendario provee segundos, minutos, horas, día, mes y año. El fin de mes es actualizado automáticamente incluso para aquellos meses de menos de 31 días e incluye corrección de año bisiesto. El reloj opera en formato militar de 24 horas o en formato de 12 horas con un indicador AM/PM. El DS1307 incluye internamente un circuito que detecta fallas de alimentación, este circuito cambia la alimentación a la fuente de reserva. Mientras el reloj opera con la fuente de reserva, la continuidad de tiempo y fecha no se ve afectada. [15]

TYPICAL OPERATING CIRCUIT

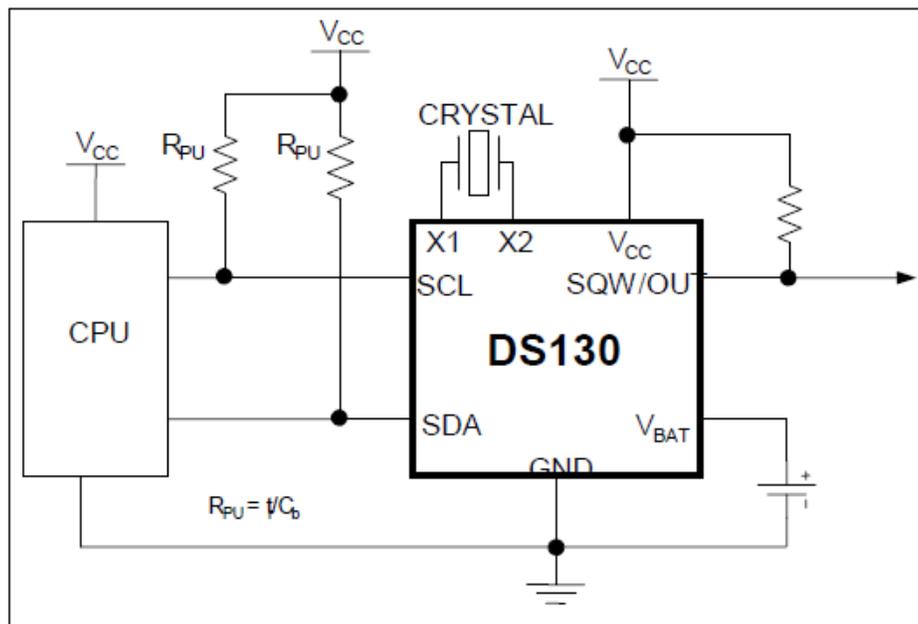


Figura 2.23 Circuito de operación del DS1307

Fuente: Vallejo H. (2009) [15].



PIN CONFIGURATIONS

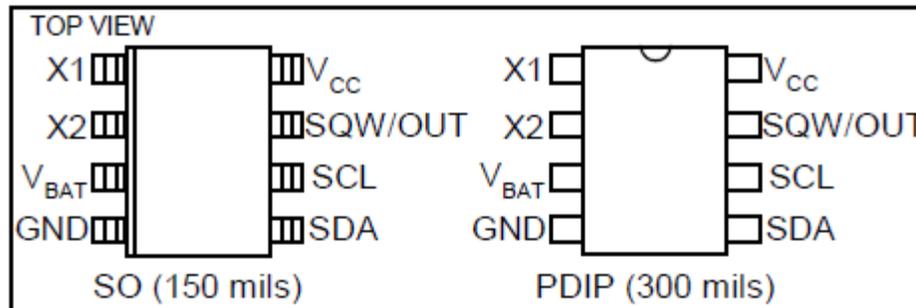


Figura 2.24 Configuración de los pines DS1307

Fuente: Vallejo H. (2009). [15]

2.2.5 Programación del PIC

a. Compiladores

Los compiladores son herramientas en las cuales el integrador, o persona que se encuentra desarrollando un proyecto con un *IC*, desarrolla el software del programa que ejecutará el integrado. Actualmente estas herramientas vienen en un entorno integrado, donde se brinda soporte en cada tópico de programación, “Ayuda”, y se encuentran comandos y librerías ya desarrolladas, los cuales configuran operaciones relacionadas al microcontrolador. Esto se denomina “Entorno de Desarrollo Integrado” *IDE* (*Integrated Development Environment* en inglés). En el mismo se incluye el software para un ensamblador, editor, simulador y programador, todo en un paquete dedicado. [15] En la figura 2.25 se muestra un ejemplo de este tipo, que es el entorno de programación del Proton Basic.

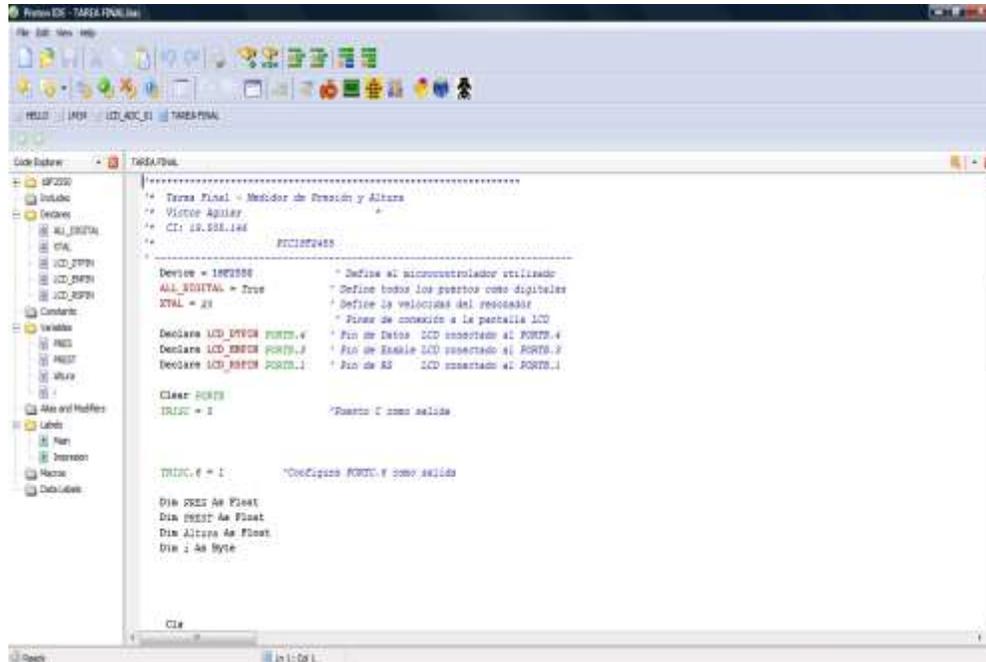


Figura 2.25 Entorno del Proton Basic

b. Simuladores

Simulador ISIS-Proteus

Es un programa para simular circuitos electrónicos complejos integrando inclusive desarrollos realizados con microcontroladores de varios tipos, en una herramienta de alto desempeño con unas capacidades graficas impresionantes.

Presenta una filosofía de trabajo semejante al *SPICE*, arrastrando componentes de una barra e incrustándolos en la aplicación. El Proteus puede simular diseños realizados para microcontroladores PIC incluyendo dispositivos electrónicos discretos e integrados (teniendo disponible en los últimos los más usados comercialmente. [18]

En la figura 2.26 se aprecia una imagen del entorno de Proteus

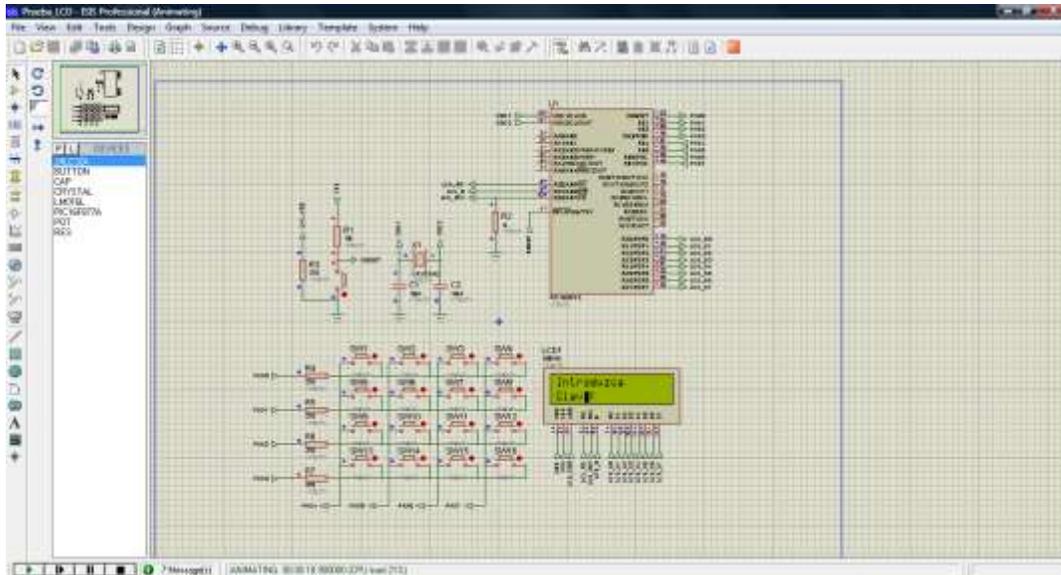


Figura 2.26 Entorno Proteus

2.2.6 Neumática

2.2.6.1 Principios Básicos

Es una tecnología capaz de hacer uso del aire comprimido para forzar el movimiento de mecanismos en procesos automatizados procesos de la industria. Para obtener el aire comprimido se sustrae aire del ambiente, el cual se encuentra a presión atmosférica, luego es filtrado, y finalmente elevado a determinada presión. Entre sus propiedades se pueden enumerar: 1) No volátil, 2) no explosivo, 3) de baja sensibilidad al aumento de la temperatura. La viscosidad del mismo se incrementa al aumentar su presión, lo cual incide directamente en el rozamiento de los elementos mecánicos, disminuyendo el rendimiento óptimo de toda la instalación neumática. Existe un rango óptimo de operación con el aire, el cual se encuentra entre 87 a 116 psi.

Al igual que la energía eléctrica, la energía neumática suele encontrarse fácilmente en cualquier taller o industria. Para ello generalmente se dispone de un compresor que



alimenta a diferentes puntos de la planta de producción, de manera de poder suministrar aire comprimido a cualquier dispositivo neumático que así lo requiera. El sistema neumático se compone de: 1) Compresor, 2) Pulmón de Aire, 3) Tuberías, 4) Purgas, 5) Válvulas, 6) Actuadores, 5) Unidades de mantenimiento, 6) Pulmón. [19]

2.2.6.2 Partes del Sistema

a. Válvulas

Son elementos que permiten controlar el arranque, parada, dirección, sentido y flujo del aire en un circuito neumático. [20] Según el tipo de construcción se pueden clasificar en:

- Válvulas de Asiento
- Válvulas de Corredera

Según su accionamiento hay varios tipos de válvula que son de vital importancia nombrar:

Electroválvulas neumáticas

Una electroválvula es un dispositivo diseñado para controlar el flujo de un fluido a través de un conducto como puede ser una tubería. No se debe confundir la electroválvula con válvulas motorizadas, que son aquellas en las que un motor acciona el cuerpo de la válvula.

Una electroválvula tiene dos partes fundamentales: el solenoide y la válvula. El solenoide convierte energía eléctrica en energía mecánica para actuar la válvula. Existen varios



tipos de electroválvulas. En algunas electroválvulas el solenoide actúa directamente sobre la válvula proporcionando toda la energía necesaria para su movimiento. Es común que la válvula se mantenga cerrada por la acción de un muelle y que el solenoide la abra venciendo la fuerza del muelle. Esto quiere decir que el solenoide debe estar activado y consumiendo potencia mientras la válvula deba estar abierta.

También es posible construir electroválvulas biestables que usan un solenoide para abrir la válvula y otro para cerrar o bien un solo solenoide que abre con un impulso y cierra con el siguiente. Las electroválvulas pueden ser cerradas en reposo o normalmente cerradas lo cual quiere decir que cuando falla la alimentación eléctrica quedan cerradas o bien pueden ser del tipo abiertas en reposo o normalmente abiertas que quedan abiertas cuando no hay alimentación. Hay electroválvulas que en lugar de abrir y cerrar lo que hacen es conmutar la entrada entre dos salidas. Este tipo de electroválvulas a menudo se usan en los sistemas de calefacción por zonas, lo que permite calentar varias zonas de forma independiente utilizando una sola bomba de circulación. [19]

Válvula de escape rápido

La válvula de escape rápido permite elevar la velocidad de los émbolos de cilindros. Con ella se ahorran largos tiempos de retornos, especialmente en cilindros de simple efecto. En la figura 2.27 se observa un esquema de la válvula. Esta tiene un empalme de alimentación P disponible para ser bloqueado, y una salida A. Cuando se le aplica presión al empalme P la junta se desliza y cubre el escape R. El aire comprimido circula entonces hacia A. Si se deja de aplicar aire comprimido a P, el aire proveniente de A empuja la junta contra el empalme P, cerrando éste. Puede escapar rápidamente por R, sin recorrer conductos largos y quizá estrechos, hasta la válvula de mando. Es recomendable acoplar la válvula de escape rápido directamente al cilindro, o bien lo más cercano posible. [19] La figura 2.28 muestra la representación de la válvula de escape rápido según la norma ISO.

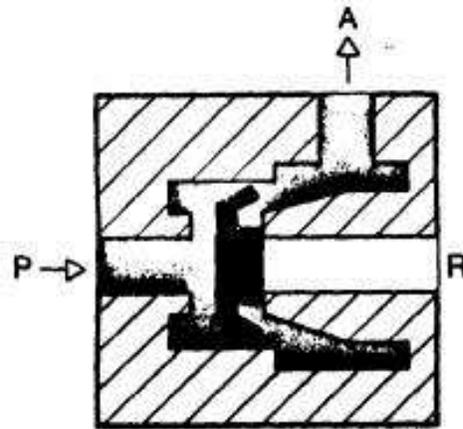


Figura 2.27 Esquema de válvula de escape rápido.
Fuente: Sapiensman (2012), documento en línea. [19]

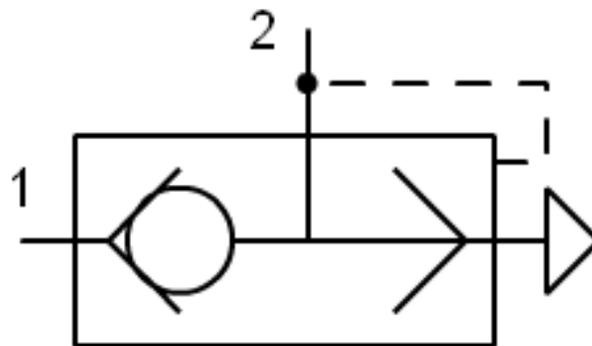


Figura 2.28 Diagrama de válvula de escape rápido.
Fuente: Bueno (2001), documento en línea. [21]



Válvula de 3/2 vías

Es un regulador de tipo distribuidor, de un circuito de aire comprimido, compuesto por tres orificios, el cual puede adaptar dos posiciones. Realiza la conexión entre tres señales neumáticas, conmutando entre dos estados. Este permite realizar la conmutación de un actuador por medio del cambio de dirección de las vías que interconecta. [19] En la figura 2.29 se muestra una muestra de la estructura de la válvula 3/2 vías.

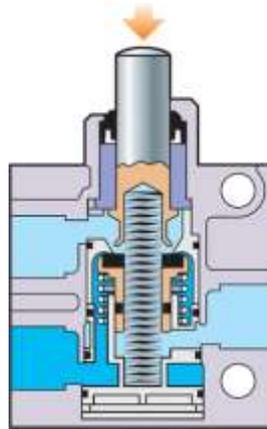


Figura 2.29. Válvula 3/2 Vías

Fuente: Kalipedia (2012), documento en línea. [22]

Cilindro Neumático

Dispositivo mecánico que permite el desplazamiento de un émbolo a través de la aplicación de presión por medio de uno de sus extremos, dependiendo del movimiento requerido [19]. Para realizar su función, se le imparte una fuerza para convertir la energía potencial de gas comprimido en energía cinética (en movimiento). Esto se alcanza por medio del gas comprimido, que es debido a la diferencia de presión. Esta diferencia o gradiente de presión del aire acciona un pistón para moverse en la dirección deseada.



Una vez que esté actuado, el aire comprimido entra en el tubo por un extremo del pistón y, por lo tanto, imparte la fuerza a través del pistón. Por lo tanto, el pistón se desplaza por el aire comprimido que se amplía en un intento por alcanzar presión atmosférica. [23]

Los cilindros neumáticos pueden funcionar en una variedad de maneras. Los ejemplos incluyen tener la capacidad de realizar movimientos múltiples sin la necesidad de la intervención de un operador, de realizar un movimiento completo. Aunque los cilindros neumáticos variarán de aspecto, tamaño y la función, Dentro este tipo de cilindros, los más usados son:

➤ Cilindro de simple efecto: genera un movimiento rectilíneo, en la carrera del émbolo, que puede ser de avance y de retroceso. En este tipo de cilindro, el aire comprimido sólo actúa sobre una de las caras del émbolo, y por tanto, sólo se puede producir trabajo en un sentido de carrera del émbolo. [19] En la figura 2.30 se puede observar este tipo de cilindro

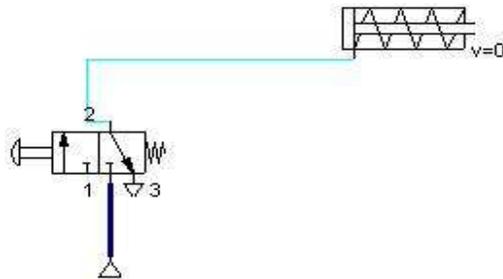


Figura 2.30 Cilindro de simple efecto.

Fuente: Martín (2007), documento en línea [23]



➤ Cilindro de doble efecto: El cilindro de doble efecto genera un movimiento rectilíneo (carrera del émbolo) que puede ser de avance y de retroceso. Este tipo de cilindro puede producir trabajo en los dos sentidos de carrera del émbolo, ya que posee dos tomas de aire comprimido, situadas a ambos lados del émbolo [19]. En la figura 2.31 se puede observar este tipo de cilindro

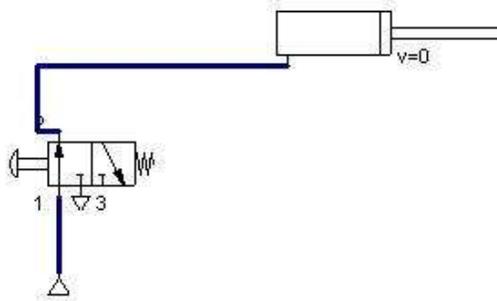


Figura 2.31 Cilindro de doble efecto.

Fuente: Martín (2007), documento en línea [23]

b. Generador de Vacío

Es el encargado en el sistema neumático de generar vacío o depresión de aire basándose en el principio del eyector o efecto Venturi. Estos generadores necesitan la presión de la red, y lo por tanto, de un compresor general.

En la figura 2.32 puede observarse el principio de funcionamiento del generador de vacío. El aire a presión procedente de la red entra por (1) y al llegar a (2) se estrangula, aumentando notablemente su velocidad. Este aumento de velocidad crea una depresión en la cámara (3) y en el conducto (5), que se aprovecha para conectar las ventosas de vacío. El aire a presión generador del vacío sale expulsado al exterior a través del conducto (4) [20].

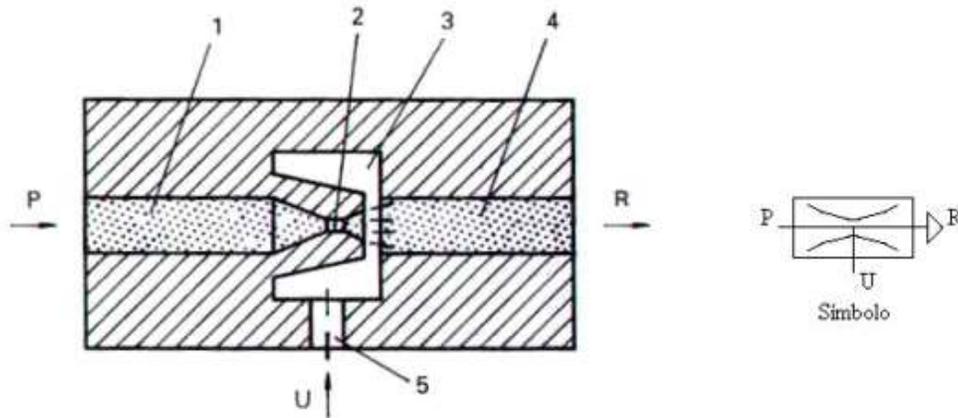


Figura 2.32 Generador de Vacío

Fuente: Escobar (2006), documento en línea [20]

Existen muchas aplicaciones en donde la utilización de la técnica del vacío implica el uso de las llamadas ventosas.

c. Ventosa

Su función principal es adherirse a los objetos que serán manipulados. Tal y como lo indica la figura 2.33, la ventosa se compone esencialmente de un cuerpo metálico (1), el cual está roscado en su parte superior, y en su parte inferior está unido a una junta de caucho en forma de vaso (2), cuyos labios (5) producen la estanqueidad cuando entran en contacto con la pieza (4) a sujetar. El vacío (U) creado en el conducto (6) y en la cámara (3), hace que la pieza se adhiera a los labios de la ventosa.

En resumen, la ventosa tiene por objetivo fundamental crear una cámara de vacío con la pieza a sujetar, de forma tal que la adherencia que se produzca entre los labios de la ventosa y la pieza sea capaz de soportar el peso de esta última. [20]

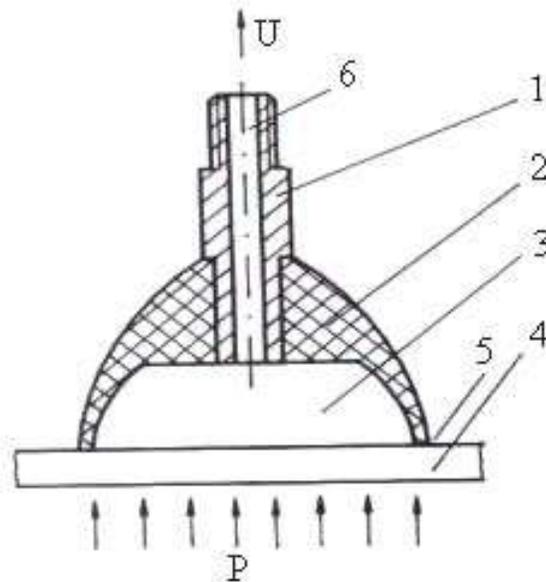


Figura 2.33. Ventosa

Fuente: Escobar (2006), documento en línea [20]

a. Unidades de mantenimiento

También conocidas como unidades FRL (Filtro, Regulador, Lubricador), las unidades de mantenimiento son aquellas que permiten una circulación de aire con las características ideales para los dispositivos que utilizados en algún circuito neumático.

La operación de esta unidad inicia en los filtros de aire, los cuales retienen las partículas sólidas y las gotas de humedad contenidas en el aire. Inicialmente pasa por los filtros ciclónicos. En estos, al entrar, el aire pasa a través de placas que fuerzan una circulación rotativa, de esta manera las grandes partículas sólidas y el líquido se depositan en las paredes del vaso o copa, por la acción centrífuga. Posteriormente el aire atraviesa el elemento filtrante principal, de malla metálica, papel, o metal sinterizado. Esta acción de



filtrado se denomina "mecánica" ya que, afecta a la contaminación mecánica del aire, y no a su contenido de humedad.

Las partículas más grandes, son retenidas por el filtro sinterizado, mientras que los líquidos son desviados al vaso del filtro. El líquido condensado en el vaso o copa del filtro se debe vaciar periódicamente, ya que si no podría ser arrastrado por la corriente del aire comprimido al circuito. Los filtros más finos, de hasta 0.01 micras, se encargan de filtrar las partículas más pequeñas e incluso mínimas gotas de agua que pudieran quedar en el aire comprimido.

Para hacer independiente la salida neumática del FRL de las variaciones en la red, se encuentra la válvula reguladora de presión. La cual mantiene la presión de trabajo constante en el lado del usuario. Para alcanzar ese objetivo, la presión de entrada del regulador debe ser siempre superior a la de trabajo.

El lubricador del aire comprimido sirve para la lubricación de los elementos neumáticos, especialmente los activos. El aceite que se utiliza en la lubricación es aspirado de un pequeño depósito de la misma unidad de mantenimiento, mezclado con la corriente del aire comprimido, y distribuido por medio de atomización. [24]

En la figura 2.34 se observa un ejemplo de FRL. Así mismo, en la figura 2.35 se muestra el esquema representativo de la unidad en norma ISO.

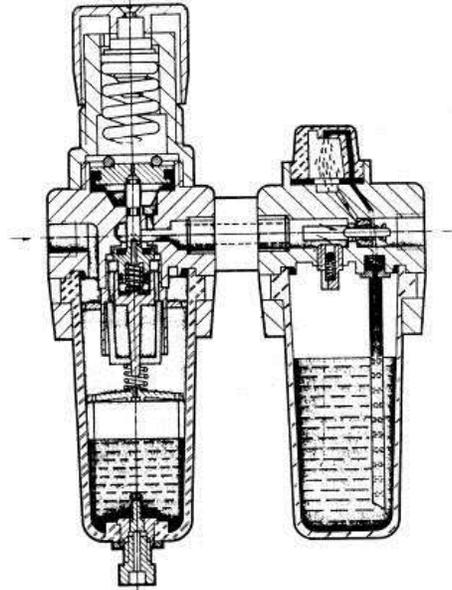


Figura 2.34. Sistema de FRL

Fuente: Martín (2007), documento en línea [23]

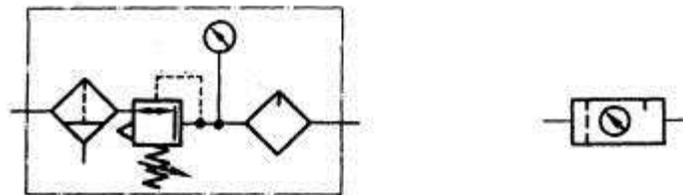


Figura 2.35. Simbología de la unidad de mantenimiento

Fuente: Sapiensman (2012), documento en línea [19]

2.2.7 Señales

Las señales constituyen una herramienta fundamental en el proceso de adquisición de datos de un sistema. Para el diseño de un dispositivo de control de un sistema es necesario



conectarlo a señales que le indiquen el estado de las variables a controlar en el sistema. Es por ello que el uso de sensores para la adquisición de datos se vuelve una herramienta imperativa. [25]

2.2.7.1 Sensores

Instrumento que detecta y señala una condición de cambio. Se puede contar con una detección discreta, como una condición de cambio, o bien una detección analógica, la cual sea capaz de sensar la variación de una magnitud física. Los sensores permiten la comunicación entre el mundo físico o campo, y los sistemas de medición y/o de control, tanto eléctricos como electrónicos, utilizándose extensivamente en todo tipo de procesos industriales y no industriales para propósitos de monitoreo, medición, control y procesamiento. [19]

Se pueden distinguir algunos tipos de sensores, entre los cuales destacan: inductivos, capacitivos, magnéticos y fotoeléctricos. A continuación se describirá el sensor fotoeléctrico, por ser el de mayor relevancia en el presente trabajo, resaltando antes que, la distancia nominal de detección varía de acuerdo al sensor:

- a) Sensores de Barrera. Cuando existe un receptor y un emisor apuntados uno al otro. Tiene este método el más alto rango de detección
- b) Sensores Reflex. Cuando la luz es reflejada por un reflector especial cuya particularidad es que devuelve la luz en el mismo ángulo que la recibe.
- c) Sensores Auto Reflex. Cuando el emisor tiene un lente que polariza la luz en un sentido y el receptor otro que la recibe mediante un lente con polarización a 90° del primero. Con esto, el control no responde a objetos muy brillosos que pueden reflejar la señal emitida.



- d) Sensores de Foco Fijo. Cuando la luz es reflejada difusamente por el objeto y es detectado por el hecho de que el transmisor y el receptor están estereoscópicamente acoplados, evitando con ello interferencia del fondo.
- e) Sensores de detección difusa. Iguales a los anteriores pero los lentes son divergentes, y se usan para detectar objetos muy próximos.
- f) Sensores de Fibra Óptica. En este tipo, el emisor y receptor están interconstruídos en una caja que puede estar a varios metros del objeto a detectar. Para la detección emplean los cables de fibra óptica por donde circulan los haces de luz emitido y recibido. La mayor ventaja de estos sensores es el pequeño volumen o espacio ocupado en el área de detección. [20]

Detalles sobre los sensores para robots

Existe una amplia variedad de dispositivos diseñados para percibir la información externa de una magnitud física y transformarla en un valor electrónico que sea posible introducir al circuito de control, de modo que el robot sea capaz de cuantificarla y reaccionar en consecuencia.

Entre las magnitudes físicas que son necesarias medir para que un robot tenga algún conocimiento del entorno están la luz, el sonido y el ultrasonido, la gravedad (inclinación, posición), la temperatura, la humedad, la presión, la velocidad, la proximidad y la distancia. [26]

2.2.7.2 Encoder

En sistemas mecánicos en donde se implementan servomecanismos, para poder controlar el movimiento del mismo es necesario controlar el movimiento de los elementos que los accionan. Dado que los servomecanismos están integrados por motores de corriente



continua, estos cambian su velocidad según la carga mecánica que tienen aplicada. Es por ello que se necesita utilizar un sistema de medición cuya salida de señal, realimentada en el circuito de manejo de potencia, sea proporcional a la velocidad. La solución aplicada a este caso, es un *encoder*. Este es un artefacto que genera una señal proporcional a la velocidad real del motor con la máxima precisión posible. Si se aplica una carga mecánica al eje del motor de corriente continua y la velocidad disminuye, esta disminución se refleja en la señal del mismo. [27] En la figura 2.36 se observa un disco de *encoder*.



Figura 2.36 Disco de *encoder*.

Fuente: Robots Argentina (2012), documento en línea [27]

El *encoder* consta de un disco con ranuras o franjas alternadas de color claro y oscuro, con un sistema óptico de lectura que entrega pulsos al enfrentarse las ranuras o bandas sobre un fotosensor. La señal producida por el sensor óptico, que básicamente es un pulso, debe ser procesada digitalmente. Para luego ser usada con el fin deseado, que puede ser control realimentado de la velocidad del motor, o bien visualización del valor de la velocidad por medio de una pantalla *LCD*. [26]



En la figura 2.37 se observa un motor de CC con *encoder* en sus ranuras.



Figura 2.37 Motor de CC con *encoder* en ranuras
Fuente: Robots Argentina (2012), documento en línea [27]

2.2.8 Robots

2.2.8.1 Definición

El robot es una máquina capaz de interactuar con su entorno. En caso de ser móvil, a menos que se mueva en un espacio absolutamente acotado y preparado para él, deberá ser capaz de adaptar sus movimientos y sus acciones de interacción en base a las características físicas de los ambientes con los que se encuentre y los objetos que hay en ellos.

Para lograr esta capacidad de adaptación, lo primero que necesitan los robots es tener conocimiento del entorno. Para ello, estos deben poseer sensores que les permitan definir ciertos parámetros como: 1) Determinar su ubicación, 2) Cómo es el lugar en el que están, 3) A qué condiciones físicas se enfrentan, 4) Dónde están los objetos con los que deben interactuar, 5) sus parámetros físicos. [26]



2.2.8.2 Estructura

Está formado por los siguientes elementos: estructura mecánica, transmisiones, actuadores, sensores, elementos terminales y controlador.

La estructura física de la mayor parte de los robots industriales tiene cierta similitud con la anatomía de las extremidades superiores del cuerpo humano, por lo que, en ocasiones, para hacer referencia a los distintos elementos que componen el robot, se usan términos como cintura, hombro, brazo, codo, muñeca, etc. [26] En la figura 2.38 se puede apreciar la relación de Morfología del Robot con la morfología de las extremidades humanas.

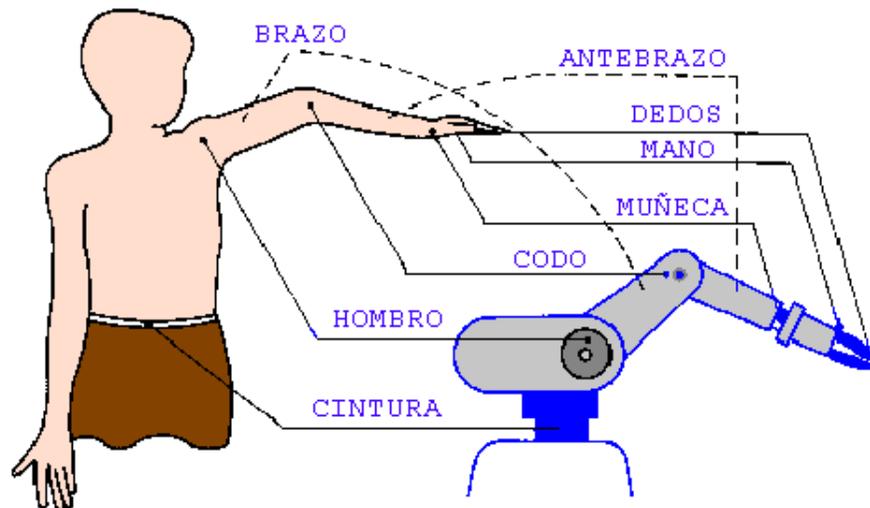


Figura 2.38 Morfología del Robot.

Fuente: CFIE (2012), documento en línea [26]

Los elementos que forman parte de la totalidad del robot son: manipulador, controlador, dispositivos de entrada y salida de datos, y dispositivos especiales. [26]



Manipulador

Desde el punto de vista mecánico es el elemento más importante del robot. Se constituye por una serie de elementos estructurales sólidos o eslabones unidos mediante articulaciones los cuales hacen posible el movimiento relativo entre cada dos eslabones consecutivos. [26] En la figura 2.39 se observa la estructura del manipulador.

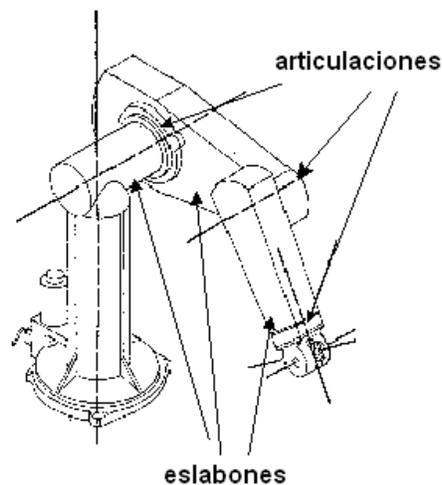


Figura 2.39 Estructura del manipulador.

Fuente: Fuente: CFIE (2012), documento en línea [26]

Sus partes son: cuerpo, brazo, muñeca, actuador final, también denominado elemento terminal, pinza, o *gripper*. Cada articulación provee al robot de, al menos, un grado de libertad. Estas articulaciones permiten realizar dos tipos de movimientos: Lineales y Angulares. Así mismo hay dos tipos de articulación utilizados en las juntas del manipulador: 1) Prismática o Lineal. En esta junta, el eslabón se apoya en un deslizador lineal. Actúa linealmente mediante los tornillos sinfín de los motores, o los cilindros. 2) Rotacional. Junta giratoria generalmente manejada por los motores eléctricos y las transmisiones, o por los cilindros hidráulicos y palancas. [26]



Controlador

Dispositivo encargado de manejar los movimientos del manipulador, las acciones, cálculos y procesado de la información. El controlador recibe y envía señales a otras máquinas y almacena programas. [26]

Dispositivos de entrada/salida

Se comprenden principalmente por teclado, monitor y caja de comandos (*teach pendant*). En la figura 2.40 se tiene un controlador (*computer module*) que envía señales a los motores de cada uno de los ejes del robot y la caja de comandos (*teach pendant*) la cual sirve para enseñarle las posiciones al manipulador del robot. [26]

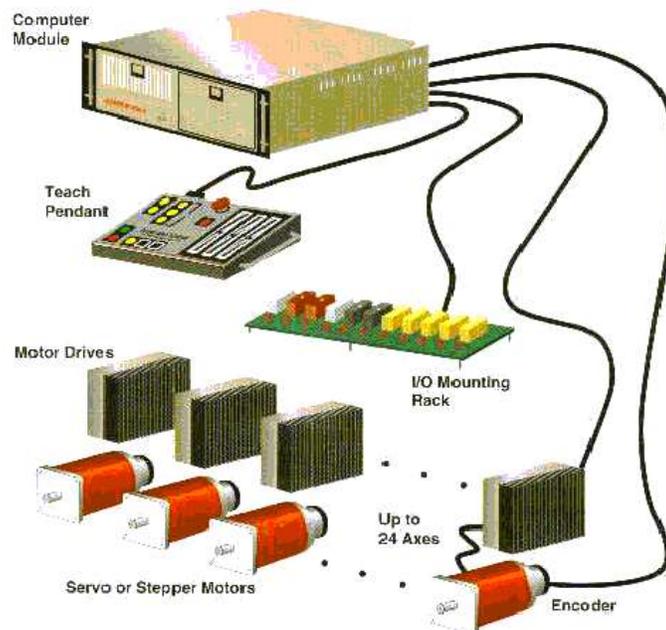


Figura 2.40 Conexión de computador con actuadores y teach pendant

Fuente: CFIE (2012), documento en línea [26]



La figura 2.41 muestra un *teach pendant* para un modelo comercial de robot industrial.



Figura 2.41 Teach Pendant

Fuente: Direct Industry (2012), sitio web [28]

Los dispositivos de entrada y salida permiten introducir y ver los datos del controlador. Para mandar instrucciones al controlador y para dar de alta programas de control, comúnmente se utiliza una computadora adicional. Hay casos en los que algunos robots poseen uno de estos componentes únicamente. En estos casos, uno de los componentes de entrada y salida permite la realización de todas las funciones.

Las señales de entrada y salida se obtienen mediante tarjetas electrónicas instaladas en el controlador del robot las cuales le permiten tener comunicación con otras máquinas-herramientas.

Se pueden utilizar estas tarjetas para comunicar al robot, por ejemplo, con las máquinas de control numérico CNC. Estas tarjetas se componen de relevadores, los cuales mandan señales eléctricas que después son interpretadas en un programa de control. Estas señales



nos permiten controlar cuándo debe entrar el robot a cargar una pieza a la máquina, cuando deben empezar a funcionar la máquina o el robot, etc. [26]

2.2.8.3 Tipos de Robots según su morfología

La estructura del manipulador y la relación entre sus elementos proporcionan una configuración mecánica, que da origen al establecimiento de los parámetros que hay que conocer para definir la posición y orientación del elemento terminal. Fundamentalmente,

Se tienen cuatro estructuras clásicas en los manipuladores relacionadas con los modelos correspondientes de coordenadas en el espacio. Estas son: cartesianas, cilíndricas, esféricas, angulares. De acuerdo a estas estructuras, el brazo del manipulador puede presentar seis configuraciones:

- Cartesiana
- Cilíndrica
- Asférica
- Antropomórfica
- SCARA (*Selective Compliance Assembly Robot Arm*).

Implementando diversas combinaciones de articulaciones en un robot, se obtienen distintas configuraciones, que arrojan características a considerar tanto en el diseño y construcción del robot como en su aplicación. Las combinaciones más frecuentes son con tres articulaciones, que son las más importantes a la hora de posicionar su extremo en un punto en el espacio. A continuación se presentan las características principales de las configuraciones del brazo manipulador.

- Cartesiana: Teniendo tres grados de libertad, esta configuración es ideal para aplicaciones que utilicen gran espacio, cuya exactitud consista en la espera del



robot. Tiene tres movimientos lineales los cuales corresponden a los movimientos localizados en los ejes X, Y, y Z.

Los movimientos que realiza este robot entre un punto y otro son con base en interpolaciones lineales. Interpolación, en este caso, significa el tipo de trayectoria que realiza el manipulador cuando se desplaza entre un punto y otro. A la trayectoria realizada en línea recta se le conoce como interpolación lineal y a la trayectoria hecha de acuerdo con el tipo de movimientos que tienen sus articulaciones se le llama interpolación por articulación. [26] En la figura 2.42 se observa un esquema y un ejemplo de robot cartesiano.



Figura 2.42 Ejemplo de robot Cartesiano

Fuente: CFIE (2012), documento en línea [26]

- Cilíndrica: Con tres grados de libertad, capacidad para realizar dos movimientos lineales y uno rotacional, permite un movimiento de rotación sobre una base, una articulación prismática para la altura, y una prismática para el radio. Los espacios de trabajo circulares son ideales para tal configuración.



Bajo este diseño, su operación es óptima en la ejecución de movimientos conocidos como interpolación lineal e interpolación por articulación. La interpolación por articulación se lleva a cabo por medio de la primera articulación, ya que ésta puede realizar un movimiento rotacional. [26] En la figura 2.43 se observa un esquema y un ejemplo de robot cilíndrico.



Figura 2.43 Ejemplo de robot Cilíndrico.

Fuente: CFIE (2012), documento en línea [26]

- Esférica / Polar: tiene los movimientos: rotacional, angular y lineal. La configuración usa la interpolación por articulación para moverse en sus dos primeras articulaciones y la interpolación lineal para la extensión y retracción. Dos juntas de rotación y una prismática permiten al robot apuntar en muchas direcciones, y extender la mano a un poco de distancia radial. [26].



En la figura 2.44 se observa la estructura de un robot esférico.

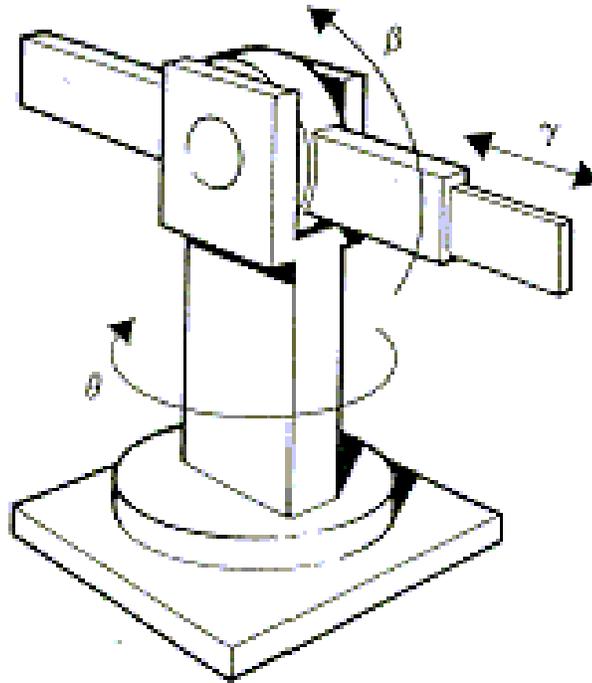


Figura 2.44 Estructura de robot Esférico.

Fuente: CFIE (2012), documento en línea [26]

- Antropomórfico: Se posiciona usando tres (3) juntas de rotación. Generalmente, el volumen de trabajo es esférico. Similar al brazo humano, presenta una articulación con movimiento rotacional y dos angulares. Aunque el brazo articulado puede realizar el movimiento llamado interpolación lineal para lo cual requiere mover simultáneamente dos o tres de sus articulaciones, el movimiento natural es el de interpolación por articulación, tanto rotacional como angular. [26]



En la figura 2.45 se observa un ejemplo de robot de brazo articulado.



Figura 2.45 Robot de Brazo Articulado.

Fuente: CFIE (2012), documento en línea [26]

- SCARA: Permite realizar movimientos horizontales de mayor alcance debido a sus dos articulaciones rotacionales, así mismo puede hacer un movimiento lineal mediante su tercera articulación. Es parecido al de configuración cilíndrica, pero el radio y la rotación se obtiene por uno o dos eslabones. [26]



En la figura 2.46 se observa un esquema y un ejemplo de robot de brazo articulado.

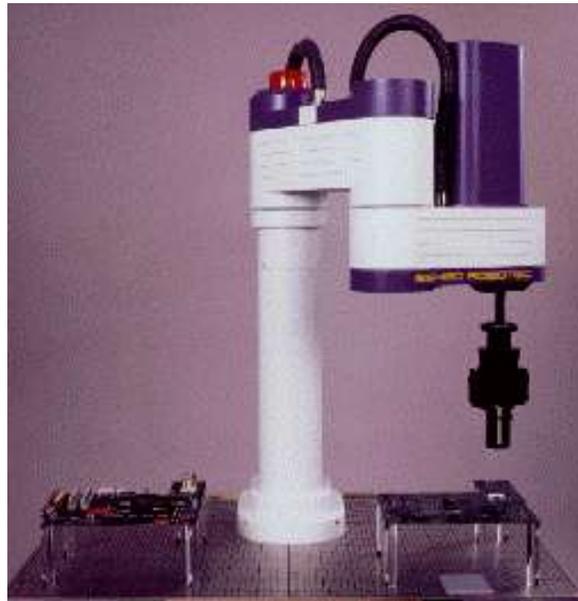


Figura 2.46 Robot SCARA.

Fuente: CFIE (2012), documento en línea [26]

2.2.8.4 Cinemática de los Robots

Definición del problema cinemático directo

Se conoce con esta denominación al dilema de determinar la posición final de elemento terminal de un Robo Manipulador en relación con un sistema de referencia fijo, sobre la base del conocimiento previo de sus articulaciones y los parámetros geométricos de sus elementos. Es decir, dada una posición inicial del efector final de un robot, se desea determinar su posición luego de aplicar distintas acciones que dependen de su configuración particular.



La geometría y configuración referidas en el párrafo anterior son el resultado del tipo de eslabones y articulaciones que componen al robot. El punto final de un eslabón articulado puede realizar un movimiento respecto al eslabón anterior, susceptible de descripción mediante una matriz de Transformación Homogénea, para lo cual se requiere escoger un sistema de referencia unido con cada enlace en forma indispensable. La nomenclatura aquí utilizada para hablar del movimiento relativo entre eslabones nombra la Matriz de Transformación Homogénea (Matriz de Paso Homogénea) con la letra A, un súper índice para indicar al eslabón que sirve de base al movimiento relativo entre dos elementos; y un subíndice para señalar la localización del sistema de referencia asociado al elemento 1, respecto al sistema de referencia asociado al elemento 0 (base del robot); A_2^1 , la de 2 respecto a 1 y A_i^{i-1} , de i-ésimo respecto al (i-1)-ésimo.

Para un robot con seis articulaciones y siete elementos (Incluyendo la base), por ejemplo para la posición del elemento final referida a la base, se describe mediante un conjunto de post-multiplicaciones matriciales equivalentes a una sola matriz de Transformación (T):

$$T = A_1^0 \times A_2^1 \times A_3^2 \times A_4^3 \times A_5^4 \times A_6^5 \quad (2.1)$$

La caracterización de un Robot Manipulador a través de la relación entre sus elementos y articulaciones se conoce como Cadena Cinemática (Richard Paul, 1981). Con ella se pretende asignar cantidades que permiten definir el movimiento relativo entre cuerpos consecutivos (coordenadas generalizadas) y generar un modelo del Robot Manipulador (modelo geométrico) para aplicar un método de obtención sistemática de la situación del efector final respecto a un sistema de referencia fijo. Ese método es conocido como Algoritmo DH, siglas de Denavit y Hartenberg, quienes lo propusieron en 1955. [29]



Parámetros Denavit-Hartenberg

El Algoritmo de Denavit y Hartenberg fundamenta su utilidad en una forma específica para escoger los sistemas de coordenadas y determinar un conjunto de parámetros (parámetros DH), lo cual será objeto de tratamiento detallado en el resto de esta sección.

En principio, cada elemento de una cadena cinemática está asociado a dos articulaciones y pueden describirse mediante dos parámetros: la distancia normal (a) a los ejes de las articulaciones y un ángulo de torsión (α). En la figura 2.47 se aprecian los parámetros anteriormente mencionados, en relación con los ejes de las articulaciones.

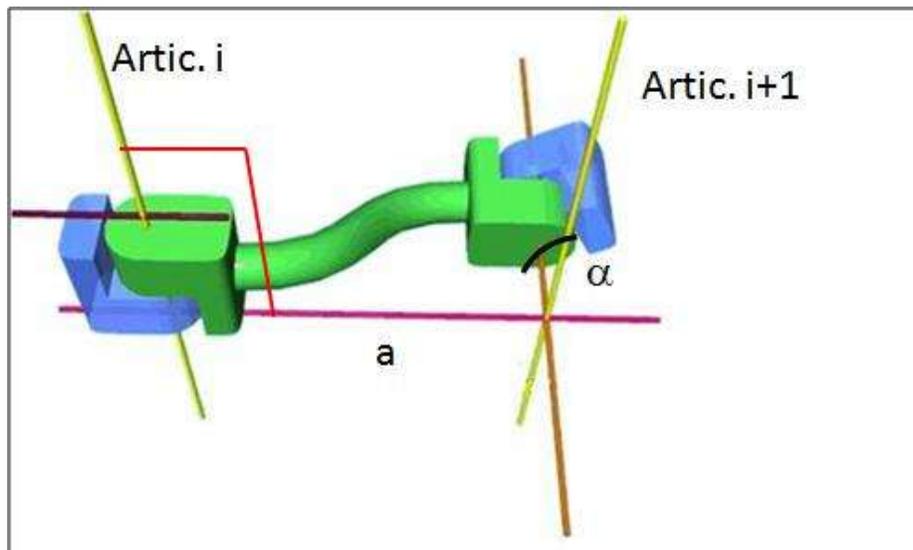


Figura 2.47 Parámetros a y α .

Fuente: Denavit y Hartenberg, 1955. [29]

Es importante notar que el ángulo α , no está en el plano de la vista frontal de la figura 2.47. Se ha intentado resaltar que la distancia “ a ”, es normal a ambos ejes, pero que los planos definidos por ella y cada eje son diferentes.



El eje de cada articulación comparte dos líneas normales (\mathbf{a}) con ejes contiguos. La distancia entre ambas normales, en la dirección del de la articulación, se conoce como " \mathbf{d} ". Finalmente la distancia angular entre las dos normales, medida según el eje de la articulación se denomina θ .

En resumen, los parámetros DH son cuatro: θ , \mathbf{d} , \mathbf{a} , y α . Los dos últimos (\mathbf{a} y α) definen las características de cada eslabón o elemento de la cadena cinemática, mientras que los dos primeros (θ y \mathbf{d}) determinan su relación con el elemento anterior. [29]

Matrices de paso homogéneas

Como se dijo en un párrafo anterior, una Matriz de paso Homogénea describe el movimiento relativo entre dos eslabones de una cadena cinemática. Tal descripción permite relacionar el sistema de referencia del elemento " i " con el del elemento " $i-1$ ". Ahora bien, la localización de un punto del elemento " i " (asociado a un sistema tomado como móvil), puede definirse respecto al sistema del elemento previo ($i-1$), tomado como sistema fijo, mediante una Transformación Homogénea que considere cuatro eventos:

1. Rotación alrededor del eje Z_{i-1} en un ángulo θ_i
2. Traslación a lo largo de Z_{i-1} , una distancia d_i
3. Traslación a lo largo del eje X_i , una distancia a_i
4. Rotación alrededor del eje X_i , un ángulo α_i

Estos cuatro movimientos permiten llevar un sistema de ejes desde una condición plenamente coincidente con un sistema de referencia (fijo) ubicado en la articulación " $i-1$ ", hasta una situación final en la que coincide con el sistema de articulación " i ". Como cada movimiento se realiza a partir de la condición alcanzada con el movimiento previo, el modelo explicativo de todo procedimiento se obtiene mediante



Transformaciones que se post-multiplican (relativas), tal como se muestra a continuación [29]:

$$A_i^{i-1} = T(z, \theta_i) \times T(0, 0, a_i) \times T(d_i, 0, 0) \times T(x, \alpha_i) \quad (2.2)$$

$$A_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$A_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \times \sin \theta_i & \sin \alpha_i \times \sin \theta_i & a \times \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \times \cos \theta_i & -\sin \alpha_i \times \cos \theta_i & a \times \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Convenciones de Denavit-Hatemberg

Además de la definición de los parámetros que llevan su nombre, Denavit y Hartenberg establecieron un conjunto de convenciones sobre los sistemas de ejes asociados que permiten sistematizar el análisis de los Robots Manipuladores. A continuación se listan y comentan tales convenciones:

1. Los eslabones se enumeran crecientemente, desde la base hacia el efector final, asignando el n°1 al eslabón que se encuentre unido a la base. La base recibirá el n° cero (0).
2. Las articulaciones se enumeran también en forma creciente comenzando por uno (1), desde la que une la base con el primer elemento hasta la que sirve de junta al elemento terminal.



3. Localizar los ejes de las articulaciones en las rotoides el eje será el eje de giro y en las prismáticas, el eje de desplazamiento.
4. Definir los ejes Z_i . Empezando desde la base, cada Z_i estará ubicado en el eje de la articulación correspondiente al extremo final del elemento (articulación $i+1$)
5. El origen del sistema de coordenadas de la base (S_0), se escogerá arbitrariamente, como cualquier punto del eje Z_0 . También será arbitrado el sentido de X_0 , pero Y_0 quedará determinado por las reglas del producto vectorial, tal que $Z = X \times Y$.
6. El origen de cada sistema " S_i " quedará determinado por la intersección de la normal (a) común a cada eje Z_i y Z_{i-1} . El eje X_i estará en la dirección de esa normal y el correspondiente Y_i cumplirá con la regla de todo sistema dextrógiro (regla de la mano derecha).
7. El sistema " S_n " se ubicará en el elemento terminal del Robot y el eje Z_n conservará la dirección Z_{n-1} .
8. θ_i será el ángulo en que debe girarse el eje X_{i-1} , alrededor de Z_{i-1} , para quedar paralelo con X_i .
9. d_i será la distancia que deberá desplazarse al eje X_{i-1} , en la dirección de Z_{i-1} , para quedar paralelo con X_i .
10. a_i será la distancia que deberá desplazarse el sistema S_{i-1} , en la dirección de X_i , para que su origen coincida con el de S_i .
11. α_i será el ángulo en que debe girarse a Z_{i-1} , alrededor de X_i , para quedar alineado con Z_i .
12. Deben obtenerse tantas Matrices de Paso Homogéneas como elementos existen en la cadena cinemática, incluyendo la matriz, correspondiente a la base.
13. La Matriz de Transformación del Robot se obtendrá como el producto de todas las Matrices de Paso Homogéneas:

$$T = A_1^0 \times A_2^1 \times \dots \times A_n^{n-1} \quad (2.5)$$



En resumen, la solución del problema cinemático directo es la Matriz de Transformación del Robot (T), la cual se encuentra mediante las Matrices de Paso Homogéneas determinadas sobre la base de los parámetros DH de cada eslabón. Con estas herramientas y el control de los parámetros de las articulaciones se logra posicionar el elemento terminal donde se decida. [29]

2.2.8.4.1 Cinemática del Robot Antropomórfico

El robot antropomórfico es una configuración muy común en los manipuladores industriales, 5 Ejes Rotacionales. Esta configuración es de especial interés, puesto que el robot a utilizar en el desarrollo del proyecto sigue estas especificaciones. Para poder apreciar de mejor forma el funcionamiento del mismo se mencionarán los parámetros correspondientes a tal configuración:

Se toma un modelo de robot construido por el Profesor Jesús Pérez como parte de su Tesis doctoral de Postgrado en Instrumentación de la Facultad de Ciencias de la Universidad Central de Venezuela, denominado Robot Initium. [29]

La figura 2.48 ilustra un modelo semejante al indicado anteriormente, este posee 5 ejes, tal como el indicado anteriormente, y pertenece al Diagrama de Ejes del Manual del Robot RV-2AJ, perteneciente a la marca Mitsubishi-FESTO.

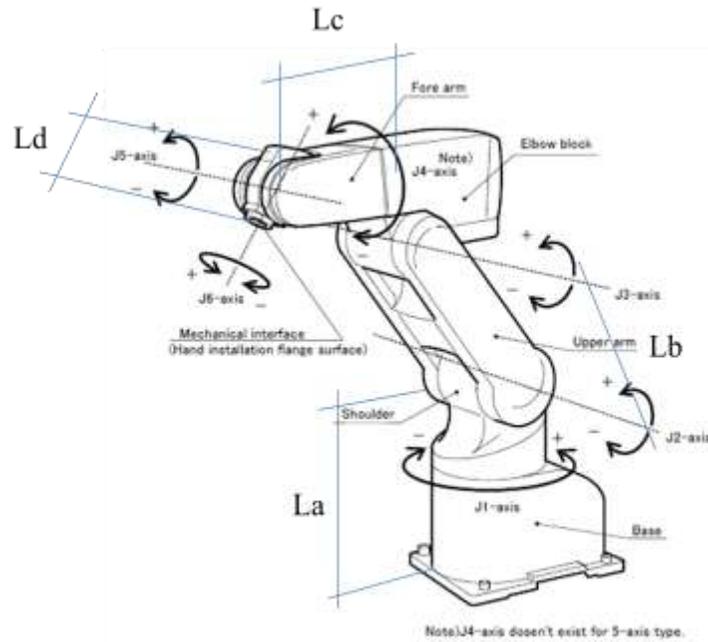


Figura 2.48 Robot Antropomórfico de 5 ejes RV-2AJ Mitsubishi-FESTO.

Fuente: Manual RV-2AJ (2012), documento en línea. [30]

Parámetros DH

Tabla 2.3 Parámetros DH del robot antropomórfico de 5 ejes

Articulación	θ	d	a	α	q
1	q_1	La	0	90	θ_1
2	q_2	0	Lb	0	θ_2
3	q_3	0	Lc	0	θ_3
4	q_4	0	0	-90	θ_4
5	q_5	Ld	0	0	θ_5

Fuente: Cinemática de Robots Industriales. Sanz, W.



Matrices de paso homogéneas

Ecuación general

$$A_i^{i-1} = \begin{bmatrix} \text{Cos}\theta & \text{Cosa} \times \text{Sen}\theta & \text{Sena} \times \text{Sen}\theta & a \times \text{Cos}\theta \\ \text{Sen}\theta & \text{Cosa} \times \text{Cos}\theta & -\text{Sena} & a \times \text{Sen}\theta \\ 0 & \text{Sena} & \text{Cosa} & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Matrices Particulares:

$$A_1^0 = \begin{bmatrix} \text{CosI} & 0 & \text{SenI} & 0 \\ \text{SenI} & 0 & -\text{CosI} & 0 \\ 0 & 1 & 0 & \text{La} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$A_2^1 = \begin{bmatrix} \text{CosII} & -\text{SenII} & 0 & \text{LbCosII} \\ \text{SenII} & \text{CosII} & 0 & \text{LbSenII} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$$A_3^2 = \begin{bmatrix} \text{CosIII} & -\text{SenIII} & 0 & \text{LcCosII} \\ \text{SenIII} & \text{CosIII} & 0 & \text{LcSenII} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

$$A_4^3 = \begin{bmatrix} \text{CosIV} & 0 & -\text{SenIV} & 0 \\ \text{SenIV} & 0 & \text{CosIV} & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

$$A_5^4 = \begin{bmatrix} \text{CosV} & -\text{SenV} & 0 & 0 \\ \text{SenV} & \text{CosV} & 0 & 0 \\ 0 & 0 & 1 & \text{Ld} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$



Elementos de la matriz T

Del producto de las matrices de Paso Homogéneas se obtiene la matriz T. Dado su tamaño, es necesario mostrar sus elementos de forma desglosada. Sin embargo, para reducir las doce expresiones anteriores se puede valer de identidades trigonométricas correspondientes a la suma y diferencia de ángulos. De éste procedimiento resulta el modelo cinemático directo que se representa a continuación:

$$nx = \cos I \times \cos(\theta_2 + \theta_3 + \theta_4) \times \cos V - \sin I \times \sin V \quad (2.12)$$

$$ny = \sin I \times \cos(\theta_2 + \theta_3 + \theta_4) \times \cos V + \cos I \times \sin V \quad (2.13)$$

$$nz = \sin(\theta_2 + \theta_3 + \theta_4) \times \cos V \quad (2.14)$$

$$sx = -\cos I \times \cos(\theta_2 + \theta_3 + \theta_4) \times \sin V - \sin I \times \cos V \quad (2.15)$$

$$sy = -\sin I \times \cos(\theta_2 + \theta_3 + \theta_4) \times \sin V + \cos I \times \sin V \quad (2.16)$$

$$sz = \cos(\theta_2 + \theta_3 + \theta_4) \times \sin V \quad (2.17)$$

$$ax = -\cos I \times \cos(\theta_2 + \theta_3 + \theta_4) \quad (2.18)$$

$$ay = -\sin I \times \sin(\theta_2 + \theta_3 + \theta_4) \quad (2.19)$$

$$az = \cos(\theta_2 + \theta_3 + \theta_4) \quad (2.20)$$

$$px = \cos I \times [Lb \times \cos II + Lc \times \cos(\theta_2 + \theta_3) + Ld \times \sin(\theta_2 + \theta_3 + \theta_4)] \quad (2.21)$$

$$py = \sin I \times [Lb \times \cos II + Lc \times \cos(\theta_2 + \theta_3) + Ld \times \sin(\theta_2 + \theta_3 + \theta_4)] \quad (2.22)$$

$$pz = Ld \times \cos(\theta_2 + \theta_3 + \theta_4) I + Lc \times \sin(\theta_2 + \theta_3) + Lb \times \sin II + La \quad (2.23)$$



Análisis de resultados

El diagrama funcional que se utilizó para obtener el modelo (matriz T) muestra como una condición del robot el cual $px = 0$, $py = 0$, $pz = La + Lb + Lc + Lc$. Esta posición implica según pueden observarse, que los parámetros de las articulaciones son nulos ($\theta_1 = \theta_3 = \theta_5 = 0^\circ$); excepto θ_2 y θ_4 que valen 90° y -90° , respectivamente. Ciertamente la posición se obtiene al evaluar dichos valores en las ecuaciones anteriores coincide con lo esperado.

Al evaluar las ecuaciones de posición se obtiene que:

$$px = Lb + Lc \quad (2.24)$$

$$py = 0 \quad (2.25)$$

$$pz = La + Ld \quad (2.26)$$

la cual es ciertamente la posición observable del *TCP* (“Tool Center Point” en inglés) respecto del sistema de coordenadas único en la base.

Cinemática Inversa

Sin dudas resulta un problema laborioso obtener expresiones cerradas para los ángulos, en función de vectores de posición y orientación. Sin embargo, es posible sin mucho esfuerzo determinar la solución para algunos de ellos.

Por ejemplo al dividir las ecuaciones (2.22) y (2.21) es sencillo determinar θ_1 :

$$\theta_1 = \tan^{-1} \left(\frac{py}{px} \right) \quad (2.27)$$

Así mismo es fácil calcular θ_5 reemplazando el valor explícito de $\cos(\theta_2 + \theta_3 + \theta_4)$ de la ecuación (2.20) en (2.17):



$$\theta_5 = \sin^{-1} \left(\frac{nz}{az} \right) \quad (2.28)$$

De aquí en adelante se puede apelar a herramientas de cómputo para determinar los ángulos restantes. Una vía factible sería utilizar las ecuaciones (2.21) o (2.22), junto a (2.23) planteando un sistema que tenga como incógnitas a la suma de $\theta_2 + \theta_3$ y a θ_2 . Luego de esto la determinación individualizada de $\theta_3 + \theta_4$ no representa problema alguno.

A modo de conclusión se puede suscribir lo que muchos otros autores ya han afirmado: la dificultad de obtener modelos cinemáticos inversos se debe a la necesidad de despejar variables que se encuentran en el argumento de funciones trascendentes. En casos como el de este robot la dificultad se hace mayor porque todas las Matrices de Paso Homogéneas tienen como parámetros variables los ángulos de las articulaciones rotoides, de manera que el manejo de funciones “Seno” y “Coseno” debe afrontarse continuamente. En robots con articulaciones prismáticas la dificultad será menor, pero no por ello desdeñable. En todo caso es fundamental comprender que, pese a que puedan determinarse expresiones algebraicas para el modelo cinemático inverso, estas expresiones no siempre arrojarán valores únicos y a veces resultarán inexistentes para algunas posiciones y orientaciones que no puedan ser definidas en la práctica. [29]

2.2.8.5 Planificación de la trayectoria

Definición del problema cinemático directo

Tradicionalmente, una tarea de planificación de trayectoria suele tener tres tipos de resultados: rutas, caminos y trayectorias.

Una ruta es un conjunto ordenado de configuraciones que deben ser alcanzadas por el robot. Por camino se entiende la discretización de una función continua que interpola las



configuraciones definidas en una ruta. Finalmente, cuando se habla de trayectoria temporal, el cual es llamado trayectoria por simplicidad, se está haciendo referencia a un camino que tiene asociado un perfil cinemático; es decir, a cada configuración perteneciente al camino se le asocia una velocidad.

El resultado de la planificación se utiliza en sistemas de navegación planificada. Una característica común de todos ellos es la utilización de un controlador que, en tiempo real, permita que el sistema siga la trayectoria previamente obtenida. Dependiendo del tipo de controlador será necesario planificar caminos o trayectorias. En este artículo, se entiende indistintamente por planificar, obtener un camino o una trayectoria, cuya ruta está definida por una configuración inicial y una configuración final.

Cuando se trata de generar trayectorias o caminos en sistemas no holónomos, hay una característica relevante que diferencia éste de otros problemas de planificación. Quizás la característica más sobresaliente esté vinculada al tipo de trayectorias que pueden seguir estos sistemas. Quizás la característica más sobresaliente esté vinculada al tipo de trayectorias que pueden seguir estos sistemas. En efecto, una configuración inicial y una final no pueden unirse mediante cualquier trayectoria. Las restricciones cinemáticas del sistema imponen unas condiciones que sólo algunos caminos cumplirán.

En dichos casos, el problema de planificación consiste en encontrar un camino que conecte la configuración inicial con la configuración final y que, además, a lo largo del mismo se satisfagan las restricciones no holónomas.

Existen numerosos trabajos que abordan el problema de planificación desde un punto de vista geométrico. Algunos desarrollos logran obtener caminos cinemáticamente admisibles a partir de la técnica de grafos de visibilidad. En ellos se calcula, mediante la utilización de grafos, una ruta libre de colisiones. Posteriormente, ésta se aproxima mediante curvas β – *splines* cúbicas. Otros trabajos se basan en la aplicación de métodos de búsqueda de caminos libres de colisiones sin considerar las restricciones no



holónomas; más tarde incorporan las restricciones y obtienen caminos cinemáticamente admisibles. También cabe citar los métodos de búsqueda basados en la discretización del mapa del entorno, estas técnicas permiten generar una serie de células libres de obstáculos que permiten la obtención de una ruta libre de colisiones.

En algunos otros casos, el problema de planificación se aborda desde un punto de vista de la teoría de control. Para ello, se diseñan leyes de control que aplicadas al modelo del sistema, generan una trayectoria que une la configuración inicial con la b final. De esta manera, tras realizar una integración, analítica o numérica, se obtienen trayectorias que satisfacen las restricciones cinemáticas. Dentro de este tipo de métodos hay técnicas que no tienen en cuenta los condicionantes impuestos por el entorno. Si se desea considerar esta clase de restricciones, como por ejemplo posibles colisiones, una vez que se ha planificado el camino, es necesario comprobar si la trayectoria generada presenta algún tipo de conflicto o colisión.

Por contra, dentro de este mismo grupo existen otros métodos donde el problema de evitación de colisiones puede tenerse en cuenta previamente a la generación de la trayectoria y no es necesario incorporar técnicas de detección de colisiones. En este artículo se describen algunas de las técnicas más representativas de los métodos de planificación basados en técnicas de control de sistemas no holónomos. [31]

Sistemas Encadenados

Tradicionalmente, el problema de control asociado a un sistema con restricciones no holónomas suele plantearse como un problema cinemático. En este caso es posible obtener un modelo de representación de estado como el de la ecuación (2.29):

$$q = [P(q)][u] \quad (2.29)$$

De forma genérica, con este tipo de técnicas, se pretende encontrar una evolución temporal del vector de control $[u]$ (ley de control en bucle abierto) de forma que la



integración numérica de (2.29) permita obtener una trayectoria cinemáticamente admisible.

Existen numerosas ocasiones en las que, dado un sistema de n variables de estado y m variables de control, es posible encontrar una transformación de coordenadas que convierta el modelo del sistema en una representación de estado de la forma:

$$q_i = u_i \quad (2.30)$$

$$q_{n+j} = q_{n+1}u_i \quad (2.31)$$

Donde $i = 1 \dots m$ y $j = 1 \dots n - m$. Esta expresión se corresponde con un tipo de sistemas denominados formas encadenadas. Las m variables (q_1, q_m) que dependen únicamente de una de las componentes del vector de entrada, se denominan variables independientes. El resto de variables reciben el nombre de variables dependientes. El subespacio formado por las variables independientes recibe el nombre de Espacio Base. [31]

El aspecto del modelo presentado en el sistema (2.30; 2.31) es un caso de transformación específico. En general, la forma que toman dichas ecuaciones dependerá del grado de no holonomía del sistema.

Como ejemplo ilustrativo para este método se utilizará en este artículo un robot del tipo tractorremolque. Un sistema de estas características está compuesto por un vehículo tractor delantero cuya cinemática se corresponde con el de un automóvil y un vehículo remolcado que se mueve sobre dos ruedas, ilustrado en la figura 2.49.

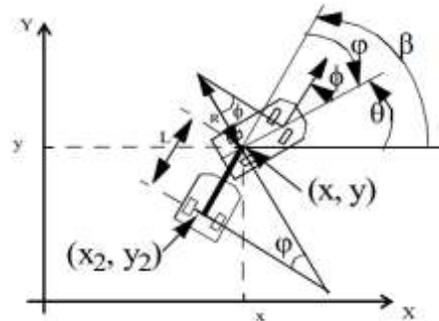


Figura. 2.49 Sistema tractor-remolque.

Fuente: Gómez, Cuesta y Ollero (2011) [31]

Para la aplicación de esta técnica se adopta el siguiente modelo:

$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \\ \dot{\theta} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos \beta \times \cos(\theta - \beta) & 0 \\ \sin \beta \times \cos(\theta - \beta) & 0 \\ 0 & 1 \\ \frac{\sin(\theta - \beta)}{L} & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ v(t)\rho(t) \end{bmatrix} \quad (2.32)$$

Merece la pena resaltar que se han escogido como variables de estado: la posición cartesiana del punto central trasero del remolque (x_2, y_2) , la orientación del remolque (θ) y la orientación del tractor (β) . La transformación de este modelo en un modelo encadenado del sistema (2.30, 2.31) se realiza de acuerdo a los siguientes cambios:

$$q_1 = x_2 \quad (2.33)$$

$$q_2 = \frac{\tan(\beta - \theta)}{\cos \beta^3 \times L} \quad (2.34)$$

$$q_3 = \tan \beta \quad (2.35)$$

$$q_4 = y_2 \quad (2.36)$$



La idea fundamental de este método de planificación consiste en generar, en primer lugar, una trayectoria que permita alcanzar los valores finales deseados para las m variables independientes. Una vez alcanzada esta configuración, se planifican, para las variables independientes, trayectorias cerradas en el Espacio Base, de forma que el valor de cada variable dependiente alcance el valor final deseado, mientras que las variables independientes mantienen el valor que presentaban antes de realizar la maniobra. [31]

Ejecución de maniobras sucesivas

La primera etapa de esta técnica consiste en planificar una trayectoria que permita alcanzar los valores finales deseados para las m variables independientes.

Una vez alcanzada esta configuración, se planifican $n - m$ trayectorias que conducen, de forma secuencial, el valor de cada variable dependiente hasta el valor deseado. En cada una de estas $n - m$ trayectorias, las trayectorias del sistema en el Espacio Base deben ser curvas cerradas. Es decir, los valores iniciales y finales de las variables independientes deben ser iguales. Para ello, se escoge un vector de entrada que haga que las variables independientes varíen según una función sinusoidal. El argumento de esta función ha de ser múltiplo entero de $2\pi t$ donde t es un parámetro que varía entre 0 y 1.

Además, la planificación se realiza de modo que en la maniobra k -ésima, las variables dependientes q_d , que cumplan relación:

$$d - m < k \quad (2.37)$$

Y que también varíen de acuerdo a una función sinusoidal cuyo argumento es un múltiplo entero de $2\pi t$. De esta manera, una vez que la variable dependiente de orden d ha alcanzado el valor deseado en la trayectoria $d - m = k$, en las siguientes maniobras, el valor final de esta variable no varía.



En el caso del modelo empleado como ejemplo en este artículo, la ley de control propuesta para cada maniobra es de la forma:

$$q_1' = u_1 = a \times \sin(2\pi t) \quad (2.38)$$

$$q_2' = u_2 = b_j \times \cos(j2\pi t) \quad (2.39)$$

$$j = 1,2 \quad (2.40)$$

donde j indica el orden de la maniobra a realizar. De esta forma, para llevar la variable dependiente $m+k$ hasta el valor deseado bastará integrar el sistema de ecuaciones (2.30, 2.31) aplicando como señales de entrada las funciones definidas en (2.38, 2.39, 2.40) haciendo

$$j = k \cdot A \quad (2.41)$$

a partir de las expresiones obtenidas, y del incremento deseado, será posible dar valores correspondientes a los parámetros a y b_j .

Este método se ha aplicado a un sistema tractorremolque en el que se ha planificado una maniobra de reorientación del remolque donde el ángulo relativo entre tractor y remolque es:

$$\varphi = -0.5 \text{ rad} \quad (2.42)$$

Como puede apreciarse en la figura 2.50 la trayectoria resultante es compleja.

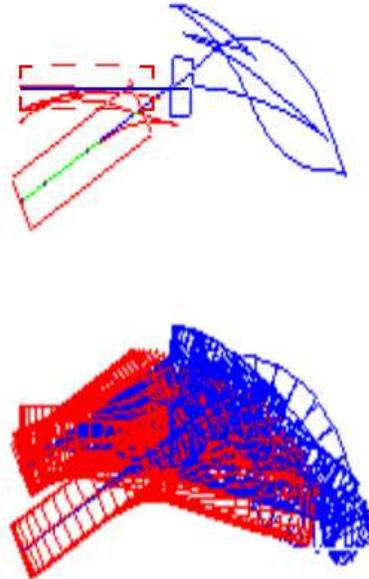


Figura. 2.50 Maniobras secuenciales para la reorientación del remolque.

Fuente: Gómez, Cuesta y Ollero (2011) [31]

Ejecución de una única maniobra

En este método se realiza una única maniobra para llevar todas las variables, dependiente e independientes, a alcanzar el valor deseado. De forma similar al método anterior se trata de planificar, para las dos variables independientes responsables del control, q_1 y q_2 , una trayectoria n el Espacio Base que lleve al resto de variables independientes hasta los valores deseados. [31] Con este fin se propone la siguiente ley de control:

$$q_1' = u_1 = a_0 + a_1 \times \sin(2\pi t) \quad (2.43)$$

$$q_2' = u_2 = b_0 + \sum_{i=1}^2 b_i \times \cos(i2\pi t) \quad (2.44)$$

Los valores de a_0 y b_0 se utilizan para llevar las variables q_1 y q_2 hasta el valor deseado. Por otra parte, una vez integrado el modelo, de forma similar a como se detalló en el apartado anterior, los distintos términos de la serie que representan la evolución



temporal de la variable dependiente $z+i$ se anulan, a excepción del término i (que involucra los parámetros a_1 y b_i). De este modo, dado un valor de a_1 basta tener en cuenta el incremento deseado en la variable i para obtener b_i .

Este método también se ha aplicado a un sistema tractor-remolque en el que se ha planificado una maniobra de reorientación del remolque donde el ángulo relativo final entre tractor y remolque es también $\varphi = -0.5 \text{ rad}$.

Como puede apreciarse en la figura 2.51 la trayectoria resultante es mucho menos compleja.

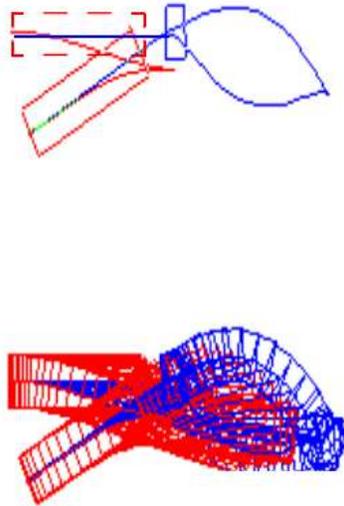


Figura. 2.51 Maniobra única de reorientación.

Fuente: Gómez, Cuesta y Ollero (2011) [31]

Evitación de colisiones

Como se verá en la última sección de este artículo, la forma de las maniobras obtenidas con cualquiera de los dos métodos anteriores depende, de forma radical, de los valores que tomen los parámetros a_i y b_i . En Tilbury (1995) [32] puede encontrarse un



algoritmo que permite, dado un entorno, calcular los valores de dichos parámetros que generan una maniobra libre de colisiones. Básicamente, dicho algoritmo implementa un procedimiento de evaluación de parámetros basado en un método de detección de colisiones y de acotación de la curvatura. Si la trayectoria generada por un conjunto de valores presenta colisión o viola la restricción de acotación de la curvatura, ésta es desechada y los valores de los parámetros evaluados son modificados con el fin de encontrar una nueva maniobra.

2.2.8.5.1 Trayectorias Canónicas

Esta técnica se fundamenta en conceptos de control y planificación geométrica. Se denominan trayectorias canónicas a aquellas generadas por un vector de control que permite asegurar que el sistema recorre una trayectoria en la que todos los vehículos mantienen una curvatura constante. En el caso del sistema tractor-remolque se utiliza también el sistema (2.32).

Resulta claro que las trayectorias canónicas en este tipo de sistemas son líneas rectas y arcos de circunferencia. En este caso, dichas trayectorias están determinadas por la configuración inicial del sistema. Por tanto, cada configuración posible presenta una trayectoria canónica. Para el modelo presentado en (2.32) el vector de control estará dado por un valor genérico para la velocidad y un valor de curvatura definido por:

$$\rho(t) = \frac{\sin(\theta_0 - \beta_0)}{L} \quad (2.45)$$

donde θ_0 y β_0 representan, respectivamente, los valores iniciales de la orientación del tractor y del remolque. El método de planificación se basa en la generación de una trayectoria que es combinación convexa de las trayectorias canónicas asociadas a la configuración inicial y final en la forma:

$$P(t) = (1 - \alpha(t)) \times \gamma(X_i, t) + \alpha(t) \times \gamma(X_f, t) \quad (2.46)$$



donde $\gamma(X_i, t)$ y $\gamma(X_f, t)$ son las trayectorias canónicas asociadas a la configuración inicial y final, y $\alpha(t)$ es una función que aplica $[0,1]$ en $[0,1]$ y que cumple:

$$\begin{cases} \alpha(0) = 0 \\ \alpha(1) = 0 \\ \dot{\alpha}(0) = \ddot{\alpha}(0) = \dot{\alpha}(1) = \ddot{\alpha}(1) = 0 \end{cases} \quad (2.46)$$

El resultado de planificar una maniobra de reorientación del remolque, donde el ángulo relativo final entre tractor y remolque es $\varphi = -0.5 \text{ rad}$, es el que se presenta en la figura 2.52. [31]

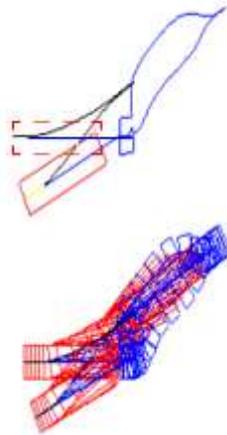


Figura. 2.52 Maniobra de reorientación con trayectorias canónicas.

Fuente: Gómez, Cuesta y Ollero (2011) [31]

Evitación de colisiones

La generación de un camino libre de colisiones se consigue mediante la combinación de esta técnica con un planificador que utiliza métodos geométricos. La idea consiste en



encontrar una ruta de configuraciones libre de colisiones, sin preocuparse de las características cinemáticas del vehículo. Posteriormente se generan caminos que conecten los distintos puntos de la ruta utilizando el método descrito anteriormente. Los caminos resultantes entre dos configuraciones son evaluados, de forma que se comprueba que en el camino generado no aparecen colisiones ni se violan las restricciones de curvatura máxima. Si se detecta alguna colisión, se escogen otros dos nuevos puntos de la ruta. El proceso es repetido hasta ir encontrando caminos libres de colisiones a lo largo de la ruta previamente obtenida. [31]

2.2.8.5.2 Planificación Basada en el Concepto de Maniobra Restringida

Una maniobra restringida es una trayectoria al final de la cual sólo cambian los valores de ciertas variables de configuración; en el resto de variables, los valores finales son iguales a los valores iniciales. Este concepto puede considerarse aplicable de forma genérica a todos los sistemas no holónomos.

La generación de maniobras restringidas está vinculada a la aplicación de una sucesión de vectores campo que satisfacen las restricciones cinemáticas; además, cada uno de estos vectores está asociado a un vector de control determinado. Por tanto, puede considerarse que esta técnica se basa en el diseño de una ley de control en bucle abierto, discontinua en el tiempo, que permite llevar el sistema desde cualquier configuración inicial a una configuración final.

La generación de maniobras restringidas para un sistema tractor-remolque se define con el siguiente modelo cinemático:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \\ -\frac{\sin(\varphi)}{L} & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ v(t)\rho(t) \end{bmatrix} \quad (2.47)$$



Donde $\varphi = \theta - \beta$.

En el caso de un sistema tractor remolque con este modelo será posible planificar maniobras restringidas para variables x, y, θ, φ . La planificación basada en esta técnica genera una sucesión de maniobras restringidas, que permiten alcanzar cualquier configuración final desde una configuración inicial arbitraria. En la figura 2.53 se ilustra el resultado obtenido para una reorientación donde el ángulo relativo entre tractor y remolque es $\varphi = -0.5 \text{ rad}$. [31]

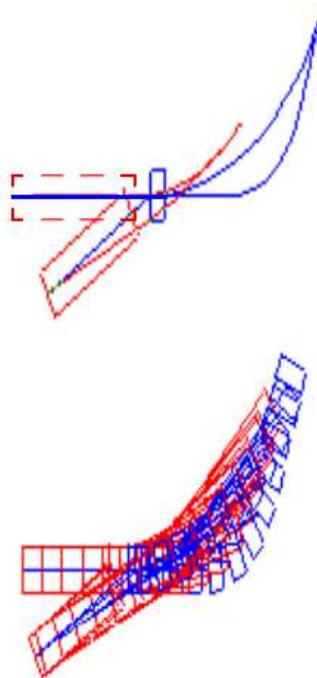


Figura. 2.53 Maniobra restringida para reorientación.

Fuente: Gómez, Cuesta y Ollero (2011) [31]

Evitación de colisiones

La selección de vectores campo adecuados permite conseguir trayectorias que son, en parte, el resultado de integrar analíticamente el modelo cinemático. Básicamente se



propone el uso de una familia de vectores que proporciona trayectorias para el vehículo tractor que están compuestas por arcos de circunferencia y líneas rectas. Esto permite que, para escenarios donde los objetos están modelados mediante polígonos rectangulares, sea posible aplicar criterios, en la selección de las curvaturas, que permiten encontrar trayectorias libres de colisiones para el vehículo tractor.

Asimismo, en el caso del vehículo remolque, pueden encontrarse trayectorias resultantes de integración analítica o integración numérica. En ambos casos, el movimiento del remolque es lo suficientemente regular como para establecer una relación entre cada maniobra y la región espacial que contiene al remolque a lo largo de la trayectoria. Esta relación puede identificarse mediante técnicas de lógica borrosa. Gracias a esta identificación y la utilización del correspondiente modelo inverso, puede implementarse un planificador como el apreciado en la figura 2.54, en el cual, dadas las restricciones espaciales definidas por los obstáculos y las especificaciones determinadas por la maniobra, se genera un perfil cinemático, secuencia de valores para la velocidad y curvatura, correspondiente a una maniobra libre de colisiones. [31]

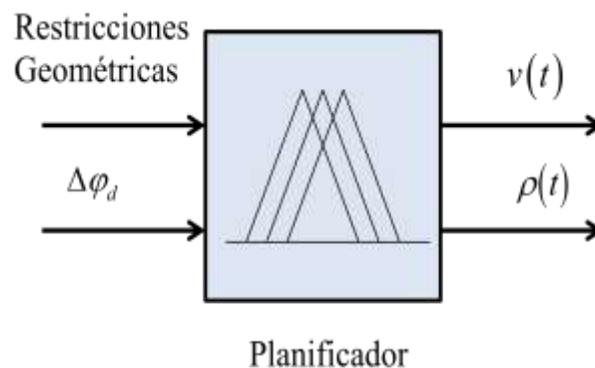


Figura. 2.54 Sistema Planificador



2.2.8.5.3 Comparación de los métodos de planificación

La generación de maniobras utilizando el método de *Ejecución de maniobras sucesivas* se presenta el inconveniente de proporcionar maniobras muy complejas. En cambio, las maniobras generadas con la técnica de *Ejecución de una única maniobra* muestran una forma mucho menos compleja. Además, dicha forma sólo depende del valor seleccionado para los parámetros a_i . Sin embargo, este método presenta algunos inconvenientes. Por ejemplo, hay configuraciones en las que las ecuaciones de transformación del modelo (2.33, 2.34, 2.35, 2.36) presentan singularidades. Por otro lado, el perfil cinemático asociado al cambio de la curvatura es bastante complejo, generando una función continua pero poco regular. Además, no es inmediato conseguir trayectorias que respeten la restricción de acotación de la curvatura; de hecho hay que chequear esta restricción en cada maniobra generada.

Finalmente, La forma de la trayectoria depende enormemente del valor tomado por los parámetros a_i , de modo que no existe, como puede apreciarse en la figura 2.55 un comportamiento regular en dicha forma. Es decir, no puede saberse a priori como será el movimiento del vehículo a lo largo de la maniobra. En la figura 2.55 se presenta una maniobra que produce el mismo valor de reorientación que la mostrada en la figura 2.51 pero donde se ha considerado un valor distinto para el parámetro a_1 .

Por tanto, es necesario examinar todas las maniobras generadas tanto si se desea comprobar que son libres de colisión como si respetan la restricción de acotación de la curvatura. En cuanto al método de planificación basado en trayectorias canónicas es importante resaltar que la forma de la trayectoria depende mucho de las trayectorias canónicas asociadas a la configuración inicial y final. Por otra parte, en numerosas circunstancias, es necesario detener el vehículo y navegar hacia atrás (es el caso de la maniobra que se presenta en este artículo). En estas circunstancias la forma de la



maniobra depende mucho del punto intermedio seleccionado para iniciar la marcha atrás.
[31]

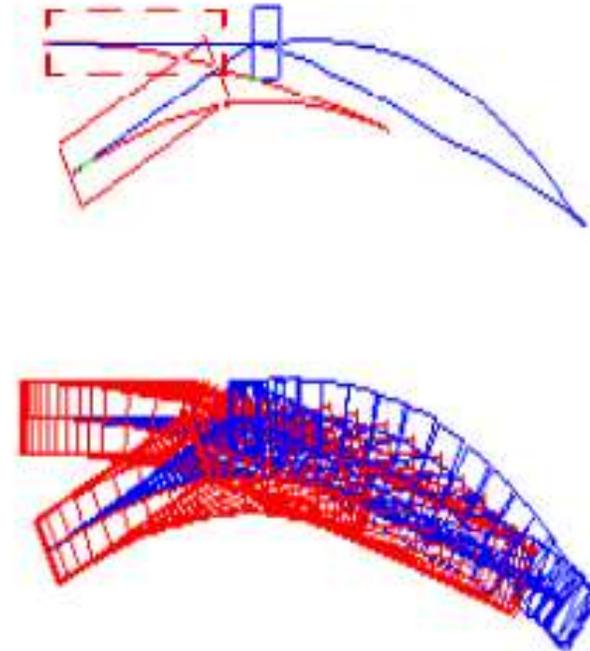


Figura. 2.55 Reorientación basada en modelo encadenado con distinto a_1 .

Fuente: Gómez, Cuesta y Ollero (2011) [31]

El método presentado en Lamiroux y Laumond (2001) [33] no proporciona criterios para la elección de este punto. De hecho, en la maniobra mostrada en la figura 2.52 dicho punto ha sido seleccionado en base a consideraciones heurística fundadas en la experiencia de los autores del artículo. Así, en la figura 2.56 puede observarse una maniobra generada con este método. Esta maniobra provoca el mismo valor de reorientación que la presentada en *Trayectorias canónicas*, sin embargo no se han tenido



en cuenta los criterios heurísticos. Como puede apreciarse, la forma de la maniobra se complica más. [31]

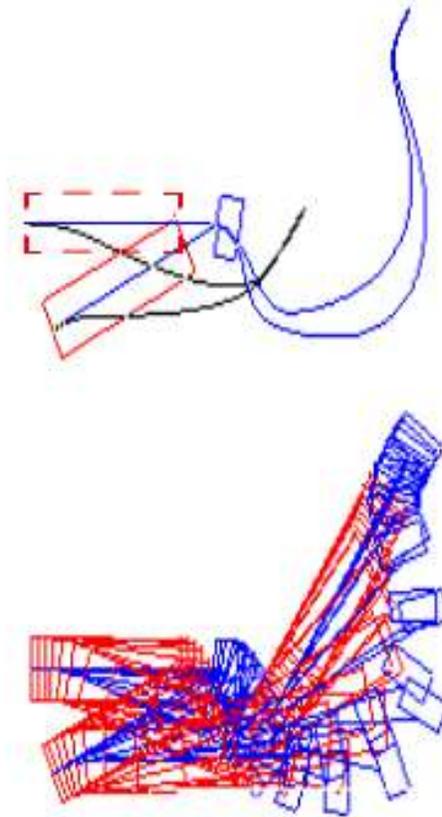


Figura 2.56 Reorientación basada en trayectoria canónica con diferente punto intermedio.

Fuente: Gómez, Cuesta y Ollero (2011) [31]

En los dos métodos anteriores la evitación de colisiones supone la implantación de técnicas de detección de colisiones. Este hecho puede representar una carga computacional costosa cuando se trata de realizar planificación “on-line”. Además, en ellos no existe un conocimiento a priori de la mayor o menor complejidad que puede presentar la maniobra. En la figura 2.57 se muestra una maniobra que produce el mismo



valor de reorientación que la maniobra presentada en *Planificación basada en el concepto de maniobra restringida*, pero con distintos valores en las restricciones geométricas. [31]

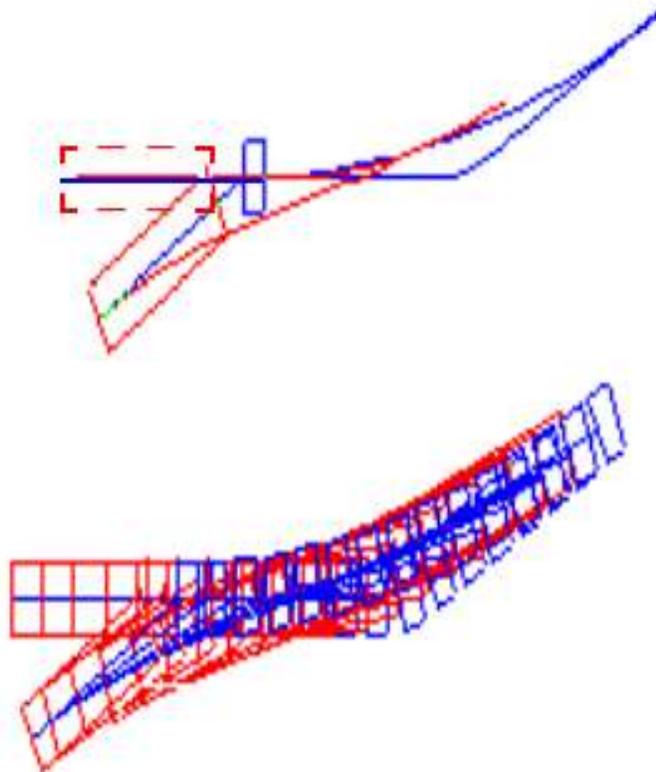


Figura 2.57 Reorientación mediante maniobra restringida.

Fuente: Gómez, Cuesta y Ollero (2011) [31]

Como puede observarse, la forma de la maniobra es similar a la anterior aunque de distintas dimensiones. Este comportamiento ha permitido el desarrollo de un planificador como el mostrado en la figura 2.54. Este mismo hecho hace posible implantar un método de planificación en el que no es necesario utilizar ningún algoritmo de detección de colisiones, ya que la maniobra generada es de por sí libre de colisión. Por último cabe resaltar que el comportamiento regular del sistema permite tener un conocimiento a priori



de cómo va a ser la maniobra. Además, el perfil cinemático generado es regular y garantiza que el valor de la curvatura está acotado. Como inconveniente cabe señalar que la trayectoria presenta discontinuidades cinemáticas, lo que supone que el vehículo ha de detenerse para cambiar el ángulo de dirección de las ruedas. Sin embargo, esto no resulta un gran inconveniente ya que en los procesos de maniobras es frecuente que el vehículo tenga que detenerse. De hecho han sido desarrollados mecanismos que permiten la ejecución de este tipo de maniobras con gran facilidad. [31]

2.2.8.6 Aplicaciones

El proceso de implantación de un robot industrial en una determinada aplicación requiere del estudio previo detallado del proceso en cuestión. Se deberá realizar un análisis, examinando las ventajas e inconvenientes que implicarían la implementación del robot. Durante el proceso se deberá estar abierto a realizar cambios en el desarrollo del proceso primitivo tales como modificaciones en el diseño de piezas, sustitución de unos sistemas por otros, que faciliten y hagan viable la aplicación del robot.

Entre las aplicaciones más comunes de los robots a nivel industrial se encuentran: los trabajos en fundición, soldaduras, aplicación de materiales, aplicación de sellantes y adhesivos, alimentación de máquinas, procesado, corte, montaje, paletización, control de calidad, y la manipulación en las salas blancas.

La aplicación que resulta de interés en el presente proyecto es la de paletización. La paletización es un proceso básicamente de manipulación, consistente en disponer piezas sobre una plataforma o bandeja (palet). Las piezas en un palet ocupan normalmente posiciones predeterminadas, procurando asegurar la estabilidad, facilitar su manipulación y optimizar su extensión. Los palets son transportados por diferentes sistemas, como



cintas transportadoras, carretillas, llevando su carga de piezas, a lo largo del proceso de fabricación, ó hasta el almacén o punto de expedición. [26]

En la figura 2.58 se observa un esquema y un ejemplo de robot de brazo articulado.



Figura 2.58 Robot paletizador.

Fuente: OTAI (2012), Sitio web [34]

Dependiendo de la aplicación concreta, un palet puede transportar piezas idénticas (para almacenamiento por lotes por ejemplo), conjuntos de piezas diferentes, pero siempre los mismos subconjuntos procedentes de ensamblados) o cargas de piezas diferentes y de composición aleatoria (formación de pedidos en un almacén de distribución).

Existen diferentes tipos de máquinas específicas para realizar operaciones de paletizado. Éstas frente al robot, presentan ventajas en cuanto a velocidad y coste, sin embargo, son



rígidas en cuanto a su funcionamiento, siendo incapaces de modificar su tarea de carga y descarga.

Así pues, los robots realizan con ventaja aplicaciones de paletización en las que la forma, número o características generales de los productos a manipular, cambian con relativa frecuencia. En estos casos, un programa de control adecuado permite resolver la operación de carga y descarga, optimizando los movimientos del robot, aprovechando la capacidad del palet o atendiendo a cualquier otro imperativo.

Generalmente, las tareas de paletización implican el manejo de grandes cargas, de peso y dimensiones elevadas. Por este motivo, los robots empleados en este tipo de aplicaciones acostumbran a ser robots de gran tamaño, con una capacidad de carga de 10 a 100 kg. No obstante, se pueden encontrar aplicaciones de paletización de pequeñas piezas, en las que un robot con una capacidad de carga de 5 kg. es suficiente.

Las denominadas tareas de *pick and place*, aunque en general con características diferentes al paletizado, guardan estrecha relación con este. La misión de un robot trabajando en un proceso de *pick and place* consiste en recoger piezas de un lugar y depositarlas en otro. La complejidad de este proceso puede ser muy variable, desde el caso más sencillo en el que el robot recoge y deja las piezas en una posición prefijada, hasta aquellas aplicaciones en las que el robot precisa de sensores externos, como visión artificial o tacto, para determinar la posición de recogida y colocación de las piezas. Al contrario que en las operaciones de paletizado, las tareas de *picking* suelen realizarse con piezas pequeñas (peso inferior a 5 Kg) necesitándose velocidad y precisión.

Un ejemplo típico de aplicación de robot al paletizado sería la formación de palets de cajas de productos alimenticios procedentes de una línea de empaquetado. En estos casos, cajas de diferentes productos llegan aleatoriamente al campo de acción del robot. Ahí son identificadas bien por una célula de carga, por alguna de sus dimensiones, o por



un código de barras. Conocida la identidad de la caja, el robot procede a recogerla y a colocarla en uno de los diferentes palets que, de manera simultánea, se están formando.

El propio robot gestiona las líneas de alimentación de las cajas y de palets, a la vez que toma las decisiones necesarias para situar la caja en el palet con la posición y orientación adecuadas de una manera flexible. El robot podrá ir equipado con una serie de ventosas de vacío y su capacidad de carga estaría entorno a los 50 kg. [26]

2.2.8.7 Programación

a. Métodos de programación

Programar un robot consiste en indicar paso por paso las diferentes acciones (moverse a un punto, abrir o cerrar la pinza, etc.) que éste deberá realizar durante su funcionamiento, la flexibilidad en la aplicación del robot y, por lo tanto, su utilidad van a depender en gran parte de las características de su sistema de programación.

Actualmente no existe normalización en relación a los procedimientos de programación de robots, cada fabricante desarrolla su método particular, el cual es válido solamente para sus propios robots. Sin embargo, algunos han servido de modelo para el desarrollo de otros, por ejemplo: el lenguaje AL (Finkel-74).

Existen varios criterios para clasificar los métodos de programación. Algunos lo hacen según la potencia del método, y otros lo hacen según el sistema utilizado para indicar la secuencia de acciones a realizar, éste último es el más ilustrativo al momento de dar a conocer las alternativas existentes para programar un robot. [26]

Según este último criterio, un robot puede ser programado por:

Programación por Guiado



- Guiado Pasivo
 - Guiado Pasivo Directo
 - Guiado Pasivo por Maniquí
- Guiado activo:
 - Guiado Básico.
 - Guiado Extendido.

Programación textual

- Nivel robot: las órdenes se refieren a los movimientos a realizar por el robot.
- Nivel objeto: las órdenes se refieren al estado en que deben ir quedando los objetos.
- Nivel tarea: las órdenes se refieren al objetivo a conseguir.

Requerimientos de un sistema de programación de robots

Tradicionalmente los requerimientos generales que se han establecido para un sistema de programación de robots son Entorno de Programación, Modelado del Entorno, Tipo de Datos, Manejo de Entradas/Salidas (digital y análoga), Control del Movimiento del Robot, Control del flujo de ejecución del programa. [26]

b. Software de programación

El programa que se utilizará en el desarrollo del proyecto será el MELSOFT-RT ToolBox2 de Mitsubishi. Y se describe a continuación

Software MELSOFT – RT ToolBox2

RT ToolBox2 es el entorno de programación para todos los robots de Mitsubishi. En sólo unos pocos minutos es posible generar programas en el lenguaje de robots MELFA



BASIC IV ó MOVEMASTER COMMAND, y transferirlos entonces a los robots reales con un solo clic después de la comprobación y la optimización. La transmisión se realiza de forma muy eficaz gracias al enlace directo entre el PC y los robots a través de una interface de red o de un enlace serie.

Durante la ejecución de los programas se supervisa y visualiza a los robots por medio de las amplias funciones de control y de diagnóstico de RT ToolBox2. Las velocidades de eje y las corrientes de motor momentáneas se presentan de forma clara e ilustrativa junto a los estados actuales de todas las entradas y salidas del robot. Es posible darle seguimiento en vivo todos los programas procesados actualmente por el control, pudiendo reconocer así posibles errores de forma sencilla y segura.

Además, RT ToolBox2 pone a la disposición herramientas para archivar programas y para realizar respaldos (*backups*) de los parámetros ajustados del robot.

Entre otras funciones ventajosas se encuentran la Función "teach in" online de posiciones, la Visualización de las posiciones en la representación 3D del robot, el Control de sintaxis, el Monitor de entradas/salidas, el Monitor de variables, la Ejecución de órdenes online, el Diagnóstico de errores (*debugger*), el Editor de posiciones y el Administración de proyectos. [35]



Capítulo III

3.1. Metodología

El tipo de investigación está enmarcado en la modalidad de proyecto factible soportado en la definición de la UPEL “como un estudio que consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales” [36]

Para la presente investigación, una vez determinada la necesidad de la empresa Dafoinca, de mejorar su proceso productivo en las fases de impresión y embalaje se aplicó la metodología adecuada para el campo de ingeniería en la programación de micro-controladores y robots así como para el diseño de hardware del dispositivo controlador y el desarrollo de la aplicación de computadoras para la comunicación con el dispositivo.

En tal sentido para el desarrollo del programa del dispositivo controlador y del Robot Industrial se aplicó la metodología de programación estructural definida por Corrado y Böhm en la cual establece que toda función computable puede ser implementada en un lenguaje de programación que combine sólo tres estructuras lógicas. Esas tres formas también llamadas estructuras de control. La primera es la secuencia, que consiste en la ejecución de una instrucción tras otra. La segunda es la selección en la cual se ejecutan una de dos instrucciones (o conjuntos), según el valor de una variable booleana. Por último la iteración que se basa en la ejecución de una instrucción o conjunto de instrucciones, mientras una variable booleana sea 'verdadera'. [37]

Para el desarrollo del hardware del dispositivo controlador se aplicó la metodología de Top-Down que según Niklaus Wirth consiste en capturar una idea con un alto nivel de abstracción, implementarla partiendo de la misma, e incrementar el nivel de detalle según



sea necesario. El sistema inicial se va subdividiendo en módulos, estableciendo una jerarquía. Cada módulo se subdivide cuantas veces sea necesario hasta llegar a los componentes primarios del diseño [38]

Para el desarrollo de la aplicación de computadoras para la comunicación con el dispositivo se aplicó la metodología de Programación Orientada a Objeto o POO que de acuerdo a Olaf Owe; es una forma de programación que usa los objetos en sus interacciones para diseñar aplicaciones y programas informáticos, en contraposición a la visión tradicional en la cual un programa es una colección de subrutinas, funciones o procedimientos, o simplemente una lista de instrucciones para el computador. Un objeto está compuesto de datos y características descritas por una clase a la que este pertenece. Utilizando funciones, llamadas métodos, que manejen y modifiquen los datos en estos objetos se logra estructurar la programación. [39]

3.1.1. Procedimiento para la recolección de información.

Para la UPEL las investigaciones de tipo proyecto factible deben de tener un apoyo ya sea de tipo documental, de campo o una modalidad que los incluya a ambos. Y define a la investigación documental como “El estudio de un problema con el propósito de ampliar y profundizar el conocimiento de su naturaleza principal, en trabajos previos, así como información y datos divulgables por medios impresos”. y a la investigación de campo como “El análisis sistemático de problemas en la realidad, con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y efectos o predecir su ocurrencia, haciendo uso de métodos característicos de cualquiera de los paradigmas o enfoques de investigación conocidos o en desarrollo. La fuente principal de datos es el sitio donde se presenta el problema, los datos de interés son



recogidos en forma directa de la realidad, en este sentido se trata de investigaciones a partir de datos originarios o primarios”. [36]

Esta definición permite el pluralismo metodológico y se refleja en la manera de abordar la investigación, en la selección de los sujetos, en la forma como se analizan los resultados y en el uso que se les da a los mismos.

En tal sentido se realizó una investigación sobre materiales teóricos referentes al proceso de impresión industrial, uso de Robots en procesos industriales, uso de micro-procesadores como elemento de automatización, adquisición de datos digitales. Todo esto fue sustentado en trabajos previos, así como información y datos divulgables por medios impresos, todo esto en función del problema de investigación al que se refiere el proyecto y con la finalidad de ampliar y profundizar el conocimiento de su naturaleza principal, en tal sentido permitió fortalecer el diseño de la propuesta para la mejora del proceso a través de la automatización.

Para la documentación del proceso se realizó una inspección a las máquinas involucradas en la parte del proceso a automatizar. Se tomaron datos de funcionamiento, modelos de máquinas, valores nominales de funcionamiento, alimentación, rango de trabajo, capacidad de producción máxima, fuentes de energía, entre otros, así como cualquier otra documentación importante como los manuales, guías y planos, existentes en la empresa.

También se determinó mediante observación los sistemas usados por máquinas y operarios para realizar el proceso de impresión. Horarios de trabajo, número de pedidos por día, tareas desempeñadas por operarios y cualquier otro elemento resaltante.



3.1.2. Técnicas de procesamiento y análisis de información.

Se registró la información mediante fotografías, registros escritos y material informativo verbal o impreso proporcionado por la empresa.

Se utilizó la herramienta de la entrevista en los operarios, se registró de manera escrita. Posteriormente se analizó esa información para conocer qué fallas de proceso han sido notadas por los operarios.

Con la información recolectada se detectaron los problemas existentes en el proceso de impresión. Que partes del proceso presentaban pérdidas de tiempo o material, y se estudió la posibilidad de minimización en las mismas.

3.1.3 Fases

Para lograr la solución propuesta para la actualización del sistema de control del proceso de impresión y la automatización del embalaje, el proyecto se desarrolló en las siguientes fases:

Fase I: Estudio del proceso de impresión y embalaje.

En esta fase se recopiló toda la información, referente al proceso de impresión y al proceso de embalaje, proveniente de registros de la compañía y del personal especializado.

Actividades:

- Se recopiló información sobre las características de operación de la máquina rotativa de impresión.



- Se obtuvo información del proceso de impresión en registros de la compañía o del personal especializado
- Se obtuvo información del proceso de embalaje en registros de la compañía o del personal especializado
- Se identificaron las normas de seguridad que se deben seguir en el departamento de embalaje.

Fase II: Análisis del control de los procesos de impresión y embalaje.

Se estudiaron los controles aplicados para el funcionamiento de los procesos de impresión y embalaje. Haciendo énfasis en los controles manuales de los procesos.

Actividades:

- Se identificaron los sistemas de control u operación manual en los procesos de impresión y embalaje.
- Se determinó si los procesos o controles identificados perjudicaban la producción y/o productividad.

Fase III: Diseño y prueba de un dispositivo controlador-registrador para la automatización de la máquina de impresión.

Actividades:

- Se seleccionó un microcontrolador y periféricos adecuados para su aplicación en la automatización de la máquina de impresión.
- Se desarrolló un programa que permita el funcionamiento del dispositivo.
- Se desarrolló una aplicación para computadora que permita extraer la información registrada por el dispositivo.



- Se simuló por software el dispositivo en operación para probar el funcionamiento del programa.
- Se construyó un prototipo del dispositivo para probar el correcto funcionamiento del mismo.

Fase IV: Desarrollo de un sistema de control automatizado para el proceso de embalaje, mediante un Robot Industrial

Actividades:

- Se seleccionó un robot adecuado para la automatización del proceso de embalaje.
- Se diseñó una herramienta de agarre del robot que se adecue a las características de los paquetes.
- Se desarrolló el programa para la ejecución automatizada del proceso de embalaje a utilizar en el robot.
- Se mostró el funcionamiento mediante la utilización de un Robot Industrial a escala fuera del proceso. Este robot fue proporcionado por la empresa FESTO.



Capítulo IV

En este capítulo se describen los resultados alcanzados, derivados de la aplicación de la metodología propuesta en el Capítulo III, lo cual permitió cumplir con los objetivos planteados para llevar a cabo el trabajo de grado.

4.1 Estudio del proceso de impresión y embalaje.

El estudio del proceso de impresión y embalaje estuvo sujeto al desarrollo de las siguientes actividades:

4.1.1 Recopilación de información sobre las características de operación de la máquina rotativa de impresión.

La máquina utilizada para el proceso de impresión, es una impresora rotativa Thimson TRS2000 de fabricación Inglesa cuyas especificaciones se detallan en la tabla 4.1.

Tabla 4.1. Datos técnicos Impresora Thimson TRS 2000

Marca	Thimson
Modelo	TRS 2000
Serial	18026
Fabricación	Engineers Keterin & London. UK
Año	1982
Motor	40 HP
Velocidad de impresión	305 metros x minuto
Ancho de impresión	20,5"

Fuente: DAFOINCA



Tabla 4.1. Datos técnicos Impresora Thimsons TRS 2000. Continuación.

Accesorios:	Desbobinador 127 cm de diámetro Elevador de rollos hidráulico Alineador lateral electrónico 4 unidades de impresión offset 4 casetes de impresión de carrera 22” 4 casetes de impresión de carrera 17” Barras de inversión Troquelado de huecos de archivo Troquelado de huecos de arrastre Perforadora transversal y longitudinal Dobladora de carrera variable
-------------	--

Fuente: DAFOINCA

Para identificar las características de operación de la máquina rotativa de impresión se destacan los siguientes componentes: un cuerpo de alimentación, cuatro módulos de impresión y un módulo de control.

El cuerpo de alimentación es donde se carga la bobina de papel que constituye la materia prima para el proceso de impresión. Este proceso se lleva a cabo con un desbobinador y un elevador de rollos hidráulico. También se encuentran los motores que desdoblan e impulsan el papel a través de la impresora y el sistema de alineado que asegura que el papel este posicionado correctamente para la transferencia a los módulos de aplicación de colores. Ver figura 4.1



Figura 4.1: Cuerpo de alimentación.

Los cuatro módulos de impresión, también llamados módulos de aplicación de colores están integrados con 4 casetes de impresión de carrera 22” y 17” que tienen la función de transferencia de tinta al papel. Cada una se abastece con un color de acuerdo a lo determinado por el diseño y las instrucciones del colorista. Ver figura 4.2



Figura 4.2: Módulos de Impresión.



Cada uno de los módulos de impresión posee depósitos de tinta y una serie de motores controlados por tarjetas electrónicas que mezclan la tinta y la adhieren al rodillo que impregna la plancha de fotgrabado. Este rodillo transfiere el diseño a la mantilla de impresión y al papel o sustrato de impresión. Este módulo posee controladores manuales para el ajuste de rodillos y botones de arranque para carga del papel. Ver figura 4.3



Figura 4.3: Módulo de Impresión offset

Por último se tiene el módulo de control. En este módulo se encuentran los controles operativos de la máquina, como lo son el arranque, la parada, selector de velocidad. Todos estos controles son manuales. Ver figura 4.4



Figura 4.4: Módulo de Control

Esta máquina opera en dos turnos al día, de acuerdo a los programas de producción suministradas por el departamento de programación de la empresa. La impresora produce en promedio 50.000 impresiones por cada turno, calculado para formularios de tamaño estándar de $9\frac{1}{2} \times 11$ pulgadas y de 100.000 para formas de $9\frac{1}{2} \times 5\frac{1}{2}$ pulgadas.

La máquina tiene paradas programadas para los cambios de papel, de acuerdo al número de copias que contenga el formulario. También ocurren en el caso de fin de turno o terminación de la orden de producción. Otras paradas se originan por ruptura de papel, falta de tinta o por fallas mecánicas.

Según registros de la compañía, semanalmente la máquina imprime 112.000 metros de papel y utiliza como materia prima 140.000 metros de papel. Observando así que existe una pérdida del 20% del papel procesado; de los cuales 18.3% corresponde a sobreproducción por paradas tardías de la máquina y el 0.7% a desperdicio por montura de puesta a punto y rupturas de papel.



4.1.2 Recopilación de información del proceso de impresión

De acuerdo a los registros de la compañía y datos suministrados por el personal especializado de la empresa, el proceso de impresión se lleva a cabo de la siguiente manera:

La máquina utiliza rollos de papel como materia prima, estos son cargados manualmente en el cuerpo de alimentación de la máquina donde se introducen en un desbobinador y un elevador de rollos hidráulico donde se encuentran los motores que desdoblan e impulsan el papel a través de la impresora. El papel se fija a los rodillos de la máquina. Al encender la impresora rotativa, esta imprime directamente a partir de un cilindro al que se le adhieren unas láminas fotograbadas que contienen del diseño del formulario. Esta operación es repetida para cada uno de los cuerpos (hasta cuatro cuerpos) dependiendo del número de colores que contiene el diseño.

Una vez montado el papel, se llenan los contenedores de tinta en el cuerpo de impresión según la formulación de colores y se arranca la máquina en modo prueba. A medida que giran los rodillos, van preparando la tinta en cuanto a viscosidad y características de secado para su transferencia al tambor que contiene la plantilla de fotograbado con el diseño del formato a imprimir. Un juego de rodillos elimina el exceso de tinta y distribuye uniformemente en el tambor. El tambor transfiere el diseño a una plantilla de impresión, de este modo la tinta es absorbida por la superficie del papel cuando entra en contacto con la plantilla. Una vez ajustado los colores y registros, se inicia el proceso de impresión.

El proceso de impresión se lleva a cabo con dos operadores, uno de los cuales lleva con ayuda de un montacargas, una bobina de papel hasta la impresora y procede al montaje manual en un eje del cuerpo de alimentación. Una vez instalada la bobina, el operador se retira y desempeña otras funciones en el almacén de papel. Ver figura 4.5



Figura 4.5: Instalación de bobina de papel en cuerpo de alimentación

El operador de la impresora se encarga de colocar adecuadamente el papel en la impresora y rellenar los contenedores de cada cuerpo con la tinta respectiva. Hace las pruebas de registro y arranca la máquina. Durante la operación, supervisa que funcione normalmente, y la detiene cuando se produzca alguna falla o se hayan procesado los metros necesarios de papel según lo indicado en la orden de producción. Ver figura 4.6



Figura 4.6: Operador manipulando controles de arranque y parada de máquinas.



Una vez impreso el papel, otro operador traslada el rollo impreso a la máquina empacadora para su agrupación, doblado y empaque. Ver figura 4.7



Figura 4.7: Transferencia de rollos a la máquina empacadora.

4.1.3 Recopilación de información del proceso de embalaje

De acuerdo a las informaciones de la compañía y del personal especializado, el proceso de embalaje se desarrolla de la siguiente manera:

Los paquetes de formularios impresos salen de una máquina Empacadora, llamada también en la empresa como “Colectora”, estos paquetes caen en una bandeja recolectora.



Dos o tres trabajadores llevan estos paquetes desde la bandeja a una mesa de trabajo, en donde se insertan o embalan en cajas de cartón. El proceso es completamente manual y depende de la velocidad de los trabajadores del departamento de embalaje para transportar los paquetes y meterlos en las cajas. Ver figura 4.8



Figura 4.8: Embalaje manual

4.1.4 Identificación de las normas de seguridad que se deben seguir en el departamento de embalaje.

Según el manual de seguridad de la empresa las normas de seguridad a seguir en la empresa están publicadas en un cartel del departamento de producción y son de obligatorio cumplimiento para los trabajadores (ver figura 4.9). Además en el departamento de embalaje se incorporan las siguientes normas:



- Todos los trabajadores que manipulen los paquetes deben ejecutar técnicas apropiadas de levantamiento
- No se debe correr
- No se debe manipular más de un paquete a la vez



Figura 4.9: Normas de seguridad e higiene ambiental

4.2 Análisis del control de los procesos de impresión y embalaje.

Se analizaron los controles aplicados para el funcionamiento de los procesos de impresión y embalaje con especial énfasis en los controles manuales de los procesos.



4.2.1. Identificación de los sistemas de control u operación manual en los procesos de impresión y embalaje.

Control del proceso de impresión:

La impresora rotativa Thimson posee controles completamente manuales. Se arranca y se detiene por medio de pulsadores y la velocidad es controlada mediante unas perillas. Todos estos controles están ubicados en el cuerpo de control de impresión.

El control es realizado por un solo operador que se encarga del funcionamiento de la máquina. Este ubica el papel en las guías de impresión de los rodillos. Tensa el papel para comenzar la impresión y se encarga de ajustar los soportes de la máquina al tamaño de papel a imprimir, incorpora los colores en los cuerpos de impresión según las indicaciones del diseño y se asegura que los módulos de aplicación de color tengan suficiente tinta para comenzar la impresión. Cuando enciende la máquina, el operador se encarga de supervisar el funcionamiento de la máquina, revisa que el papel esté alineado y se asegura de que la impresión y registro sean uniformes.

El mismo operador es también encargado de detener la máquina. Antes de imprimir un pedido el operador se le informa cuantos metros de papel impreso son necesarios para completar el pedido. El operador tiene que detener la máquina cuando el indicador analógico de metros procesados indique que se alcanzaron la cantidad de metros necesarios, o eventualmente por roturas de papel o fallas mecánicas.

Control del proceso de embalaje:

Como se explico en el punto 4.1.3, el proceso de embalaje es completamente manual, por lo que el control es aplicado por los trabajadores del departamento de embalaje. La única acción de control destacable existe cuando el pedido del cliente es grande y no puede realizarse este proceso de manera continua. La máquina empacadora hace los paquetes más rápido de lo que los trabajadores pueden transportarlos y embalarlos por lo que se



sobrellena la bandeja al final de la máquina. En este sentido, el operador de la máquina empacadora tiene que parar regularmente la máquina y esperar a que se desocupe la bandeja. Ver figura 4.10



Figura 4.10: Detención de la máquina empacadora por sobrecarga de bandeja

4.2.2 Determinación de si los procesos o controles identificados perjudican la producción y/o productividad.

Luego de analizar la información de los procesos y los controles de los mismos se encontraron elementos que perjudican la producción y productividad. A continuación se detallan estos elementos perjudiciales.



Control de proceso de impresión:

El control de la impresora rotativa es manual y es realizado por solo un operador. Entre las responsabilidades de este operador (mencionadas en el punto 4.2.1) también tiene que detener la máquina cuando el indicador analógico de metros procesados indique que se alcanzó la cantidad de metros necesarios para completar la orden de producción. El proceso de impresión puede tardar horas y es muy difícil para el operador detener la máquina justo cuando se procesen la cantidad de metros necesarios. En muchas ocasiones acciona el botón de detención cuando se han procesado cientos de metros por encima de lo necesario.

Otra situación que se presenta en este proceso es que cuando hay poco tiempo para completar el pedido el operador aumenta la velocidad de la máquina por encima de los valores nominales de operación, esto debido que los operadores no se les permite el sobretiempos si no que se paga un bono de producción si logran completar la producción en los tiempos estipulados, de acuerdo a los estándares de producción. Esta situación ocasiona rupturas del papel e interrupción del proceso.

El registro de los metros procesados es completamente manual. Un trabajador toma nota al final de la jornada de la cantidad de metros impresos, sin identificar cuántos de esos metros pertenecen a los pedidos procesados en el día. Esto dificulta el pago de los incentivos de producción ya que no se puede conocer con precisión la magnitud de la producción que tuvo el área de impresión en relación con cada orden de producción.

Proceso de embalaje:

Este proceso representa un problema en si mismo ya que depende exclusivamente en la mano de obra de los trabajadores, lo que lleva al atraso de la producción de los pedidos y situaciones más graves como lesiones sufridas por los trabajadores.



Cuando el pedido del cliente es grande, no puede realizarse este proceso de manera continua. La máquina empacadora hace los paquetes más rápido de lo que los trabajadores pueden transportarlos y embalarlos por lo que se tiene que parar regularmente la máquina y esperar a que se desocupe la bandeja de la empacadora. Esto representa un atraso en la producción que se traduce en pérdidas monetarias ya que el proceso no puede continuar si no se embalan los paquetes.

Otra situación también problemática surge en el proceso de embalaje. Al igual que en el proceso de impresión, cuando hay poco tiempo para completar el pedido los trabajadores corren con los paquetes para cumplir el tiempo, lo que representa un gran riesgo a la salud de los operarios ya que pueden lesionarse. Según registro de la compañía en lo que va del año 2012 han ocurrido 3 accidentes en el área de embalaje, causados por la prisa de cumplir los pedidos. También se han originado pérdidas de paquetes por sobresaturación en la bandeja de la empacadora y daños a la máquina empacadora por excesivos arranques y paradas.

4.3 Diseño y prueba de un dispositivo controlador-registrador para la automatización la máquina de impresión.

Para el diseño se procede a construir una idea general del dispositivo, según la metodología Top-Down indicada en el Capítulo III. El dispositivo controlador-registrador está constituido de varios elementos que se explican como bloques de función y se muestran en un diagrama de bloques en la figura 4.11

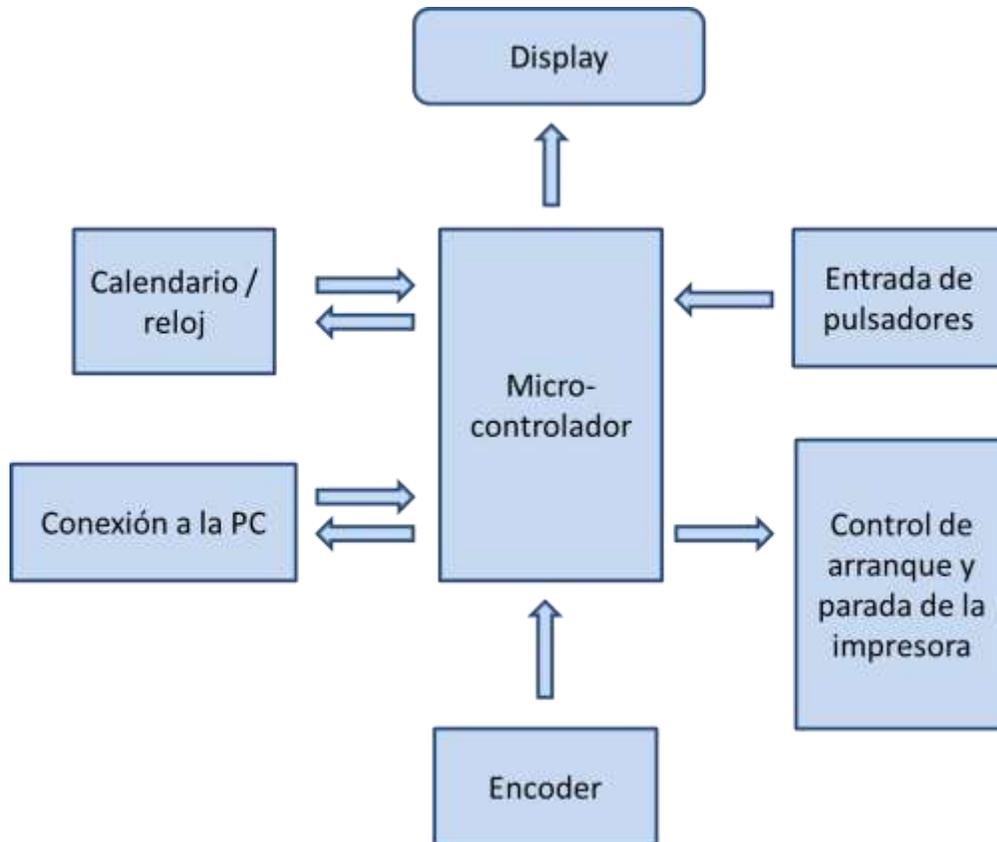


Figura 4.11: Diagrama de bloques controlador-registrador

En la figura 4.11 se observan:

- Un bloque de función llamado microcontrolador en el cual se encarga de ejecutar una lógica que permita la interacción de todos los bloques de funciones.
- Un bloque llamado encoder, este representa la entrada de la señal que viene del encoder que se instalaría en el rotor de la máquina impresora.
- El bloque “entrada de pulsadores” representa a los pulsadores que utilizan los usuarios para controlar las acciones que desempeña el dispositivo.



- El bloque de comunicación con PC, como su nombre indica, permite la transmisión y recepción de datos entre el dispositivo y la PC.
- El bloque calendario que permite al dispositivo conocer la fecha y hora actual.
- Un bloque de control de arranque y parada de la impresora que permite el control de la misma.
- El bloque de display representa una manera de interacción con el usuario para mostrar información.

Se procedió a la aplicación de las actividades necesarias para el diseño y prueba de un dispositivo controlador-registrador para la automatización de la máquina de impresión rotativa Thimson TRS2000.

4.3.1 Selección de un microcontrolador y periféricos adecuados para su aplicación en la automatización en la máquina de impresión.

Para la selección del microcontrolador se tomó en cuenta las capacidades requeridas para realizar las funciones propuestas

- Capacidad para procesar señales cuadradas, su frecuencia y ciclo útil. Esto debido a que la velocidad del rotor de la máquina se obtiene a través de un encoder adherido al rotor.
- Compatibilidad con protocolos de comunicación estándar a través de puerto serial RS-232. Es necesaria esta comunicación para enviar los datos de los registros de las órdenes procesadas a la computadora.
- Compatibilidad con protocolo de comunicación I2C, para extraer datos de un chip calendario que permita obtener hora y fecha de realización de los pedidos.
- Memoria EEPROM integrada, para poder guardar los datos de los pedidos de manera segura en caso de una falla de alimentación.
- Capacidad para realizar cálculos numéricos básicos con variables tipo Float.



- Suficientes entradas y salidas para todas las tareas mencionadas anteriormente a demás de entradas para pulsadores, salidas para indicadores como LEDs.
- Capacidad para soportar pantallas LCD.
- Este microcontrolador debe estar disponible y ser de fácil adquisición en el mercado local.

Con estas capacidades identificadas, se seleccionó el microcontrolador: PIC18F4520, ver figura 4.12



Figura 4.12: PIC18F4520

Fuente: Msebilbao (2012), documento en línea [40]

Este micro-procesador de la marca Microchip posee todas las características que se identificaron anteriormente, incluyendo:

- USART mejorada que soporta el protocolo serial RS-232
- Dos módulos de captura/comparación/PWM para el proceso de señales cuadradas.
- Módulo maestro I2C.
- Memoria EEPROM integrada
- Capacidad para realizar cálculos numéricos básicos con variables tipo Float.
- Capacidad para soportar pantallas LCD.
- Cinco puertos de entradas y salidas digitales.



En cuanto a los periféricos se seleccionaron los siguientes:

Pantalla LCD: Esta pantalla es de 2x16 caracteres, tecnología LCD con un bus de datos de 8 pines, luz de fondo y control de contraste. Ver figura 4.13



Figura 4.13: Pantalla LCD

Fuente: Loging (2012), documento en línea [41]

Chip Calendario: Se usó el integrado DS1307, el cual se comunica por protocolo I2C y provee la hora y fecha al PIC. La fecha y la hora son programables por el usuario. Necesita un cristal como oscilador de 32.768kHz y una fuente separada de 3VDC. . Ver figura 4.14

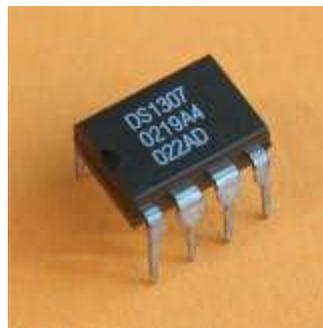


Figura 4.14: Chip calendario

Fuente: Bkbelectronics (2012), documento en línea [42]



4.3.2 Desarrollo de un programa que permita el funcionamiento del dispositivo.

Se utilizó la metodología de programación estructurada como se estipula en el Capítulo III para el desarrollo del programa del microcontrolador. El entorno de desarrollo seleccionado para la creación del programa es Proton IDE. Este entorno soporta lenguaje Basic y posee un compilador Basic que crea los archivos .hex utilizados por el PIC, como programas. Este entorno es una excelente herramienta ya que es versátil, eficiente y el editor de programa es amigable al usuario.

Para acceder a Proton IDE se selecciona el icono del escritorio y se muestra la ventana de trabajo de Proton como se muestra en la figura 4.15

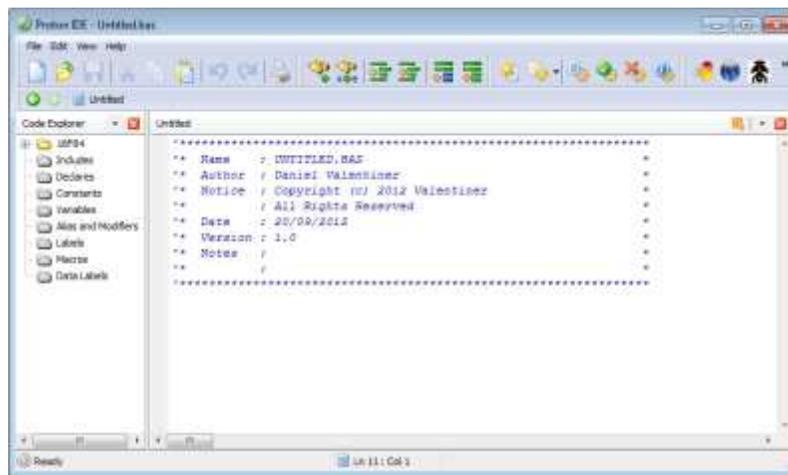


Figura 4.15: Pantalla de inicio Proton

Para obtener las librerías con los registros adecuados al PIC usado, se selecciona dentro de Proton IDE el PIC a usar. Una vez seleccionado el dispositivo, la librería se cambia automáticamente y aparecen los registros pertinentes al PIC, como se muestra en la figura 4.8. En esta se muestra como al seleccionar el dispositivo aparece en la ventana “Code explorer” a la izquierda todos los registros pertenecientes al PIC18F4520. Ver figura 4.16

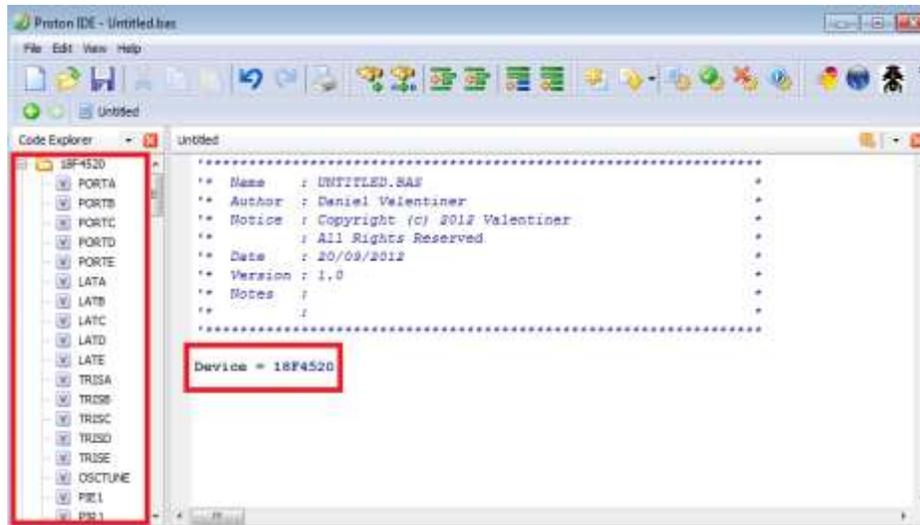


Figura 4.16: Librería

Se realizaron las configuraciones pertinentes para el funcionamiento. Primero se configuraron los registros que controlan el oscilador interno del PIC. Ver figura 4.17

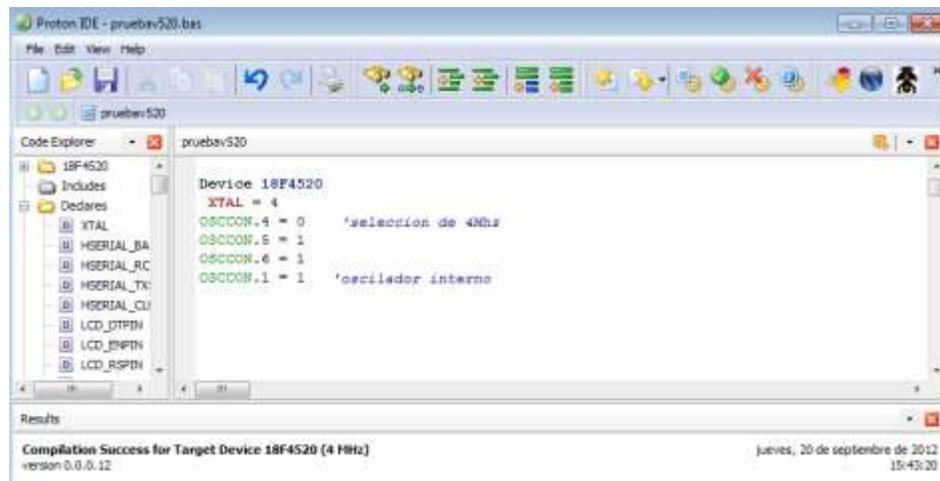


Figura 4.17: Oscilador PIC



La configuración de la comunicación serial en el programa se hace directamente en el comando que ejecuta la acción. Los comandos de recepción y transmisión serial son “Serin” y “Serout” respectivamente, y puede identificarse la configuración de los mismos en la figura 4.18.

```
GoTo Main
Sinc:
  Cls
  Print At 1,2,"Conectando.."
  SerIn PORTC.7,16468,7000, Mal,[portin] 'Recepción RC7, 9600bps, 7s antes de saltar etiqueta
  If portin <> "g" Then GoTo Sinc
  Print At 1,2,"Sincronizando.."
  SerOut PORTC.6,16468,[114] 'Transmisión RC6, 9600bps
```

Figura 4.18: Configuración comunicación serial

Se seleccionó el puerto que servirá como conexión a la pantalla LCD. Ver figura 4.19

```
Declare LCD_DTPIN PORTD.4 * Pin de Datos LCD conectado al PORTC.4
Declare LCD_ENPIN PORTD.2 * Pin de Enable LCD conectado al PORTC.3
Declare LCD_RSPIN PORTD.0 * Pin de RS LCD conectado al PORTC.0
```

Resultado
Compilation Success for Target Device 18F4520 (4 MHz)
version 0.9.0.12
jueves, 20 de septiembre de 2012
10:43:20

Figura 4.19: LCD PIC



A continuación se declararon las entradas y salidas a utilizar en los puertos a utilizar, en este caso son los puertos “B” y “D”. Ver figura 4.20

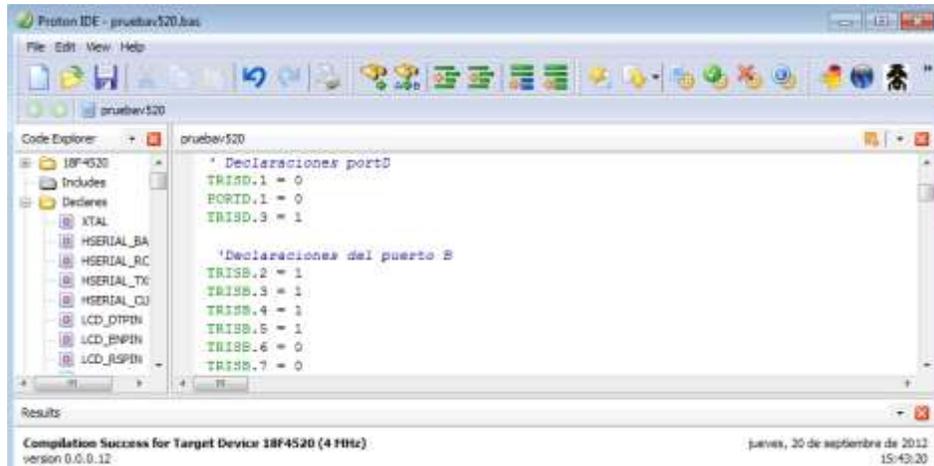


Figura 4.20: Puerto PIC

Luego se declararon las variables a utilizar en el programa, especificando nombre y tipo de variable, entre ellas byte, palabras y flotantes o float. Ver figuras 4.21, 4.22 y 4.23.

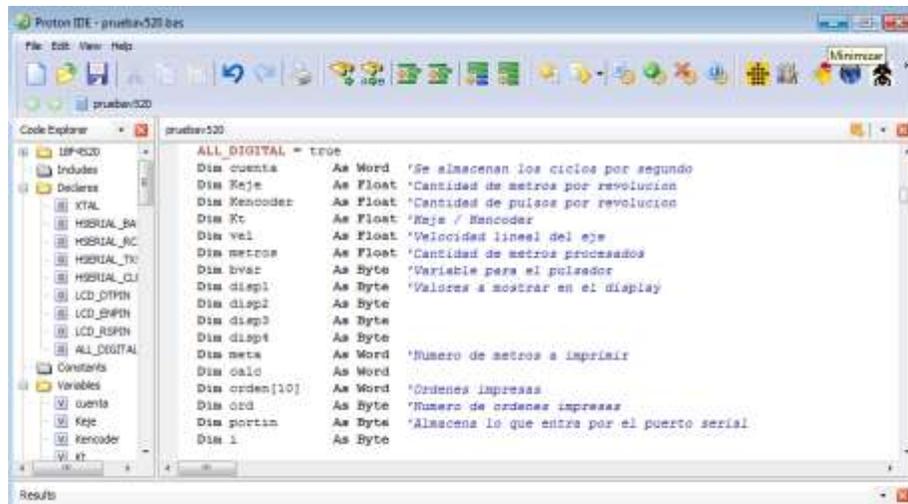


Figura 4.21: Variable 1 PIC

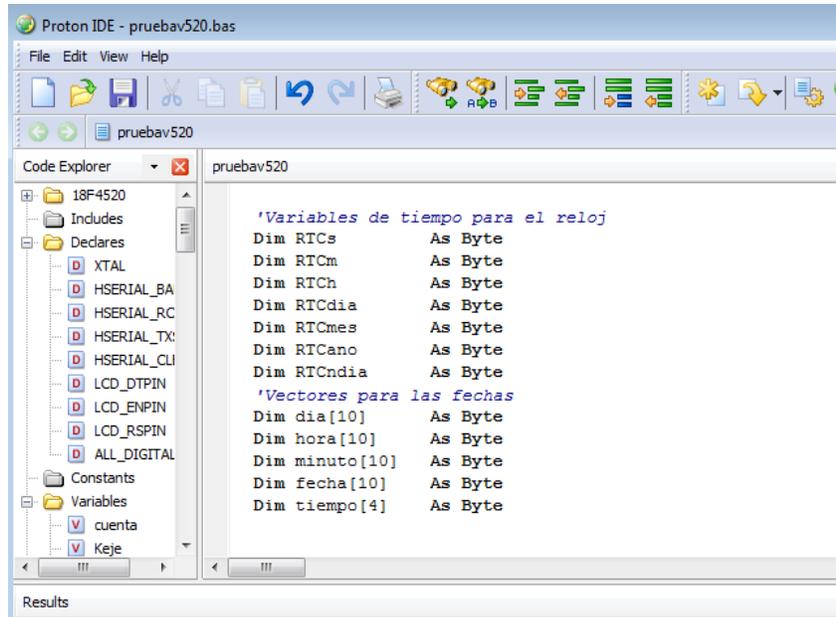


Figura 4.22: Variable 2 PIC

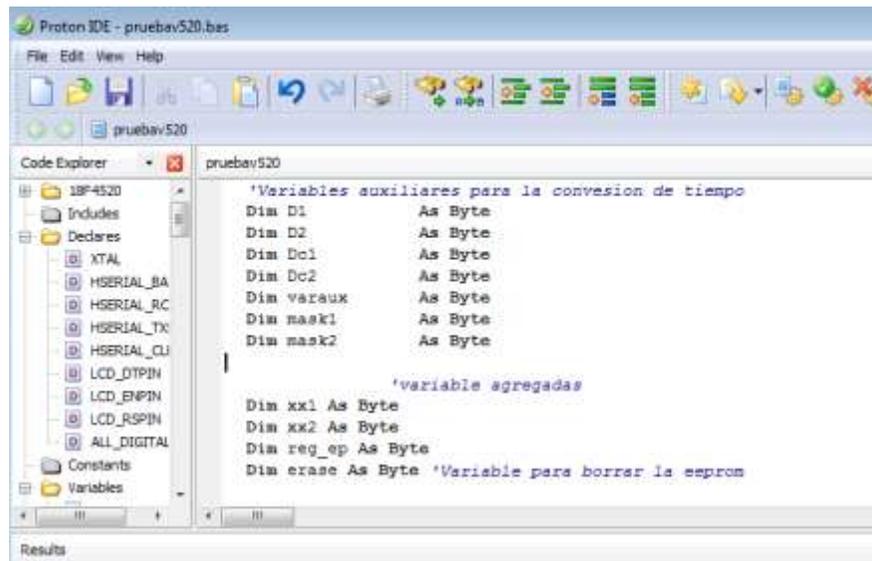


Figura 4.23: Variable 3 PIC



Después se declararon los símbolos del sistema, que se utilizan como un cambio en la referencia de una palabra reservada del editor, por preferencia del usuario. También se declararon las constantes. Ver figura 4.24

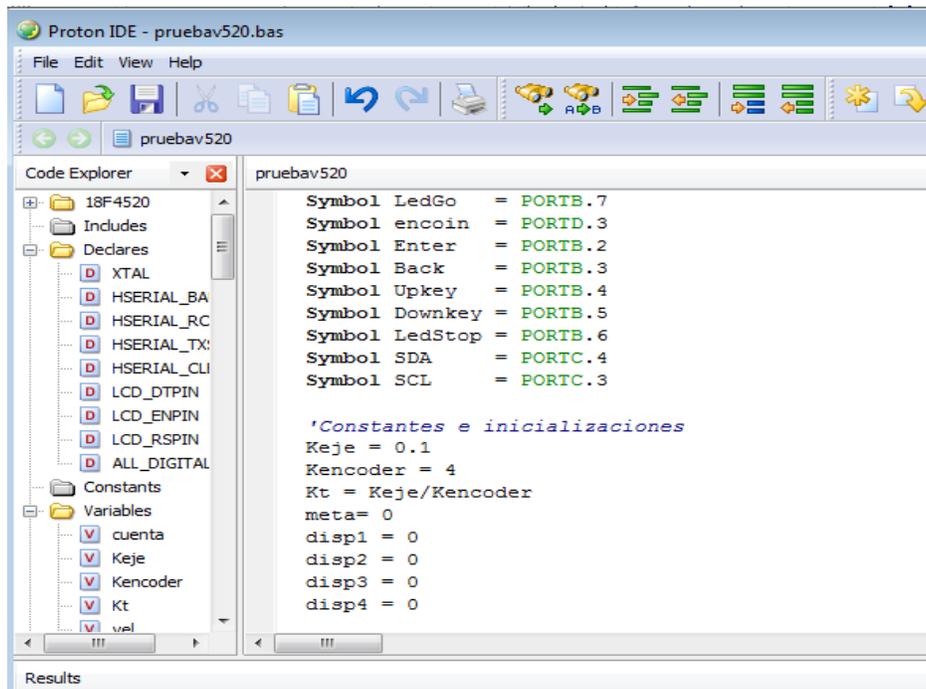


Figura 4.24: Constantes PIC

Se continuo a desarrollar el programa principal, bajo la metodología de programación estructurada se ideó bajo condiciones, saltos y subprogramas, una lógica de ejecución de programa que permite al microcontrolador desempeñar las tareas requeridas. El código del programa se encuentra plasmado en su totalidad en el Apéndice A y en el Apéndice B se encuentra el diagrama de flujo que muestra la lógica del programa.



4.3.3 Desarrollo de una aplicación para computadora que permita extraer la información registrada por el dispositivo.

Esta aplicación se desarrolló utilizando la metodología de programación dirigida a objetos, tal como se indica en el Capítulo III. Para el desarrollo de esta aplicación se seleccionó como entorno de desarrollo Microsoft Visual Studio 2010. Este entorno permite programar en distintos lenguajes como C++, C, Visual Basic, entre otros. Para esta investigación se utilizó el lenguaje de programación Visual Basic como se muestra en la figura 4.25

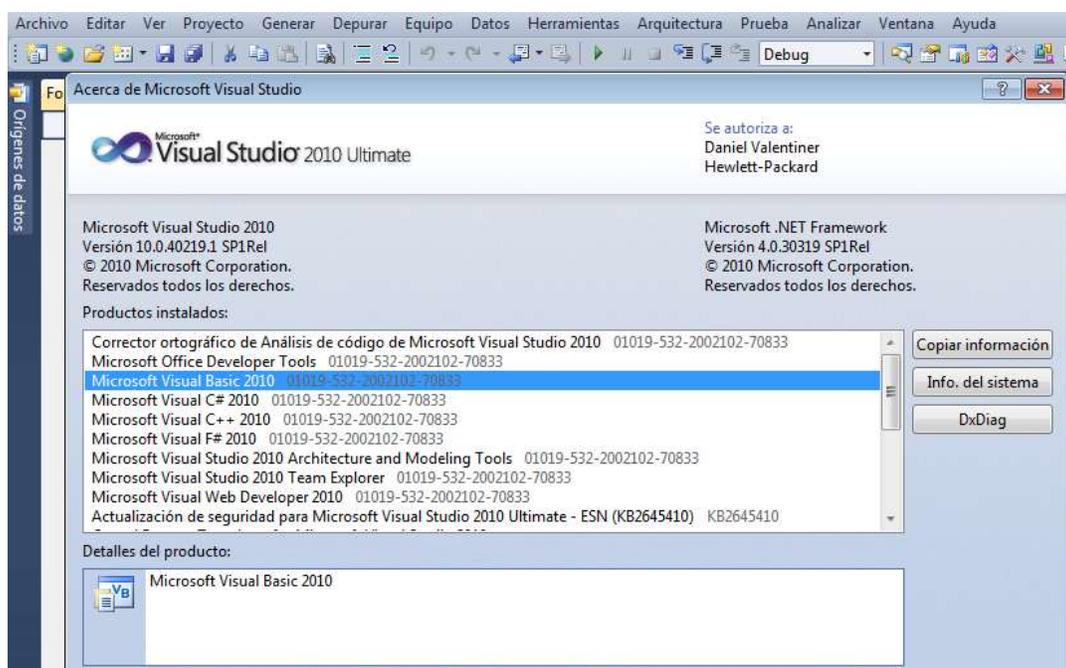


Figura 4.25 Selección del lenguaje Basic

El entorno de desarrollo Visual Studio permite la creación de una Interfaz Gráfica de Usuario (IGU) utilizando un editor en el que se encuentran objetos comúnmente utilizados con características predefinidas, estas pueden cambiarse para adaptarse a las



necesidades del usuario. Los manejadores de eventos también son predeterminados, pero las acciones que realizan los objetos cuando ocurre un evento son libres para codificar. En la figura 4.26 se muestra una barra de herramientas donde se encuentran los objetos más comunes utilizados para el desarrollo de la Interfaz Gráfica de Usuario.

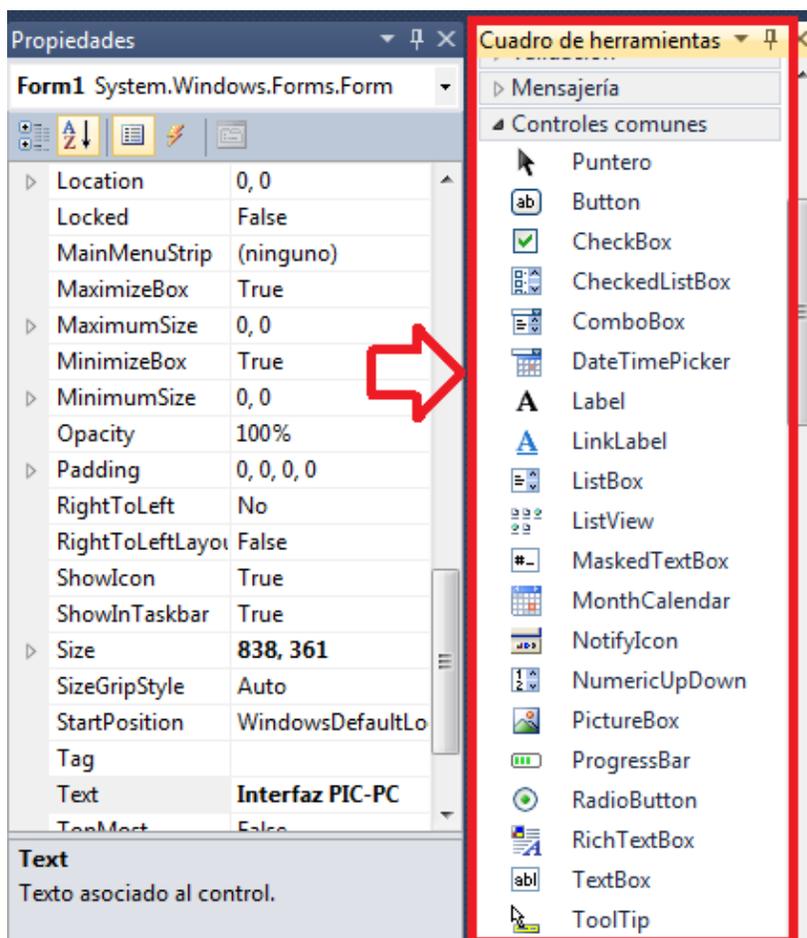


Figura 4.26: Cuadro de herramientas



Se selecciona un objeto y se coloca en la forma donde se realiza la Interfaz Gráfica de Usuario. Luego se selecciona y se modifican los parámetros. Como se observa en la figura 4.27, existen varios campos de características del objeto que se pueden modificar.

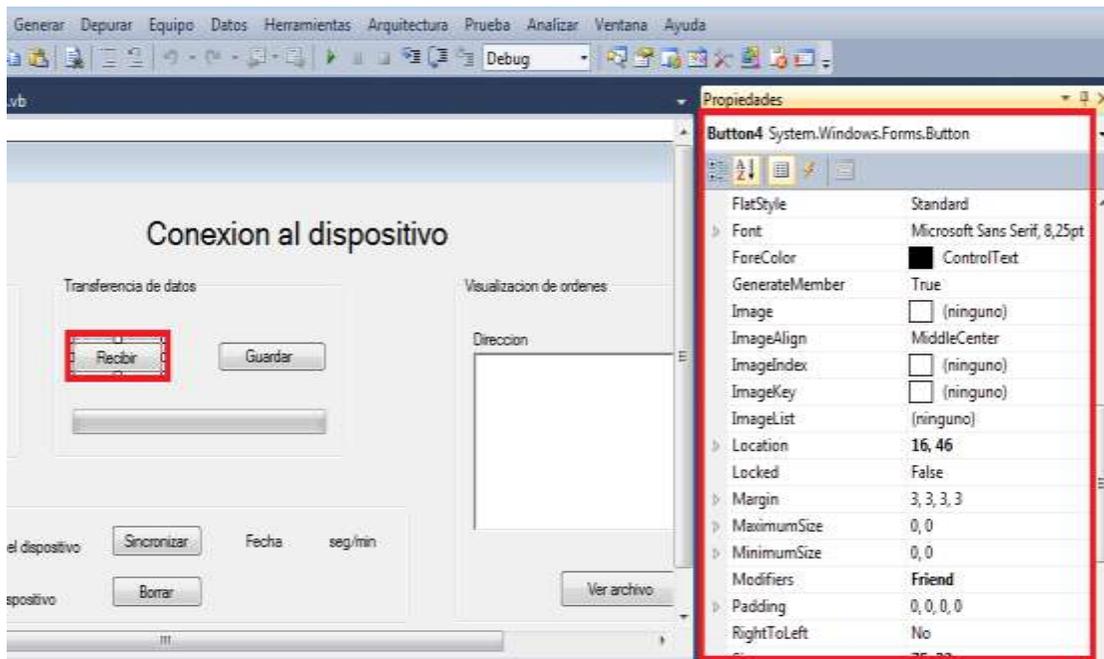


Figura 4.27 Características del objeto

Al seleccionar cualquier objeto ubicado en la forma de trabajo, el editor abre una ventana donde existe el código generado al ubicar o crear un objeto. Se crea una función para el manejo de eventos y dentro de esta se puede codificar para que realice la acción deseada. En la figura 4.28 se nota una función de manejo de evento creada para el botón recibir (mostrado en la figura 4.27), en la cual se codifica para que haga la función de recibir los datos de microcontrolador al presionar el botón.



```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    Dim datosHigh() As Byte 'Aqui se guardan los datos recibidos del pic
    Dim datosLow() As Byte
    Dim sal As Byte 'Con esta variable se asegura que funciona la conexion
    Dim ordenes As Byte 'Numero de ordenes
    Dim paso As Decimal 'Para la barra de progreso

    SerialPort1.ReadTimeout = 8000 ' Timeout para la lectura de los datos
    Saludo:
    SerialPort1.Write("g")
    Try
        sal = SerialPort1.ReadByte()
    Catch ex As TimeoutException
        MsgBox(Err.Description)
        GoTo Dataerror
    End Try

    If sal <> 114 Then GoTo Saludo:
    Try
        ordenes = SerialPort1.ReadByte()
    Catch ex As TimeoutException
        MsgBox(Err.Description)
        GoTo Dataerror
    End Try
End Sub
```

Figura 4.28 Código de recibir

Esta acción se realizó para todos los objetos, también se incluyeron variables globales para el manejo de los datos, como se indica en la figura 4.29. Estas variables son usadas por cada función, pero a diferencia de las variables locales (como la variable “sal” en la figura 4.28), estas no dependen de las funciones, son modificadas por ellas pero pueden ser utilizadas por todas las funciones en el programa.



```
Public Class Form1
    Dim datos1() As Integer 'Guarda los metros procesados
    Dim dia() As Byte 'Dia en que fue hecha la orden
    Dim hora() As Byte 'Hora de termino de la orden
    Dim min() As Byte 'Minuto de terminada la orden
    Dim ordgen As Byte 'Guarda el numero de ordenes
    Dim registro As String 'Guarda la direccion del archivo con las ordenes grabadas
End Class

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    SerialPort1.PortName = TextBox1.Text
    Button1.Enabled = False
    Button4.Enabled = True
    Button2.Enabled = True
    Button7.Enabled = True
    Try
        SerialPort1.Open()
    Catch ex As Exception
    End Try
End Sub
```

Figura 4.29: Variables globales

Para algunas acciones que realiza esta interfaz se tuvo que configurar ciertos controles especiales. Uno de estos es el manejo de puerto serie para la comunicación con el microcontrolador; el control de puerto serie puede ser encontrado en el cuadro de herramientas, se agrega a la forma (ver figura 4.30), esto permite que el editor identifique el código utilizado al referirse a este puerto.

Otro control es el explorador de archivos de Windows. El cual se activa para guardar en la computadora el archivo de datos creado a partir de los datos recibidos del microcontrolador. De la misma forma que el puerto serial, en el cuadro de herramientas se encuentra el control del explorador de archivos de Windows y se agrega a la forma. Como se indica en la figura 4.30.

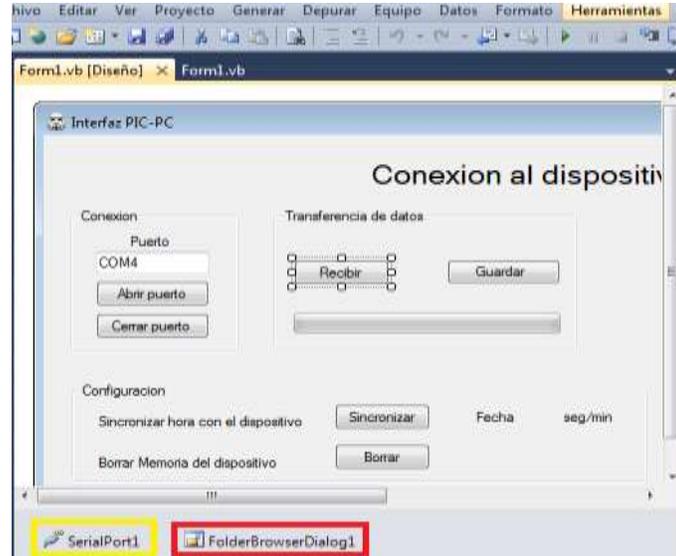


Figura 4.30: Controles especiales

Luego de realizada la programación, la interfaz gráfica luce de la siguiente manera (ver figura 4.31).

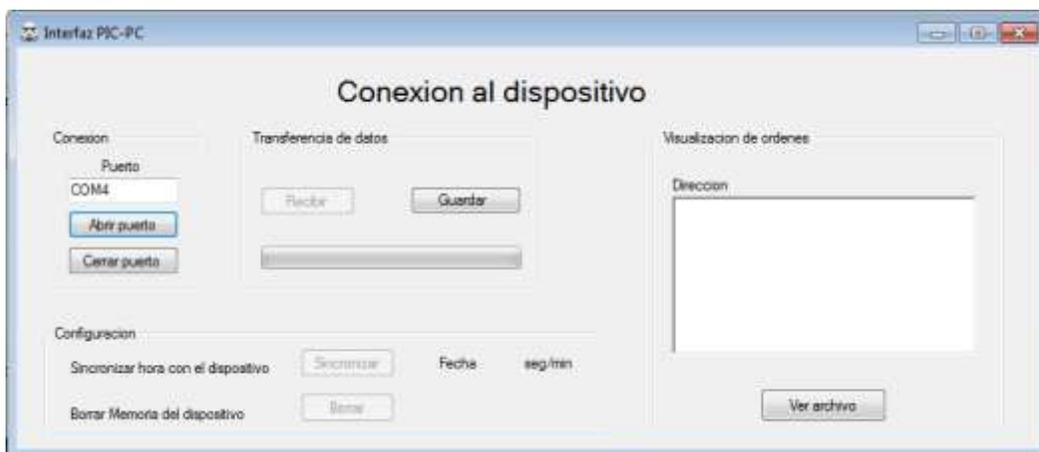


Figura 4.31: Interfaz PIC-PC



Se realizó toda la programación pertinente a manejo de eventos, creación y lectura de archivos, comunicación serial y una estructura de programación dedicada a objetos, como lo se indica en el Capítulo III, que permitió a la aplicación desempeñar todas las actividades necesarias para servir como interfaz para comunicación con el microcontrolador. El programa se encuentra plasmado en su totalidad en el Apéndice C.

4.3.4 Simulación por software del dispositivo en operación.

Para la simulación del funcionamiento se utilizó el software de simulación de circuitos ISIS Professional. Este software permite simular circuitos con componentes comerciales como microcontroladores, compuertas lógicas, pulsadores, entre otros. También es posible cargar un programa a un microcontrolador simulado y observar paso a paso la ejecución del mismo. Otra característica muy importante es la habilidad de simular conexión a cualquier puerto serial de la computadora (puertos COM).

El circuito realizado en el programa se fundamentó en el diagrama de bloques de la figura 4.32. Se cambiaron los bloques de función por los dispositivos seleccionados en el punto 4.3.1 y se conectaron.

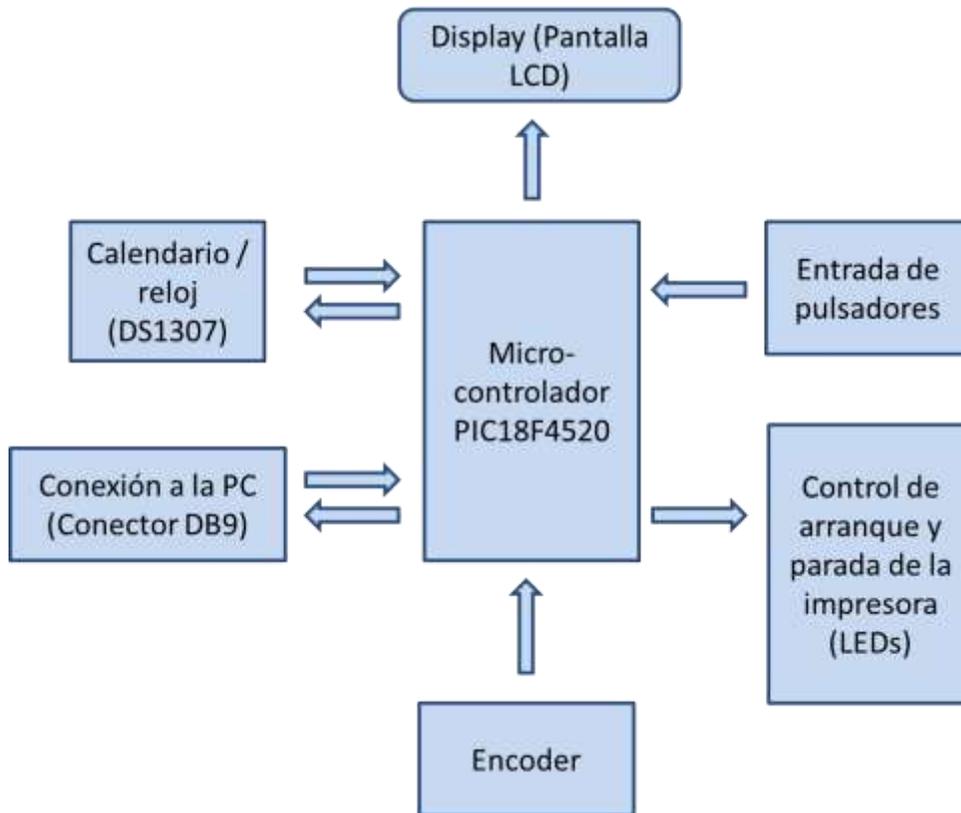


Figura 4.32: Diagrama de bloques controlador-registrador con componentes

Se inicio colocando los dispositivos seleccionados, como se menciona en el punto 4.3.1. Se busca en las librerías existentes en el programa cada uno de los dispositivos (ver figura 4.33).

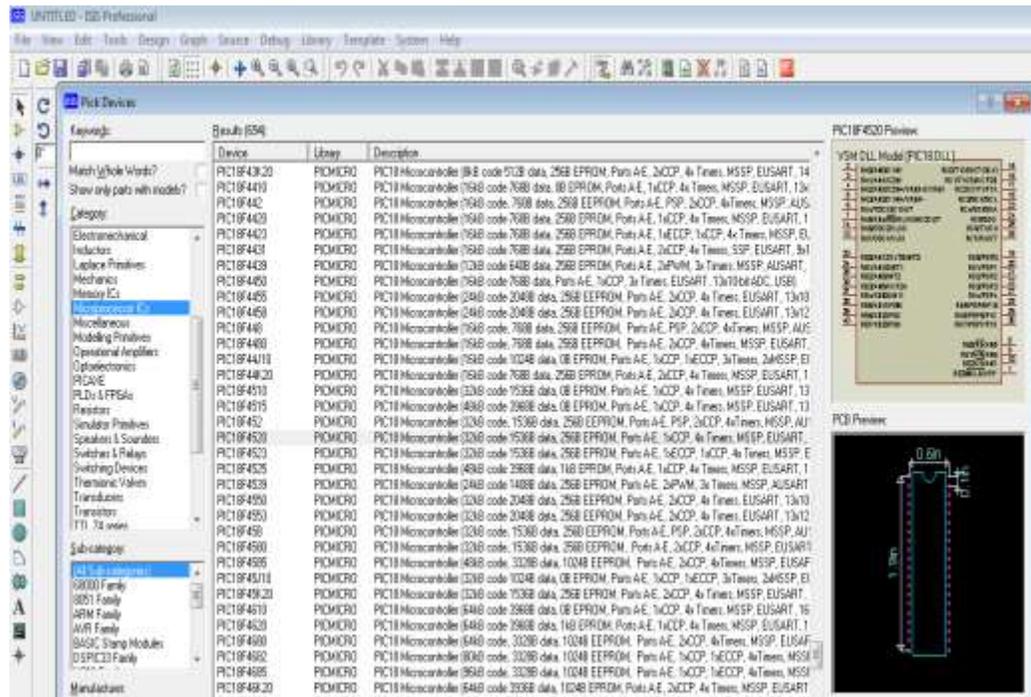


Figura 4.33: Búsqueda de dispositivos

Una vez que ubicados en el área de trabajo todos los dispositivos necesarios encontrados en la librería, se procedió a conectar los dispositivos entre sí, de acuerdo a la configuración de los puertos hechas en el programa para el microcontrolador desarrollado en el punto 4.3.2, obteniendo así el circuito que se muestra en la figura 4.34, donde se muestran las conexiones entre componentes.

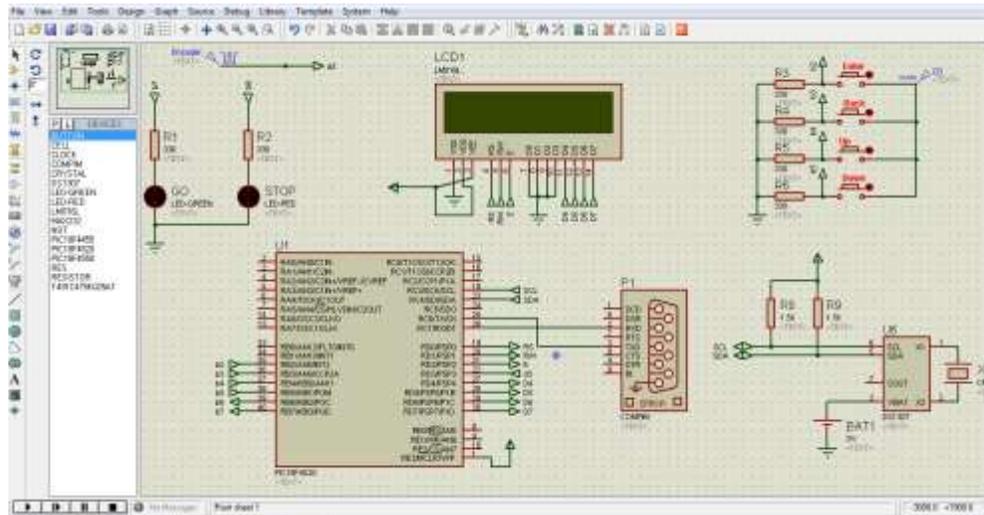


Figura 4.34: Circuito para simulación

En la figura 4.35 se muestra una parte del circuito a simular. Se encuentran indicados mediante cuadros numerados los componentes usados en la simulación y se describen a continuación:

1. Esta es una fuente de voltaje, de señal cuadrada y amplitud 5V. Se utilizó para simular el encoder que mide la cantidad de revoluciones en el eje de la máquina impresora.
2. El LED identificado como “Go” representa la señal enviada a un relé que activa la señal de control que enciende la máquina impresora. El otro LED identificado como “Stop” representa una señal enviada a otro relé que activa una señal de control que detiene la máquina.
3. El dispositivo indicado es la pantalla LCD.



4. Aquí se encuentran los 4 pulsadores que controlan el dispositivo. Los pulsadores están identificados como “Enter” (continuar), “Back” (atrás), “Up” (arriba) y “Down” (abajo).

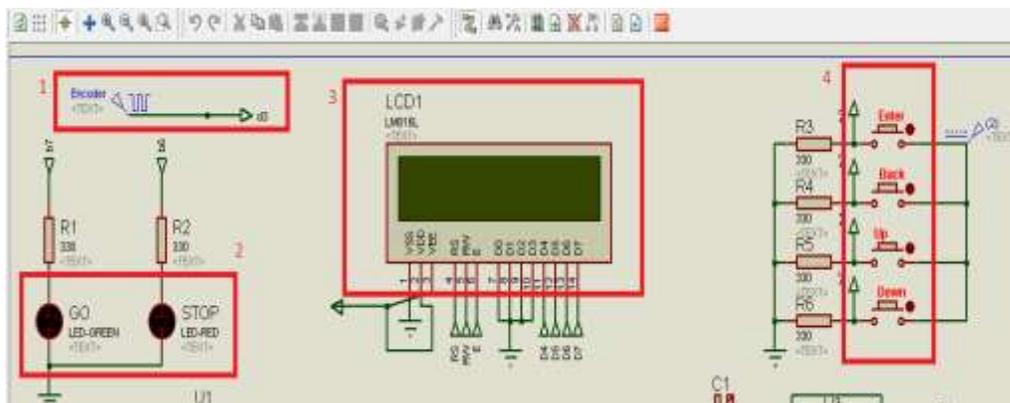


Figura 4.35: Circuito identificado controlador-registrador interfaz

En la figura 4.36 se muestra otra parte del circuito a simular. Se encuentran indicados mediante cuadros numerados los componentes usados en la simulación y se describen a continuación:

5. Este es el modelo del PIC18F4520
6. Conector DB9. Configurado para 9600bps de velocidad de transmisión.
7. Circuito integrado DS1307 (chip calendario).

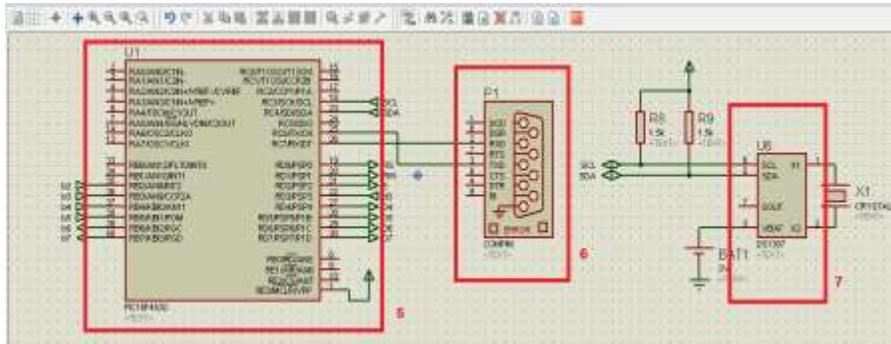
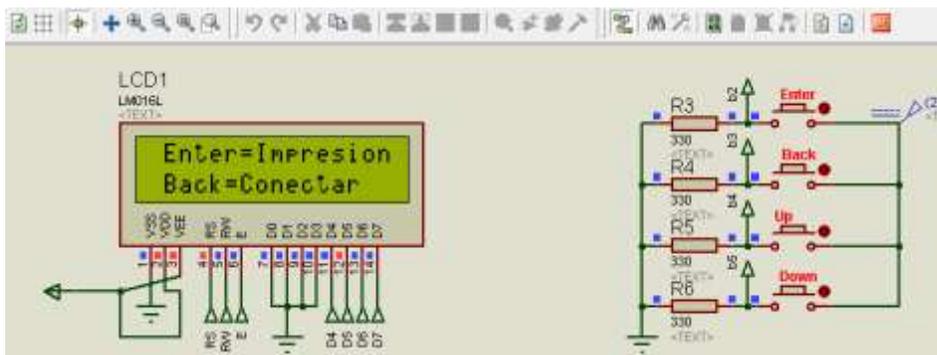


Figura 4.36: Circuito identificado controlador-registrador comunicación

Se le cargó al modelo de microcontrolador, el programa desarrollado en el punto 4.3.2 y se comenzó la simulación. Lo primero que se observa es la pantalla (ver figura 4.37) mostrando un menú donde se le da la opción al usuario de seleccionar la acción que desee realizar con el dispositivo. Si se presiona el pulsador “Enter”, se comienza el proceso de impresión y si se presiona el pulsador “Back”, se entra en otro menú de comunicaciones.

Figura 4.37: Pantalla principal controlador-registrador



El primer proceso probado fue el de impresión, Así que se presionó el pulsador “Enter” Al hacer esto, se accede a un mensaje en la pantalla que dice “Introduzca metros” (ver figura 4.38). Esta pantalla solo aparece momentáneamente para indicarle al usuario que lo que tiene que hacer a continuación es introducir la cantidad de metros a imprimir.

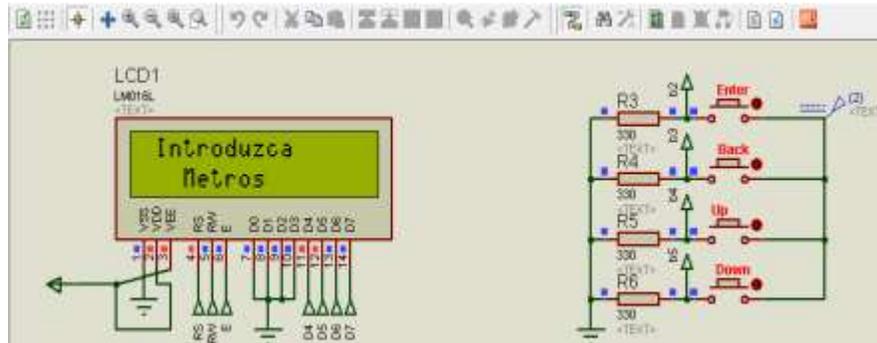


Figura 4.38: Indicación de metros

Luego la pantalla muestra la cantidad de metros introducidos (ver figura 4.39). Para modificar este número se utilizan los pulsadores “Enter” y “Back” para seleccionar que dígito de los 4 dígitos disponibles se quiere modificar, Enter es para seleccionar hacia la derecha y Back para seleccionar hacia la izquierda. Para cambiar el valor del dígito seleccionado se deben utilizar el pulsador Up para aumentar el valor y Down para disminuir el valor.

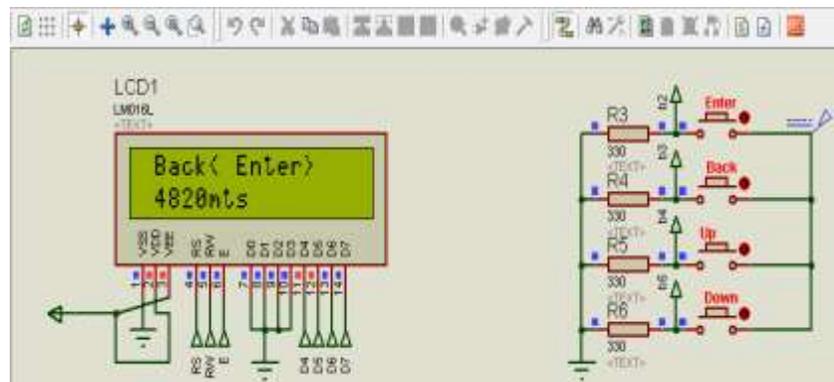


Figura 4.39: Pantalla de metros

Una vez que se introdujeron el números de metros a imprimir se presiona Enter y aparece en la pantalla un mensaje (ver figura 4.40) en donde se pide confirmar el número de



metros a imprimir. Si se presiona Enter se continúa a la impresión y si se presiona Back se regresa a la pantalla principal.

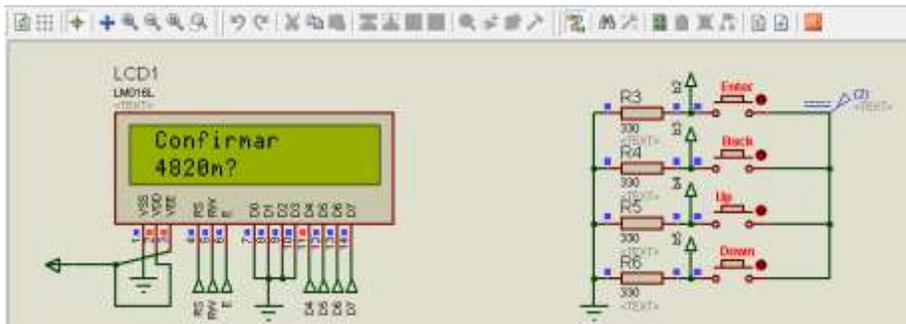


Figura 4.40: Confirmación de metros

Una vez presionado Enter se comienza la impresión. El LED “Go” se enciende, indicando que se arrancó la impresora. En la parte superior izquierda de la pantalla se encuentra la frecuencia de la señal que proviene del encoder. En la parte superior derecha se observa la velocidad de impresión en metros sobre segundos. En la parte inferior izquierda de la pantalla se indican los metros impresos y la parte inferior derecha se muestran los metros totales a imprimir, como se muestra en la figura 4.41.

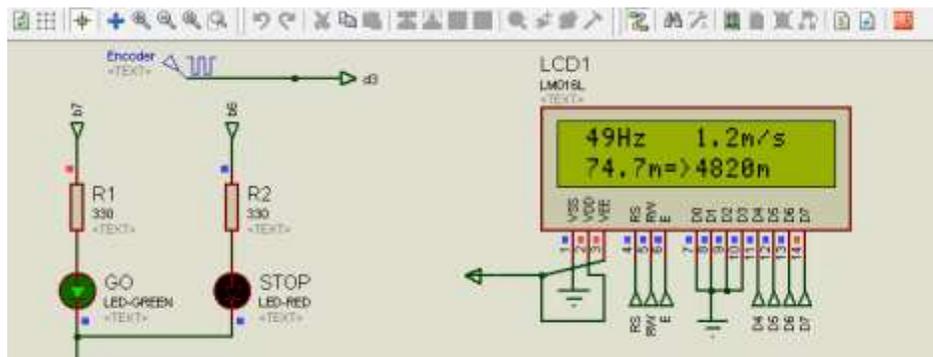


Figura 4.41 Pantalla de impresión



Una vez terminada la impresión, el LED Go se apaga, es decir que la señal de encendido se apaga, y se enciende el LED Stop, señal que indica la parada de la máquina. La pantalla muestra un mensaje que indica que el pedido está listo e indica la cantidad de metros impresos, como se muestra en la figura 4.42

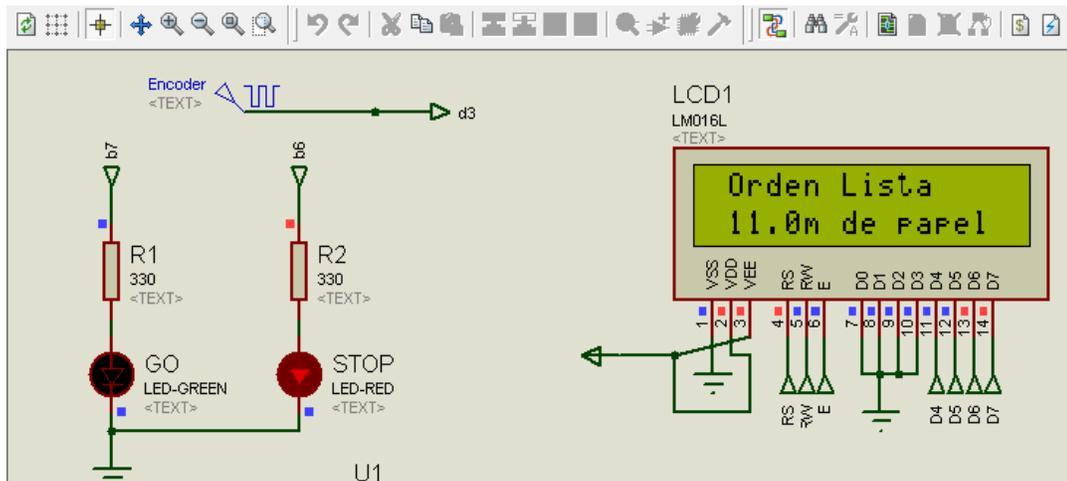


Figura 4.42: Impresión lista

Luego de mostrar momentáneamente el mensaje mencionado anteriormente en la pantalla se muestra el menú principal antes mencionado. En este menú se presionó el pulsador Back para acceder a otro menú de conexión, como se demuestra en la figura 4.43. Cualquier opción de conexión debe realizarse con la conexión a la interfaz desarrollada en el punto 4.3.3. Para esta simulación se conectó el puerto serial de la simulación a un puerto de la computadora (internamente) y así la interfaz PIC-PC tiene acceso al puerto para simular la conexión. Se eligió el puerto COM4, se seleccionó este puerto en la interfaz y luego se pulsó el botón “Abrir” que abre el puerto y realiza la conexión.

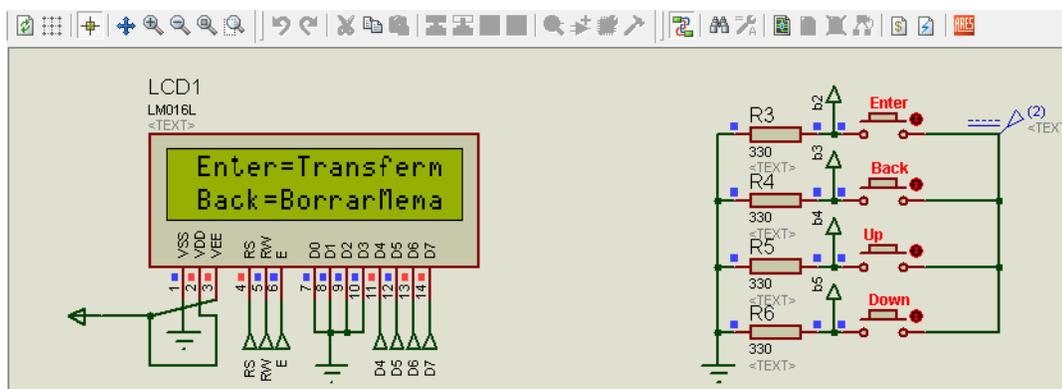


Figura 4.43: Pantalla de conexión

En este menú existen 3 opciones que no se muestran completamente en la pantalla, estas son:

- **Transferm:** Esta opción transfiere los datos registrados de los pedidos impresos, a la computadora a través de la interfaz desarrollada en el punto 4.3.3.
- **BorrarMem:** Esta opción borra la memoria EEPROM del microcontrolador a través de la interfaz con la computadora.
- **Sincroniza:** Esta opción sincroniza el reloj del microcontrolador con el reloj de la computadora a través de la interfaz antes mencionada.

Para recorrer las opciones disponibles se debe desplazar las opciones utilizando los pulsadores Up y Down. Una vez ubicada la opción a elegir se presiona Back o Enter según lo indique la pantalla. Se eligió primero la opción “Transferm”, en la cual la pantalla muestra un mensaje que indica que se esta conectando como se muestra en la figura 4.44.

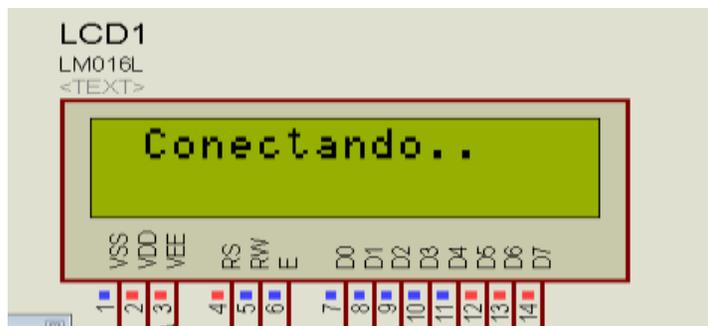


Figura 4.44: Conexión transferencia

Al elegir esta opción se pulsó el botón “Recibir” de la interfaz PIC-PC, una vez hecho esto se realiza la conexión entre el microcontrolador y el PC. Mientras se transmiten los datos, en la pantalla del dispositivo aparece un mensaje que indica que se transfieren los datos. En la interfaz una barra de progreso se llena hasta que la transferencia se completa y la barra se llena completamente como se observa en la figura 4.45.

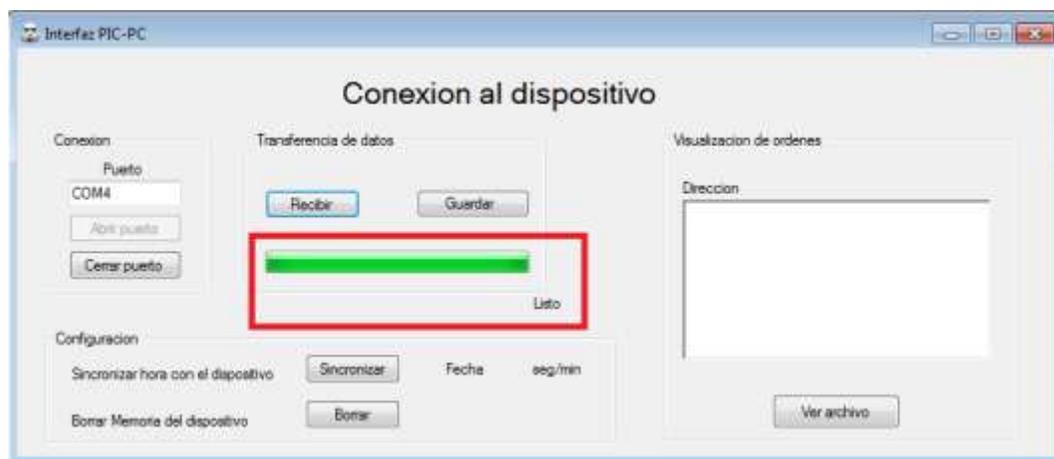


Figura 4.45: Indicador de conexión transferencia



Luego de recibir los datos, estos deben guardarse en la computadora. Para esto se presionó el botón “Guardar”. Al hacer esto se mostró el explorador de Windows para seleccionar el destino en el cual guardar el archivo (ver figura 4.46). El archivo se guarda en la dirección especificada como un archivo de extensión .txt, puede abrirse con la mayoría de los programas de procesamiento de palabras, pero también puede abrirse con hojas de cálculo de manera de procesar los datos directamente.

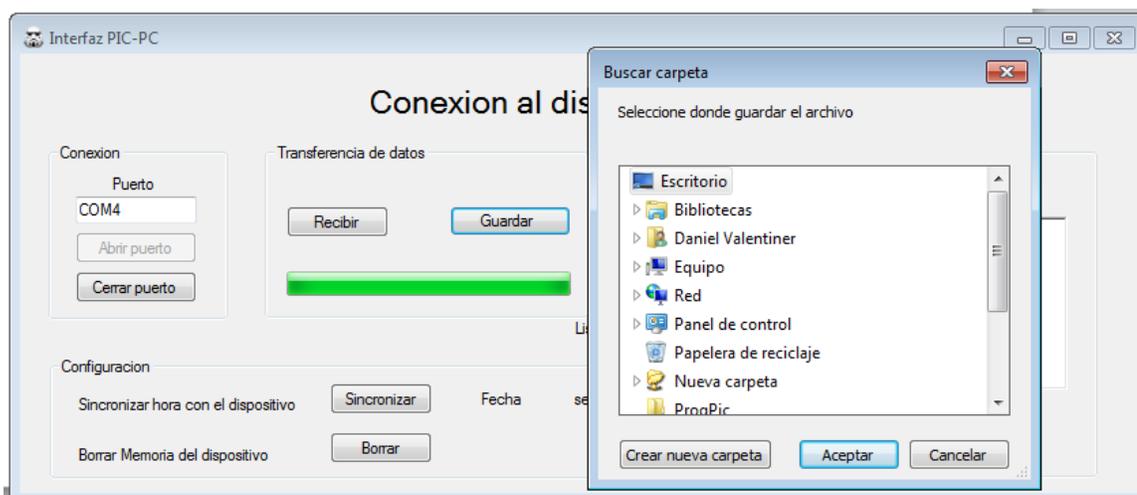


Figura 4.46: Guardado de archivo

Otra opción para la visualización de este archivo es a través de la interfaz PIC-PC. En la parte inferior derecha de la interfaz hay un botón etiquetado con las palabras “Ver archivo”. Se presionó este botón y la interfaz mostró en el cuadro blanco sobre el botón, el contenido del archivo guardado que contiene el número de órdenes realizadas así como la hora y día de finalización de la orden y la cantidad de metros impresos en cada una. Esto se observa en la figura 4.47.



Figura 4.47: Visualización de archivo

La segunda opción de conexión seleccionada fue BorrarMem. Se seleccionó esta opción y se presionó el botón Borrar de la interfaz PIC-PC, al hacer esto la interfaz muestra una ventana de confirmación, se presiona aceptar y se comienza a borrar la memoria EEPROM de dispositivo. Durante el proceso se observó en la pantalla del dispositivo el mensaje “Borrando Memoria”. Una vez finalizado el proceso la interfaz muestra un mensaje como es muestra en la figura 4.48



Figura 4.48: Visualización de archivo (memoria borrada)



La última opción de conexión probada fue la opción Sincroniza. Esta opción permite sincronizar la fecha y hora del dispositivo con la de la computadora. Esto es muy conveniente ya que en el caso de una pérdida de alimentación del chip calendario, este puede volver a ser programado con la fecha y hora actual.

Al seleccionar esta opción se seleccionó en la interfaz el botón “Sincronizar”. Al hacer esto el dispositivo mostró en la pantalla la hora y fecha que poseía la computadora en el momento. Luego se actualizó la fecha y hora en la ventana que permite ver los registros de tiempo del chip calendario en el entorno de simulación Proteus (ver figura 4.49).

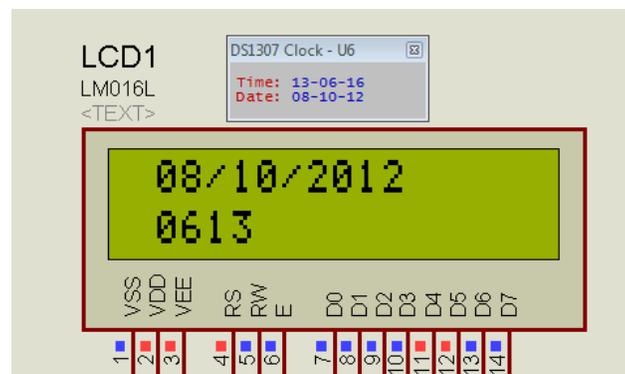


Figura 4.49: Visualización de archivo LCD1

El resultado de la ejecución de estas acciones prueba el correcto funcionamiento del programa del microcontrolador en la simulación. A demás de probar la comunicación y el correcto funcionamiento de la Interfaz PIC-PC. Se logro leer la velocidad e manera simulada, también realizar impresiones y guardar los registros de estas en la memoria. La simulación también fue exitosa en la comunicación. Se pudieron extraer y visualizar los datos de las impresiones simuladas.



4.3.5 Construcción de un prototipo del dispositivo para probar el correcto funcionamiento del mismo.

Debido al correcto funcionamiento observado del circuito simulado, la construcción del prototipo dispositivo se baso en los esquemáticos mostrados en el punto 4.3.4. Se agregaron aquellos componentes operacionales que no se consideran en los esquemáticos de la simulación. En la figura 4.50 se puede observar el esquemático con los cambios para el funcionamiento físico.

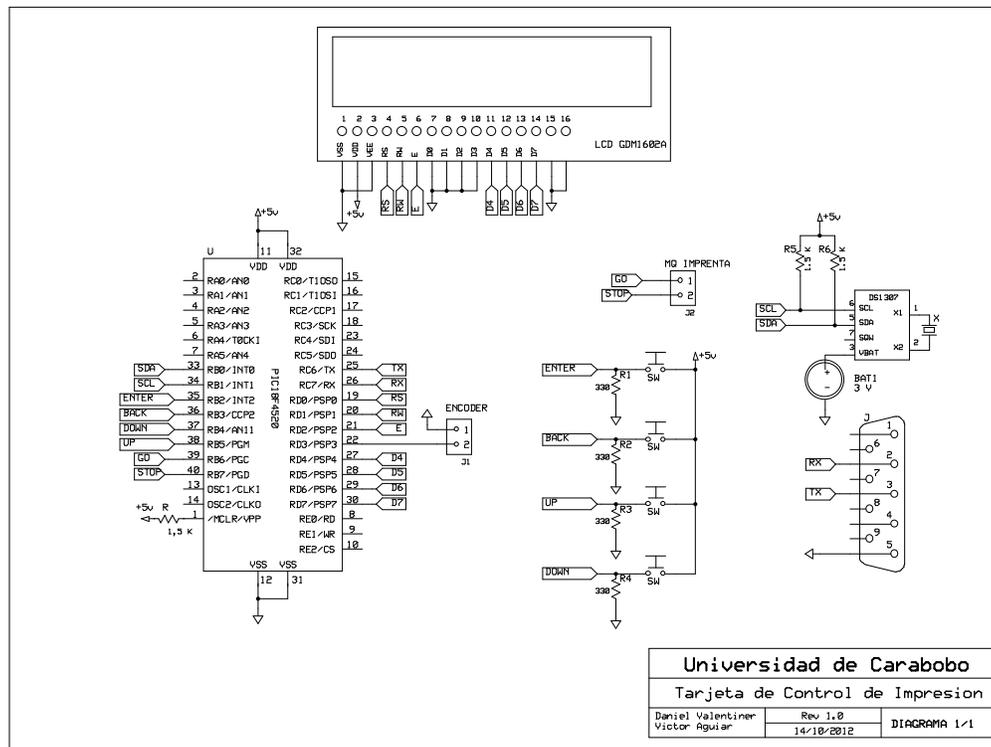


Figura 4.50: Esquemático final



A continuación se enlistan todos los componentes utilizados en la construcción del prototipo:

- Microcontrolador PIC18F4520
- Pantalla LCD 2x16
- Integrado reloj/calendario DS1307
- Cuatro pulsadores
- Conector serial DB9 hembra
- Cristal resonador de 32.768kHz
- Batería tipo botón CR2032
- Base de batería tipo botón
- Cuatro resistores de 330ohms
- Tres resistores de 1.5Kohms
- Dos conectores tipo terminal block
- Conector para alimentación tipo Jack

Con el esquemático se procedió a armar el circuito utilizando un protoboard. Se realizaron las conexiones entre componentes de acuerdo a lo señalado en el esquemático.

En la figura 4.51 se observa el circuito armado en el protoboard.

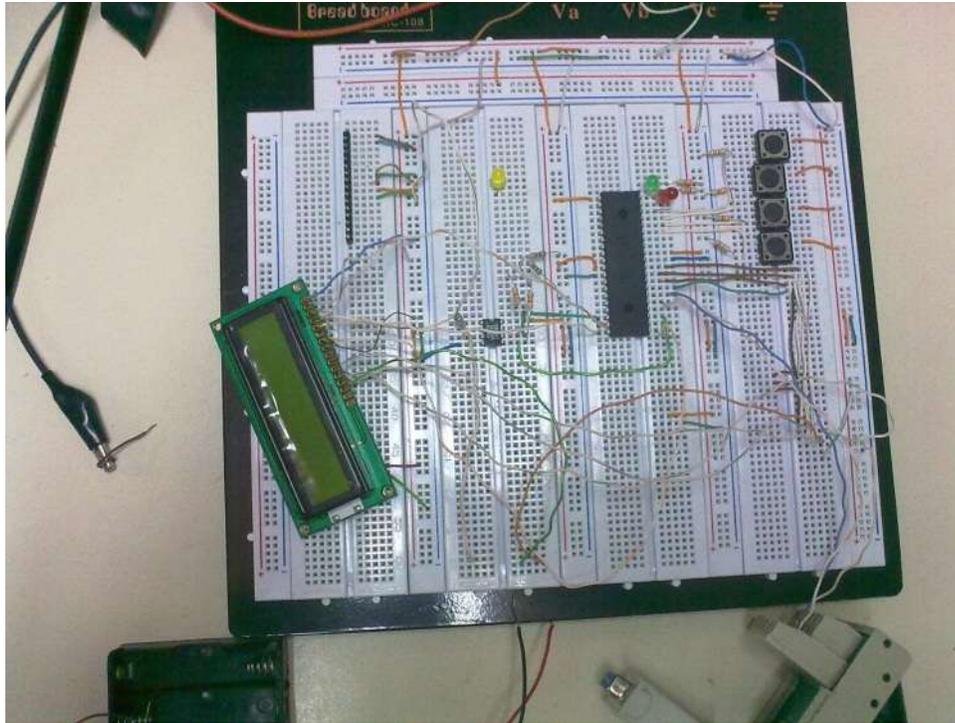


Figura 4.51: Protoboard

Se probaron todas las funciones del programa en el circuito del protoboard como se muestra en el punto 4.3.4. Todas las funciones se llevaron a cabo de manera exitosa, corroborando el buen funcionamiento del dispositivo.

Luego se diseñó la tarjeta a partir del circuito armado en el protoboard. Se colocaron todos los componentes y se conectaron. Se seleccionó el ancho de las pistas, el recorrido de las mismas y se señaló en el diseño.

La figura 4.52 muestra el diseño de la tarjeta del dispositivo.

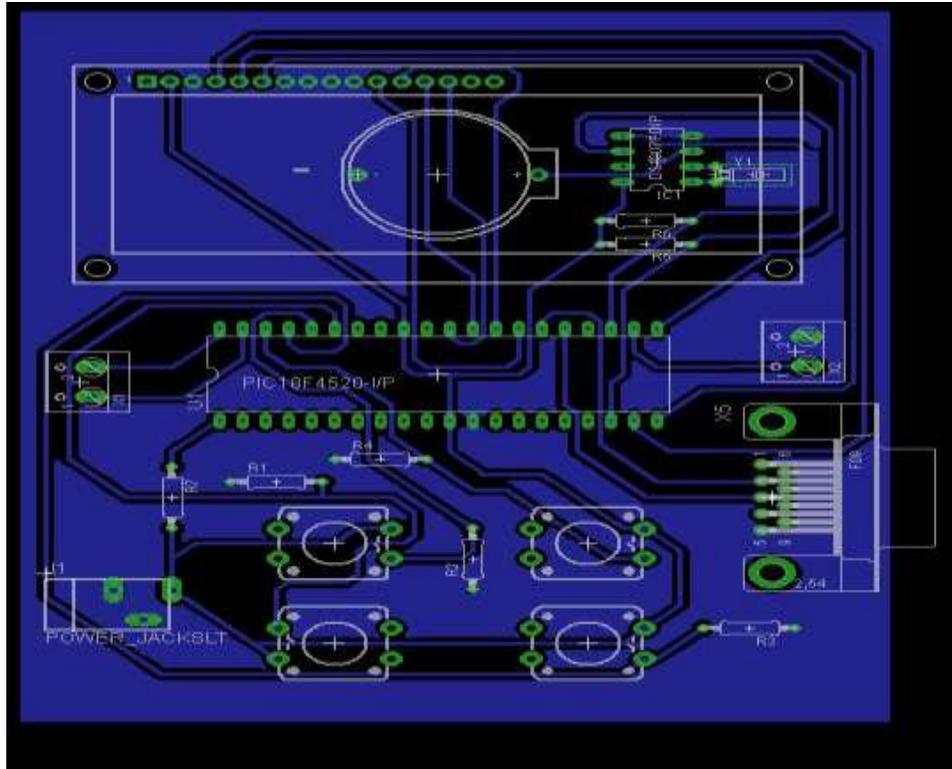


Figura 4.52: Tarjeta del prototipo

Este diseño fue impreso en acetato con una impresora laser. Este acetato se colocó sobre la placa “virgen” y se procedió a aplicar calor con una plancha. Esto permitió que las pistas se marcaran sobre la placa. Se realizaron las perforaciones correspondientes a los agujeros donde se colocarían los componentes y se repasó con un marcador de tinta indeleble el interior de los mismos.

Luego se procedió a corroer el cobre excedente sumergiendo la placa en ácido durante 20 min. Posteriormente se limpió la placa con acetona para remover cualquier resto de ácido y poder comenzar a colocar los componentes (ver figura 4.53).

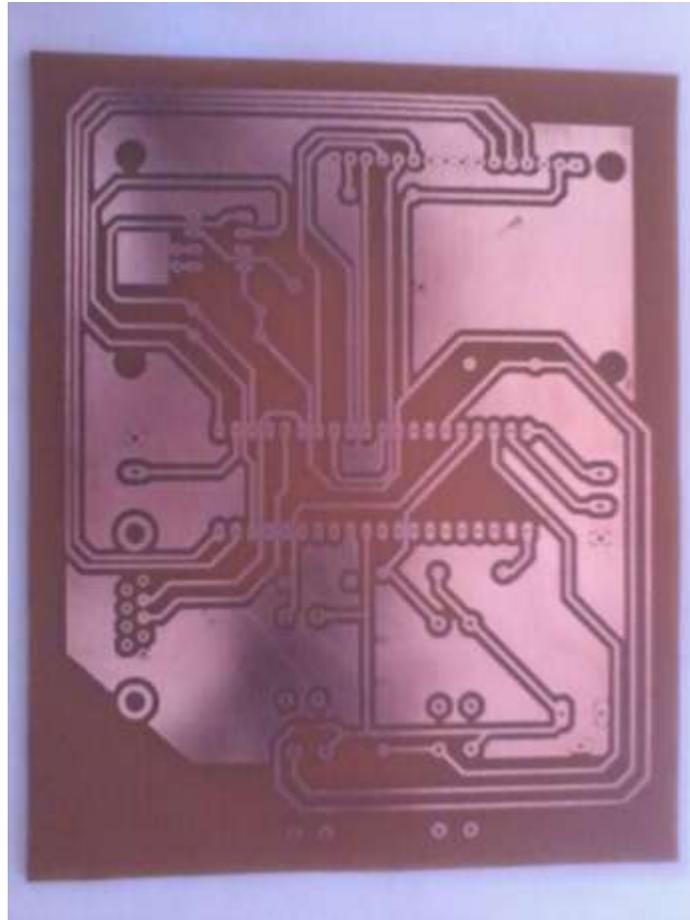


Figura 4.53: Placa sin soldar

Los componentes se colocaron en los espacios correspondientes, fueron soldados a la placa utilizando un cautín de 30 w y estaño como metal de unión. En la figura 4.54 se muestra la placa con los componentes soldados lo que finaliza la construcción del prototipo del dispositivo controlador-registrador para la automatización de proceso de impresión.

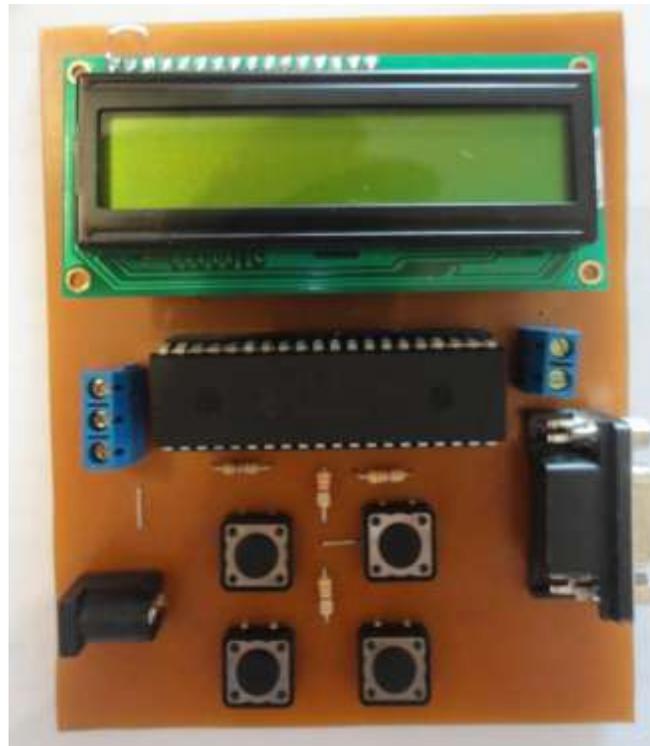


Figura 4.54: Prototipo terminado

Con el prototipo construido se procedió a probarlo para verificar su correcto funcionamiento.

Para la prueba realizaron las siguientes conexiones:

- Con una fuente de 5 Vdc-2A se alimentó el prototipo.
- Se conectó un generador variable de señales cuadradas al prototipo para simular el encoder que va sujeto al eje de la impresora.
- Se conectó al puerto serial del prototipo un adaptador RS232 a USB, con este se realizó la comunicación con la PC
- Se conectaron las salidas de control de impresión del dispositivo a dos LEDs que simulaban el arranque y parada de la impresora.



Primero se procedió a sincronizar la hora y fecha del prototipo con la de la PC. Para esto se realizaron los mismos pasos que en el punto 4.3.4 en el que el dispositivo actualiza su fecha y hora. En la figura 4.55 se muestra al dispositivo cuando termina de sincronizarse correctamente con la PC, en la segunda línea de la pantalla se observa minutos y luego horas (15:44 en formato militar o 3:44pm).



Figura 4.55: Sincronización con dispositivo

Luego se probó la impresión, para esto se encendió el generador y se seleccionó la opción “Imprimir” y se realizaron las mismas acciones que en el punto 4.3.4. Realizando de esta manera cuatro impresiones. En la figura 4.56 se muestra la pantalla del prototipo cuando se realiza la segunda impresión en la cual consta de 36 metros de papel.



Figura 4.56: Prototipo impresión

En cada una de las impresiones, el prototipo mostró la frecuencia a la que se encontraba el encoder (generador de señales) incluso cuando se variaba durante el proceso. También encendió la impresora (LED de arranque) cuando se comenzaba el proceso y la apagaba (LED de parada) cuando finalizaba el proceso.

Otro aspecto probado en el dispositivo fue la transferencia de datos a la PC. Luego de las impresiones se transfirieron los datos a la PC según el proceso mostrado en el punto 4.3.4 en el cual por medio de la interfaz PIC-PC se recibían los datos del dispositivo. Se guardaron estos datos y se visualizaron en la interfaz PIC-PC como lo muestra la figura 4.57.

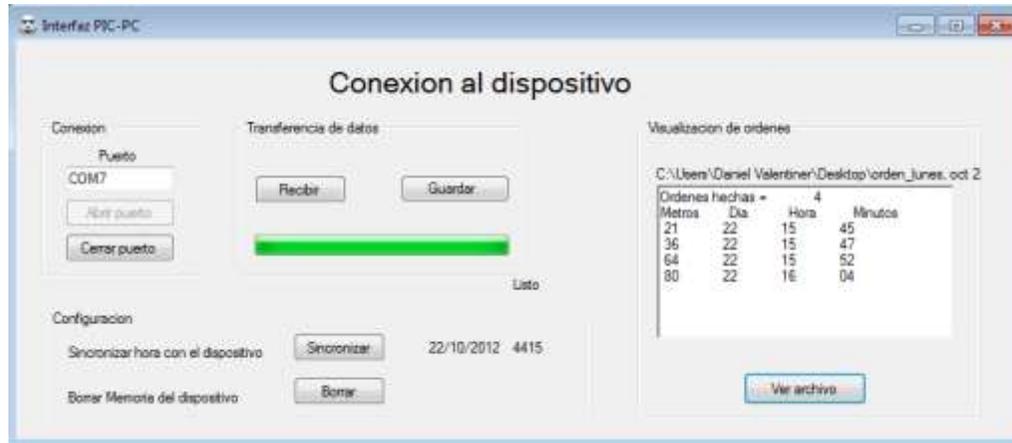


Figura 4.57: Datos extraídos del prototipo.

Los datos de impresión coinciden con lo que se hizo, por lo que se probó el correcto funcionamiento de la transferencia de datos y del proceso de impresión.

Por ultimo se probó el borrado de memoria siguiendo los mismos pasos que en el punto 4.3.4 para el circuito simulado. En este se selecciona la opción de borrar memoria en el dispositivo y en la interfaz PIC-PC. En la figura 4.58 se muestra el mensaje mostrado por la interfaz al terminar el proceso de borrado de memoria.

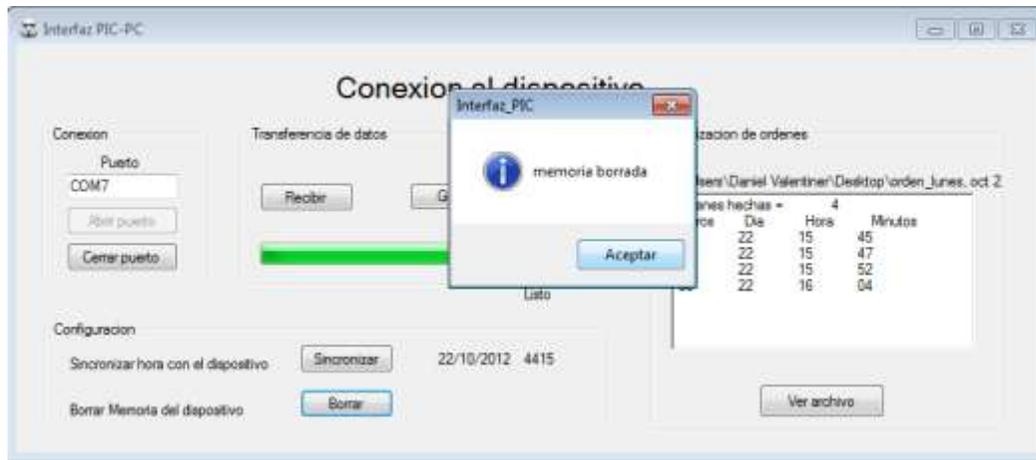


Figura 4.58: Borrado de memoria del prototipo

Con esta prueba se corroboró el correcto funcionamiento del prototipo del dispositivo controlador-registrador para la automatización del proceso de impresión.



4.4 Desarrollo de un sistema de control automatizado para el proceso de embalaje, mediante un Robot Industrial.

4.4.1 Selección de un robot adecuado para la automatización del proceso de embalaje.

Para la selección de un robot que se adecue a la automatización del proceso de embalaje se identificaron las características principales que el robot debe poseer, estas se describen a continuación:

- El robot debe estar diseñado para soportar un peso de 10 kg, este es el peso máximo que puede tener un paquete a embalar.
- Debe poseer al menos 4 grados de libertad
- El diseño del robot tiene que ser robusto, para soportar un entorno de planta.
- El robot debe ser diseñado para operaciones de pick and place.

Considerando las características mencionadas, el robot seleccionado fue “Mitsubishi RV-12SD” (ver figura 4.59), el cual cumple con todas las características necesarias.

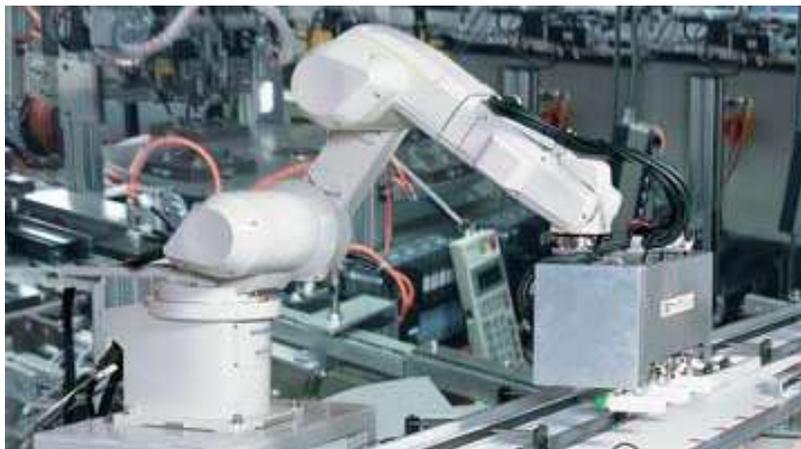


Figura 4.59: RV-12SD

Fuente: FESTO



Este es un robot es fabricado por la compañía Mitsubishi, posee 6 grados de libertad y capacidad nominal de hasta 12 kg. Está diseñado para operaciones de manejo de piezas, ensamble y pick and place. En la tabla 4.2 se muestran las especificaciones técnicas.

Tabla 4.2: Especificaciones técnicas de Robot RV-12Sd

Modelo	RV-12SD
Número de ejes	6
Montaje	En suelo, pared o techo
Velocidad [mm/s]	9600
Fuerza de elevación [kg]	12
Repetibilidad [mm]	0,05
Peso [kg]	93
Alcance sin pinza [mm]	1085

Fuente: FESTO

El robot posee un módulo de entradas y salidas digitales que permite la conexión con otros elementos de control que manejen señales digitales, como sensores, válvulas neumáticas, entre otras. A demás cuenta con tuberías neumáticas internas para conexión de fuentes neumáticas que alimenten a la herramienta de agarre que se le instale. De manera similar posee cableado eléctrico interno para el manejo de herramientas de agarre accionadas por electricidad. Existen tomas neumáticas y eléctricas que viajan desde la base del robot hasta el extremo donde se encuentra la base para instalar la herramienta de agarre.

Según lo expuesto anteriormente este robot posee todas las características necesarias para considerarse adecuado para la automatización del proceso de embalaje. Pero además de



contar con los elementos necesarios, la línea a la que pertenece este robot es distribuida por la empresa FESTO, la cual es el consultor tecnológico de la empresa DAFOINCA.

4.4.2 Diseño de una herramienta de agarre del robot que se adecue a las características de los paquetes.

Las herramientas de agarre del robot se diseñan de acuerdo a la tarea que desempeñen, en el caso de esta investigación se quiere levantar, transportar y posicionar paquetes de papel de 10kg de peso. Estos paquetes tiene forma de paralelepípedo, constan de hojas de papel impreso de 9 ½” x 11” o de 9 ½” x 5 ½”, puestas una encima de otra. La máquina empacadora coloca una tapa rectangular de cartón y sujeta con cinta engomada uniendo las hojas con el cartón formando un paquete solido (ver figura 4.60).



Figura 4.60: Paquetes del formato 9 ½” x 11”



Las dimensiones físicas de los paquetes se encuentran en la tabla 4.3

Tabla 4.3: Dimensiones físicas de paquetes.

Formato	9 ½" x 11"	9 ½" x 5 ½"
Peso [kg]	10	5
Largo [mm]	279,4	139,7
Ancho [mm]	241,3	241,3
Alto [mm]	100	100

Fuente: DAFOINCA

Para sujetar los paquetes se pensó utilizar una ventosa, ya que la tapa de cartón en la parte externa proporciona una superficie dura y lisa ideal para una ventosa, además la forma de sujeción de las tapas con la cinta engomada hace que el paquete permanezca unido aun cuando éste es sujetado solo por la tapa de cartón. De esta manera el robot solo tiene que posicionar el paquete, accionar la ventosa, levantar el paquete, transportarlo, posicionarlo y desactivar la ventosa sin necesidad de algún otro accesorio mecánico.

Una ventosa requiere también de un generador de vacío lo suficientemente potente para soportar el peso del paquete a levantar, y de otros elementos para que pueda funcionar. Este análisis de piezas requeridas para la herramienta de agarre se realizó con un software especializado proporcionado por la empresa FESTO.

El software es "FESTO – Selección de vacío". A través del se puede diseñar una herramienta de vacío, conociendo las dimensiones y el peso de lo que se quiere levantar. Además se contemplan otras preferencias de usuario como el material de la ventosa, la forma, la cantidad de ventosas, incluso el color.



A continuación se mostrará en detalle el proceso llevado a cabo para el diseño de esta herramienta:

Primero se ejecuta el programa antes “Selección de vacío”. Al ejecutarlo se mostró una pantalla de inicio, se presionó el botón “página siguiente” (representado por una flecha a la derecha) y se mostró una pantalla en la que se introdujeron los datos de los paquetes (ver figura 4.61). Debido a que la ventosa debe poseer suficiente succión para levantar el paquete más pesado, se introdujeron las dimensiones del paquete de 10kg.

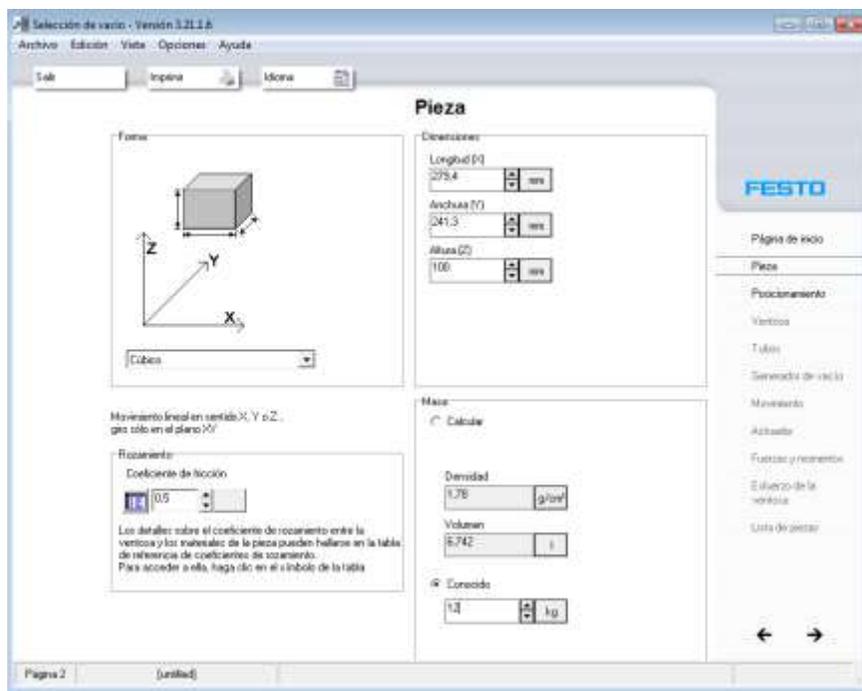


Figura 4.61: Programa de vacío paquete



Al avanzar a la siguiente página se decide la distribución, el número de las ventosas y los límites con el borde de la superficie de contacto. La distribución debe ser tal que con la misma herramienta se puedan sujetar los dos tipos de paquetes. Para esto se selecciono una distribución con dos ventosas dispuestas linealmente, las cuales no pueden ocupar un área mayor a la del paquete pequeño, es decir 240 mm x 140 mm. Esto se introduce en el programa como se muestra en la figura 4.62

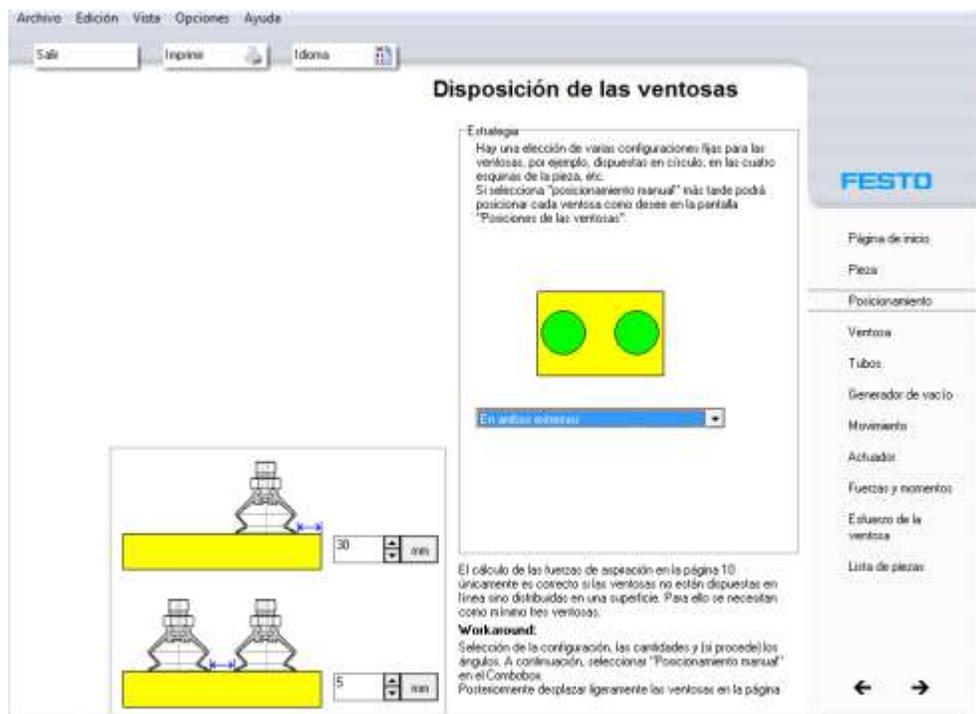


Figura 4.62: Programa de vacío distribución

Posteriormente se procedió a seleccionar la ventosa, para esto se escogen una serie de características como diámetro, material y tipo. Para cubrir el área necesaria se seleccionaron dos ventosas de 100 mm, las cuales (según figura 4.62) están separadas



entre si 5 mm y por lo que el área que cubren es de 205 mm x 100 mm, lo cual es menor a lo establecido de 240 mm x 140 mm. Se seleccionaron ventosas tipo estándar y las demás especificaciones se marcaron “sin preferencia” (ver Figura 4.63).

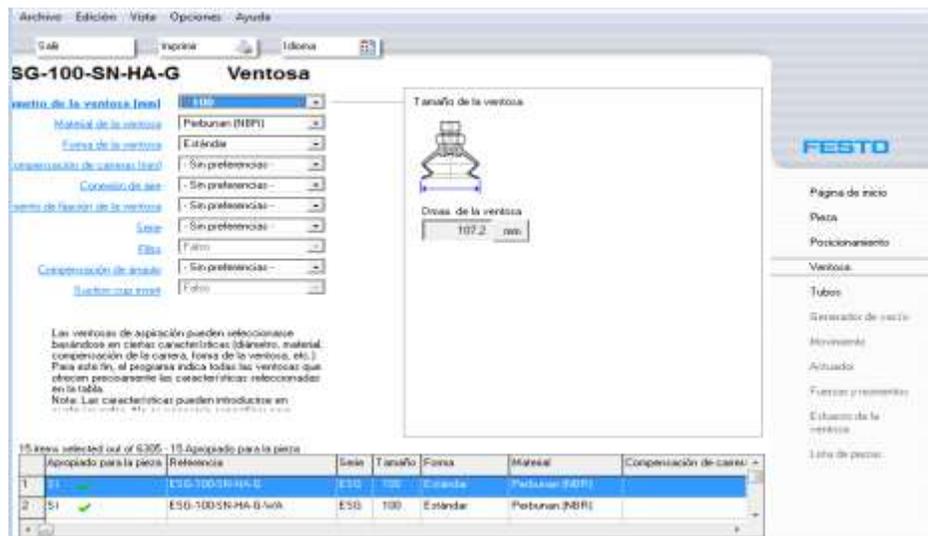


Figura 4.63: Programa de vacío ventosa

Una vez elegida la ventosa se procedió a seleccionar los tubos neumáticos a utilizar. Se seleccionaron tubos de 6 mm (ver figura 4.64).



Figura 4.64: Programa de vacío tubería



A continuación se seleccionó el generador de vacío. Se escogieron los valores predeterminados mostrados por el programa para el generador de vacío, debido a que este software recomienda la mejor opción posible según la información proporcionada en los pasos anteriores (ver figura 4.65).

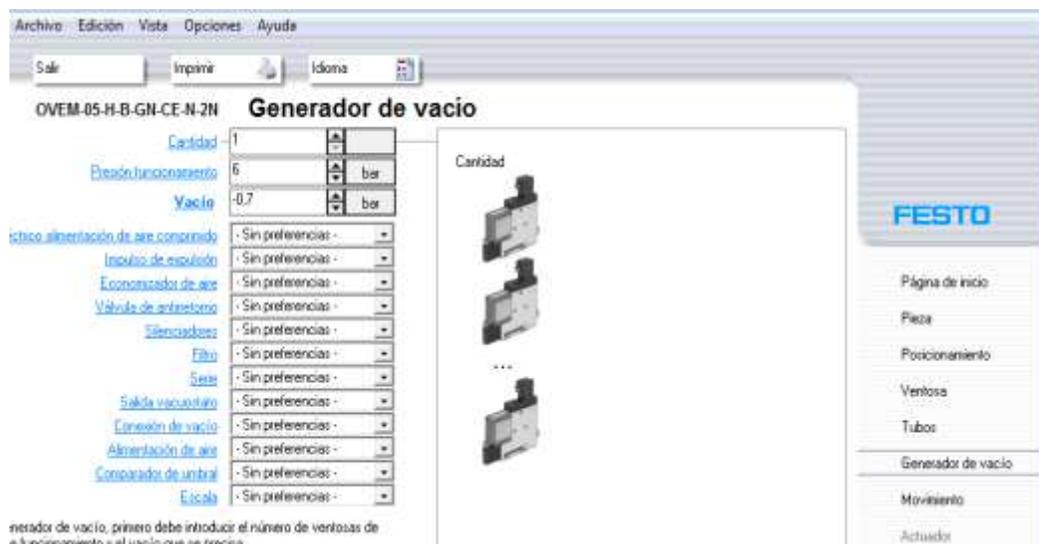


Figura 4.65 Programa de vacío generador

Luego se determinó la velocidad a la que se desplazara el paquete y la dirección. Este paquete se moverá en varias direcciones pero la única de estas direcciones en la cual el movimiento alcanzara una velocidad considerable; es en el desplazamiento de un lado del área de trabajo al otro mediante la rotación de una de las articulaciones del robot. El programa simula este movimiento angular, y solo se necesita proporcionar una longitud de brazo la cual se conoce del manual del robot. Estos datos proporcionados al programa se pueden observar en la figura 4.66



Capítulo IV

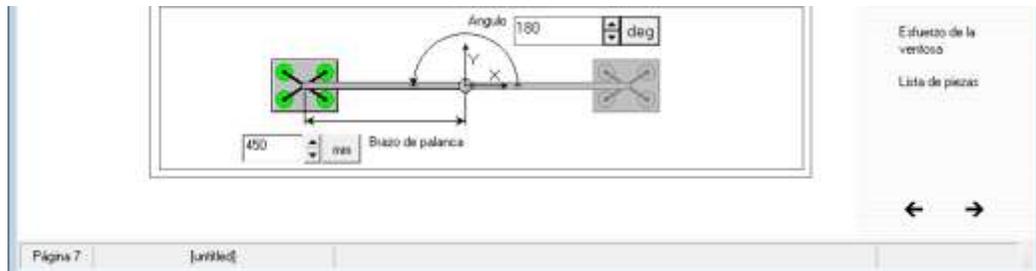


Figura 4.66: Programa de vacío brazo

Para comprobar el correcto funcionamiento de la herramienta, el programa realiza una simulación del comportamiento en operación. Simula el movimiento del paquete con la herramienta diseñada, a una velocidad de rotación promedio y calcula las fuerzas que soportan el paquete y la ventosa.

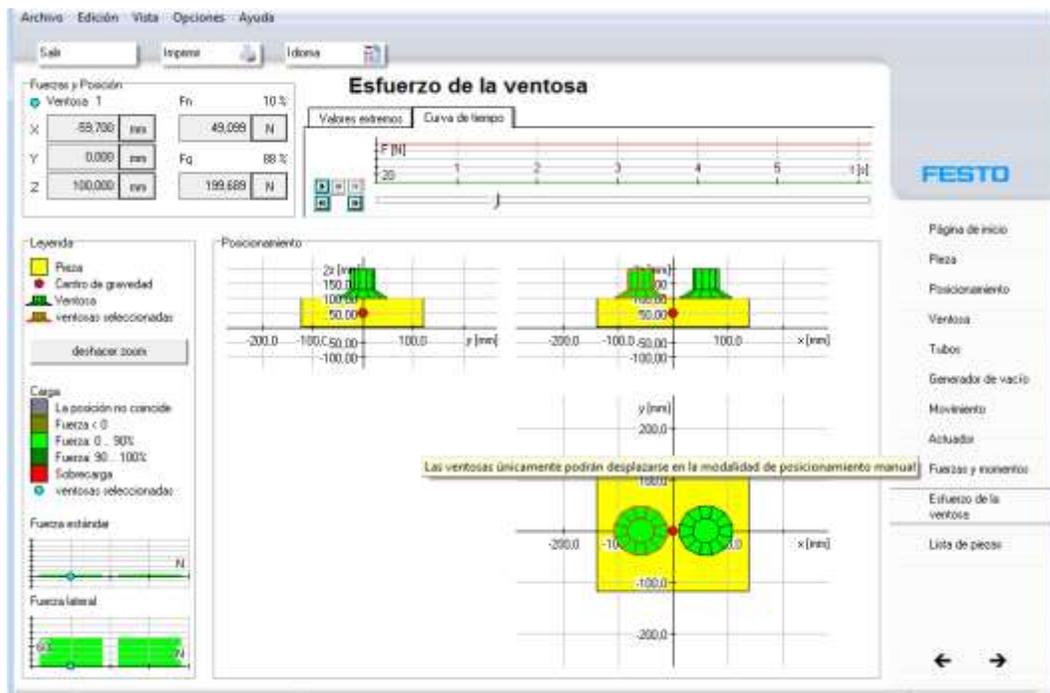


Figura 4.67: Programa de vacío simulación



En la figura 4.67 se muestra en tres vistas distintas el paquete y al herramienta sujetándolo. Para iniciar la simulación se pulso el botón “simular” representado por un triangulo ubicado en la parte superior central de la pantalla debajo de un. Al hacer esto se simula el movimiento completos y se grafican las fuerzas soportadas en cada una de las ventosas. En la parte inferior izquierda de la pantalla hay dos gráficos Fuerza [N] vs tiempo [s] (ver figura 4.67). En el grafico inferior se registran las fuerzas soportadas por las ventosas durante la simulación, durante la simulación se observo que la gráfica se mantenía de un color verde, esto significa que la fuerza soportada por la ventosa es menor al 90% de la fuerza aplicada por la ventosa, por lo que la herramienta diseñada es acertada para la aplicación.

El programa finaliza con una lista de piezas que conforman la herramienta, cada elemento posee un código que representa el modelo de la pieza. Las piezas obtenidas para la herramienta se muestran en la figura 4.68.

	Nº de artículo	Referencia	Descripción	Longitud	Cantidad	Packaging unit
1	3862	PL-6-BL	Tubos		50 m	50 m
2	189179	ESG-100-SN-HA-G	Ventosa		2	1
3	529074	OVEM-05-H-B-EN-CE-N-3N	Generador de vacío		1	1

Figura 4.68: Programa de vacío resultado

Para completar el diseño de la herramienta estas piezas deben ensamblarse en una pieza sólida que pueda unirse al robot. Para esto se realizó un diseño de la herramienta con las piezas seleccionadas. Se realizó un plano en 3 dimensiones de la herramienta utilizando modelos de las piezas. Estos modelos se hicieron de acuerdo a las especificaciones de



cada pieza, estas especificaciones se encuentran en las pantallas de selección en el software mostrado anteriormente. El modelo 3D de la herramienta fue hecho en el software de dibujo técnico AutoCAD (ver figura 4.69).

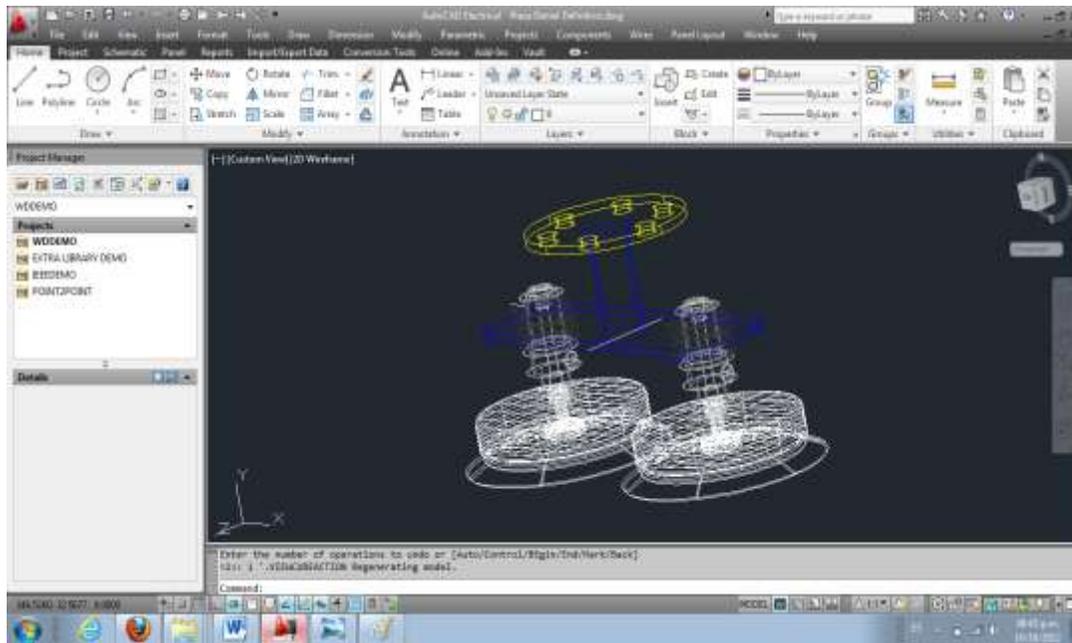


Figura 4.69: Vista AutoCAD

Este modelo fue realizado con las medidas reales de las piezas. Se incluyó una brida metálica para poder conectar la herramienta con el robot, también se incluyó un cilindro metálico y una pletina que se soldarían para crear una pieza solidada. Luego roscada a la pletina se encuentra el conector de la ventosa. Roscado a cada conector se encuentran las ventosas de 100 mm seleccionadas.

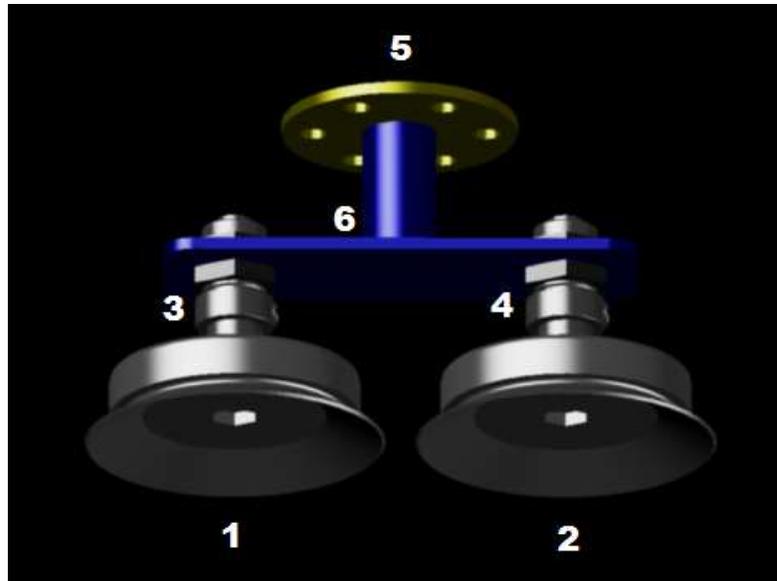


Figura 4.70: Herramienta isometría identificación de partes

En la figura 4.70 se tiene el modelo de la herramienta en una vista isométrica, en esta imagen se enumeran las piezas que componen la herramienta y se explican a continuación:

- Componentes #1 y #2: Ambas son ventosas de 100 mm de diámetro tipo estándar, están hechas de un material a base de goma llamado Perbunan (NBR).
- Componentes #3 y #4: Estos son conectores roscados para ventosas FESTO ESG-100.
- Componente #5: Esta es una brida metálica de 80 mm de diámetro que se ajusta a las medidas de la brida de conexión de herramienta del robot.
- Componente #6: Esta es una pieza soldada a la brida que consta de un cilindro metálico unido por soldadura a una platina con dos orificios para colocar los conectores de las ventosas.



La alimentación neumática de las ventosas proviene de tubos neumáticos que se sujetan a la parte superior de los conectores de las ventosas como se señala mediante flechas en al figura 4.71.

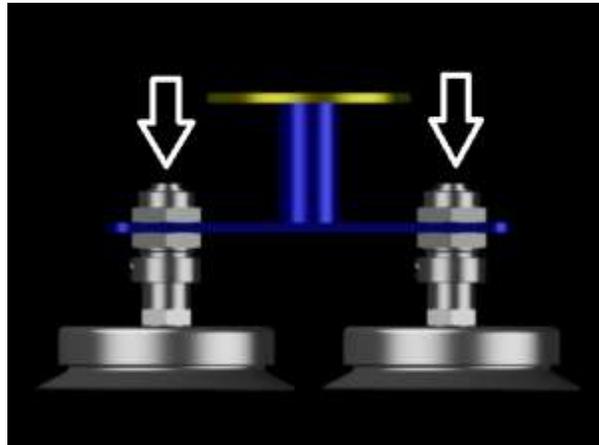


Figura 4.71: Herramienta ubicación tubos neumáticos

Estos tubos neumáticos se conectan a la toma neumática del brazo de robot. Esta toma se extiende por el interior del robot hasta llegar a su base donde se encuentra otra toma la cual se conecta, a través de un tubo neumático, con el generador de vacío seleccionado. El mapa de conexiones neumáticas internas del robot RV-12SD se muestra en la figura 4.72.

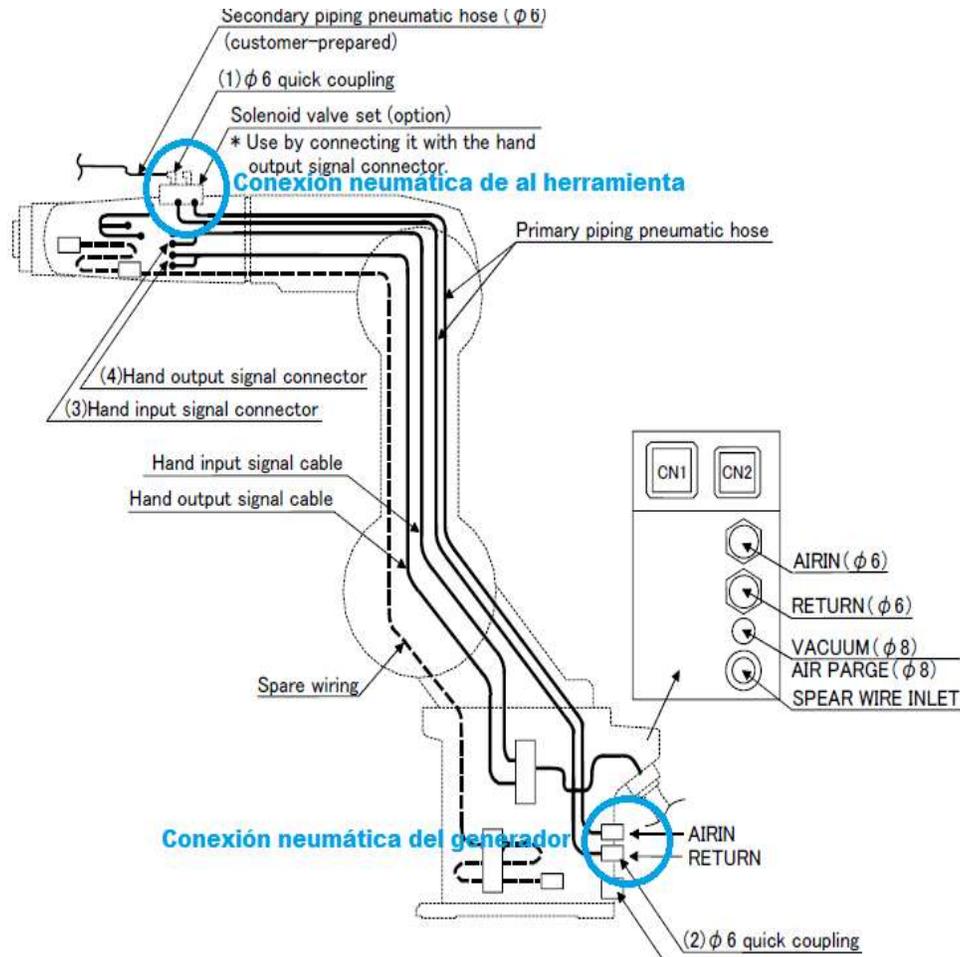


Figura 4.72. RV-12SD neumática interna

Fuente: FESTO

En la figura 4.73 se muestra el diseño final de la pieza en varias vistas. De esta manera se tiene toda la información necesaria para su elaboración.

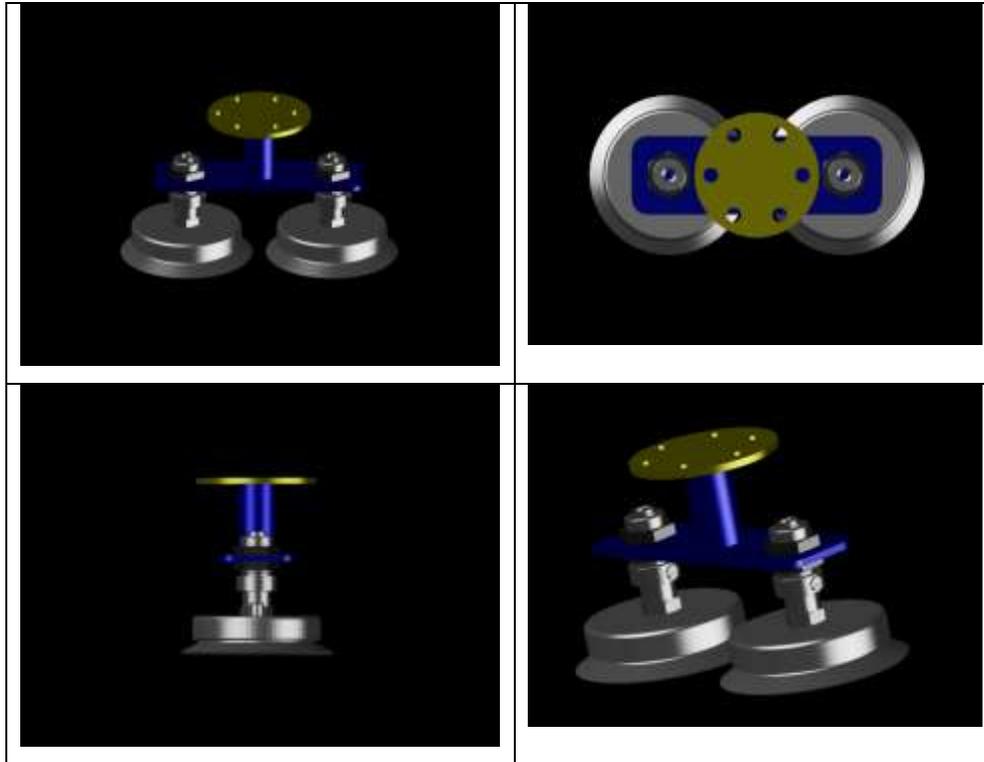


Figura 4.73: Vistas herramienta

Al poseer la selección de piezas y el modelo en tamaño real de la herramienta, se obtiene un diseño completo de la herramienta de agarre del robot. La herramienta diseñada se ajusta perfectamente a los parámetros necesarios para la automatización del proceso de embalaje con el Robot Industrial.

4.4.3 Desarrollo del programa para la ejecución automatizada del proceso de embalaje a utilizar en el robot.

Se utilizó la metodología de programación estructurada como se estipula en el Capítulo III para el desarrollo del programa del Robot Industrial. El programa consiste en tomar el paquete de la bandeja de la máquina empaquetadora y colocarlo dentro de una caja de



cartón, repetir el proceso de manera ordenada hasta llenarla. Luego se expulsa la caja y el proceso comienza de nuevo.

Para realizar el programa se utilizó el entorno “MELSOFT – RT ToolBox2”, este software es usado para todos los robots Mitsubishi. Utiliza los lenguajes de programación para robots Melfa Basic IV y Movemaster Command. En el caso de esta investigación se utilizó el lenguaje Melfa Basic IV.

Se necesitó el módulo de entradas y salidas del robot, las cuales se conectan con sensores y actuadores para desempeñar todas las tareas necesarias para la automatización. Se necesita conocer si hay paquetes en la bandeja, si hay cajas en posición para embalar, y se necesita expulsar de alguna manera las cajas embaladas para que otra pueda tomar su lugar y continuar con el proceso.

En la tabla 4.4 se muestran las variables de entrada y salida usadas en este programa y que representan.

Tabla 4.4: Variables de entrada y salida del Robot

Variable	Tipo	Descripción
M_IN(11)	Entrada	Sensor de proximidad, detecta si hay o no una caja de cartón
M_IN(12)	Entrada	Pulsador de parada n/c (Stop)
M_IN(13)	Entrada	Pulsador de arranque n/a (Start)
M_IN(14)	Entrada	Sensor de proximidad, detecta si hay o no un paquete
M_OUT(9)	Salida	Activa un cilindro neumático que desplaza la caja embalada



En el Apéndice D, se muestra a través de un diagrama de flujo el funcionamiento del programa.

Por razones de seguridad se incluyó una subrutina en el programa la cual se encarga de posicionar el robot de manera segura en caso de una parada de emergencia a través del pulsador “Stop”; esta subrutina se comporta como una interrupción al programa principal. Si se pulsa Stop el programa se detiene, si luego se pulsa Start el programa continúa desde donde dejó de ejecutarse, si en vez de Start se pulsa Stop, el robot se posiciona de manera segura y en caso de tener sujetado un paquete, este es depositado en un área de seguridad.

En el Apéndice E, existe un diagrama de flujo que explica la lógica de la subrutina de interrupción de seguridad.

Utilizando la misma lógica de programación que la presentada en los diagramas de flujo, la cual se basa en la metodología de programación estructurada indicada en el Capítulo III, se desarrollaron 3 programas que ejecuta el robot en el proceso de embalaje. Uno de los programas es para el embalaje de los paquetes de formato $9 \frac{1}{2}' \times 11'$, el código de este programa se encuentra en el Apéndice F. El segundo programa es para el embalaje de los paquetes de formato $9 \frac{1}{2}' \times 5 \frac{1}{2}'$ y su código se puede ver en Apéndice G. El último programa es un subprograma que llama cualquiera de los otros 2 programas anteriores para mover el robot a la posición segura. El código del subprograma se encuentra en el Apéndice H.



4.4.4 Prueba de funcionamiento mediante la utilización de un Robot Industrial a escala fuera del proceso.

Para la realización de esta prueba se utilizó un Robot Industrial Mitsubishi RV-2AJ (ver figura 4.74) de la misma línea que el robot RV-12SD, seleccionado en el punto 4.4.1. Al ser de la misma línea, el RV-2AJ es ideal para probar los programas y el desempeño del robot fuera del proceso ya que este modelo de robot es igual al RV-12SD a una escala menor. Este robot fue proporcionado por la empresa FESTO.

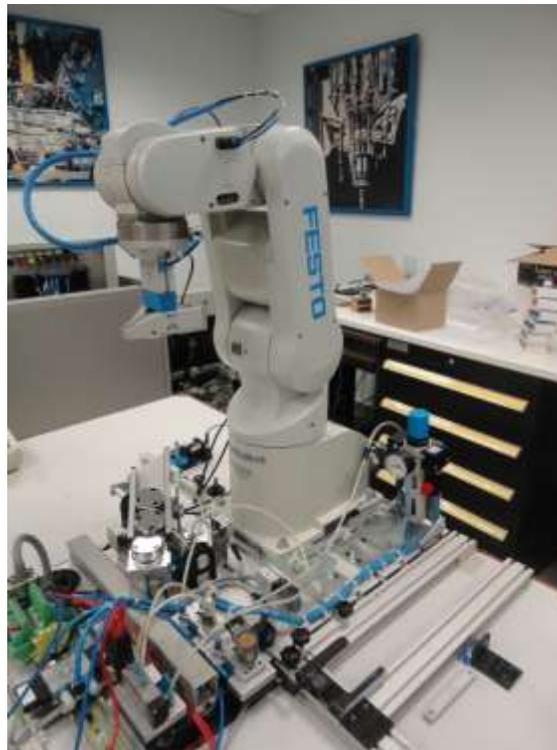


Figura 4.74: RV-2AJ

El RV-2AJ es un robot de 5 grados de libertad, soporta una carga de 3 kg y está diseñado para operaciones de pick and place. Está construido para realizar el mismo tipo de



movimientos que el RV-12SD y puede realizar las mismas funciones. La diferencia principal radica en el peso que puede elevar. En la tabla 4.5 se muestran sus especificaciones técnicas comparadas con las del robot RV-12SD

Tabla 4.5: Datos técnicos del robot RV-2AJ

Modelo	RV-12SD	RV-2AJ
Numero de ejes	6	5
Montaje	En suelo, pared o techo	En suelo, pared o techo
Velocidad [mm/s]	9600	4400
Fuerza de elevación [kg]	12	3
Repetibilidad [mm]	0,05	0,02
Peso [kg]	93	17
Alcance sin pinza [mm]	1085	504

La automatización del proceso de embalaje contempla sensores y actuadores que se comunican con el robot. La función de cada uno de estos sensores y su relación con el programa del robot se encuentra especificada en el punto 4.4.3. A continuación se mostrarán los sensores y actuadores utilizados en el robot para la prueba.

Como sensor de paquetes se utilizó un sensor óptico de barrera tipo SOOF (ver figura 4.75). Este envía una señal lógica al robot. Si el sensor detecta a un paquete este activa una señal (1 lógico), si no detecta el paquete desactiva la señal (0 lógico)



Figura 4.75: Sensor de paquetes

Para el sensor de cajas se utilizó el sensor óptico SOEG-RT-M18 (ver figura 4.76). Este sensor envía una señal lógica al robot. Si el sensor detecta una caja, este activa una señal (1 lógico), si no detecta la caja desactiva la señal (0 lógico)



Figura 4.76: Sensor de cajas



En el caso del actuador se utilizó un cilindro neumático doble efecto (ver figura 4.77). Este es el cilindro que empuja la caja cuando ya están embalados los paquetes.



Figura 4.77: Cilindro neumático

Para controlar al cilindro neumático se utilizó una válvula electro-neumática monoestable 5/2 vías retorno por muelle (ver figura 4.78).



Figura 4.78: Válvula electro-neumática

En el caso de los pulsadores de arranque “Start” y parada “Stop” se utilizó un cajetín de pulsadores en el cual se escogieron 2 pulsadores. Uno de ellos normalmente abierto para el Start y otro normalmente cerrado para el Stop. En la figura 4.79 se muestran los pulsadores, el Start es el segundo pulsador en orden descendente (conexión 13 y 14) y el Stop es el tercer pulsador en orden descendente (conexión 31-32).

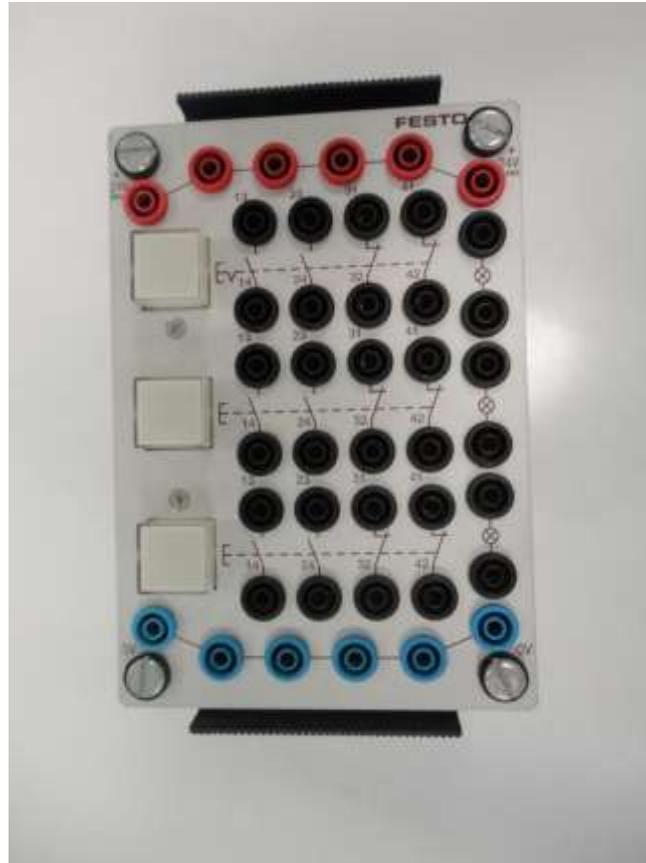


Figura 4.79: Panel de pulsadores

Para que el vástago del cilindro neumático retrocediera con rapidez para permitir el ingreso de otra caja se utilizó una válvula de escape rápido. (Ver figura 4.80)



Figura 4.80: Válvula de escape rápido

También fue necesario construir los paquetes a escala para la prueba. Estos paquetes de prueba se hicieron con pequeñas cajas de cartón y se pintaron de colores alusivos al paquete real. El robot utilizado para la prueba posee una herramienta de agarre en forma de pinza diseñada para sostener piezas que forman parte de módulos didácticos de automatización utilizados por la empresa FESTO. En la figura 4.81 se muestra la herramienta de agarre del robot en cuatro vistas. Se observa la forma circular de agarre ideada para sostener las piezas del módulo didáctico. En la figura 4.82 se muestran estas piezas de agarre del módulo didáctico.

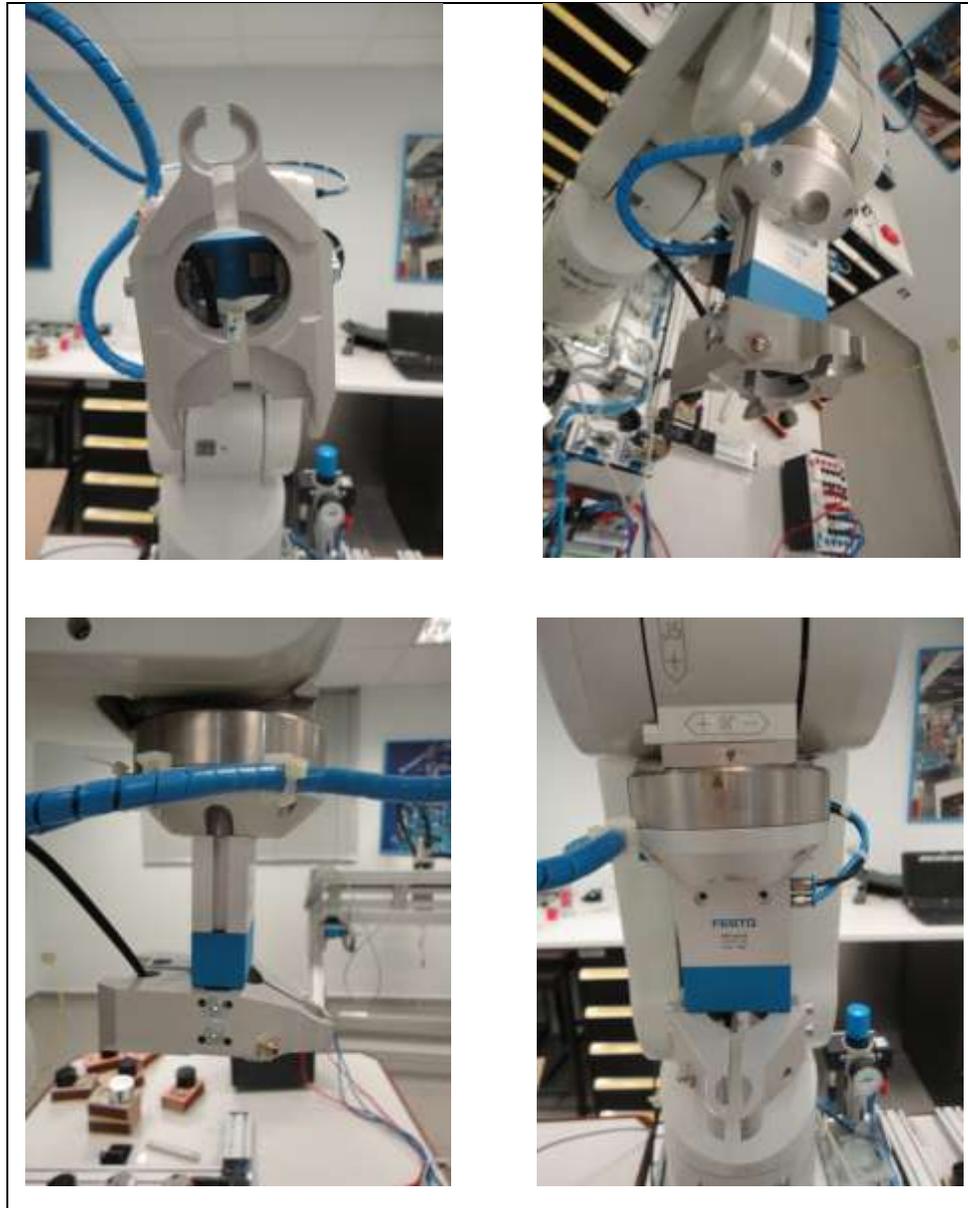


Figura 4.81 Herramienta RV-2AJ



Figura 4.82: Piezas de agarre

Esta herramienta no puede ser sustituida debido a políticas de la empresa, por lo cual en vez de remplazar la herramienta se adaptaron los paquetes de prueba para que puedan ser sujetados por la herramienta. Para lograrlo se unieron los paquetes de prueba con las piezas del módulo didáctico. De manera que la herramienta al sujetar la pieza también sujete el paquete de prueba. En la figura 4.83 se observa un paquete de prueba terminado, se puede notar que en la parte superior se encuentra una pieza del módulo didáctico, la cual está firmemente sujeta al paquete de prueba.

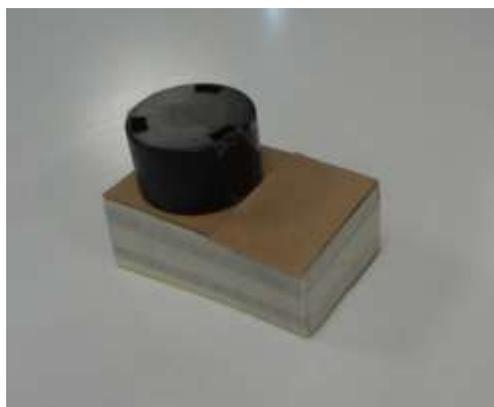


Figura 4.83: Paquete de prueba



Para se pueda colocar un paquete encima de otro de manera que ambos paquetes calcen, se hizo una hendidura en la parte inferior de cada paquete de prueba para que se pudiera introducir la pieza de agarre sujeta al paquete de prueba que se encuentre debajo. La figura 4.84 muestra estos paquetes de prueba uno encima de otro asemejando a dos bloques de construcción.



Figura 4.84 Dos paquetes de prueba

Otro aspecto que se necesitó para la prueba del robot fue la caja en la cual los paquetes son embalados. Debido a que toda la prueba fue hecha a escala del proceso real, también tuvo que construirse una caja tamaño escala. Esta tiene capacidad para 6 paquetes de formato $9 \frac{1}{2}' \times 11'$ (paquetes de prueba) y la misma tiene capacidad para 12 paquetes de formato $9 \frac{1}{2}' \times 5 \frac{1}{2}'$ (paquetes de prueba). En la figura 4.85 se muestra la caja.



Figura 4.85: Caja de prueba

Para representar la bandeja de trabajo en la cual se encuentran los paquetes que salen de la máquina empaquetadora, se utilizaron barras metálicas como guías para los paquetes de prueba. Los paquetes se posicionan dentro de estas barras y de allí el robot sujeta los paquetes. En la figura 4.86 se muestran las guías metálicas a la izquierda, a la derecha se encuentran en las guías con los paquetes de prueba.

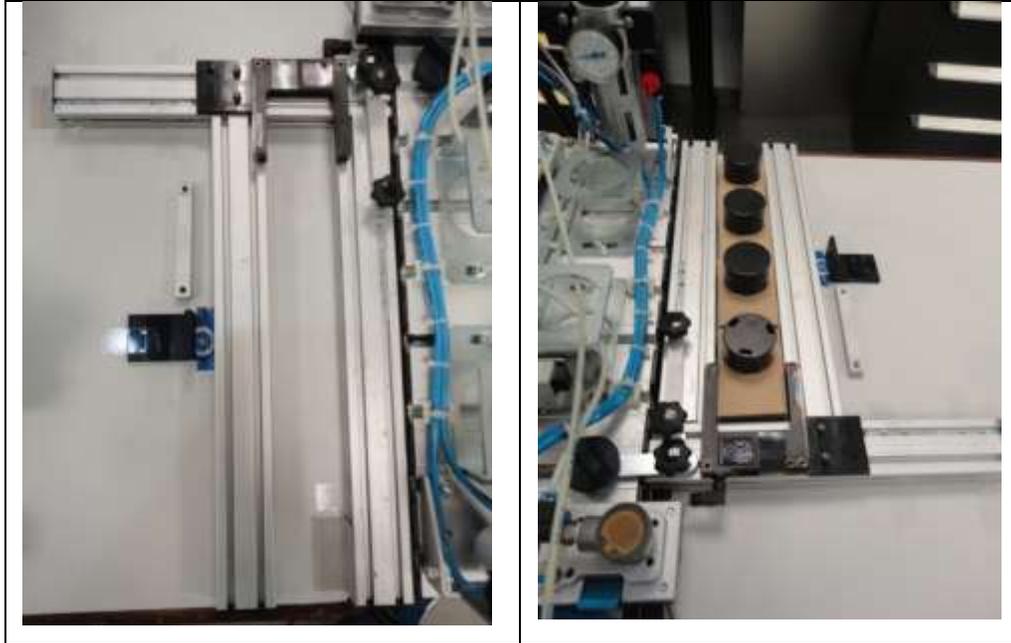


Figura 4.86: Guías metálicas

Para comenzar la prueba se transfirieron los programas desarrollados al robot RV-2AJ, mediante el software de programación “MELSOFT – RT ToolBox2”. Se posicionaron en las guías metálicas los paquetes de prueba, alineados uno detrás del otro. Se ubicó la caja al lado contrario de donde se encontraban los paquetes de prueba que es donde está ubicado el actuador neumático que expulsa la caja cuando la misma está embalada (ver figura 4.87).

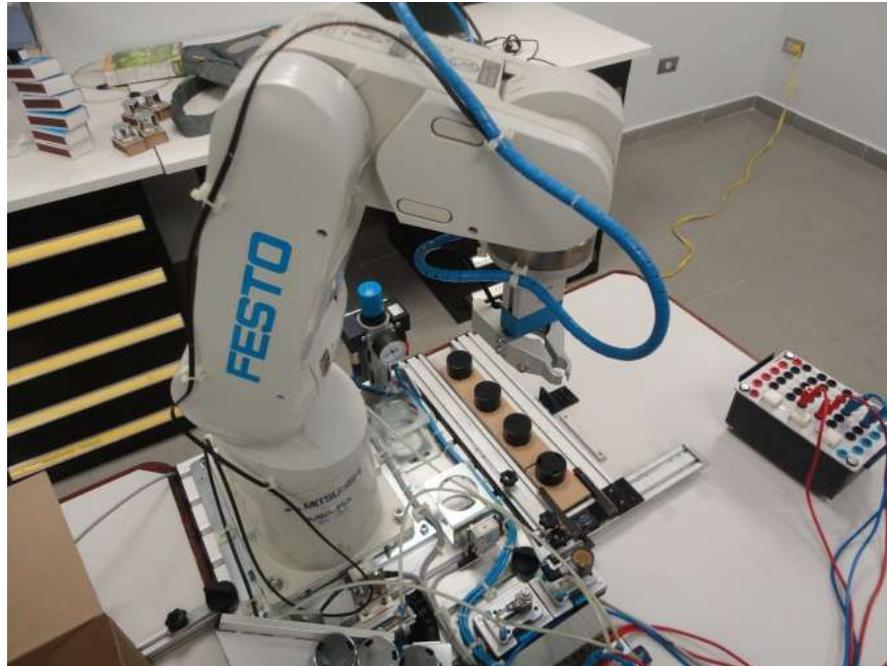


Figura 4.87: Prueba inicial

Se comprobó el funcionamiento de los sensores observando en el módulo de entradas y salidas del robot. Este módulo posee un indicador por cada entrada y salida, este indica si la señal correspondiente a esa salida o entrada esta activa o no. Se interactuó con los sensores y se observó que los indicadores cambiaban su estado conforme a la interacción con los sensores. (Ver figura 4.88)



Figura 4.88: Módulo de entradas y salidas



Se seleccionó el programa a probar y se comenzó la ejecución del mismo. Para esto se presionó el pulsador de Start. El robot se dirigió al paquete ubicado en la zona de agarre de la guía de metal donde se encuentra el sensor de paquete, lo sujetó y lo comenzó a transportar al otro lado del área de trabajo. Esto se muestra en la figura 4.89

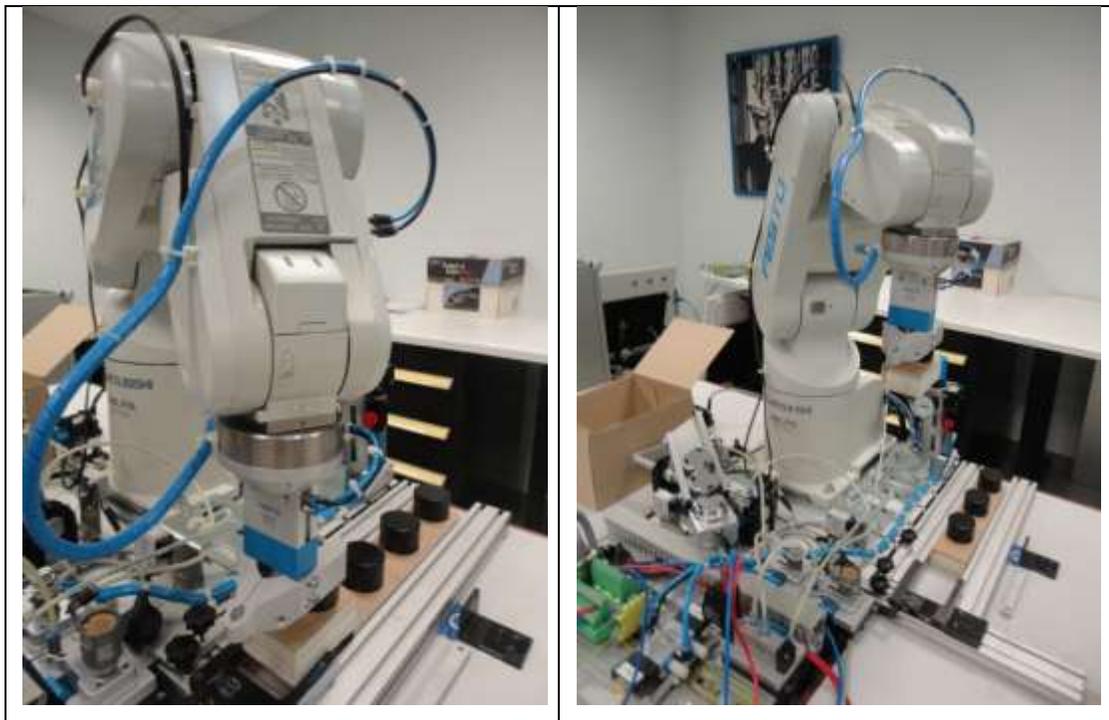


Figura 4.89: Prueba de sujeción paquete

Luego se dirigió a la caja y depositó el paquete dentro de ésta en el primer punto de depósito de paquete (ver figura 4.90). Después de depositarlo se dirigió al área de los paquetes a tomar el siguiente y repitió el proceso hasta llenar la caja con los 12 paquetes de capacidad que tiene la misma.

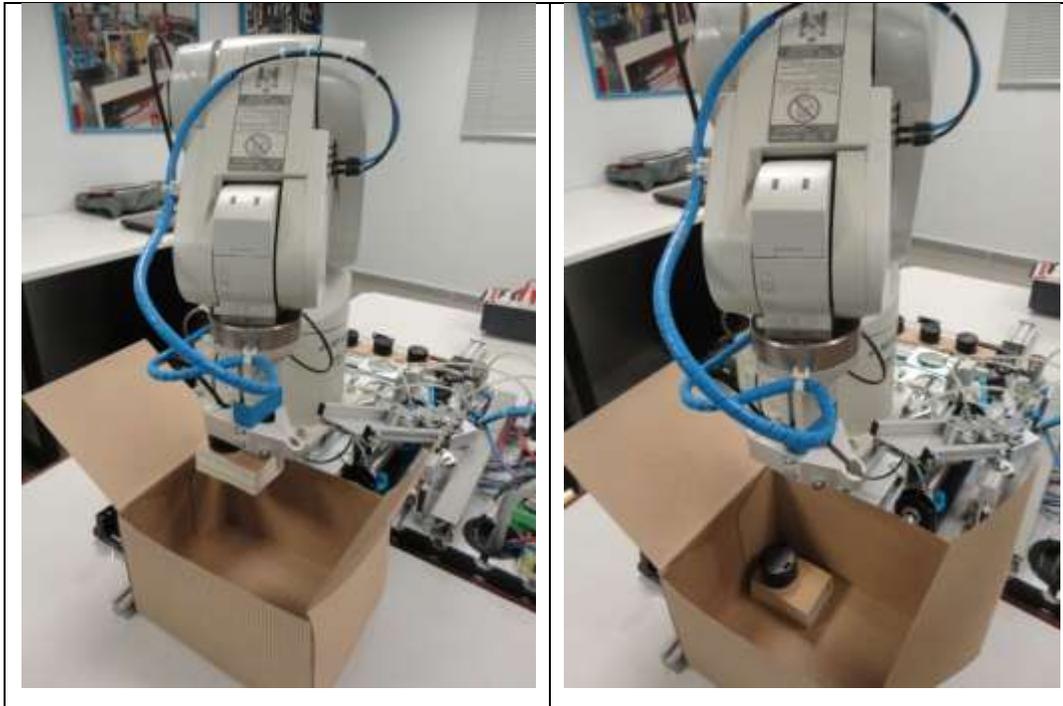


Figura 4.90: Prueba soltar paquete

Luego de llenar la caja el robot activó el cilindro neumático que expulsó la caja del área de embalaje (ver figura 4.91).



Figura 4.91: Prueba caja llena

El proceso se llevó a cabo de manera fluida y sin errores lo que comprobó el correcto funcionamiento del programa para el proceso. Para probar el resto de las situaciones que manejan los programas se llevaron a cabo las siguientes acciones:

- Se quitaron los paquetes del área de agarre de paquetes, al hacer esto el robot se mantuvo en posición de espera hasta que se suministró un paquete al área de agarre, esto prueba el correcto funcionamiento de la lógica del programa en el



evento de que se suspenda temporal o permanentemente la alimentación de paquetes.

- La caja de fue removida durante la ejecución del proceso. El robot sujetó un paquete correspondiente al área de agarre, y esperó a que la caja fuera colocada, lo que prueba el correcto funcionamiento de la lógica del programa en el evento de que se extraiga del proceso la caja a embalar.

Para probar la subrutina de interrupción por Stop se realizaron las siguientes acciones:

- Se presionó el pulsador Stop. Al hacer esto el robot detuvo totalmente su movimiento y no realizó ninguna acción. Se presionó el pulsador Start y el robot continuó inmediatamente su funcionamiento normal.
- Se presionó el pulsador Stop mientras el robot no sujetaba ningún paquete. Se presionó Stop nuevamente, entonces el robot expulsó la caja y se dirigió a la posición segura especificada en el programa.
- Se presionó el pulsador Stop mientras el robot sujetaba un paquete. Se presionó Stop nuevamente y el robot expulsó la caja. Luego se dirigió al área de depósito seguro de paquetes. Dejó allí el paquete que sostenía. Posteriormente se dirigió a la posición segura especificada en el programa.

El comportamiento del robot al realizar estas acciones fue el que se esperaba. Por lo que se demuestra que el sistema de automatización desarrollado para el proceso de embalaje funciona correctamente sin ningún error.



Capítulo V

Conclusiones y recomendaciones

5.1 Conclusiones.

Luego de realizada la investigación y obtenidos los resultados se llegó a las siguientes conclusiones:

Se construyó el prototipo de un dispositivo controlador-registrador que permite la automatización de una impresora rotativa industrial Thimsom TRS2000. Este dispositivo consta de un circuito impreso conformado por un microcontrolador PIC18F4520, un puerto serial para comunicación con la PC y una interfaz formada por pulsadores y una pantalla LCD, que permiten la interacción directa con el operador de la impresora. Dicho dispositivo controla el arranque de la máquina, detecta la cantidad de papel impreso a través de un encoder y detiene la máquina cuando se llega a la cantidad de papel a imprimir preestablecida por el operador. También registra todas las impresiones realizadas con fecha de culminación y hora de culminación. La información de las impresiones puede obtenerse para su posterior procesamiento por parte de la empresa DAFOINCA.

Se desarrolló un programa que permitió el funcionamiento del dispositivo, a través de la programación del microcontrolador PIC18F4520 mediante el software Proton IDE. Este programa permitió al dispositivo controlar el arranque y parada de la máquina impresora, el registro de la información de las impresiones y la comunicación vía puerto serial con la PC.

La automatización de la máquina impresora rotativa mediante el dispositivo controlador – registrador permite una mayor eficiencia del proceso de impresión y un mejor control de



la impresora disminuyendo el desperdicio y eliminando la necesidad de monitoreo constante de parte del operador. También se puede llevar un registro preciso de los datos de las impresiones, proporcionando así indicadores de producción que puedan sustentar mejor los cálculos y beneficios a los trabajadores a los efectos de bonos de producción y otros incentivos otorgados por la empresa a sus trabajadores.

Para la extracción de la información del dispositivo controlador-registrador se desarrolló una aplicación de computadora mediante la herramienta de software Visual Studio 2010. Dicha aplicación consta de una Interfaz Gráfica de Usuario que permite comunicar el dispositivo controlador-registrador con la PC, extraer la información de las impresiones almacenadas por el dispositivo, guardar esta información en un archivo de texto en una ruta seleccionada por el usuario, borrar la memoria del dispositivo así como también sincronizar la hora y fecha del dispositivo con la de la PC.

Para la automatización del proceso de embalaje, se seleccionó el Robot Industrial Mitsubishi modelo RV-12SD, adecuado a las características del sistema de embalaje de la empresa DAFOINCA y se desarrolló un programa mediante el software MELSOFT – RT ToolBox2. Este programa permite al robot tomar los paquetes que provienen de la máquina empacadora y embalarlos en cajas de acuerdo a especificaciones. En este sentido se desarrolló la programación para dos formatos diferentes de impresión 9 ½” x 11” y 9 ½” x 5 ½”. Esta automatización mejora la productividad ya que elimina el problema del retraso en el proceso cuando los pedidos son grandes; realiza con mayor rapidez el proceso de lo que los hacen los operarios y evitaría lesiones a los trabajadores. Además la empresa podrá reubicar a estos trabajadores en fases de producción en la cual poseen un déficit de personal.

Se diseñó una herramienta de agarre para el robot que permite el manejo de los paquetes que el mismo embalará. Esta herramienta consta de dos ventosas instaladas en un marco



metálico que se une al brazo del robot. Las ventosas se conectan a un generador de vacío mediante conexiones neumáticas internas del brazo del robot lo que permite sujetar mediante succión cualquiera de los paquetes de formato estipulado en el programa.

Se realizó exitosamente una prueba del programa en funcionamiento con un Robot Industrial a escala, fuera del proceso. Para esta prueba se utilizó el robot Mitsubishi RV-2AJ, el cual pertenece a la misma línea que el RV-12SD. Se logró hacer funcionar el robot y que este llevara a cabo el proceso de embalaje con la ayuda de sensores y actuadores para interactuar con el entorno de trabajo. Se verificó el correcto funcionamiento del programa desarrollado así como la correcta ejecución de todas las tareas para las que fue programado el robot.

5.2. Recomendaciones

- Se sugiere a la empresa DAFOINCA la implementación del dispositivo controlador-registrador para la automatización del proceso de impresión. Por medio de ello se aumentará la eficiencia operacional del proceso, reduciendo los costos por concepto de impresión de papel en exceso. Se eliminaría la necesidad de la supervisión manual del proceso por parte del operador. También se deriva una ventaja administrativa ya que los datos de las impresiones que proporciona del dispositivo permiten un cálculo más eficiente de los bonos de productividad pagados por la empresa.
- Para la implementación del dispositivo controlador-registrador, será necesario la adquisición de cableado, para la conexión a la impresora. Para el registro, y estudio posterior de la data recolectada por el dispositivo, será necesario el uso de una computadora portátil en donde se correrá la aplicación desarrollada para la comunicación. Con la ejecución de la misma se podrá descargar el archivo tipo



texto con el registro de impresión, y permitirá almacenarlo en la memoria de la computadora.

- Para el proceso de embalaje se recomienda la adquisición de un Robot Industrial modelo Mitsubishi RV-12SD, el cual cumple con los requerimientos de dicho proceso. Este robot puede ser suministrado por la empresa FESTO que es su proveedora de servicios tecnológicos.
- Para la implementación del robot en el proceso es necesaria la fabricación la pieza de agarre diseñada en el punto 4.4.2, y su instalación como herramienta de agarre, conectándola al sistema neumático. Esta pieza permitirá al robot manipular eficientemente los paquetes manejados durante el proceso.
- En la implementación de ambos sistemas de automatización es recomendable llevar a cabo un período de prueba, previo a la puesta en marcha, de modo de afinar detalles como lo son la medición del metraje impreso, para el primer sistema, y la verificación de operación del robot dentro de los parámetros deseados, para el segundo.



Bibliografía y Referencias

- [1] Simone, A. (2011). **Desarrollo de una estación robotizada para el cardado de calzados, usando el lenguaje Rapid para un robot ABB IRB 6400.** Caso de estudio: Fábrica de Calzados INDUSIM C. A. Universidad de Carabobo. Valencia, Venezuela
- [2] Estévez, E., & Escalona, H. (2010). **Diseño y construcción de un prototipo de unidad de control para un sistema de transferencia de desconexión y reposición automática a la red de potencia basado en microcontroladores PIC.** Universidad de Carabobo. Valencia, Venezuela
- [3] Guape, Á. (2010). **Desarrollar un prototipo de telemetría para el proyecto “Baja” de la organización SAEUC Venezuela donde se transmitan y registren los datos de velocidad, revoluciones por minuto del motor y nivel de gasolina del vehículo.** Universidad de Carabobo. Valencia Venezuela.
- [4] Patiño, L. & Simanca, C. (2009). **Reingeniería de la línea de envasado del laboratorio de automatización Industrial II con inclusión de dispositivos de bus de campo.** Universidad de Carabobo. Valencia, Venezuela
- [5] Janampa, J. (2008). **Desarrollo de un Banco de Control de Procesos Portátil para el Laboratorio de Automatización Industrial I mediante un Micro PLC Simatic S7-200.** Universidad de Carabobo. Valencia, Venezuela.
- [6] Montoya, H. (2007). **Diseño de un sistema emulando el movimiento articulado de una mano, brazo y antebrazo.** CENIDET. Cuernavaca, México
- [7] Gridling, G., & Weis, B. (2012). **Introducción a los microcontroladores.** Universidad Tecnológica de Viena. Viena.



- [8] Mikroelektronika. (2009). <http://www.mikroe.com>. Recuperado en Mayo de 2012, de <http://www.mikroe.com/eng/products/view/285/book-pic-microcontrollers-programming-in-c/>
- [9] Silva (2007). **Cursos Complementarios de Software** - Universidad Autónoma de Nayarit. Obtenido de http://www.fv.uan.edu.mx/file.php/77/tema1/Primera_sesion.pdf
- [10] Malpica, N. (2008). lambio08.escet.urjc.es. Universidad Rey Juan Carlos. Recuperado en Mayo de 2012, de lambio08.escet.urjc.es/assets/files/tema1.pdf
- [11] Galeano, L. (2008). PIC 16F84 **una guía para el aprendizaje**. Recuperado en Mayo de 2012, de <http://entrnadorpic16f84.wordpress.com/1-arquitectura-harvard/>
- [12] Microchip. (2008). <http://www.microchip.com>. Recuperado en Octubre de 2012, de <http://ww1.microchip.com/downloads/en/devicedoc/39631e.pdf>
- [13] Rodríguez, J. (2008). **Manual de Sistema de Desarrollo MDK_B28**. Valencia.
- [14] DITEL C. A. (2012). <http://www.ditel.com.ve>. Recuperado en Mayo de 2012, de http://www.ditel.com.ve/productos.php?id=212&_pagi_pg=3
- [15] Vallejo, H. (2009). **Todo sobre PICs**. Editorial Quark. Argentina
- [16] MicroC, p. P. (2012). <http://www.programarpicenc.com>. Recuperado en Mayo de 2012, de http://www.programarpicenc.com/libro/cap10_rs232.html
- [17] Franco, E. (2010). infomundo.blogspot.com. Recuperado el Mayo de 2012, de <http://informmundo.blogspot.com/2010/04/que-llevan-por-dentro-los-cables-usb.html>



- [18] Frino.com. (2012). Frino.com. Recuperado en Mayo de 2012, de <http://www.frino.com.ar/proteus.htm>
- [19] Sapiensman. (2012). Sapiensman. Recuperado en Mayo de 2012, de <http://www.sapiensman.com/neumatica/neumatica22.htm>
- [20] Escobar. (2006). emescobar. (Universidad de Carabobo) Obtenido de <http://www.ing.uc.edu.ve/~emescobar>
- [21] Bueno, A. (2001). <http://www.portaleso.com>. Recuperado en Mayo de 2012, de http://www.portaleso.com/usuarios/Toni/web_neumatica/neumatica_indice.html
- [22] Kalipedia. (2012). Kalipedia, la enciclopedia en línea. Recuperado en Mayo de 2012, de http://www.kalipedia.com/tecnologia/tema/graficos-valvulas.html?x1=20070821klpinginf_44.Ges&x=20070821klpinginf_112.Kes
- [23] Martín, M. (2007). Wikifab - Contenido sobre ingeniería de fabricación. (Wikifab) Recuperado en Mayo de 2012, de http://wikifab.dimf.etsii.uom.es/wikifab/index.php/tarea_2:_cilindros_neum%C3%A1ticos_en_fluidsim_por_1463720
- [24] Solo mantenimiento. (Mayo de 2012). Solomantenimiento.com. Recuperado mayo de 2012, de <http://www.solomantenimiento.com/articulos/acondicionamiento-aire-presion.htm>
- [25] Instrumentación y Control. (2012). Instrumentacionycontrol.net. Recuperado en Mayo de 2012, de <http://www.instrumentacionycontrol.net/es/curso-completo-de-plcs/106-capitulo-34-tipos-de-senales-de-un-plc>
- [26] Centro de Formación del Profesorado e Innovación Educativa de Valladolid. (2012). cfievalladolid2.net. Recuperado en Mayo de 2012, de http://cfievalladolid2.net/tecno/cyr_01/robotica/



- [27] Robots Argentina. (2012). Robots Argentina. Recuperado en Mayo de 2012, de http://robots-argentine.com.ar/MotorCC_circuitosrealimentados.htm
- [28] Direct Industry. (2012). <http://www.directindustry.com>. Recuperado en Mayo de 2012, de <http://www.directindustry.com/prod/comau-robotics/robot-controllers-with-teach-pendant-15481-548711.html>
- [29] Manual Robot RV-A2J (2002). Manual de instrucciones. MELFA Industrial Robots. Mitsubishi Electric – Industrial Automation.
- [29] Sanz, W. (2009). **Cinemática de Robots Industriales**. Dirección de Medios y Publicaciones de la Universidad de Carabobo. Valencia
- [31] Gómez, F., Cuesta, F., y Ollero, A. (2012). **Planificación de trayectorias en robots móviles basada en técnicas de control de sistemas no holónomos**. Universidad de Sevilla. España
- [32] Tilbury D. M. (1995). “**Exterior Differential Systems and Nonholonomic Motion Planning**”. Tesis Doctoral. Universidad de California en Berkley.
- [33] Lamiriaux, F. y Laumond, J. (2001). “**Smooth Motion Planning for Car-Like Vehicules**”. IEEE Transactin on Robotics and Automation. Vol 12, pp. 498-501.
- [34] OTAI. (2012). OTAI - Automatización Industrial. Recuperado en Mayo de 2012, de <http://www.otai.es/paletizado/KR.aspx>
- [35] Mitsubishi - Automation. (2011). Mitsubishi - Automation. Recuperado el Mayo de 2012, de http://www.mitsubishi-automation.es/products/software_rttoolbox2_content.htm
- [36] UPEL. (1998). **Manual de trabajos de grado de especialización, maestrías y tesis doctorales**. FEDUPEL .Caracas



- [37] Bohm C. y Jacopini G. (1966). **Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules**. University of Waterloo. Ontario
- [38] Wirth N. (1978), Algorithms + Data Structures = Programs, Upper Saddle River, Prentice Hall. NJ, USA
- [39] Owe, O. Krogdahl, S. y Lyche, T. (2004). **From Object-Orientation to Formal Methods**, Springer. Berlin Germany.
- [40] Msebilbao (Julio 2012). Msebilbao.com . Recuperado en Julio de 2012. Pagina web en <http://www.msebilbao.com>
- [41] Loging (Julio 2012). Loging.mk. Recuperado en Julio de 2012. Pagina web en <http://www.loging.mk>
- [42] Bkbelectronics (Julio 2012). Bkbelectronics.com. Recuperado en Julio de 2012. Pagina web en <http://www.bkbelectronics.com>
- [43] Electronicamagnabit (Julio 2012). electronicamagnabit.com. Recuperado en Julio



Bibliografía





PRESENTACIÓN DE APÉNDICES



Apéndices





Apéndice A

Programa del PIC

```
Device 18F4520
XTAL = 4
OSCCON.4 = 0    'seleccion de 4Mhz
OSCCON.5 = 1
OSCCON.6 = 1
OSCCON.1 = 1    'oscilador interno

Declare LCD_DTPIN PORTD.4    'Pin de Datos LCD conectado al PORTC.4
Declare LCD_ENPIN PORTD.2    'Pin de Enable LCD conectado al PORTC.3
Declare LCD_RSPIN PORTD.0    'Pin de RS LCD conectado al PORTC.0

' Declaraciones portD
TRISD.1 = 0
PORTD.1 = 0
TRISD.3 = 1

' Declaraciones del puerto B
TRISB.2 = 1
TRISB.3 = 1
TRISB.4 = 1
TRISB.5 = 1
TRISB.6 = 0
TRISB.7 = 0

ALL_DIGITAL = true
Dim cuenta      As Word    'Se almacenan los ciclos por segundo
Dim Keje        As Float   'Cantidad de metros por revolucion
Dim Kencoder    As Float   'Cantidad de pulsos por revolucion
Dim Kt          As Float   'Keje / Kencoder
Dim vel         As Float   'Velocidad lineal del eje
Dim metros     As Float   'Cantidad de metros procesados
Dim bvar        As Byte    'Variable para el pulsador
Dim disp1      As Byte    'Valores a mostrar en el display
Dim disp2      As Byte
Dim disp3      As Byte
Dim disp4      As Byte
Dim meta       As Word    'Numero de metros a imprimir
Dim calc       As Word
Dim orden[10] As Word    'Ordenes impresas
Dim ord        As Byte    'Numero de ordenes impresas
```



```
Dim portin As Byte 'Almacena lo que entra por el puerto serial
Dim i As Byte
'Variables de tiempo para el reloj
Dim RTCs As Byte
Dim RTCm As Byte
Dim RTCh As Byte
Dim RTCdia As Byte
Dim RTCmes As Byte
Dim RTCano As Byte
Dim RTCndia As Byte
'Vectores para las fechas
Dim dia[10] As Byte
Dim hora[10] As Byte
Dim minuto[10] As Byte
Dim fecha[10] As Byte
Dim tiempo[4] As Byte
'Variables auxiliares para la convesion de tiempo
Dim D1 As Byte
Dim D2 As Byte
Dim Dc1 As Byte
Dim Dc2 As Byte
Dim varaux As Byte
Dim mask1 As Byte
Dim mask2 As Byte

'variable agregadas
Dim xx1 As Byte
Dim xx2 As Byte
Dim reg_ep As Byte
Dim erase As Byte 'Variable para borrar la eeprom

Symbol LedGo = PORTB.7
Symbol encoin = PORTD.3
Symbol Enter = PORTB.2
Symbol Back = PORTB.3
Symbol Upkey = PORTB.4
Symbol Downkey = PORTB.5
Symbol LedStop = PORTB.6
Symbol SDA = PORTC.4
Symbol SCL = PORTC.3
Symbol Rx = PORTC.6
Symbol Tx = PORTC.7

'Constantes e inicializaciones
Keje = 0.1
Kencoder = 4
Kt = Keje/Kencoder
meta= 0
```



```
disp1 = 0
disp2 = 0
disp3 = 0
disp4 = 0

EWrite 0x00,[0x11]
DelayMS 20

Main:

metros = 0
meta = 0
calc = 0
LedGo = 0
LedStop = 1
Cls
Print At 1,2, "Enter=Impresion"           'Menu #1, en este menu se
selecciona entre
Print At 2,2, "Back=Conectar"           'realizar una orden y
conectar con la computadora
DelayMS 600

Ciclo1:
Button Enter,1,255,0,bvar,1, Configurar
Button Back,1,255,0,bvar,1, Ciclo2
GoTo Ciclo1

Ciclo2:
Print At 1,2, "Enter=Transfer"           'Menu #1.1 este menu
esta dentro de la opcion conectar
Print At 2,2, "Back=BorrarMem"
DelayMS 300
Button Enter,1,255,0,bvar,1, Conectar
Button Back,1,255,0,bvar,1, Borrar
Button Downkey,1,255,0,bvar,1,Ciclo3
Button Upkey,1,255,0,bvar,1,Ciclo4
GoTo Ciclo2

Ciclo3:
Print At 1,2, "Enter=BorrarMem"           'Menu #1.2 este menu esta
dentro de la opcion conectar
Print At 2,2, "Back=Sincroniza"
DelayMS 300
Button Enter,1,255,0,bvar,1, Borrar
Button Back,1,255,0,bvar,1, Sinc
Button Downkey,1,255,0,bvar,1,Ciclo4
Button Upkey,1,255,0,bvar,1,Ciclo2
GoTo Ciclo3
```



```
Ciclo4:
Print At 1,2, "Enter=Sincroniza"           'Menu #1.3 este menu esta
dentro de la opcion conectar
Print At 2,2, "Back=Transferir"
DelayMS 300
Button Enter,1,255,0,bvar,1, Sinc
Button Back,1,255,0,bvar,1, Conectar
Button Downkey,1,255,0,bvar,1,Ciclo2
Button Upkey,1,255,0,bvar,1,Ciclo3
GoTo Ciclo4

Configurar:                               'Se accesa en la opcion
imprimir. Se introduce la cantidad
Cls                                         'De metros a imprimir
Print At 1,2,"Introduzca"
Print At 2,4,"Metros"
DelayMS 1200
Cls
Print At 1,2,"Back< Enter>"

LoopA:                                       'Adquisicion de los metros a
traves de los pulsadores
Print At 2,2,Dec disp1
Print At 2,3,Dec disp2
Print At 2,4,Dec disp3
Print At 2,5,Dec disp4,"mts"
Button Enter,1,255,0,bvar,1,LLL2
Button Back,1,255,0,bvar,1,Main
Button Upkey,1,255,0,bvar,1,LL1
Button Downkey,1,255,0,bvar,1,LLL1
GoTo L1
LL1:
If disp1 < 9 Then Inc disp1
DelayMS 200
GoTo L1
LLL1:
If disp1 <>0 Then Dec disp1
DelayMS 200
L1:
GoTo LoopA

LoopB:
Print At 2,2,Dec disp1
Print At 2,3,Dec disp2
Print At 2,4,Dec disp3
Print At 2,5,Dec disp4,"mts"
Button Enter,1,255,250,bvar,1, LLL3
Button Back,1,255,250,bvar,1, LL1
Button Upkey,1,255,250,bvar,1, LL2
Button Downkey,1,255,250,bvar,1, LLL2
```



```
GoTo L2
LL2:
If disp2 < 9 Then Inc disp2
DelayMS 200
GoTo L2
LLL2:
If disp2 <>0 Then Dec disp2
DelayMS 200
L2:
GoTo LoopB

LoopC:
Print At 2,2,Dec disp1
Print At 2,3,Dec disp2
Print At 2,4,Dec disp3
Print At 2,5,Dec disp4,"mts"
Button Enter,1,255,250,bvar,1, LLL4
Button Back,1,255,250,bvar,1, LL2
Button Upkey,1,255,250,bvar,1, LL3
Button Downkey,1,255,250,bvar,1, LLL3
GoTo L3
LL3:
If disp3 < 9 Then Inc disp3
DelayMS 200
GoTo L3
LLL3:
If disp3 <>0 Then Dec disp3
DelayMS 200
L3:
GoTo LoopC

LoopD:
Print At 2,2,Dec disp1
Print At 2,3,Dec disp2
Print At 2,4,Dec disp3
Print At 2,5,Dec disp4,"mts"
Button Enter,1,255,250,bvar,1, Preciclo
Button Back,1,255,250,bvar,1, LL3
Button Upkey,1,255,250,bvar,1, LL4
Button Downkey,1,255,250,bvar,1, LLL4
GoTo L4
LL4:
If disp4 < 9 Then Inc disp4
DelayMS 200
GoTo L4
LLL4:
If disp4 <>0 Then Dec disp4
DelayMS 200
L4:
GoTo LoopD
```



```
Preciclo:
Cls
ord = ERead 0x02 'leer numero de ordenes en la eeprom
If ord = 0xFF Then GoTo Mem1 '17-9 cambio para memoria full
If ord > 9 Then GoTo Mem_full
Mem1:
calc = (disp1*1000)+(disp2*100)+(disp3*10)+(disp4) 'Se calcula el
numero ingresado de mts
meta = calc
DelayMS 200
Loop2:
Print At 1,2,"Confirmar"
Print At 2,2,Dec calc,"m?"
Button Enter,1,255,250,bvar,1, cle
Button Back,1,255,250,bvar,1, Configurar
GoTo Loop2

cle:
Cls
LedStop = 0
LedGo = 1 'Se enciende la impresora

ciclo:
cuenta = Counter encoin,1000 'Mide ciclos por segundo
vel = cuenta*Kt 'Convierte en velocidad lineal
Print At 1,2, Dec cuenta,"Hz"
Print At 1,9, DECl vel,"m/s"
metros = metros+(vel) 'Suma los metros de ese segundo
Print At 2,2, DECl metros, "m=>"
Print At 2,9, Dec meta,"m"
If metros>= meta Then GoTo Listo
GoTo ciclo

Listo:
LedStop = 1
LedGo = 0
'-----
'Prueba
ord = ERead 0x02 'leer numero de ordenes en la eeprom
If ord = 0xFF Then
ord = 0
EndIf
'-----
orden[ord]=metros 'Se guarda la orden en memoria
GoSub Leer_reloj 'Leer hora de la orden y se guarda en vectores
paralelos
varaux = RTCdia
GoSub Mascara
GoSub DesConv1
```



```
GoSub DesConv2
dia[ord]= Dc1+Dc2           'Fecha de la orden
varaux = RTCh
GoSub Mascara
GoSub DesConv1
GoSub DesConv2
hora[ord]= Dc1+Dc2         'Hora de la orden
varaux = RTCm
GoSub Mascara
GoSub DesConv1
GoSub DesConv2
minuto[ord]=Dc1 +Dc2      'Minuto de la orden

'<<-----
'escritura en eeprom
reg_ep = (0x03)+(ord*5)
EWrite reg_ep, [orden[ord]]
DelayMS 20
EWrite (reg_ep+2), [dia[ord]]
DelayMS 20
EWrite(reg_ep+3), [hora[ord]]
DelayMS 20
EWrite (reg_ep+4), [minuto[ord]]
DelayMS 20
'----->>

ord = ord +1               'Se incrementa el numero de ordenes
'prueb---                 'Se guarda en eeprom
EWrite 0x02, [ord]
DelayMS 20
'-----

Cls
Print At 1,2,"Orden Lista"
Print At 2,2, DEC1 metros,"m de papel"
DelayMS 3000
LedStop = 0
GoTo Main

Conectar: 'Transferencia de datos a la PC

'<<-----
'copia de registro de eeprom en vectores
ord = ERead 0x02
For i= 0 To (ord-1)
reg_ep = (0x03)+(i*5)
xx1 = ERead reg_ep
xx2= ERead (reg_ep+0x01)
orden[i]= (xx2*0x10)+xx1
dia[i] = ERead (reg_ep+0x02)
```



```
hora[i]= ERead (reg_ep+0x03)
minuto[i] = ERead (reg_ep+0x04)
Next
'----->>>

Cls
Print At 1,2,"Conectando.."
SerIn Rx,16468,7000,Mal,[portin] ' HSerIn 7000, Mal, [portin]
If portin <> "g" Then GoTo Conectar
SerOut Tx,16468,[114] 'HSerOut [114]
SerOut Tx,16468,[ord] 'HSerOut [ord]
Print At 1,2,"Transfiriendo"
For i=0 To (ord-1) Step 1
SerIn Rx,16468,1000,Mal,[portin] 'HSerIn 1000, Mal, [portin]
If portin= 104 Then
SerOut Tx,16468,[orden[i]] 'HSerOut [orden[i]]
SerOut Tx,16468,[dia[i]] 'HSerOut [dia[i]]
SerOut Tx,16468,[hora[i]] 'HSerOut [hora[i]]
SerOut Tx,16468,[minuto[i]] 'HSerOut [minuto[i]]
EndIf
Next
GoTo Main

Mal: ' Fallo de conexion
Cls
Print At 1,2,"Error de "
Print At 2,2,"Conexion"
DelayMS 1200
GoTo Main

Borrar: ' Borrado de la eprom a traves de la PC
Cls
Print At 1,2, "Conectando.."
SerIn Rx,16468,7000, Mal,[erase] 'HSerIn 7000, Mal, [erase]
If erase = "e" Then
Cls
Print At 1,2,"Borrando"
Print At 2,2,"Memoria"
' 'Borrado de memoria EEPROM
EWrite 0,
[0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0
xffff]
EWrite 0x8,
[0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0
xffff]
EWrite 0x10,
[0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0
xffff]
EWrite 0x18,
```



```
[0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0
xffff]
  EWrite 0x20,
[0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0
xffff]
  EWrite 0x28,
[0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0
xffff]
  EWrite 0x30,
[0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0xffff,0
xffff]

Clear orden
Clear dia
Clear hora
Clear minuto
SerOut Tx,16468,[222] 'HSerOut [222]      'Indica que se borro la
memoria
Else
GoTo Borrar
EndIf

GoTo Main
Sinc:
  Cls
  Print At 1,2,"Conectando.."
  SerIn Rx,16468,7000, Mal,[portin] 'HSerIn 7000, Mal, [portin]
  If portin <> "g" Then GoTo Sinc
  Print At 1,2,"Sicronizando.."
  SerOut Tx,16468,[114] 'HSerOut [114]

  For i= 1 To 10 Step 1
  SerOut Tx ,16468, ["x"]
  SerIn Rx,16468,3000,Mal,[fecha[i-1]]
  Next

  For i= 1 To 4 Step 1
  SerOut Tx, 16468, ["x"]
  SerIn Rx,16468,3000,Mal,[tiempo[i-1]]
  Next
' SerIn PORTC.7,16468,5000, Mal,[Str fecha] 'HSerIn 5000,Mal,[Str
fecha]
' SerIn PORTC.7,16468,5000, Mal,[Str tiempo] 'HSerIn 5000,Mal,[Str
tiempo]
  Cls
  Print At 1,2,Str fecha
  Print At 2,2,Str tiempo
  DelayMS 1000

  D1 = fecha[0]
```



```
GoSub Conv1
D2 = fecha[1]
GoSub Conv2
RTCdia = D1 + D2
D1 = fecha[3]
GoSub Conv1
D2 = fecha[4]
GoSub Conv2
RTCmes = D1 + D2
D1 = fecha[8]
GoSub Conv1
D2 = fecha[9]
GoSub Conv2
RTCano = D1 + D2
D1 = tiempo[0]
GoSub Conv1
D2 = tiempo[1]
GoSub Conv2
RTCm = D1 + D2
D1 = tiempo[2]
GoSub Conv1
D2 = tiempo[3]
GoSub Conv2
RTCh = D1 + D2
RTCs=0
GoTo Esc_reloj
```

```
GoTo Main
```

```
Leer_reloj:
```

```
I2Cin SDA, SCL, 0xD0 , 0x01 , [ RTCs ]
I2Cin SDA, SCL, 0xD0 , 0x01 , [ RTCm ]
I2Cin SDA, SCL, 0xD0 , 0x02 , [ RTCh ]
I2Cin SDA, SCL, 0xD0 , 0x01 , [ RTCndia ]
I2Cin SDA, SCL, 0xD0 , 0x04 , [ RTCdia ]
I2Cin SDA, SCL, 0xD0 , 0x05 , [ RTCmes ]
I2Cin SDA, SCL, 0xD0 , 0x06 , [ RTCano ]
DelayMS 100
Return
```

```
Esc_reloj:
```

```
I2COUT SDA, SCL, 0xD0 , 0x00 , [ RTCs ]
I2COUT SDA, SCL, 0xD0 , 0x01 , [ RTCm ]
I2COUT SDA, SCL, 0xD0 , 0x02 , [ RTCh ]
I2COUT SDA, SCL, 0xD0 , 0x03 , [ 0X05 ]
I2COUT SDA, SCL, 0xD0 , 0x04 , [ RTCdia ]
I2COUT SDA, SCL, 0xD0 , 0x05 , [ RTCmes ]
```



```
I2COUT SDA, SCL, 0xD0 , 0x06 , [ RTCano ]
```

```
I2COUT SDA, SCL, 0xD0 , 0x07 , [ 0x00 ]
```

```
DelayMS 300
```

```
Return
```

Conv1 :

```
Select D1
```

```
Case 48
```

```
D1=0X0
```

```
Return
```

```
Case 49
```

```
D1=0X10
```

```
Return
```

```
Case 50
```

```
D1=0X20
```

```
Return
```

```
Case 51
```

```
D1=0X30
```

```
Return
```

```
Case 52
```

```
D1=0X40
```

```
Return
```

```
Case 53
```

```
D1=0X50
```

```
Return
```

```
EndSelect
```

Conv2 :

```
Select D2
```

```
Case 48
```

```
D2=0X00
```

```
Return
```

```
Case 49
```

```
D2=0X01
```

```
Return
```

```
Case 50
```

```
D2=0X02
```

```
Return
```

```
Case 51
```

```
D2=0X03
```

```
Return
```

```
Case 52
```

```
D2=0X04
```

```
Return
```

```
Case 53
```

```
D2=0X05
```

```
Return
```

```
Case 54
```

```
D2=0X06
```



```
Return
Case 55
D2=0X07
Return
Case 56
D2=0X08
Return
Case 57
D2=0X09
Return
EndSelect

Mascara:
    mask1= varaux & 0xF0
    mask2= varaux & 0x0F
Return

DesConv1:
    Select mask1
    Case 0x0
Dc1=00
    Return
    Case 0x10
Dc1=10
    Return
    Case 0x20
Dc1=20
    Return
    Case 0x30
Dc1=30
    Return
    Case 0x40
Dc1=40
    Return
    Case 0x50
Dc1=50
    Return
EndSelect

DesConv2:
Dc2 = mask2
Return

Mem_full:      'Memoria llena.
Cls
Print At 1,2, "Exceso de"
```



Apéndice A



Print At 2,4, "ordenes"

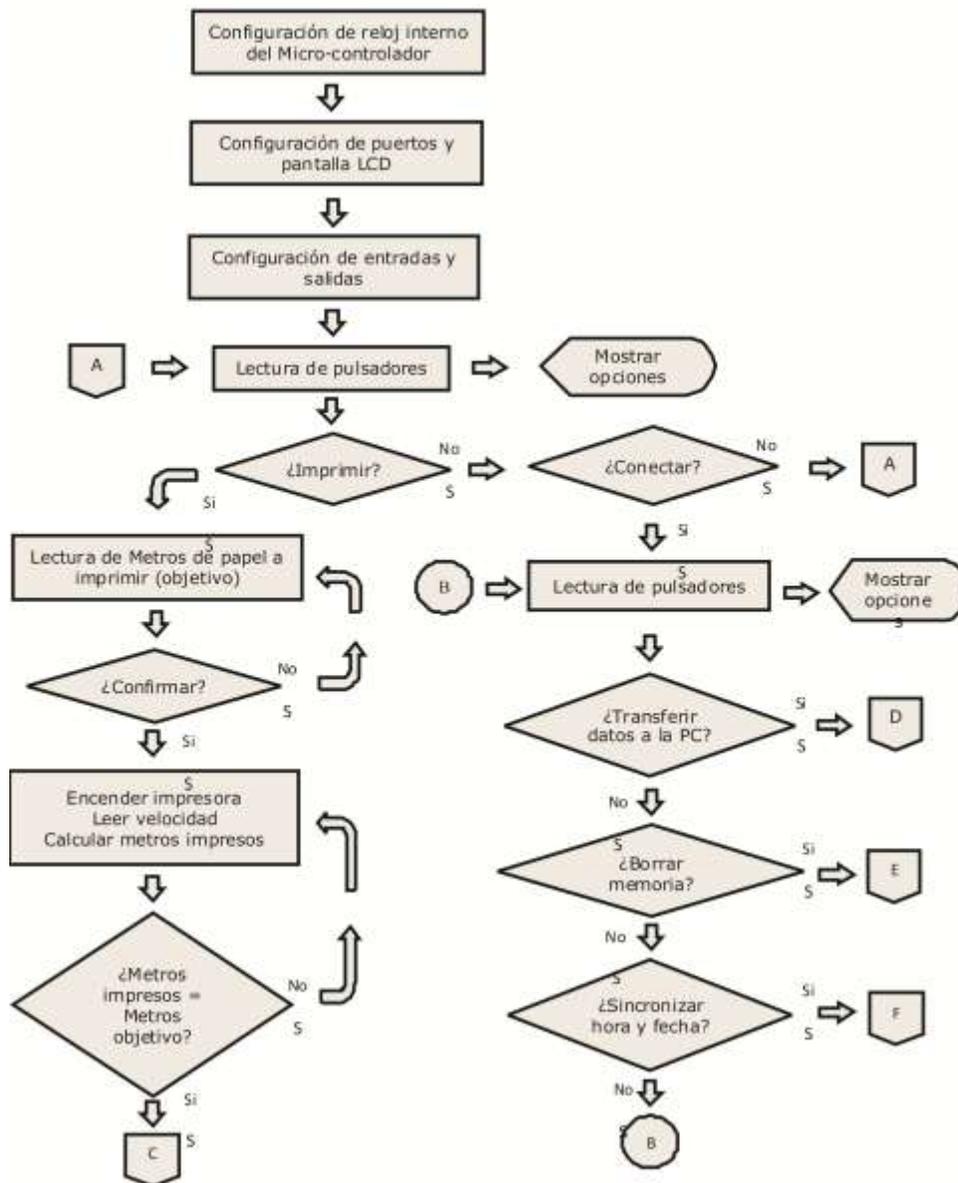
DelayMS 2000

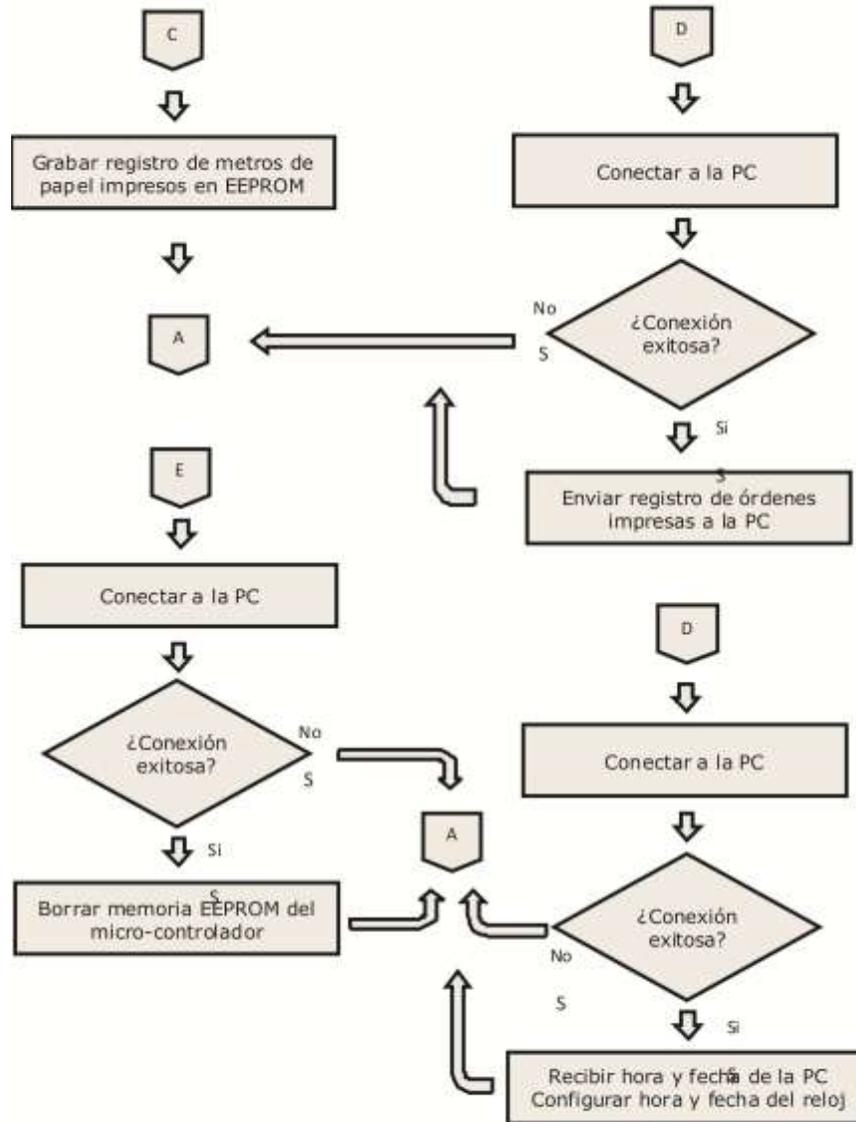
GoTo Main



Apéndice B

Diagrama de flujo para el programa del PIC







Apéndice C

Programa de la aplicación para computadora “Interfaz PIC-PC”

```
Public Class Form1
    Dim datos1() As Integer 'Guarda los metros procesados
    Dim dia() As Byte 'Dia en que fue hecha la orden
    Dim hora() As Byte 'Hora de termino de la orden
    Dim min() As Byte 'Minuto de terminada la orden
    Dim ordgen As Byte 'Guarda el numero de ordenes
    Dim registrog As String 'Guarda la direccion del archivo con las ordenes
    grabadas

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        SerialPort1.PortName = TextBox1.Text
        Button1.Enabled = False
        Button4.Enabled = True
        Button2.Enabled = True
        Button7.Enabled = True
        Try
            SerialPort1.Open()
        Catch ex As Exception
            MsgBox(Err.Description, MsgBoxStyle.Critical)
        Finally
            If SerialPort1.IsOpen = False Then
                MsgBox("Este puerto no pudo ser abierto", MsgBoxStyle.Exclamation)
            End If
        End Try
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        SerialPort1.Close()
        Button1.Enabled = True
        Button4.Enabled = False
        Button2.Enabled = False
        Button7.Enabled = False

        Label2.Text = ""
        ProgressBar1.Increment(-100)
    End Sub
```



```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Dim datosHigh() As Byte 'Aqui se guardan los datos recibidos del pic
    Dim datosLow() As Byte
    Dim sal As Byte 'Con esta variable se asegura que funcion la
conexion
    Dim ordenes As Byte 'Numero de ordenes
    Dim paso As Decimal 'Para la barra de progreso

    ProgressBar1.Increment(-(ProgressBar1.Value)) 'Limpia la barra de carga

    SerialPort1.ReadTimeout = 8000 ' Timeout para la lectura de los datos
Saludo:
    SerialPort1.Write("g")
    Try
        sal = SerialPort1.ReadByte()
    Catch ex As TimeoutException
        MsgBox(Err.Description)
        GoTo Dataerror
    End Try

    If sal <> 114 Then GoTo Saludo
    Try
        ordenes = SerialPort1.ReadByte()
    Catch ex As TimeoutException
        MsgBox(Err.Description)
        GoTo Dataerror
    End Try
    ReDim datosHigh(0 To (ordenes - 1))
    ReDim datosLow(0 To (ordenes - 1))
    ReDim dia(0 To (ordenes - 1))
    ReDim hora(0 To (ordenes - 1))
    ReDim min(0 To (ordenes - 1))
    datosHigh.Initialize()
    datosLow.Initialize()
    dia.Initialize()
    hora.Initialize()
    min.Initialize()
    paso = 100 / ordenes
    For i = 0 To (ordenes - 1) Step 1
        SerialPort1.Write("h")
        Try
            datosLow(i) = SerialPort1.ReadByte()
            datosHigh(i) = SerialPort1.ReadByte()
            dia(i) = SerialPort1.ReadByte()
            hora(i) = SerialPort1.ReadByte()
            min(i) = SerialPort1.ReadByte()
        Catch ex As TimeoutException
            MsgBox(Err.Description)
            GoTo Dataerror
        End Try
    Next i
End Sub
```



```
End Try
ProgressBar1.Increment(paso)
Next

ReDim datos1(0 To (ordenes - 1))
datos1.Initialize()

For i = 0 To ordenes - 1 Step 1
    datos1(i) = (datosHigh(i) * 16) + datosLow(i)
Next

Label2.Text = "Listo"
ordgen = ordenes

Dataerror:
End Sub

Private Sub Button5_Click(sender As System.Object, e As System.EventArgs)
Handles Button5.Click
Dim datostr As String
Dim registro As String
Dim fecha As String
Dim hora1 As String
registro = ""
datostr = ""
fecha = ""
hora1 = ""

FolderBrowserDialog1.Description = "Seleccione donde guardar el archivo"
FolderBrowserDialog1.ShowDialog() 'Muestra la ventana para seleccionar
carpetas
fecha = Format(DateValue(Now), "dddd, MMM d yyyy") 'Toma la
fecha de hoy
hora1 = Hour(Now)

registro = FolderBrowserDialog1.SelectedPath + "\orden_" + fecha + "_" +
hora1 + ".txt"
registro = registro

Try
FileOpen(1, registro, OpenMode.Output, OpenAccess.Write)
Catch ex As Exception
MsgBox(Err.Description)
MsgBox("Revise si la carpeta 'registro_impr' ha sido creada en C:\",
MsgBoxStyle.Exclamation)
GoTo erroropen
End Try

PrintLine(1, "Ordenes hechas = ", ordgen)
PrintLine(1, "Metros", "Dia", "Hora", "Minutos")
```



```
For i = 0 To (ordgen - 1) Step 1
    PrintLine(1, datos1(i), dia(i), hora(i), min(i))
Next
FileClose(1)
erroropen:
End Sub

Private Sub Button6_Click(sender As System.Object, e As System.EventArgs)
Handles Button6.Click
    Dim texto As String
    Dim texto2 As String
    Dim registro As String
    texto = ""
    texto2 = ""

    registro = registrog

    Try
        FileOpen(1, registro, OpenMode.Input, OpenAccess.Read)
    Catch ex As Exception
        MsgBox(Err.Description)
        MsgBox("El archivo no ha sido creado ", MsgBoxStyle.Exclamation)
    End Try

    Try
        While Not EOF(1)
            texto = LineInput(1)
            texto2 = texto2 + texto + vbCrLf
        End While
    Catch ex As Exception
    End Try

    RichTextBox1.Text = texto2
    FileClose(1)
    Label8.Text = registrog

End Sub

Private Sub Button2_Click(sender As System.Object, e As System.EventArgs)
Handles Button2.Click
    Dim sal As Byte
    Dim tiempo As String
    Dim fecha As String
    Dim i As Integer
    Dim saludo As Byte
    SerialPort1.ReadTimeout = 8000 ' Timeout para la lectura de los datos

    fecha = DateValue(Now) 'Toma la fecha de hoy
```



```
If Minute(Now) < 10 Then
    tiempo = "0" + CStr(Minute(Now))
Else
    tiempo = CStr(Minute(Now))
End If

If Hour(Now) < 10 Then
    tiempo = tiempo + "0" + CStr(Hour(Now))
Else
    tiempo = tiempo + CStr(Hour(Now))
End If

Label6.Text = fecha
Label7.Text = tiempo

SerialPort1.Write("g")
Try
    sal = SerialPort1.ReadByte()
Catch ex As Exception
    MsgBox(Err.Description)
    GoTo malo
End Try
If sal = 114 Then
```

```
For i = 1 To 10 Step 1
```

```
Leer1:
```

```
Try
    saludo = SerialPort1.ReadByte()
Catch ex As Exception
    MsgBox(Err.Description)
    GoTo malo
End Try
```

```
If saludo = 120 Then
    SerialPort1.Write(fecha(i - 1))
Else
    GoTo Leer1
End If
```

```
saludo = 0
```

```
Next
```

```
For i = 1 To 4 Step 1
```

```
Leer2:
```

```
Try
    saludo = SerialPort1.ReadByte()
```



```
        Catch ex As Exception
            MsgBox(Err.Description)
            GoTo malo
        End Try

        If saludo = 120 Then
            SerialPort1.Write(tiempo(i - 1))
        Else
            GoTo Leer2
        End If

        saludo = 0
    Next

End If

malo:
    End Sub

Private Sub Button7_Click(sender As System.Object, e As System.EventArgs)
Handles Button7.Click
    Dim mensaje As String
    Dim mensaje2 As String
    Dim borrado As Byte

    SerialPort1.ReadTimeout = 8000 ' Timeout para la lectura de los datos
    borrado = 0
    mensaje = "¿Esta seguro que desea borrar la memoria?"
    mensaje2 = "Si borra la memoria no podra recuperar los datos, asegurese de
haber transferido los archivos del PIC a la PC."
    MsgBox(mensaje2, MsgBoxStyle.Exclamation)

    If MsgBox(mensaje, MsgBoxStyle.YesNo) = MsgBoxResult.No Then
        GoTo noborrar
    End If
    SerialPort1.Write("e")
    Try
        borrado = SerialPort1.ReadByte()
    Catch ex As TimeoutException
        MsgBox("Error de comunicacion", MsgBoxStyle.Critical)
        GoTo noborrar
    End Try

    If borrado = 222 Then
        MsgBox("memoria borrada", MsgBoxStyle.Information)
        borrado = 0
    Else
        MsgBox("Error al borrar", MsgBoxStyle.Critical)
    End If
End If
```



Apéndice C



noborrar:

End Sub

End Class



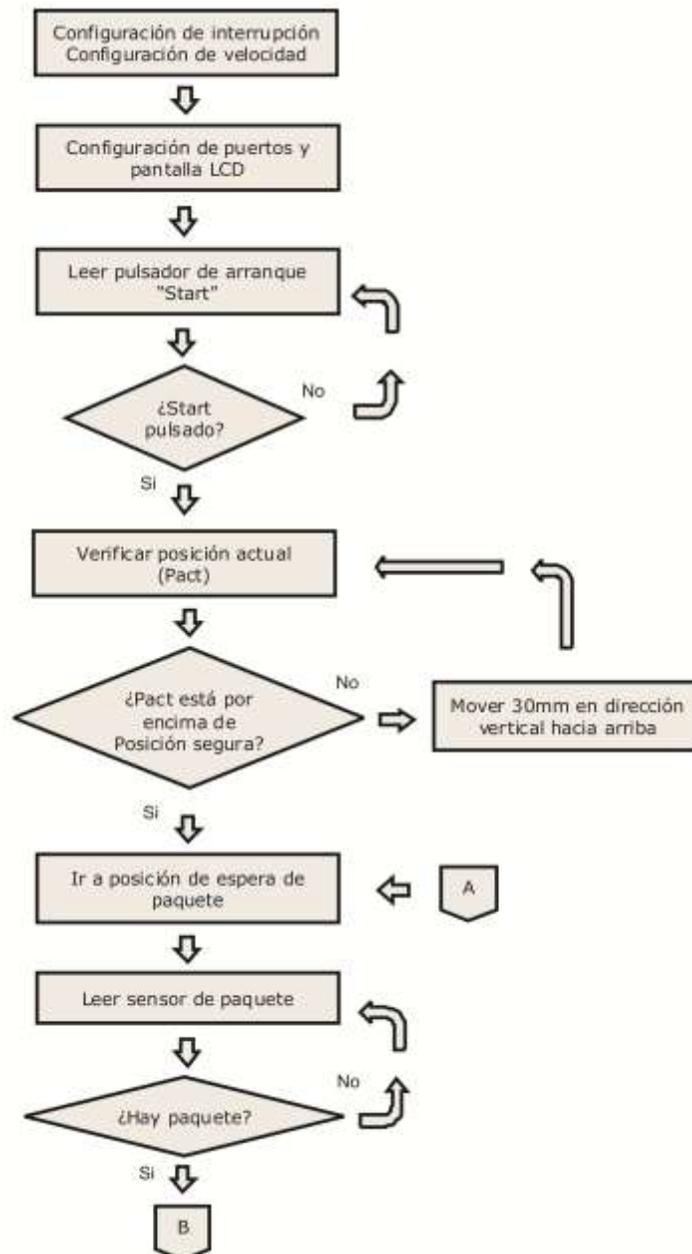
Apéndice C

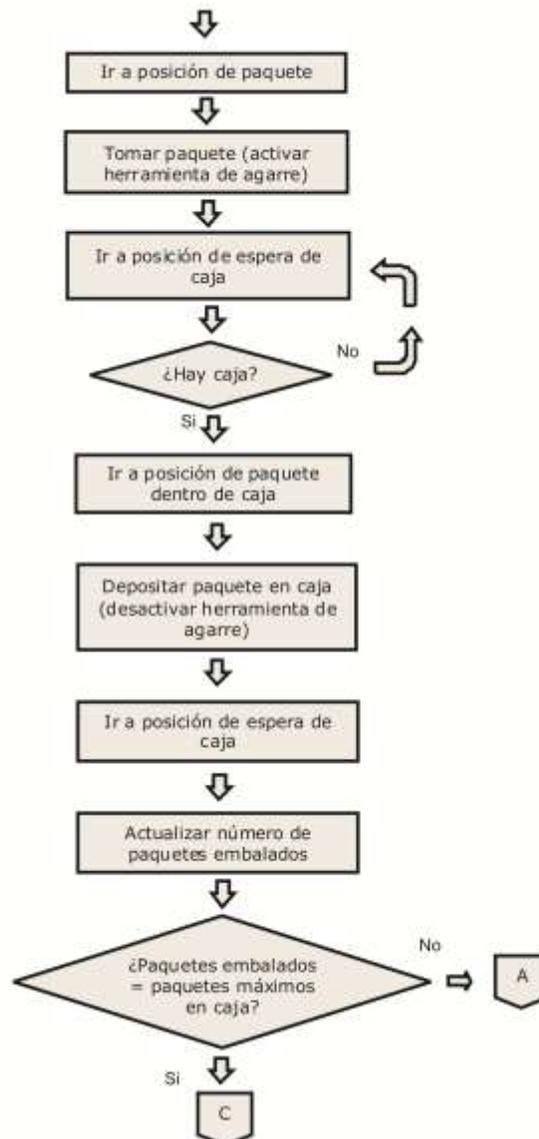


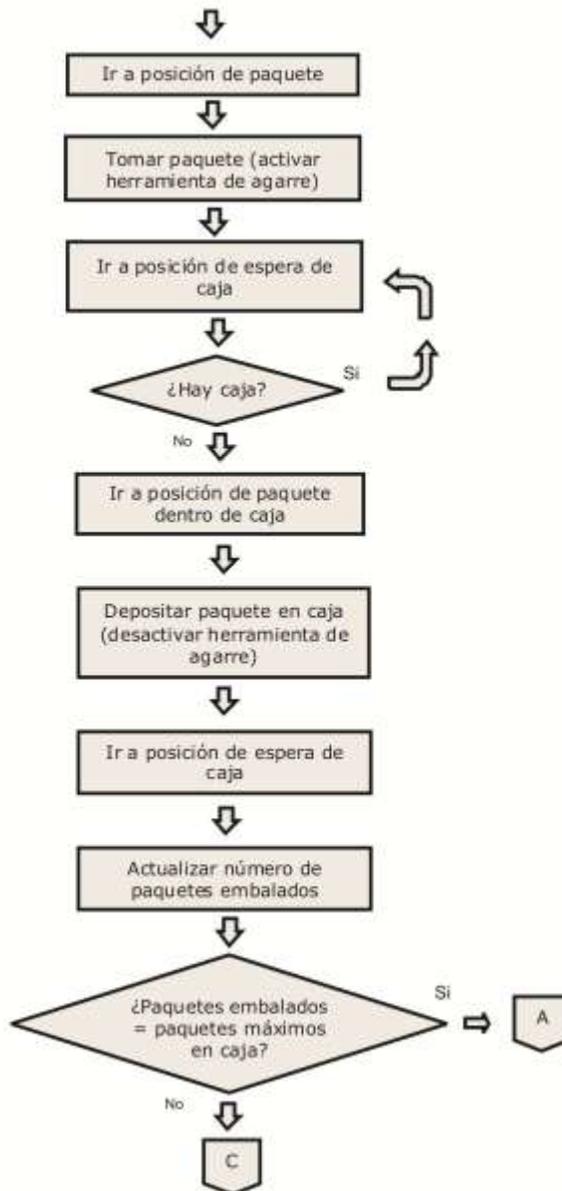


Apéndice D

Diagrama de flujo para el programa del Robot RV2AJ









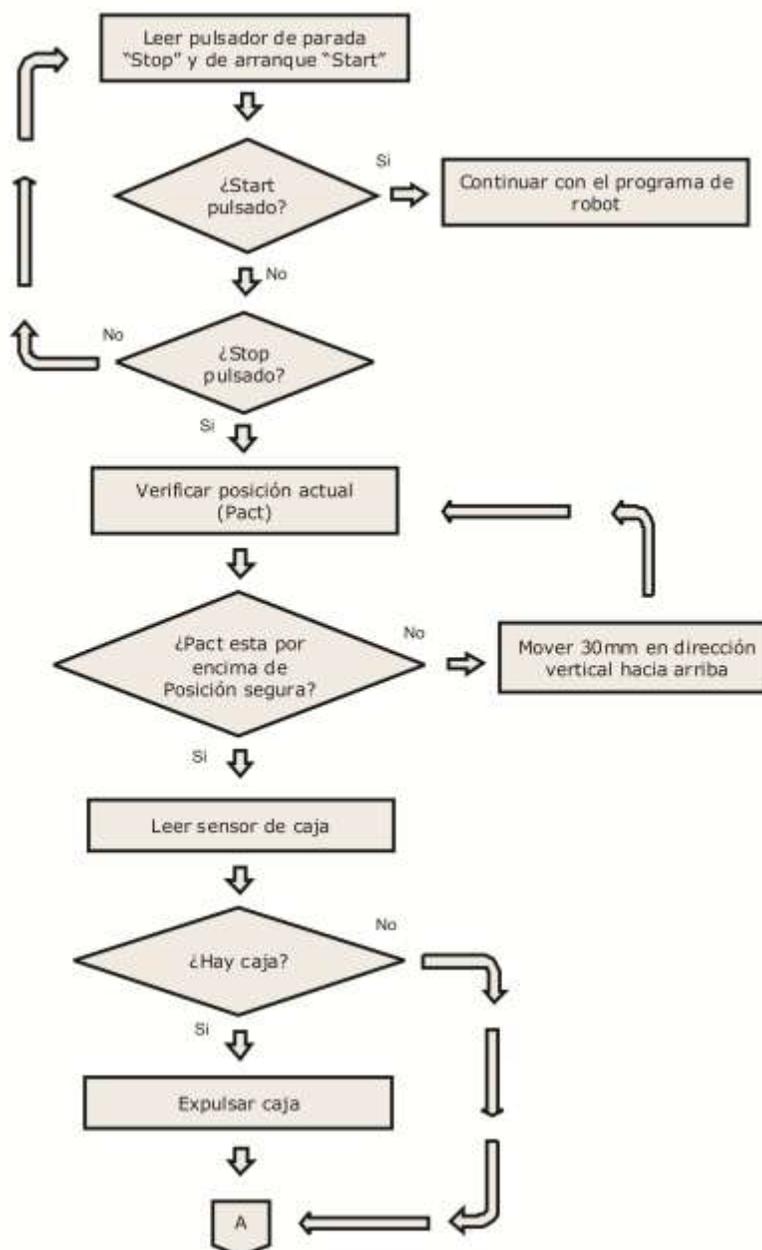
Apéndice D

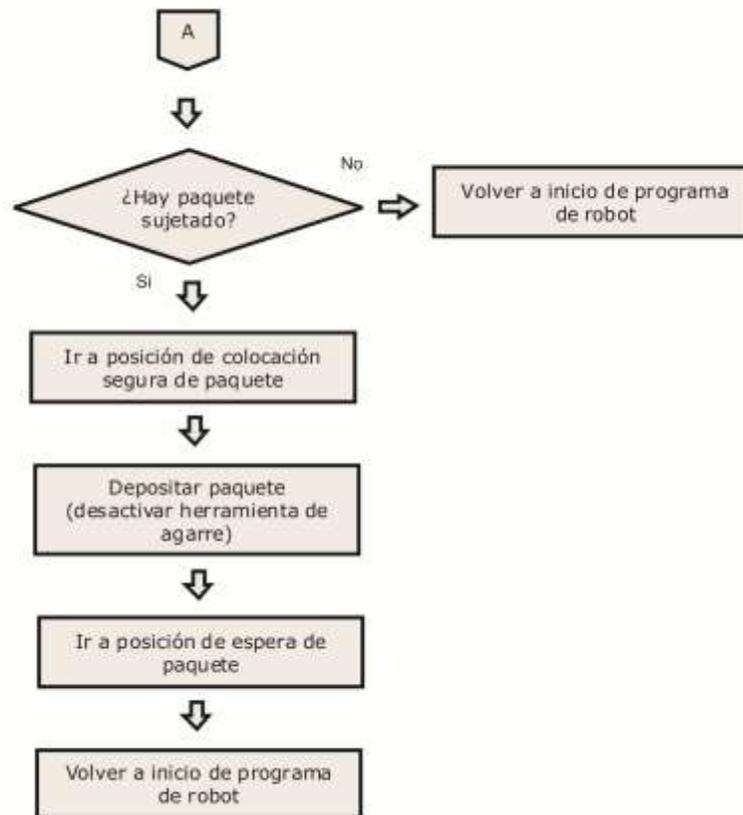




Apéndice E

Diagrama de flujo para subrutina de interrupción del programa del Robot RV2AJ







Apéndice F

Programa de embalado de robot. Formato 9 ½ x 11

Programa de embalado de robot. Formato 9 ½ x 11

```
10 'Programa de paletizado con caja (6 paquetes)
20 'Daniel Valentiner
30 OVRD 40
40 DEF ACT 1, M_IN(12)=0 GOSUB 1920 ' Definicion de interrupcion por STOP
50 ACT 1 = 1
60 M1= 0 ' M1 es el estatus de agarre ( 1= pieza agarrada, 0= sin pieza)
70 M2= 0 ' Se activa si se encuentra en el subprograma de agarre
80 M_OUT(9)=0 'Retrae el cilindro
90 WAIT M_IN(13)=1 'Espera el START
100 CALLP "T1"
110 GOSUB 1740 'Va hacia Agarre
120 MOV PI2
130 MOV P1,-40
140 OVRD 10
150 MVS P1 'Va a p1, el primer punto de la paleta
160 DLY 1
170 HOPEN 1
180 M1 = 0 ' pieza suelta
190 DLY 1
200 MVS P1,-40
210 OVRD 40
220 MOV PI2
230 GOSUB 1740 'Va hacia Agarre
240 MOV PI2
250 MOV P2,-40
260 OVRD 10
270 MVS P2 'Va hacia p2, segundo punto de la paleta
280 DLY 1
290 HOPEN 1
300 M1 = 0 ' pieza suelta
310 DLY 1
320 MVS P2,-40
330 OVRD 40
340 MOV PI2
```



350 GOSUB 1740 'Va hacia Agarre
360 MOV PI2
370 MOV P3,-40
380 OVRD 10
390 MVS P3 ' Va hacia p3, tercer punto de la paleta
400 DLY 1
410 HOPEN 1
420 M1 = 0 ' pieza suelta
430 DLY 1
440 MVS P3,-40
450 OVRD 40
460 MOV PI2
470 GOSUB 1740 'Va hacia Agarre
480 MOV PI2
490 MOV P4,-40
500 OVRD 10
510 MVS P4 ' Va hacia p4 cuarto punto de la paleta
520 DLY 1
530 HOPEN 1
540 M1 = 0 ' pieza suelta
550 DLY 1
560 MVS P4,-40
570 OVRD 40
580 MOV PI2
590 GOSUB 1740 'Va hacia Agarre
600 MOV PI2
610 MOV P5,-40
620 OVRD 10
630 MVS P5 ' Quinto punto de la paleta
640 DLY 1
650 HOPEN 1
660 M1 = 0 ' pieza suelta
670 DLY 1
680 MVS P5,-40
690 OVRD 40
700 MOV PI2
780 GOSUB 1740 'Va hacia Agarre
800 MOV PI2
810 MOV P6,-40
820 OVRD 10
830 MVS P6 'Sexto punto de la paleta
840 DLY 1



```
850 HOPEN 1
860 M1 = 0 ' pieza suelta
870 DLY 1
880 MVS P6,-40
890 OVRD 40
900 MOV PI2 '1680 MOV PI1
1690 M_OUT(9)= 1      ' Activa el actuador  OJO
1700 WAIT M_IN(11)=0  ' Espera a que salga la caja
1710 M_OUT(9)=0      ' Desactiva el actuador
1720 WAIT M_IN(11)=1  ' Espera a que entre la caja
1730 GOTO 100         ' Regresa al inicio del programa (Después del START)
1740 'Agarre (sub)
1750 M2 = 1
1760 MOV PD1
1762 MOV PD11,-25     ' Se mueve al lado de donde debería estar la pieza
1763 DLY 0.5
1764 MOV PD11
1770 WAIT M_IN(9)= 1  'Espera el paquete
1775 MOV PD11,-35
1780 MOV PD2,-35
1790 OVRD 20
1800 HOPEN 1
1810 MVS PD2
1820 DLY 1
1830 HCLOSE 1
1840 M1 = 1 ' pieza agarrada
1850 DLY 1
1860 MVS PD2,-30
1870 OVRD 40
1880 MOV PD1
1890 WAIT M_IN(11)=1 'Espera a la pieza
1900 M2 = 0
1910 RETURN           ' Fin del subprograma de Agarre
1920 ' Interrupcion (stop-sub)
1930 DLY 2
1940 IF M_IN(12) = 0 GOTO 1970 'Salta a la colocacion segura
1950 IF M_IN(13) = 1 GOTO 2170 'va al return
1960 GOTO 1940        'ciclo
1970 CALLP "T1"
1980 IF M2 = 0 THEN MOV PI1
1990 IF M2 = 1 THEN MOV PD1
2000 IF M_IN(11) = 0 THEN GOTO 2040 'Si no hay caja, salta
```



2010 M_OUT(9)= 1	'Extiende el cilindro
2020 WAIT M_IN(11)=0	'Espera que salga la caja
2030 M_OUT(9)=0	'Retrae el cilindro
2040 IF M1 = 0 THEN GOTO 10	'Si no tiene pieza va al principio del programa
2050 OVRD 30	
2060 IF M1 = 1 THEN MOV PD1	
2070 OVRD 25	
2080 MOV PS1,-170	
2090 MOV PS1	'PSAVE HAY QUE CAMBIARLO
2100 DLY 1	
2110 HOPEN 1	
2120 DLY 1	
2130 MOV PS1,-10	
2140 OVRD 40	
2150 MOV PD1	
2160 GOTO 10	'ir al inicio del programa
2170 RETURN 0	
2180	' Fin del subprograma de rutina de interrupción por STOP
2190 END	



Apéndice G

Programa de embalado paquetes 9 ½ x 5 ½

```
Programa de embalado paquetes 9 ½ x 5 ½
10 'Programa de paletizado con caja (12 paquetes)
20 'Daniel Valentiner
30 OVRD 40
40 DEF ACT 1, M_IN(12)=0 GOSUB 1920 ' Definicion de interrupcion por STOP
50 ACT 1 = 1
60 M1= 0 ' M1 es el estatus de agarre ( 1= pieza agarrada, 0= sin pieza)
70 M2= 0 ' Se activa si se encuentra en el subprograma de agarre
80 M_OUT(9)=0 'Retrae el cilindro
90 WAIT M_IN(13)=1 'Espera el START
100 CALLP "T1"
110 GOSUB 1740 'Va hacia Agarre
120 MOV PI2D
130 MOV P1,-40
140 OVRD 10
150 MVS P1 'Va a p1, el primer punto de la paleta
160 DLY 1
170 HOPEN 1
180 M1 = 0 ' pieza suelta
190 DLY 1
200 MVS P1,-40
210 OVRD 40
220 MOV PI2D
230 GOSUB 1740 'Va hacia Agarre
240 MOV PI2I
250 MOV P2,-40
260 OVRD 10
270 MVS P2 'Va hacia p2, segundo punto de la paleta
280 DLY 1
290 HOPEN 1
300 M1 = 0 ' pieza suelta
310 DLY 1
320 MVS P2,-40
330 OVRD 40
340 MOV PI2I
350 GOSUB 1740 'Va hacia Agarre
```



360 MOV PI2D
370 MOV P3,-40
380 OVRD 10
390 MVS P3 ' Va hacia p3, tercer punto de la paleta
400 DLY 1
410 HOPEN 1
420 M1 = 0 ' pieza suelta
430 DLY 1
440 MVS P3,-40
450 OVRD 40
460 MOV PI2D
470 GOSUB 1740 'Va hacia Agarre
480 MOV PI2I
490 MOV P4,-40
500 OVRD 10
510 MVS P4 ' Va hacia p4 cuarto punto de la paleta
520 DLY 1
530 HOPEN 1
540 M1 = 0 ' pieza suelta
550 DLY 1
560 MVS P4,-40
570 OVRD 40
580 MOV PI2I
590 GOSUB 1740 'Va hacia Agarre
600 MOV PI2D
610 MOV P5,-40
620 OVRD 10
630 MVS P5 ' Quinto punto de la paleta
640 DLY 1
650 HOPEN 1
660 M1 = 0 ' pieza suelta
670 DLY 1
680 MVS P1,-40
690 OVRD 40
700 MOV PI2D
780 GOSUB 1740 'Va hacia Agarre
800 MOV PI2I
810 MOV P6,-40
820 OVRD 10
830 MVS P6 'Sexto punto de la paleta
840 DLY 1
850 HOPEN 1



860 M1 = 0 ' pieza suelta
870 DLY 1
880 MVS P6,-40
890 OVRD 40
900 MOV PI2I
920 GOSUB 1740 'Va hacia Agarre
940 MOV PI2D
950 MOV P7,-40
960 OVRD 10
970 MVS P7 ' Séptimo punto de la paleta
980 DLY 1
990 HOPEN 1
1000 M1 = 0 ' pieza suelta
1010 DLY 1
1020 MVS P7,-40
1030 OVRD 40
1040 MOV PI2D
1060 GOSUB 1740 'Va hacia Agarre
1080 MOV PI2I
1090 MOV P8,-40
1100 OVRD 10
1110 MVS P8 'Octavo punto de la paleta
1120 DLY 1
1130 HOPEN 1
1140 M1 = 0 ' pieza suelta
1150 DLY 1
1160 MVS P8,-40
1170 OVRD 40
1190 MOV PI2I 'NUEVOOOO
1200 GOSUB 1740 'Va hacia Agarre
1210 MOV PI2D
1220 MOV P9,-40
1230 OVRD 10
1240 MVS P9 'Noveno punto de la paleta
1250 DLY 1
1260 HOPEN 1
1270 M1 = 0 ' pieza suelta
1280 DLY 1
1290 MVS P9,-40
1300 OVRD 40
1310 MOV PI2D
1320 GOSUB 1740 'Va hacia Agarre



1330	MOV	PI2I	
1340	MOV	P10,-40	
1350	OVRD	10	
1360	MVS	P10	'Decimo punto de la paleta
1370	DLY	1	
1380	HOPEN	1	
1390	M1 = 0		' pieza suelta
1400	DLY	1	
1410	MVS	P10,-40	
1420	OVRD	40	
1430	MOV	PI2I	
1440	GOSUB	1740	'Va hacia Agarre
1450	MOV	PI2D	
1460	MOV	P11,-40	
1470	OVRD	10	
1480	MVS	P11	' Onceavo punto de la paleta
1490	DLY	1	
1500	HOPEN	1	
1510	M1 = 0		' pieza suelta
1520	DLY	1	
1530	MVS	P11,-40	
1540	OVRD	40	
1550	MOV	PI2D	
1560	GOSUB	1740	'Va hacia Agarre
1570	MOV	PI2I	
1580	MOV	P12,-40	
1590	OVRD	10	
1600	MVS	P12	' Doceavo punto de la paleta
1610	DLY	1	
1620	HOPEN	1	
1630	M1 = 0		' pieza suelta
1640	DLY	1	
1650	MVS	P12,-40	
1660	OVRD	40	
1670	MOV	PI2I	'FIN NUEVOO
1680	MOV	PI1	
1690	M_OUT(9)= 1		' Activa el actuador OJO
1700	WAIT	M_IN(11)=0	' Espera a que salga la caja
1710	M_OUT(9)=0		' Desactiva el actuador
1720	WAIT	M_IN(11)=1	' Espera a que entre la caja
1730	GOTO	100	' Regresa al inicio del programa (Después del START)
1740			'Agarre (sub)



```
1750 M2 = 1
1760 MOV PD1
1770 WAIT M_IN(14)= 1 'Espera el paquete
1780 MOV PD2,-30
1790 OVRD 20
1800 HOPEN 1
1810 MVS PD2
1820 DLY 1
1830 HCLOSE 1
1840 M1 = 1 ' pieza agarrada
1850 DLY 1
1860 MVS PD2,-30
1870 OVRD 40
1880 MOV PD1
1890 WAIT M_IN(11)=1 'Espera a la pieza
1900 M2 = 0
1910 RETURN ' Fin del subprograma de Agarre
1920 ' Interrupcion (stop-sub)
1930 DLY 2
1940 IF M_IN(12) = 0 GOTO 1970 'Salta a la colocación segura
1950 IF M_IN(13) = 1 GOTO 2170 'va al return
1960 GOTO 1940 'ciclo
1970 CALLP "T1"
1980 IF M2 = 0 THEN MOV PI1
1990 IF M2 = 1 THEN MOV PD1
2000 IF M_IN(11) = 0 THEN GOTO 2040 'Si no hay caja salta
2010 M_OUT(9)= 1 'Extiende el cilindro
2020 WAIT M_IN(11)=0 'Espera que salga la caja
2030 M_OUT(9)=0 'Retrae el cilindro
2040 IF M1 = 0 THEN GOTO 10 'Si no tiene pieza va al principio del programa
2050 OVRD 30
2060 IF M1 = 1 THEN MOV PD1
2070 OVRD 25
2080 MOV PS1,-170
2090 MOV PS1
2100 DLY 1
2110 HOPEN 1
2120 DLY 1
2130 MOV PS1,-170
2140 OVRD 40
2150 MOV PD1
2160 GOTO 10 'ir al inicio del programa
```



Apéndice G



2170 RETURN 0

2180 ' Fin del subprograma de rutina de interrupción por STOP

2190 END



Apéndice H

Código del Subprograma del robot

Código del Subprograma del robot

10 OVRD 20

20 P1=P_ZERO 'Inicializa el punto en cero

25 P2=P_ZERO

30 P1=P_CURR 'Obtiene la posición actual y la guarda en p1

40 DLY 0.1

50 IF P1.Z>252 THEN GOTO 110 'Si la coordenada z de p1 es mayor que 252 mm sale del subprograma

70 P2= P1

80 P2.Z= P1.Z+30 'Suma 30mm a la coordenada z de p1

90 MVS P2

100 GOTO 20 'regresa al inicio

110 END
