

**UNIVERSIDAD AUTÓNOMA AGRARIA ANTONIO NARRO**  
**DIVISIÓN DE INGENIERÍA**  
**DEPARTAMENTO DE MAQUINARIA AGRÍCOLA**



*Manual de control automático de temperatura mediante  
Microcontroladores y PLC*

**POR:**

**Iván Pacheco Antonio**

**TESIS**

**Presentada como requisito parcial para  
Obtener el título de:**

**INGENIERO MECÁNICO AGRÍCOLA**

**Buenavista, Saltillo, Coahuila, México**

**Junio del 2013**

**UNIVERSIDAD AUTÓNOMA AGRARIA**

**“ANTONIO NARRO”**

**DIVISIÓN DE INGENIERÍA**

**DEPARTAMENTO DE MAQUINARIA AGRÍCOLA**

***Manual de control automático de temperatura mediante  
Microcontroladores y PLC***

**POR:**

**Iván Pacheco Antonio**

**TESIS**

Que Somete a Consideración del H. Jurado Examinador Como Requisito Parcial  
para Obtener el Título de:

**INGENIERO MECÁNICO AGRÍCOLA**

**Aprobado por el comité de tesis**

**Asesor Principal**

**Dr. Santos Gabriel Campos Magaña**

**Co-director**

**Sinodal**

**Dr. Martin Cadena Zapata**

**Ing. Juan Antonio López López**

Universidad Autónoma Agraria  
**“ANTONIO NARRO”**

**Coordinador de la división de ingeniería**

**M.C. Luis Rodriguez Gutiérrez**



## **AGRADECIMIENTOS**

A la **Universidad Autónoma Agraria Antonio Narro** por haberme abierto sus puertas y permitirme cumplir una meta más en la vida.

Al **Dr. Santos G. Campos Magaña**, por su gran apoyo y confianza, por el tiempo dedicado en el desarrollo de este proyecto, por su excelente dirección y asesoría técnica en este trabajo de tesis.

Al **Dr. Martín Cadena Zapata**, por su enseñanza y la oportunidad de compartir sus conocimientos. Por la confianza de trabajar en el Departamento de Maquinaria Agrícola.

Al **Ing. Juan López López** por su apoyo que me brindo de manera desinteresada.

A los distinguidos profesores del departamento de Maquinaria Agrícola en especial al **M. C. Héctor Uriel Serna Fernández, M. C. Juan Antonio Guerrero Hernández, Ing. Juan Arredondo Valdez, M.C. Mario Alberto Méndez Dorado Dr. Jesús Valenzuela García, Ing. Tomás Gaytán Muñiz, Ing. Jorge Flores Berrueto, Ing. Rosendo González Garza**, y a todos aquellos profesores que aportaron sus conocimientos y experiencias durante el desarrollo de mi carrera.

## **DEDICATORIA**

### **A mis padres:**

Efrén Jorge Donay Pacheco Pérez  
Juana Antonio Echeverría.

### **Hermanos:**

Tokyche Pacheco Antonio  
Igor Pacheco Antonio  
Otoniel González Pacheco

### **A mi cuñada y sobrino:**

Alma Yadira Martínez Zuviri  
Edahi Emiliano Pacheco Martínez

### **A mi abuela**

Herlinda Leonor Echeverría Antonio †

### **A mis amigos**

Esmeralda, Leandro "William", Prisciliano "Chueco", Luis "Chivo", Rafael Salas, Eduardo Reséndiz, Juan González, Daniel Rodríguez, y en especial a quienes hicieron inolvidable esta estancia en Saltillo, Cecilia Varela Zamarrón, Reynaldo Pérez "Kenan", Moisés Guadarrama "la Vaquera", Alonso "Toro" Solís, Ramaosiris "Ramaberto", Salvador Azpeitia "Charmin".

### **A mis Coaches**

Juan Javier "Brujo" González, Sergio Rubén Reséndiz "Lechero", Roberto Cepeda "el Seco", Leonel Charvel, Jaime "Pibis" Contreras.

### **Al equipo de Football Americano**

Por formar parte de un gran equipo que son los Buitres de Agricultura.

"Buitres siempre Buitres de la Narro"

# ÍNDICE DE CONTENIDO

<b>INDICE DE FIGURAS .....</b>	<b>vii</b>
<b>INDICE DE CUADROS.....</b>	<b>xii</b>
<b>RESUMEN.....</b>	<b>xiv</b>
<b>I INTRODUCCION .....</b>	<b>1</b>
<b>II REVISIÓN DE LITERATURA .....</b>	<b>3</b>
<b>2.1 Sistemas de control .....</b>	<b>3</b>
<b>2.2 Tipos de sistemas de control.....</b>	<b>4</b>
2.2.1 Sistema en lazo abierto.....	4
2.2.2 Sistema en lazo cerrado.....	4
<b>2.3 Automatización .....</b>	<b>5</b>
<b>2.4 PLC.....</b>	<b>6</b>
2.4.1 Clasificación de PLC.....	7
2.4.2 Tipos de programación de PLC .....	10
<b>2.5 Sensores .....</b>	<b>14</b>
2.5.1 Tipos de sensores.....	14
<b>2.6 Microcontrolador .....</b>	<b>14</b>
<b>III MATERIALES, EQUIPO Y MÉTODO .....</b>	<b>16</b>
<b>3.1 Materiales y equipo .....</b>	<b>16</b>
3.1.1 PLC Festo FC34.....	16
3.1.2 Termostato electrónico .....	17
3.1.3 Sensores DS-18B20 .....	17
3.1.4 PIC16F84A.....	18
3.1.5 Módulo de simulación de sensores .....	19
3.1.6 Programador PICSTAR Plus .....	19
3.1.7 Festo FST V4.10.50.....	20
3.1.8 MPLab IDE V8.10.....	20
3.1.9 Proteus V7.9. ....	21
<b>3.2 Método.....</b>	<b>22</b>
3.2.1 Programación de PLC.....	22
3.2.2 Programación de sensores de temperatura .....	23
3.2.3 Programación de PLC-UAAAN.....	23
3.2.4 Estructuración de manual de prácticas de automatización.....	24
<b>IV RESULTADOS Y DISCUSION .....</b>	<b>26</b>
<b>4.1 Ejercicio 1. Encendido y apagado de led .....</b>	<b>26</b>
4.1.1 Definición de entradas y salidas .....	27

4.1.2	Diseño del programa con variables .....	28
4.1.3	Compilación y descarga del programa al PLC .....	28
4.1.4	Conexión de entradas y salidas.....	28
4.1.5	Ejecución en línea para verificación de errores.....	29
4.1.6	Ejecución del programa para verificación del funcionamiento .....	29
<b>4.2</b>	<b>Ejercicio 2. Compuertas lógicas .....</b>	<b>29</b>
4.2.1	Definición de entradas y salidas .....	30
4.2.2	Diseño de programa con variables .....	31
4.2.3	Compilación, descarga de programa en PLC.....	32
4.2.4	Conexiones de entrada y salida .....	32
4.2.5	Ejecución en línea con la verificación de errores.....	32
4.2.6	Ejecución del programa para verificación del funcionamiento .....	32
<b>4.3</b>	<b>Ejercicio 3. Puerta de garaje.....</b>	<b>33</b>
4.3.1	Definición de entradas y salidas .....	34
4.3.2	Diseño de programa con variables .....	34
4.3.3	Compilación, descarga de programa en PLC.....	35
4.3.4	Conexiones de entrada y salida .....	35
4.3.5	Ejecución en línea con la verificación de errores.....	35
4.3.6	Ejecución del programa para verificación del funcionamiento .....	35
<b>4.4</b>	<b>Ejercicio 4. Encendido de motores.....</b>	<b>36</b>
4.4.1	Definición de entradas y salidas .....	37
4.4.2	Diseño del programa con variables .....	37
4.4.3	Compilación y descarga del programa al PLC .....	38
4.4.4	Conexión de entradas y salidas.....	38
4.4.5	Ejecución en línea para verificación de errores.....	38
<b>V</b>	<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>39</b>
<b>VI</b>	<b>BIBLIOGRAFÍAS.....</b>	<b>40</b>
<b>VII</b>	<b>ANEXOS.....</b>	<b>43</b>
<b>ANEXO 7.1</b>	<b>.....</b>	<b>44</b>
<b>ANEXO 7.2</b>	<b>.....</b>	<b>48</b>
7.2.1	Funciones de la placa de control .....	48
7.2.2	Pulsador Modo.....	49
7.2.2.1	Modo "Termostato_OFF" .....	49
7.2.2.2	Modo "Termostato_Ajuste" .....	49
7.2.2.3	Modo "Termostato_ON" .....	49
7.2.3	Interrupciones.....	50
7.2.4	Funciones del termómetro DS1820 .....	51
7.2.5	Resumen del programa.....	52

<b>ANEXO 7.3.....</b>	<b>53</b>
7.3.1 Uso del MPLAB y grabado del microcontrolador .....	53
7.3.2 Conexión del microcontrolador. ....	55
7.3.3 El quemador o grabador PICSTART Plus, Figura (7.14) .....	56
<b>ANEXO 7.4.....</b>	<b>59</b>
7.4.1 Programación en FST 4.10 .....	59
7.4.2 Instalación del PLC y módulo de simulación de sensores .....	59
7.4.3 PLC Festo FC34 .....	61
7.4.4 Instalación de Software FST 4.10 .....	62
4.5 Ejercicio 1. Encendido y apagado de un led. ....	72
7.4.6 Descarga de programa en el PLC .....	76
<b>ANEXO 7.5.....</b>	<b>80</b>
7.5.1 Puertas Logicas.....	80
7.5.2 Practica AND y OR .....	81
7.5.3 Variables .....	82
7.5.4 Compilacion del programa .....	84
7.5.5 Descarga de la Pcc al PLC .....	84
7.5.6 Ejecucion de programa .....	85
7.5.6.1 Primer caso .....	85
7.5.6.2 Segundo caso .....	87
7.5.6.3 Tercer caso.....	88
7.5.6.4 Cuarto caso .....	90
7.5.7 Practica NAN y NOR.....	92
7.5.8 Variables .....	93
7.5.9 Compilacion del programa .....	94
7.5.10 Descarga de la PC al PLC.....	95
7.5.11 Ejecucion de programa .....	96
7.5.11.1 Primer caso .....	96
7.5.11.2 Segundo caso .....	98
7.5.11.3Tercer caso.....	99
7.5.11.4 Cuarto caso .....	101
<b>ANEXO 7.6.....</b>	<b>103</b>
7.6.1 Puerta de garage .....	103

<b>7.6.2 Variables .....</b>	<b>103</b>
<b>7.6.3 Compilacion del programa .....</b>	<b>105</b>
<b>7.6.4 Descarga de la PC al PLC.....</b>	<b>105</b>
<b>7.6.5 Ejecucion de programa .....</b>	<b>107</b>
7.6.5.1 Abrir puerta desde afuera. ....	107
7.6.5.2 Cerrar puerta desde adentro. ....	109
7.6.5.3 Abrir desde adentro.....	110
7.6.5.4 Cerrar puerta desde afuera .....	112
<b>ANEXO 7.7.....</b>	<b>115</b>
<b>7.7.1 Activacion de motores. ....</b>	<b>115</b>
<b>7.7.2 Variables .....</b>	<b>116</b>
<b>7.7.3 Compilacion del programa .....</b>	<b>117</b>
<b>7.6.4 Descarga de la PC al PLC.....</b>	<b>117</b>
<b>7.7.5 Ejecución del programa .....</b>	<b>119</b>
7.7.5.1 Primer caso .....	119
7.7.5.2 Segundo caso .....	120
7.7.5.3 Tercer caso.....	121
7.7.5.4 Cuarto caso .....	123

## INDICE DE FIGURAS

<i>Figura 2.1 Sistema de control en lazo abierto.....</i>	<i>4</i>
<i>Figura 2. 2 Sistema en lazo cerrado.....</i>	<i>5</i>
<i>Figura 2.3 PLC tipo Nano.....</i>	<i>8</i>
<i>Figura 2.4 PLC tipo compacto.....</i>	<i>9</i>
<i>Figura 2.5 PLC tipo modular.....</i>	<i>10</i>
<i>Figura 2.6 Diagrama Grafcet .....</i>	<i>11</i>
<i>Figura 2.7 Diagrama de lista de instrucciones, .....</i>	<i>12</i>
<i>Figura 2.8 Texto estructurado .....</i>	<i>12</i>
<i>Figura 2.9 Diagrama de contactos .....</i>	<i>13</i>
<i>Figura 2.10 Diagrama de funciones.....</i>	<i>13</i>
<i>Figura 3.1 PLC Festo FC34 .....</i>	<i>17</i>
<i>Figura 3.2 Sensor DS18B20 .....</i>	<i>18</i>
<i>Figura 3.3 Microcontrolador PIC16F84A.....</i>	<i>18</i>
<i>Figura 3.4 Módulo de simulación de sensores .....</i>	<i>19</i>
<i>Figura 3.5 Programador PICSTAR Plus.....</i>	<i>20</i>
<i>Figura 3.6 Diagrama de bloques para la creación de programas en FST 4.10 .....</i>	<i>22</i>
<i>Figura 3.7 Diagrama de bloques para el quemado de PIC.....</i>	<i>23</i>
<i>Figura 4.1 PLC y módulo de simulación conectados .....</i>	<i>27</i>
<i>Figura 4.2 Tipos de contactos e inversor.....</i>	<i>28</i>
<i>Figura 4.3 Programa de encendido y apagado de led .....</i>	<i>28</i>
<i>Figura 4.4 Programa AND y OR.....</i>	<i>31</i>
<i>Figura 4.5 Programas NAND y NOR.....</i>	<i>31</i>
<i>Figura 4.6 Programa de puerta de Garaje .....</i>	<i>35</i>
<i>Figura 4.7 Programa de encendido de motores .....</i>	<i>37</i>
<i>Figura 7.1 Descripción física.....</i>	<i>45</i>
<i>Figura 7.2 Distribución de E/S y demás especificaciones .....</i>	<i>45</i>
<i>Figura 7.3 Estados de led .....</i>	<i>46</i>

<i>Figura 7.4 Conexión de entradas</i> .....	46
<i>Figura 7.5 Conexión de entradas de señal (sensores)</i> .....	47
<i>Figura 7.6 Conexión de salida</i> .....	47
<i>Figura 7.7 Posición de los pulsadores en las placas de control</i> .....	48
<i>Figura 7.8 Diagrama de flujo de la subrutina Cambiar_Modo</i> .....	49
<i>Figura 7.9 Diagrama de flujo de la subrutina de atención a las interrupciones</i> .....	50
<i>Figura 7.10 Diagrama de flujo principal del termostato digital</i> .....	51
<i>Figura 7.11 Selección del programa “asm”</i> .....	54
<i>Figura. 7.12 Procedimiento para el ensamblado</i> .....	54
<i>Figura 7.13 USB Adapter Keyspan</i> .....	55
<i>Figura 7.14 PICSTART Plus</i> .....	56
<i>Figura 7.15 Verificación de memoria</i> .....	57
<i>Figura 7.16 Borrar memoria del micro controlador</i> .....	57
<i>Figura 7.17 Borrar la memoria del microcontrolador</i> .....	58
<i>Figura 7.18 Programación directa</i> .....	58
<i>Figura 7.19 Fuente de poder de 24 V</i> .....	59
<i>Figura 7.20 Entradas de Señal de PLC</i> .....	60
<i>Figura 7.21 Salida de Señal</i> .....	60
<i>Figura 7.22 Modulo de simulación de sensores</i> .....	60
<i>Figura 7.23 PLC Festo FC34</i> .....	61
<i>Figura 7.23 FST 4.10</i> .....	61
<i>Figura 7.24 Explorador de archivos</i> .....	62
<i>Figura 7.26 Selección de opciones a instalar</i> .....	62
<i>Figura 7.27 Inicio de instalación de FST4.10</i> .....	63
<i>Figura 7.28 Términos y condiciones</i> .....	63
<i>Figura 7.29 Ventanas en el proceso de instalación</i> .....	64
<i>Figura 7.30 Ventanas con opciones de ubicación e idiomas a instalar</i> .....	64
<i>Figura 7.31 Opciones y copiado de archivos</i> .....	65

<b>Figura 7.32 Finalización de instalación .....</b>	<b>65</b>
<b>Figura 7.33 Ventana del programa FST 4.10.....</b>	<b>66</b>
<b>Figura 7.34 Pestaña de Preferences .....</b>	<b>66</b>
<b>Figura 7.36 FST Preferences .....</b>	<b>67</b>
<b>Figura 7.37 Propiedades del sistema.....</b>	<b>67</b>
<b>Figura 7.38 Administrador de dispositivos.....</b>	<b>68</b>
<b>Figura 7.39 Propiedades de puerto de comunicaciones (COM1) .....</b>	<b>68</b>
<b>Figura 7.40 Opciones de pestaña Project.....</b>	<b>69</b>
<b>Figura 7.41 New Project.....</b>	<b>69</b>
<b>Figura 7.42 Opciones .....</b>	<b>70</b>
<b>Figura 7.43 Allocation list entry .....</b>	<b>70</b>
<b>Figura 7.44 Insertar nuevo programa.....</b>	<b>71</b>
<b>Figura 7.45 New program .....</b>	<b>72</b>
<b>Figura 7.46 Allocation list .....</b>	<b>72</b>
<b>Figura 7.47 Secuencia para un nuevo programa.....</b>	<b>73</b>
<b>Figura 7.48 Características del programa .....</b>	<b>73</b>
<b>Figura 7.49 Programa en blanco .....</b>	<b>73</b>
<b>Figura 7.50 Barra de herramientas .....</b>	<b>74</b>
<b>Figura 7.51 Contacto en FST4.10 .....</b>	<b>74</b>
<b>Figura 7.52 Determinación de contacto.....</b>	<b>75</b>
<b>Figura 7.53 Asignación de nombre de la bobina .....</b>	<b>75</b>
<b>Figura 7.54 Programa de encendido y apagado de un led.....</b>	<b>76</b>
<b>Cuadro 7.1 Tabla de verdad .....</b>	<b>76</b>
<b>Figura 7.55 Compilación del programa (a), (b), (c).....</b>	<b>77</b>
<b>Figura 7.56 Cuadro de dialogo .....</b>	<b>77</b>
<b>Figura 7.57 Descarga del programa en PLC .....</b>	<b>77</b>
<b>Figura 7.58 Ejecución de programa Online .....</b>	<b>78</b>
<b>Figura 7.59 Estados posibles del PLC .....</b>	<b>78</b>

<b>Figura 7.60</b>	<b>Funcionamiento y ejecución del programa cargado en el PLC</b>	<b>79</b>
<b>Figura 7.61</b>	<b>Programa AND y OR</b>	<b>81</b>
<b>Figura 7.62</b>	<b>Contactos en serie, compuerta AND</b>	<b>82</b>
<b>Figura 7.63</b>	<b>Tipos de contactos, e inversor</b>	<b>83</b>
<b>Figura 7.64</b>	<b>Contactos en paralelo, compuerta OR</b>	<b>83</b>
<b>Figura 7.65</b>	<b>Salidas de las diferentes compuertas</b>	<b>83</b>
<b>Figura 7.66</b>	<b>Iconos de compilacion</b>	<b>84</b>
<b>Figura 7.67</b>	<b>Botón de descarga de programa al PLC</b>	<b>84</b>
<b>Figura 7.68</b>	<b>Cuadro de dialogo</b>	<b>85</b>
<b>Figura 7.69</b>	<b>Descarga del programa en PLC</b>	<b>85</b>
<b>Figura 7.70</b>	<b>Ejecución de programa Online</b>	<b>85</b>
<b>Figura 7.71</b>	<b>programa AND y OR en linea primer caso</b>	<b>86</b>
<b>Figura 7.72</b>	<b>Respuesta al primer caso de las compuertas AND y OR</b>	<b>86</b>
<b>Figura 7.73</b>	<b>Programa en linea segundo caso</b>	<b>87</b>
<b>Figura 7.74</b>	<b>Respuesta al segundo caso de las compuertas AND y OR</b>	<b>88</b>
<b>Figura 7.75</b>	<b>Programa en linea tercer caso</b>	<b>89</b>
<b>Figura 7.76</b>	<b>Respuesta al tercer caso de las compuertas AND y OR</b>	<b>89</b>
<b>Figura 7.77</b>	<b>Programa en linea cuarto caso</b>	<b>90</b>
<b>Figura 7.78</b>	<b>Respuesta al cuarto aso de las compuertas AND y OR</b>	<b>91</b>
<b>Figura 7.79</b>	<b>Programas NAND y NOR</b>	<b>92</b>
<b>Figura 7.80</b>	<b>Contactos con inversor en paralelo, compuerta NAND</b>	<b>93</b>
<b>Figura 7.81</b>	<b>Contactos con inversor en serie, compuerta NOR</b>	<b>94</b>
<b>Figura 7.82</b>	<b>Salidas de las diferentes compuertas</b>	<b>94</b>
<b>Figura 7.83</b>	<b>Iconos de compilacion</b>	<b>95</b>
<b>Figura 7.84</b>	<b>Botón de descarga de programa al PLC</b>	<b>95</b>
<b>Figura 7.85</b>	<b>Cuadro de dialogo</b>	<b>95</b>
<b>Figura 7.86</b>	<b>Descarga del programa en PLC</b>	<b>96</b>
<b>Figura 7.87</b>	<b>Ejecución de programa Online</b>	<b>96</b>

<i>Figura 7.88 Programa NAND y NOR en línea primer caso .....</i>	<i>97</i>
<i>Figura 7.89 Respuesta al primer caso de las compuertas NAND y NOR.....</i>	<i>97</i>
<i>Figura 7.90 Programa en línea segundo caso .....</i>	<i>98</i>
<i>Figura 7.91 Respuesta al segundo caso de las compuertas NAND y NOR.....</i>	<i>99</i>
<i>Figura 7.92 Programa en línea tercer caso .....</i>	<i>100</i>
<i>Figura 7.93 Respuesta al tercer caso de las compuertas NAND y NOR.....</i>	<i>100</i>
<i>Figura 7.94 Programa en línea cuarto caso .....</i>	<i>101</i>
<i>Figura 7.95 Respuesta al cuarto caso de las compuertas AND y OR .....</i>	<i>102</i>
<i>Figura 7.96 Programa de puerta de garaje.....</i>	<i>104</i>
<i>Figura 7.97 Iconos de compilacion .....</i>	<i>105</i>
<i>Figura 7.98 Botón de descarga de programa al PLC.....</i>	<i>105</i>
<i>Figura 7.99 Cuadro de dialogo .....</i>	<i>106</i>
<i>Figura 8.0 Descarga del programa en PLC .....</i>	<i>106</i>
<i>Figura 8.1 Ejecución de programa Online .....</i>	<i>106</i>
<i>Figura 8.2 Programa en línea puerta de garaje .....</i>	<i>107</i>
<i>Figura 8.3 Puerta abierta desde afuera .....</i>	<i>108</i>
<i>Figura 8.4 Puerta abierta desde afuera con limit switch activo .....</i>	<i>108</i>
<i>Figura 8.5 Respuesta en modulo de simulacion de sensores .....</i>	<i>109</i>
<i>Figura 8.6 Puerta cerrada desde adentro .....</i>	<i>109</i>
<i>Figura 8.7 Puerta cerrada desde adentro con limit switch activo .....</i>	<i>110</i>
<i>Figura 8.8 Respuesta en el modulo de simulacion.....</i>	<i>110</i>
<i>Figura 8.9 Puerta abierta desde adentro.....</i>	<i>111</i>
<i>Figura 8.10 Puerta abierta desde adentro con limit swicht activo .....</i>	<i>111</i>
<i>Figura 8.11 Respuesta en el modulo de simulacion.....</i>	<i>112</i>
<i>Figura 8.12 Cerrar puerta desde afuera.....</i>	<i>112</i>
<i>Figura 8.13 Cerrar puerta desde afuera con limit switch activo.....</i>	<i>113</i>
<i>Figura 8.14 Respuesta en el modulo de simulacion.....</i>	<i>113</i>
<i>Figura 8.15 Programa de encendido de motores .....</i>	<i>115</i>

<i>Figura 8.16 Iconos de compilacion .....</i>	<i>117</i>
<i>Figura 8.17 Botón de descarga de programa al PLC.....</i>	<i>117</i>
<i>Figura 8.18 Cuadro de dialogo .....</i>	<i>118</i>
<i>Figura 8.19 Descarga del programa en PLC .....</i>	<i>118</i>
<i>Figura 8.20 Ejecución de programa Online .....</i>	<i>118</i>
<i>Figura 8.21 Programa de encendido de motores en linea .....</i>	<i>119</i>
<i>Figura 8.22 Modulo de simulacion de sensores con variables definidas.....</i>	<i>119</i>
<i>Figura 8.23 Programa en linea temperatura 1 activa .....</i>	<i>120</i>
<i>Figura 8.24 Respuesta del modulo de simulacion temperatura 1 .....</i>	<i>121</i>
<i>Figura 8.25 Programa en linea temperatura 1 y 2 activa .....</i>	<i>122</i>
<i>Figura 8.26 Respuesta de modulo de simulacion temperatura 1 y 2.....</i>	<i>122</i>
<i>Figura 8.27 Programa en linea temperatura 1,2 y 3 activo .....</i>	<i>123</i>
<i>Figura 8.28 Respuesta de modulo de simulacion temperatura 1,2 y 3.....</i>	<i>123</i>
<i>Figura 8.29 Circuito en Protoboard .....</i>	<i>124</i>
<i>Figura 8.30 (a) Circuito diseñado en Ares 7 Profesional, (b) circuito en 3D, (c) circuito montado en placa fenólica. ....</i>	<i>126</i>
<i>Figura 8.31 Tablero de Automatización del Departamento de Maquinaria Agrícola....</i>	<i>126</i>

## **INDICE DE CUADROS**

<i>Cuadro 4.1 Tabla de verdad del programa encendido y apagado de un led.....</i>	<i>29</i>
<i>Cuadro 4.2 Tabla de verdad de puertas lógicas .....</i>	<i>30</i>
<i>Cuadro 4.3 Operación del programa puerta de garaje.....</i>	<i>33</i>
<i>Cuadro 4.4 Variables del programa puerta de garaje .....</i>	<i>33</i>
<i>Cuadro 4.5 Tabla de verdad encendido de motores .....</i>	<i>36</i>
<i>Cuadro 7.1 Tabla de verdad.....</i>	<i>76</i>
<i>Cuadro 7.2 Tabla de verdad de compuertas logicas.....</i>	<i>80</i>
<i>Cuadro 7.3 Tabla de verdad puertas logicas AND y OR .....</i>	<i>81</i>

<b><i>Cuadro 7.4 Variables de entradas y salidas de señal.....</i></b>	<b><i>82</i></b>
<b><i>Cuadro 7.5 Respuesta primer caso .....</i></b>	<b><i>87</i></b>
<b><i>Cuadro 7.6 Respuestas segundo caso .....</i></b>	<b><i>88</i></b>
<b><i>Cuadro 7.7 Respuestas tercer caso .....</i></b>	<b><i>90</i></b>
<b><i>Cuadro 7.8 Respuestas cuarto caso .....</i></b>	<b><i>91</i></b>
<b><i>Cuadro 7.9 Tabla de verdad puertas logicas NAND y NOR .....</i></b>	<b><i>92</i></b>
<b><i>Cuadro 7.10 Variables de entradas y salidas de señal.....</i></b>	<b><i>93</i></b>
<b><i>Cuadro 7.11 Respuesta primer caso .....</i></b>	<b><i>98</i></b>
<b><i>Cuadro 7.12 Respuestas segundo caso .....</i></b>	<b><i>99</i></b>
<b><i>Cuadro 7.13 Respuestas tercer caso .....</i></b>	<b><i>101</i></b>
<b><i>Cuadro 7.14 Respuestas cuarto caso .....</i></b>	<b><i>102</i></b>
<b><i>Cuadro 7.15 Operación del programa puerta de garaje.....</i></b>	<b><i>103</i></b>
<b><i>Cuadro 7.16 Variables del programa puerta de garaje .....</i></b>	<b><i>103</i></b>
<b><i>Cuadro 7.17 Variables de encendido de motores.....</i></b>	<b><i>116</i></b>
<b><i>Cuadro 7.18 Tabla de verdad del programa encendido de motores .....</i></b>	<b><i>116</i></b>
<b><i>Cuadro 7.19 Tabla de verdad primer caso .....</i></b>	<b><i>120</i></b>
<b><i>Cuadro 7.20 Tabla de verdad temperatura 1 activa.....</i></b>	<b><i>121</i></b>
<b><i>Cuadro 7.21 Tabla de verdad temperatura 1 y 2 activas .....</i></b>	<b><i>122</i></b>
<b><i>Cuadro 7.22 Temperaturas 1,2 y 3 activas.....</i></b>	<b><i>124</i></b>

## RESUMEN

El presente trabajo fue realizado en las instalaciones del Departamento de Maquinaria Agrícola de la Universidad Autónoma Agraria Antonio Narro, desarrollándose un manual de prácticas para el control automático de temperatura mediante microcontroladores y PLC, se emplearon microcontroladores PIC16F84A de la marca MICROCHIP, tres termostatos electrónicos, un PLC Festo FC34, módulo de simulación de sensores. El proceso de programación para microcontroladores y PLC se desarrolló en ambientes de MPlab v8.10 y FST 4.10. Las practicas consisten desde encender y apagar un led, hasta la aplicación del encendido de motores para el control automático de temperatura, las cuales son en base a ecuaciones de algebra booleana.

El presente trabajo está integrado por 4 practicas, la primera consiste en la familiarización en ambiente FST 4.10 y programación del PLC, la segunda consiste en verificar el comportamiento de compuertas lógicas AND, OR, NAND y NOR, la tercera es automatizar la apertura y cerrado de una puerta de garaje la última practica consiste en el encendido de motores mediante tres termostatos electrónicos. Las practicas que se implementaron operaron satisfactoriamente logrando los objetivos correspondientes.

Además se realizó una tarjeta de relevadores y transistores para operar a diferentes voltajes (5V, 24V y 110V) para la comunicación entre los termostatos y el PLC así como también para la activación de dos motores.

**Palabras clave:** *PLC, PIC16F84A, Automatización, Sistemas de control*

## I INTRODUCCION

El control automático de procesos es una de las disciplinas que se ha desarrollado a una velocidad vertiginosa, dando las bases a lo que hoy algunos autores llaman la segunda revolución industrial.

Su estudio y aplicación ha contribuido al reconocimiento universal de sus ventajas y beneficios asociados al ámbito industrial, que es donde tiene una de sus mayores aplicaciones debido a la necesidad de controlar un gran número de variables, sumado esto a la creciente complejidad de los sistemas. El control automático de procesos se usa fundamentalmente porque reduce el costo asociado a la generación de bienes y servicios, incrementa la calidad y volúmenes de producción de una planta industrial entre otros beneficios asociados con su aplicación. (Abarca, 2013)

Hoy en día, las necesidades de automatización en la industria también se ha reflejado en el campo y mecanización agrícola en tractores, cosechadoras, etc., así también incrementado día con día, la necesidad de controlar variables como riego, conductividad eléctrica, luminosidad, temperatura entre otras.

Un claro ejemplo de un sistema de automatización de procesos, es una seleccionadora de tomate la cual mediante sensores separa el producto por peso, tamaño, color, etc. agilizando la selección y mejorando los tiempos, así también como mano de obra, que son los principales objetivos, al igual que un sistema de fertirriego prendiendo y apagando el sistema, o en un sistema de ventilación de un invernadero el cual mantiene una temperatura deseada dentro del mismo, buscando mejorar las condiciones de clima para el desarrollo del cultivo.

En el presente trabajo se busca crear un manual para el control automático de temperatura mediante sensores, microcontroladores y PLC que permita su automatización tanto en el laboratorio como la aplicación en campo. Se aplicara la teoría de algebra booleana, para activar 2 motores a través de tres termostatos previamente calibrados en rangos de temperatura de 20 a 60 grados centígrados

## **Objetivos**

Elaborar un manual que permita controlar automáticamente la operación de ventiladores para el control de temperatura mediante microcontroladores y PLC.

## **Hipótesis**

Es posible diseñar un sistema de control automático de temperatura con fines didácticos y para la aplicación en campo.

## II REVISIÓN DE LITERATURA

### 2.1 Sistemas de control

Los sistemas de control automático en lo que consta a su operación, no tiene demasiada complejidad, algunos autores lo definen como algo esencial e indiscutiblemente primordial en nuestra sociedad, Navarro, 2004 lo define como un conjunto formado por el proceso y el sistema de control. Para conseguir este objetivo son indispensables una serie de dispositivos adicionales, que de alguna manera lleven a cabo el control o regulación del proceso con el fin de sustituir la acción del hombre por un dispositivo llamado controlador o regulador. El conjunto que forman los componentes que llevarán a cabo el control automático de un proceso se le llama sistema de control automático.

Acedo, 2006, define a un sistema de control automático el cual mide una variable y actúa de una forma determinada para que esa variable se mantenga en un valor deseado de referencia. Un sistema de control no automático, puede hacer cambiar el valor de la variable pero no llevarla al valor de referencia.

En la actualidad existen plantas industriales que solo requieren para su operación de unas cuantas personas que supervisen sus procesos de producción, los cuales se realizan automáticamente bajo el control de sistemas computarizados. Estos sistemas efectúan el control directo de las variables, ajustan los parámetros de acuerdo a las condiciones de operación, informan a los operadores de las condiciones anormales de la planta y aun sugieren las acciones de corrección en caso de falla.

El desarrollo de los sistemas de control automático se ha logrado en poco tiempo ya que podemos considerar que este se inició, como conocimiento estructurado en la década de 1920. (Cavada, 1998)

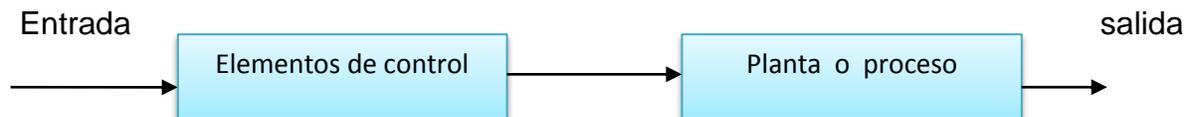
Acedo, 2006, hace referencia sobre los controladores automáticos en donde han incorporado una habilidad que corresponde al ser humano. Esa habilidad es la de

(sentir) una condición correcta o incorrecta y actuar para corregirla. Las ventajas que se derivan porque una maquina realice algunas funciones que hace al hombre son obvias, sobre todo cuando la maquina llega a superar al hombre en esa función. Un controlador automático no se cansa, no necesita dormir, ni sale de vacaciones; siempre está al cuidado del trabajo encomendado.

## 2.2 Tipos de sistemas de control

### 2.2.1 Sistema en lazo abierto

Ogata, 2003. Son los sistemas en los cuales la salida no tiene efecto sobre la acción de control. En otras palabras, en un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada. En cualquier sistema de control en lazo abierto, la salida no se compara con la entrada de referencia. Así, a cada entrada de referencia le corresponde una condición de operación fija; como resultado de ello, la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones, un sistema de control de lazo abierto no realiza la tarea deseada Figura (2.1).



*Figura 2.1 Sistema de control en lazo abierto. Fuente: Ogata 2003*

### 2.2.2 Sistema en lazo cerrado

Los sistemas de control realimentados se denominan también sistemas de control en lazo cerrado. En la practica, los términos control realimentado y control en lazo cerrado se usan indistintamente. En un sistema de control en lazo cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivadas y/o integrales), con el fin de reducir el error y llevar la salida del sistema a un valor deseado. El termino

control en lazo cerrado siempre implica el uso de una acción de control realimentada para reducir el error del sistema. (Ogata, 2003)

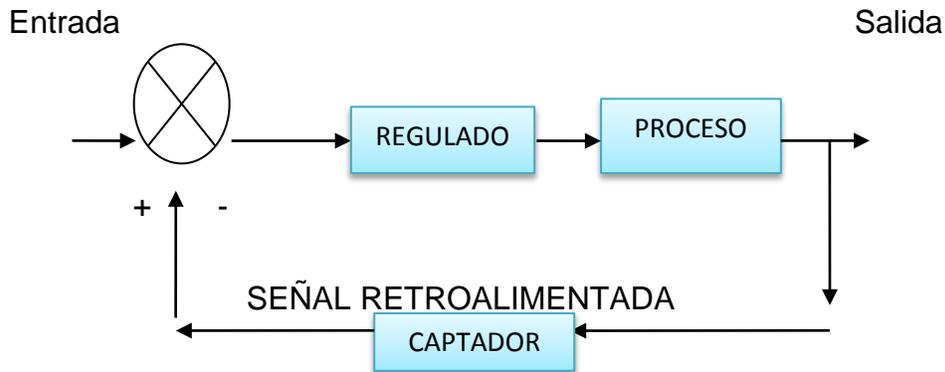


Figura 2. 1 Sistema en lazo cerrado. Fuente: Ogata 2003.

### 2.3 Automatización

La palabra automático se deriva del griego y significa automotor o auto pensante. La palabra automatización se acuñó para indicar aspectos de manufactura en los que la producción, el movimiento y la inspección se realizan o controlan por máquinas que se operan a sí mismas sin la intervención humana. (Schey, 2002)

Se define la automatización, por lo general, como el proceso de hacer que las máquinas sigan un orden predeterminado de operaciones con poca o ninguna mano de obra, usando equipo y dispositivos especializados que ejecutan y controlan los procesos de manufactura. La automatización, en todo su potencial, se logra usando diversos dispositivos, sensores, actuadores, técnicas y equipo capaces de observar y controlar todos los aspectos del proceso de manufactura, de tomar decisiones acerca de los cambios que se deben hacer en la operación y de controlar todos los aspectos de esta. (Kalpakjian y Schmid, 2002)

La Real Academia de las Ciencias Físicas y Exactas define la automática como el conjunto de métodos y procedimientos para la sustitución del operario en tareas físicas y mentales previamente programadas. De esta definición original se

desprende la definición de la automatización como la aplicación de la automática al control de procesos industriales. (Ponsa y Villnova, 2005)

A mediados de la década de 1940, la industria automotriz en Estados Unidos propuso la palabra automatización para indicar el manejo y proceso automáticos de las partes entre las máquinas de producción. Aunque no existe una definición precisa, por lo general la automatización significa la metodología y el sistema de operación de una máquina o proceso por medios altamente automáticos; proviene del término griego autómatas, que significa "de accionamiento propio".

Aunque pueden existir variaciones en la definición, la automatización suele definirse como el proceso de habilitación de las máquinas para seguir una secuencia predeterminada de operaciones con poca o ninguna mano de obra humana, utilizando equipo especializado y dispositivos que realizan y controlan los procesos de manufactura. La automatización total se logra mediante diversos dispositivos, sensores, actuadores, técnicas y equipo que tienen la capacidad de (a) supervisar todos los aspectos de la operación de manufactura; (b) tomar decisiones respecto de los cambios que deben hacerse en la operación, y (c) controlar todos sus aspectos. (Instituto Tecnológico Superior de Calkiní en el Estado de Campeche)

## **2.4 PLC**

Un autómatas programable (PLC o Programmable Logic Controller) es un sistema de control basado en un microprocesador y los elementos necesarios para que este microprocesador opere de forma conveniente.

Al estar basado en un microprocesador, permite que la función que el PLC realice sea programable por cada usuario a efectos de satisfacer cada necesidad concreta de control, lo que le convierte en una herramienta sumamente útil y flexible. Su desarrollo a lo largo del tiempo ha experimentado sucesivas mejoras en el sentido de aumentar sus prestaciones, unas prestaciones que han sido

consecuencias directas del avance en la tecnología y prestaciones de los microprocesadores. (Domingo, *et al*, 2003)

El PLC es un dispositivo de estado sólido, diseñado para controlar procesos secuenciales (una etapa después de la otra) que se ejecutan en un ambiente industrial. Es decir, que van asociados a la maquinaria que desarrolla procesos de producción y controlan su trabajo.

Como se puede deducir de la definición, el PLC es un sistema, porque contiene todo lo necesario para operar, y es industrial, por tener todos los registros necesarios para operar en los ambientes hostiles que se encuentran en la industria. (Instituto Nacional de Tecnologías Educativas y de Profesorado)

El PLC según la Norma IEC-1131 de lenguajes de programación, es un sistema electrónico de funcionamiento digital, diseñado para ser utilizado en un entorno industrial, que utiliza una memoria programable para el almacenamiento interno de instrucciones orientadas al usuario, para la realización de funciones de: enlaces lógicos, secuenciación, temporización, recuento y cálculo, para controlar a través de entradas y salidas digitales o analógicas, diversos tipos de máquinas o procesos. (IEC, 1992)

#### **2.4.1 Clasificación de PLC.**

##### **1. PLC Nano:**

Generalmente es un PLC de tipo compacto, Figura (2.3) que puede manejar un conjunto reducido de entradas y salidas, generalmente en un número inferior a 100. Este PLC permite manejar entradas y salidas digitales y algunos módulos especiales.



*Figura 2.3 PLC tipo Nano*

*Fuente: <http://www.esuppliersindia.com/emc-automation-pvt-ltd-/nano-plc-pr410779-sCATALOG-swf.html>*

## 2. PLC Compacto

El PLC compacto, Figura (2.4) tienen incorporada la fuente de alimentación, su CPU y los módulos de entrada y salida en un solo módulo principal y permiten manejar desde unas pocas entradas y salidas hasta varios cientos (alrededor de 500 entradas y salidas), su tamaño es superior a los PLC tipo Nano y soportan una gran variedad de módulos especiales, tales como:

- Entradas y salidas análogas
- módulos contadores rápidos
- módulos de comunicaciones
- interfaces de operador
- expansiones de entrada y salida



*Figura 2.4 PLC tipo compacto*

*Fuente: <http://www.directindustry.es/prod/festo/automatas-programables-4735-266911.html>*

### 3. PLC Modular:

El PLC modular, Figura (2.5) se componen de un conjunto de elementos que conforman el controlador final. Estos son:

- El Rack
- La fuente de alimentación
- La CPU
- Los módulos de entrada y salida

De estos tipos de PLC existen desde los denominados Micro-PLC que soportan gran cantidad de entradas y salida, hasta los PLC de grandes prestaciones que permiten manejar miles de entradas y salidas. (Instituto Nacional de Tecnologías Educativas y de Profesorado)



*Figura 2.5 PLC tipo modular*

Fuente: <http://www.directindustry.es/prod/eaton-moeller/automatas-programables-modulares-5067-1099317.html>

## 2.4.2 Tipos de programación de PLC

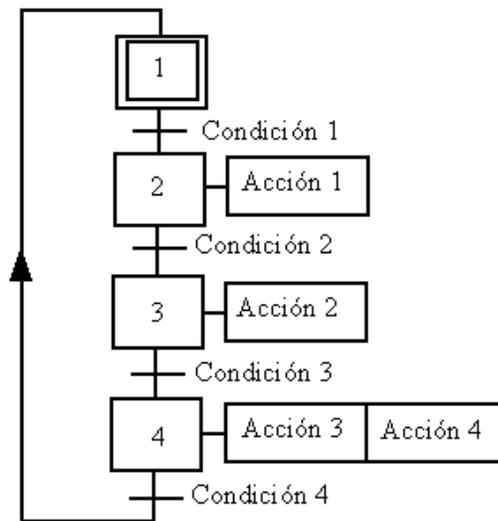
La incrementada complejidad en la programación de los autómatas programables requiere más que nunca de la estandarización de la misma. Bajo la dirección del IEC el estándar IEC 1131-3 (IEC 65) para la programación de PLC's ha sido definida. Alcanzó el estado de Estándar Internacional en Agosto de 1992. Los lenguajes gráficos y textuales definidos en el estándar son una fuerte base para entornos de programación potentes en PLC's. Con la idea de hacer el estándar adecuado para un gran abanico de aplicaciones, cinco lenguajes han sido definidos en total. (Autómatas Industriales)

- Grafico secuencial de funciones (Grafcet)
- Lista de instrucciones (LDI o AWL)
- Texto estructura
- Diagrama de contactos
- Diagrama de funciones

## 1. Gráfico secuencial de funciones (Grafcet)

El gráfico secuencial de funciones (SFC o Grafcet), Figura (2.6) es un lenguaje gráfico que proporciona una representación en forma de diagrama de las secuencias del programa. Soporta selecciones, alternativas de secuencias paralelas. Los elementos básicos son pasos y transiciones.

Los pasos consisten de piezas de programa que son inhibidas hasta que una condición especificada por las transiciones es conocida. Como consecuencia de que las aplicaciones industriales funcionan en forma de pasos, el SFC es la forma lógica de especificar y programar el más alto nivel de un programa de PLC.



*Figura 2.6 Diagrama Grafcet*

*Fuente: <http://es.wikipedia.org/wiki/GRAFCET>*

## 2. Lista de instrucciones

La lista de instrucciones (IL o AWL), Figura (2.7) es un lenguaje de bajo nivel, similar al lenguaje ensamblador. Con IL solo una operaciones permitida por línea (Ej. Almacenar un valor en un registro).

Este lenguaje es adecuado para pequeñas aplicaciones y para optimizar partes de una aplicación.

000	LD	%I0.1	Bp. inicio ciclo
	AND	%I0.0	Dp. presencia vehiculo
	AND	%M3	Bit autorización reloj calendario
	AND	%I0.5	Fc. alto rodillo
	AND	%I0.4	Fc. detrás pórtico
005	S	%M0	Memo inicio ciclo
	LD	%M2	
	AND	%I0.5	
	OR	%I0.2	Bp. parada ciclo
	R	%M0	
010	LD	%M0	
	ST	%Q0.0	Piloto ciclo

Figura 2.7 Diagrama de lista de instrucciones,

Fuente: <http://www.automatas.org/software.htm>

### 3. Texto estructurado

El texto estructurado (ST), Figura (2.8), es un lenguaje de alto nivel estructurado por bloques que posee una sintaxis parecida al pascal. El ST puede ser empleado para realizar rápidamente sentencias complejas que manejen variables con un amplio rango de diferentes tipos de datos, incluyendo valores analógicos y digitales.

```
Programa en texto estructurado para un PLC marca Telemecanique ISA-07  
  
LD [%MW10>100]  
ST %Q0.3  
AND [%MW20<%MW35]  
ST %Q0.2  
LD %I0.2  
OR [%MW30>=%MW40]  
ST %Q0.4
```

Figura 2.8 Texto estructurado

Fuente: <http://automatismoyk.blogspot.mx/2010/05/lenguajes-de-programacion.html>

#### 4. Diagrama de contactos

El diagrama de contactos (LD), Figura (2.9), es un lenguaje que utiliza un juego estandarizado de símbolos de programación. En el estándar IEC los símbolos han sido racionalizados (se ha reducido su número).

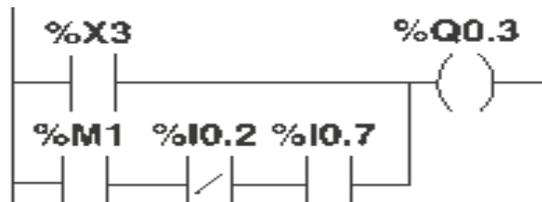


Figura 2.9 Diagrama de contactos

Fuente: <http://edublogmicros.blogspot.mx/p/grafcet.html>

#### 5. Diagrama de funciones

El diagrama de funciones (function block diagram o FBD), Figura (2.10), es un lenguaje gráfico que permite programar elementos que aparecen como bloques para ser cableados entre sí de forma análoga al esquema de un circuito. Es adecuado para muchas aplicaciones que involucren el flujo de información o datos entre componentes de control. (Aguilera, 2002)

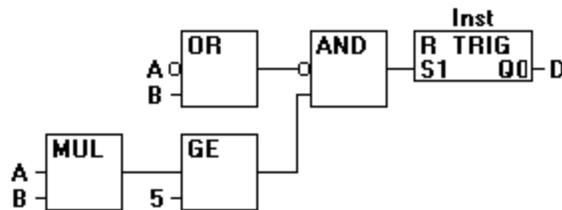


Figura 2.10 Diagrama de funciones

Fuente: [http://infosys.beckhoff.com/espanol.php?content=../content/1034/tcplccontrol/html/tcplcctrl\\_languages%20fbd.htm&id=](http://infosys.beckhoff.com/espanol.php?content=../content/1034/tcplccontrol/html/tcplcctrl_languages%20fbd.htm&id=)

## **2.5 Sensores**

Pallás (1993), menciona que un sensor es un elemento de un sistema que lo conecta con su entorno físico, excluyendo al usuario. La función de los sensores es obtener señales electrónicas en respuesta a magnitudes de entradas no eléctricas, existen varios tipos de sensores pero los más comunes son los analógicos y digitales.

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una Tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc. (Wikipedia)

### **2.5.1 Tipos de sensores.**

Analógicos, en este tipo de sensores la salida varía, a nivel macroscópico, de forma continua. La información está en la amplitud, si bien se suelen incluir en este grupo los sensores con salida en el dominio temporal. Digitales, la salida varía en forma de saltos o pasos discretos. No requiere conversión A/D (analógicos/digital) y la transmisión de su salida es más fácil. Tienen mayor fidelidad y mayor fiabilidad, y muchas veces exactitud, pero lamentablemente no hay modelos digitales para muchas de las magnitudes físicas de mayor interés, que en muchas ocasiones son las más requeridas (Pallás, 2003).

## **2.6 Microcontrolador**

Un microcontrolador de fábrica, no realiza tarea alguna, este debe ser programado para que realice desde un simple parpadeo de un led hasta un sofisticado control de un robot. Un microcontrolador es capaz de realizar la tarea de muchos circuitos

lógicos como compuertas AND, OR, NOT, NAND, conversores A/D, D/A, temporizadores, decodificadores, etc. Simplificado todo el diseño a una placa de reducido tamaño y pocos elementos. (Reyes, 2006)

Es un circuito integrado digital monolítico que contiene todos los elementos de un procesador digital secuencial síncrono programable de arquitectura Harvard o Princeton (Von Neumann). Se le suele denominar también microcomputador integrado o empotrado (embedded processor) y está especialmente orientado a tareas de control y comunicaciones. (Mandado, *et al* 2007)

## III MATERIALES, EQUIPO Y MÉTODO

### 3.1 Materiales y equipo

Los equipos y software empleados, fueron los siguientes:

- PLC Festo FC34
- Termostato Electrónico
- Sensor DS 18B20
- PIC16F84A
- Módulo de simulación
- Programador PICSTAR Plus
- Software Festo FST V4.10.50
- Software MPLab IDE V8.10
- Software Proteus V7.9.

A continuación se describen brevemente los equipos empleados.

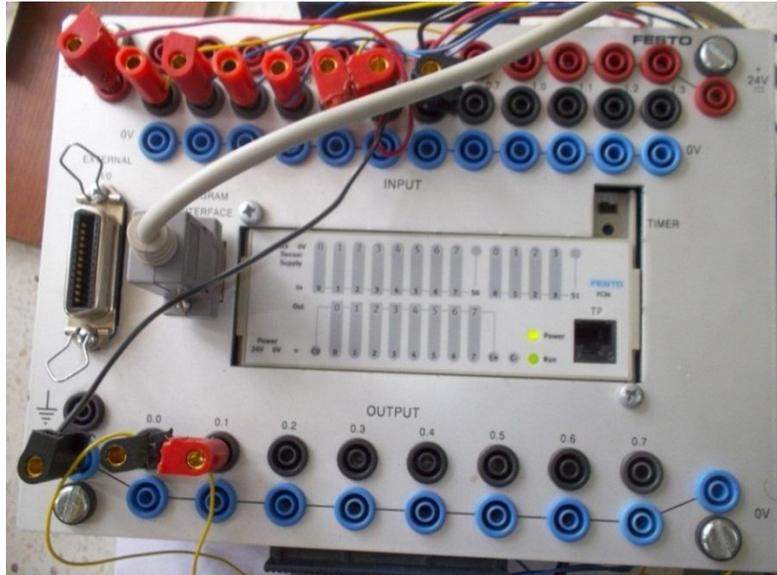
#### 3.1.1 PLC Festo FC34

El PLC FC34 es de tipo compacto el cual se muestra en la figura (3.1) se programa utilizando FST.

FST es una programación única lenguaje que es rica en tradición y muy fácil de usar, lo que permite "La programación de la forma de pensar":

*IF... THEN... OTHERWISE*

FST también soporta la operación STEP programación de secuencias. FST puede ser utilizado para la programación a través de Ethernet; un servidor web también está disponible. Las características del PLC se encuentran el Anexo 7.1, (Festo AG&Co.KG)



*Figura 3.1 PLC Festo FC34*

### **3.1.2 Termostato electrónico**

El programa de control para un termómetro y termostato digital. Utiliza el sensor de temperatura DS18B20 que transmite la información vía serie a través de un bus de una sola línea según un protocolo del fabricante de semiconductores Dallas Semiconductors.

El ajuste de la temperatura a la que conmuta el termostato se logra mediante tres pulsadores, y son: "MODO" e "INCREMENTAR", además un pulsador de "RESET" para reinicio del programa, que se conectan a pines del Puerto B y cuyo funcionamiento se basa en interrupción por cambio en la línea del Puerto B. (López, 2010), Anexo 7.2

### **3.1.3 Sensores DS-18B20**

El termómetro digital DS1820, Figura (3.2), es un pequeño sensor de temperatura de 3 pines que se asemeja a un transistor. Cada uno tiene un único "código de 64 bits de serie" y pueden comunicarse a través de un "1 Wire (alambre)" estándar,

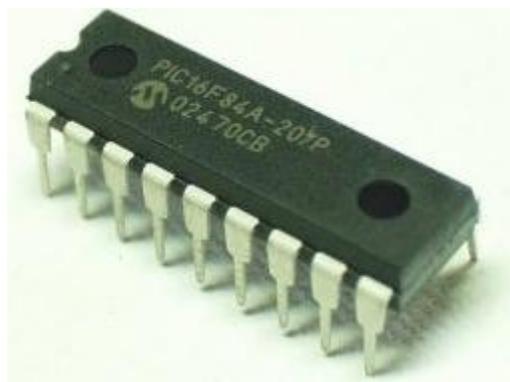
para conectarse al microcontrolador y utilizarlos en los sistemas de control. Este sensor se conecta a las diferentes placas de control.



*Figura 3.2 Sensor DS18B20*

### **3.1.4 PIC16F84A**

El PIC16F84A, Figura (3.3) es el motor de las placas de control se debe principalmente a este microcontrolador, que lleva la programación necesaria para controlar el sistema, con instrucciones lógicas, programadas previamente.



*Figura 3.3 Microcontrolador PIC16F84A.*

Microchip Technology Inc 2001, nos especifican su composición exterior de 8 bits, 18 pines, y un set de instrucciones RISC muy amigable para memorizar y fácil de entender.

### 3.1.5 Módulo de simulación de sensores

El módulo de simulación, Figura (3.4) servirá para representar diferentes ejercicios que se realizaran, enviando señal (24 V) al PLC, así también recibiendo la señal del mismo. Está constituido por leds, que indicaran el estado de la señal de entrada o de salida, la activación de la señal será mediante switches, que arrojará el voltaje de alimentación de 24 volts hacia el PLC.

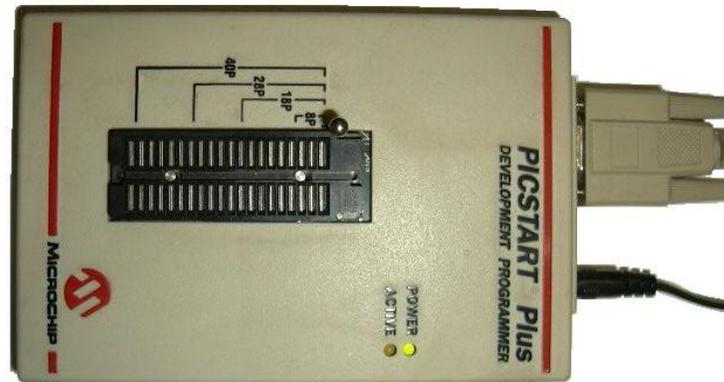


*Figura 3.4 Módulo de simulación de sensores*

### 3.1.6 Programador PICSTAR Plus

PICSTART Plus de Microchip, Figura (3.5) es un programador de desarrollo. Se conecta a través del puerto serie RS-232 a su PC y se opera con el paquete de Microchip Entorno de desarrollo integrado de software llamado MPLAB® IDE que se incluye con el programador de forma gratuita. El programador compatible con la mayoría de los productos microcontroladores DIP-empaquetados disponibles de Microchip. Es una herramienta complementaria de MPLab, para programar los

microcontroladores, el cual utilizamos, para la programación del microcontrolador (Vázquez, 2011) Anexo 7.3



*Figura 3.5 Programador PICSTAR Plus.*

### **3.1.7 Festo FST V4.10.50**

Festo FST 4 es un compilador y programador de PLC's es muy sencillo y fácil de usar, tiene la posibilidad de adoptar diferentes métodos de programación como son la lista de instrucciones y la programación tipo ladder.

Configurar, parametrizar, programar y diagnosticar, el sistema de programación FST ofrece soporte, además de la programación final de los controladores FEC Compact, FEC Standard y CPXFEC, inclusive para el proceso de desarrollo. (Festo AG&Co.KG)

### **3.1.8 MPLab IDE V8.10**

MPLAB es un ambiente de desarrollo integrado (IDE, por sus siglas en inglés) creado por Microchip Technology para dar soporte a la realización de proyectos basados en microcontroladores PIC. Incluye varios módulos que permiten llevar a

cabo las distintas etapas de un proyecto: Edición, Ensamblaje, Simulación y Programación.

### **3.1.9 Proteus V7.9.**

Proteus es un entorno de desarrollo CAD de la empresa Labcenter Electronics Ltd, el cual integra diversas herramientas de software para las tareas más comunes en el desarrollo de proyectos electrónicos tales como: captura de esquemáticos, fabricación de circuitos impresos y simulación basada en PSICE.

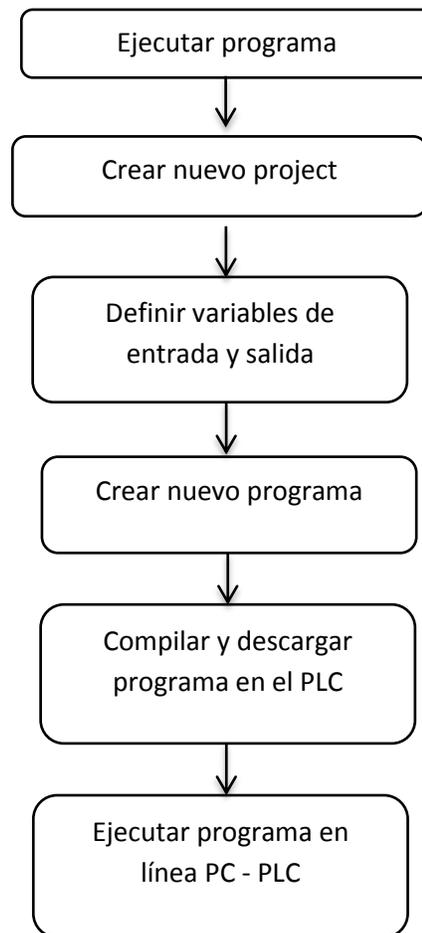
**ISIS** (“Intelligent Schematic Input System”). El cual es módulo de captura de diagramas esquemáticos.

**ARES** (“Advanced Routing and Editing Software”). El cuál es el modulo para la realización de circuitos impresos o PCB.

## 3.2 Método

### 3.2.1 Programación de PLC

Para la programación del PLC mediante FST 4.10, se diseñaron los programas en base a la programación de diagramas de contactos (ladder diagram), para lo cual se siguió el proceso mostrado en la Figura (3.6)



*Figura 3.6 Diagrama de bloques para la creación de programas en FST 4.10*

Al finalizar el proceso de la Figura (3.6) el programa se ejecuta mediante el módulo de simulación de sensores, y la PC en línea mediante una conexión del puerto serial, para verificar el correcto funcionamiento del programa. Anexo 7.4

### 3.2.2 Programación de sensores de temperatura

La programación de sensores, se establecerá una temperatura deseada según las opciones del programa grabado en el PIC16F84A de la placa del termostato electrónico, las opciones que nos muestra las placas de control de temperatura servirá para la activación, mediante una señal que arrojará para la activación del PLC, el cual tendrá un programa previamente cargado para el accionamiento de los motores, y así tener un control de temperatura deseado. Anexo 7.2

### 3.2.3 Programación de PLC-UAAAN

Es una tarjeta construida donde el principal componente para la operación es mediante un PIC16F84A, que contiene un programa para el funcionamiento del PLC-UAAAN, y mediante un programa que será cargado a través de un proceso donde se utilizara MPLab v8.10, siguiendo los pasos como se muestra en la Figura (3.7), el detalle se encuentra en el Anexo 7.3

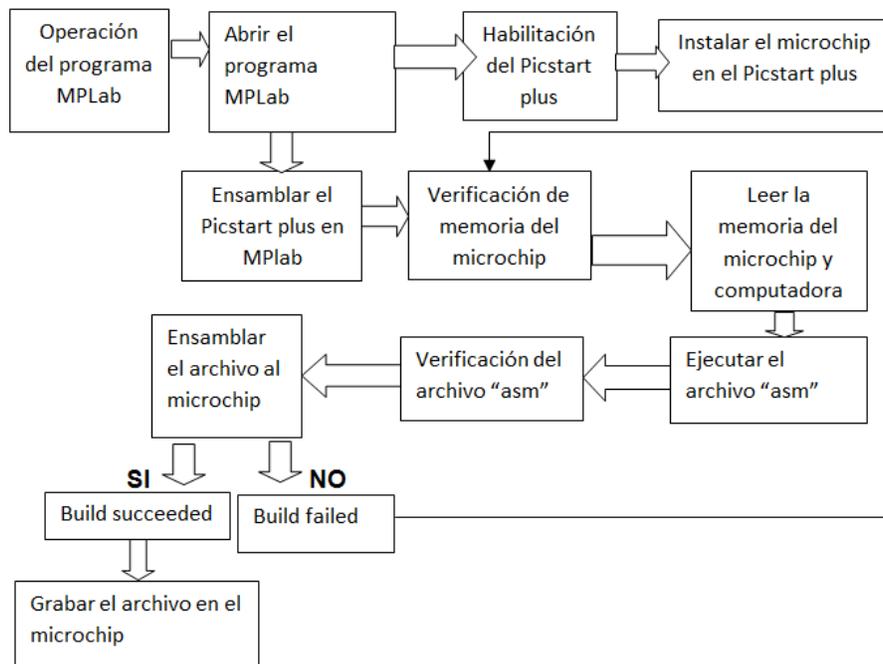


Figura 3.7 Diagrama de bloques para el quemado de PIC

### **3.2.4 Estructuración de manual de prácticas de automatización.**

Las prácticas que están incluidas en este manual fueron las siguientes:

- Encendido y apagado de led
- Compuertas lógicas
- Puerta de garaje
- Encendido de motores

La creación de programas para el PLC FC34, comprenderá de 4 apartados en los cuales cada uno ejecutara diferentes funciones, inicialmente para la comprensión del funcionamiento, así como los principios para la creación de programas más complejos, los cuales mediante la combinación de los mismos nos ayudara a la facilitación para dar solución a diferentes tipos de problemas que se puedan generar, y así generar una solución mediante un sistema de automatización.

#### **1. Encendido y apagado de led.**

El objetivo de este programa es familiarizarse con la programación del PLC así como sus comandos principales, hasta el grabado del programa al PLC. El programa comprende el encendido de un led, el cual se activara mediante una señal de 24V. Los equipos y software empleados para este ejercicio, fue una conexión serial, PLC FC34, computadora con puerto serial, módulo de simulación de sensores, software FST4.

#### **2. Compuertas lógicas**

La meta de este programa es la comprensión del funcionamiento de los diferentes tipos de compuertas lógicas en el PLC, son ecuaciones en algebra booleana, en las cuales arroja un valor de falso (0) o verdadero (1), y mediante el programa creado para el PLC, se comprenderá el comportamiento de respuesta mediante la combinación de diagramas y comandos, para tener definido cada una de las diferentes tipos de puertas lógicas como son AND, OR, NAND, NOR; al igual que

en el ejercicio anterior se utilizara conexión serial, PLC FC34, computadora con puerto serial, módulo de simulación de sensores, software FST4.

### 3. Puerta de garaje

La finalidad del programa de una puerta de garaje, es conocer la aplicación de las compuertas lógicas mediante una combinación de estas. La puerta de garaje comprende de la activación o desactivación de la misma, pero en diferentes situaciones, es decir, abrir o cerrar la puerta ya sea desde adentro o desde afuera de un domicilio. La práctica de este programa nos mostrara el resultado de la mezcla de compuertas lógicas, a través del software y hardware se ha venido utilizando anteriormente.

### 4. Encendido de motores.

El propósito de la programación consta de la activación de motores mediante una señal enviada por termostatos electrónicos, al igual que el PIC16F84A del PLC-UAAAN donde mediante una tabla de verdad, muestra una respuesta para un sistema de automatización, el cual consta del encendido y apagado de motores según se presente la combinación de señales en 3 termostatos en función de una ecuación específica en álgebra booleana.

El ejercicio que se desarrolló para el control de temperatura de un invernadero con los tres termostatos y dos motores, que trabajaron de acuerdo a la tabla de verdad mostrada en el Cuadro (4.1)

Se creara un programa el cual se descargara en el PLC FEC 34, se simulara la señal de sensores, ejecutarlo en línea con la PC, mediante el puerto serial, para al final crear un enlace entre los termostatos electrónicos del Departamento de Maquinaria Agrícola y el PLC Festo FC34, obteniendo una mejor y practica respuesta para la programación de PLC y microcontroladores.

## **IV RESULTADOS Y DISCUSION**

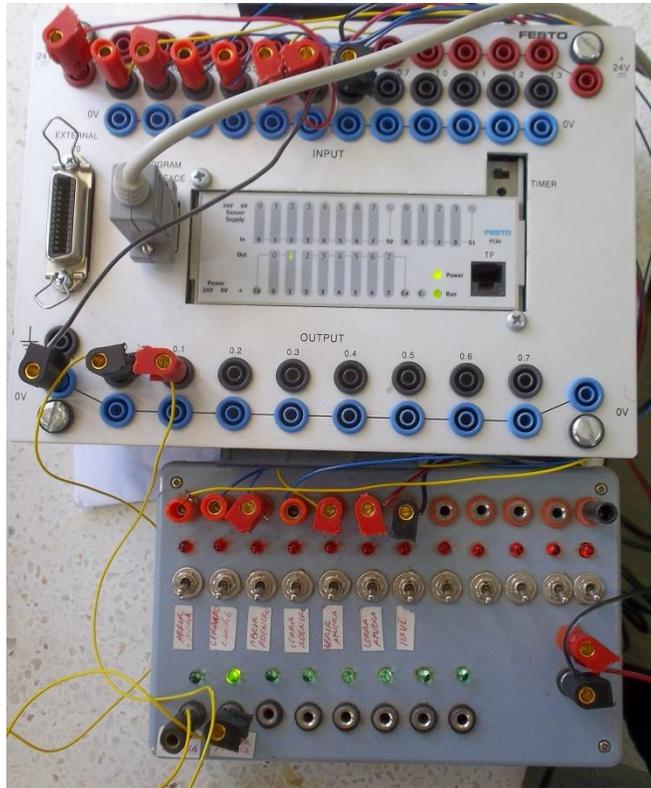
Para la creación de programas en FST 4.10, se realizó con un lenguaje de diagrama de contacto (ladder diagram), los programas se realizaron mediante el siguiente proceso:

### **4.1 Ejercicio 1. Encendido y apagado de led**

El objetivo de esta práctica es familiarizarse con la programación de PLCs y descarga de programas del software en la computadora al PLC.

Para este ejercicio el equipo y software que se emplea, es el siguiente:

- PLC Festo FC34, Figura 3.1
- Módulo de simulación de señales de E/S, Figura 3.4
- Software FST 4.10
- Fuente de 24VCD
- Cables de conexión E/S
- Programa en lenguaje Festo ladder diagram
- PC con puerto serial
- Cable de comunicación serial RS-232



*Figura 4.1 PLC y módulo de simulación conectados*

Procedimiento para la elaboración de la práctica

El PLC recibe una señal en el cual activara o desactivara la salida de señal, dependiendo del programa diseñado, en FST 4.10.

La elaboración del programa de encendido y apagado de un led, que comprende de los siguientes pasos:

#### **4.1.1 Definición de entradas y salidas**

Se ejecuta el software FST 4.10 para la creación del programa, donde se definen las entradas (I) y salidas de señal (O), que se emplearon con el módulo de simulación de sensores, Figura, (3.4) para la ejecución del programa diseñado, el detalle se encuentra en el Anexo 7.4.

Las entradas están definidas como relevadores normalmente abiertos y normalmente cerrados como se muestra en la Figura (4.2).

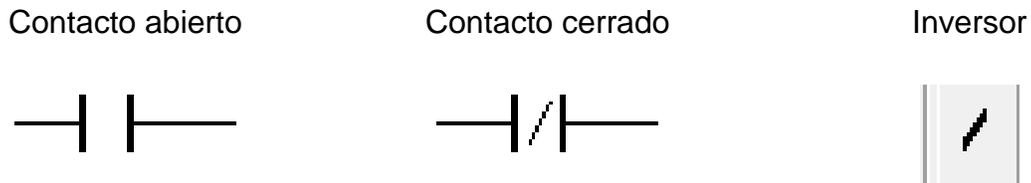


Figura 4.2 Tipos de contactos e inversor

#### 4.1.2 Diseño del programa con variables

Se creó un programa como se muestra en la Figura (4.3) que se compiló y descargó en el PLC, previamente conectado a la pc, mediante el puerto serial.

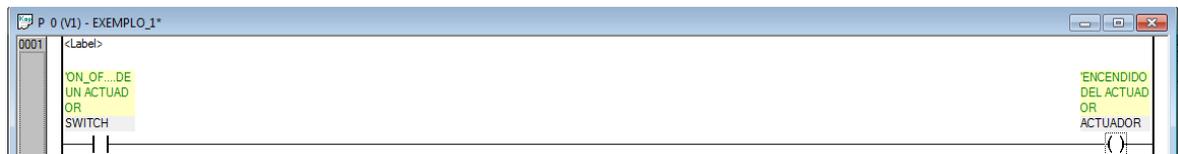


Figura 4.3 Programa de encendido y apagado de led

#### 4.1.3 Compilación y descarga del programa al PLC

La compilación del programa antes de ser descargado, verifica los errores que puedan interferir en el funcionamiento adecuado.

#### 4.1.4 Conexión de entradas y salidas

Se realiza la conexión de entradas y salidas de señales definidas del PLC y módulo de simulación como muestra la Figura (4.1).

#### 4.1.5 Ejecución en línea para verificación de errores

Se realiza una práctica en línea con el programa descargado en el PLC, para la verificación del correcto funcionamiento, mediante una tabla de verdad como muestra el Cuadro (4.1)

*Cuadro 4.1 Tabla de verdad del programa encendido y apagado de un led*

Variable	Salida
1(Verdadero)	1
0(Falso)	0

#### 4.1.6 Ejecución del programa para verificación del funcionamiento

La respuesta obtenida es al abrir el switch de señal de entrada (I), el PLC mostro un resultado en el cual la salida es verdadera (1), lo que significa que el led del módulo de simulación se mostró encendido. Anexo 7.4

#### 4.2 Ejercicio 2. Compuertas lógicas

Los programas realizados en esa sección se dividen en 4, los cuales se diseñaron programas de las compuertas lógicas, AND, OR, NAND y NOR.

Al igual que en el primer ejercicio, la creación de programas es el mismo proceso de edición, compilado y descarga; para este programa se manejó un diseño más amplio, en la simulación se utilizó el mismo hardware mencionado en el ejercicio 1.

Para este ejercicio el equipo y software que se emplea, es el siguiente:

- PLC Festo FC34, Figura (3.1)
- Módulo de simulación de señales de E/S, Figura (3.4)
- Software FST 4.10

- Fuente de 24VCD
- Cables de conexión E/S
- Programa en lenguaje Festo ladder diagram
- PC con puerto serial
- Cable de comunicación serial RS-232

Las respuestas a estos programas fueron las siguientes, como se muestra en la tabla de verdad, Cuadro (4.2)

*Cuadro 4.2 Tabla de verdad de puertas lógicas*

<b>A</b>	<b>B</b>	<b>AND</b> <b>A * B</b>	<b>OR</b> <b>A + B</b>	<b>NOR</b> $\bar{A} * \bar{B}$	<b>NAND</b> $\bar{A} + \bar{B}$
0	0	0	0	1	1
1	0	0	1	0	1
0	1	0	1	0	1
1	1	1	1	0	0

Inicialmente la creación de los programas se realizaron los mismos pasos que en el ejercicio anterior, solo que en este caso se definieron ocho variables de entrada (I) con la literales (A1, A2) para AND, para OR las literales fueron (A3, A4), para las compuertas NAND las variables que fueron (A, B) y por ultimo para la compuerta NOR las literales fueron (A1, B2).el detalle de los programas de las cuatro compuertas lógicas se encuentran en el Anexo 7.5.

#### **4.2.1 Definición de entradas y salidas**

se ejecutó el software FST 4.10 para la creación del programa, donde se definieron entradas y salidas de señal, que se emplearon con el módulo de simulación de sensores para la ejecución del programa diseñado, en este

programa se utilizaron 4 entradas de señal y 2 salidas para la respuesta, para AND y OR, al igual para NAND y NOR

#### 4.2.2 Diseño de programa con variables

Se creó un programa AND y OR, y otro programa con NAND y NOR como se muestra en la Figura (4.4) y la Figura (4.5) que se compilaron y descargaron en el PLC, previamente conectado a la pc, mediante el puerto serial.

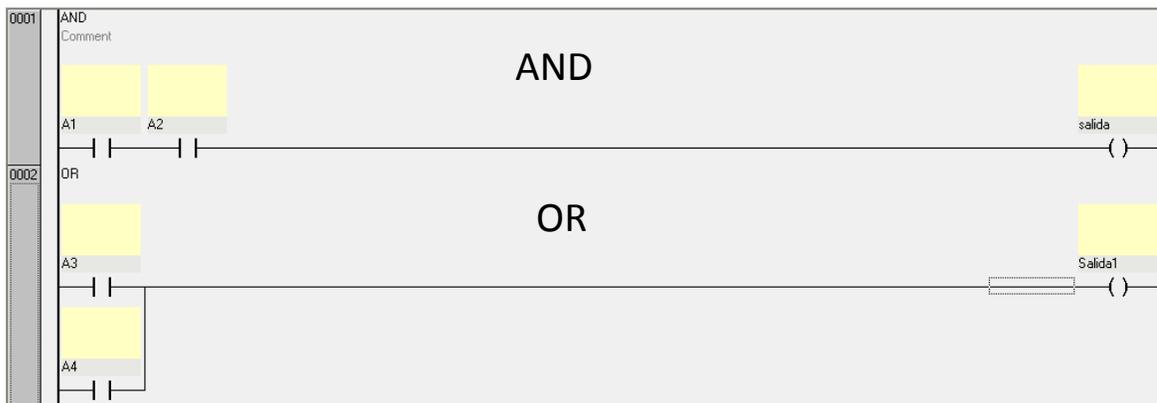


Figura 4.4 Programa AND y OR



Figura 4.5 Programas NAND y NOR

### **4.2.3 Compilación, descarga de programa en PLC**

La compilación del programa antes de ser descargado sirve para confirmar que no existen errores en el programa que se graba en el PLC

### **4.2.4 Conexiones de entrada y salida**

Se realizó la conexión de entradas y salidas de señales definidas del PLC y módulo de simulación como muestra la Figura (4.1).

### **4.2.5 Ejecución en línea con la verificación de errores.**

Se realizó una práctica en línea con el programa descargado en el PLC, para la verificación del correcto funcionamiento, mediante una tabla de verdad como muestra el Cuadro (4.2)

### **4.2.6 Ejecución del programa para verificación del funcionamiento**

las respuestas obtenidas en estos programas descargados al PLC, se manejaron 2 variables de entrada, donde la simulación de los valores se definieron como A y B, las cuales arrojaron las respuestas de la puerta lógica AND, OR, NAND y NOR según el Cuadro (4.2)

### 4.3 Ejercicio 3. Puerta de garaje

El objetivo de este ejercicio es operar en forma externa e interna una puerta de garaje en forma automática que presenta en el Cuadro (4.3)

*Cuadro 4.3 Operación del programa puerta de garaje*

Situación	Respuesta
Abrir desde afuera	Verdadero (1)
Abrir desde adentro	Verdadero (1)
Cerrar desde afuera	Verdadero (1)
Cerrar desde adentro	Verdadero (1)

Con las variables mostradas en el Cuadro (4.4)

*Cuadro 4.4 Variables del programa puerta de garaje*

<b>Salidas</b>	<b>Variable</b>	<b>Operación</b>
O0.0	Relay_op	Relay para abrir la puerta
O0.1	Relay_clo	Relay para cerrar la puerta
<b>Entradas</b>		
I0.0	Open	Limit switch cuando se abre la puerta
I0.1	Closed	Limit switch cuando se cierra la puerta
I0.2	Open_in	Abrir puerta desde adentro
I0.3	Close_in	Cerrar puerta desde adentro
I0.4	Open_out	Abrir puerta desde afuera
I0.5	Close_out	Cerrar puerta desde afuera
I0.6	Key	Llave para abrir desde afuera

Para este ejercicio el equipo y software que se emplea, es el siguiente:

- PLC Festo FC34, Figura (3.1)
- Módulo de simulación de señales de E/S, Figura (3.4)
- Software FST 4.10
- Fuente de 24VCD
- Cables de conexión E/S
- Programa en lenguaje Festo ladder diagram
- PC con puerto serial
- Cable de comunicación serial RS-232

#### **4.3.1 Definición de entradas y salidas**

se ejecutó el software FST 4.10 para la creación del programa, donde se definieron entradas y salidas de señal, que se emplearon con el módulo de simulación de sensores para la ejecución del programa diseñado, en este programa se utilizaron 7 entradas de señal y 2 salidas para la respuesta, como se muestra en el Cuadro (4.4)

#### **4.3.2 Diseño de programa con variables**

El diseño del programa de la puerta de garaje comprende de combinación de relevadores, tanto normalmente abiertos como cerrados, arreglados en forma de compuertas lógicas con el propósito de abrir y cerrar la puerta estando tanto dentro como fuera.

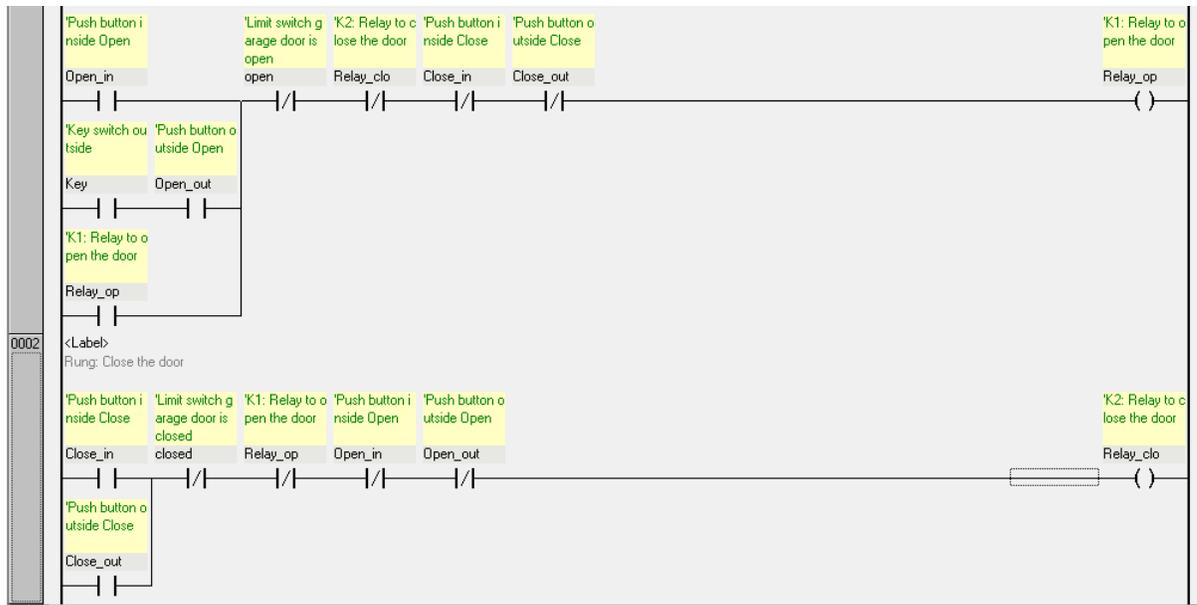


Figura 4.6 Programa de puerta de Garaje

### 4.3.3 Compilación, descarga de programa en PLC

La compilación del programa antes de ser descargado sirve para confirmar que no existen errores en el programa que se grabara en el PLC

### 4.3.4 Conexiones de entrada y salida

Se hizo una conexión de entradas y salidas de señales definidas del PLC y módulo de simulación como muestra la Figura (4.1)

### 4.3.5 Ejecución en línea con la verificación de errores.

Se realizó una práctica en línea con el programa descargado en el PLC, para conocer el funcionamiento del programa.

### 4.3.6 Ejecución del programa para verificación del funcionamiento

El programa realizado en FST 4.10 como se muestra en la Figura (4.6) y mediante el hardware descrito al principio de esta práctica, mostro funcionar

correctamente de acuerdo a los arreglos mostrados en Cuadro (4.3). El funcionamiento de este programa se encuentra en detalle en el Anexo 7.6

#### 4.4 Ejercicio 4. Encendido de motores

Para este ejercicio el equipo y software que se emplea, es el siguiente:

- PLC Festo FC34, Figura (3.1)
- Módulo de simulación de señales de E/S, Figura (3.4)
- Software FST 4.10
- Fuente de 24VCD
- Cables de conexión E/S
- Programa en lenguaje Festo ladder diagram
- PC con puerto serial
- Cable de comunicación serial RS-232
- Termostatos electrónicos
- Motores de 110V
- Tarjeta lógica con pic16f84

El ejercicio de encendido y apagado de motores debe funcionar de acuerdo al Cuadro (4.5)

*Cuadro 4.5 Tabla de verdad encendido de motores*

<b>Termostato 1</b>	<b>Termostato 2</b>	<b>Termostato 3</b>	<b>Motor 1</b>	<b>Motor 2</b>
0	0	0	0	0
1	0	0	1	0
1	1	0	0	1
1	1	1	1	1

Formula de la tabla de verdad del Cuadro (4.5)

$$M_1 = T_1 * \bar{T}_2 + T_3$$

$$M_2 = T_2$$

El Cuadro (4.5) muestra cuando se tienen apagadas las temperaturas los dos motores están apagados, cuando se enciende la temperatura (1), se enciende el motor (1), cuando se enciende la temperatura (1) y (2) se apaga el motor (1) y se enciende el motor (2), y cuando se encienden las temperaturas (1), (2) y (3) se enciende el motor (1) y (2).

#### 4.4.1 Definición de entradas y salidas

se ejecutó el software FST 4.10 para la creación del programa, donde se definió entradas y salidas de señal, que se utilizó junto con el módulo de simulación de sensores para la ejecución del programa diseñado, para este programa se utilizó 3 entradas de señal, y 2 salidas de señal.

#### 4.4.2 Diseño del programa con variables

Se creó un programa como se muestra en la Figura (4.7) que se compilo y descargo en el PLC, previamente conectado a la pc, mediante el puerto serial.

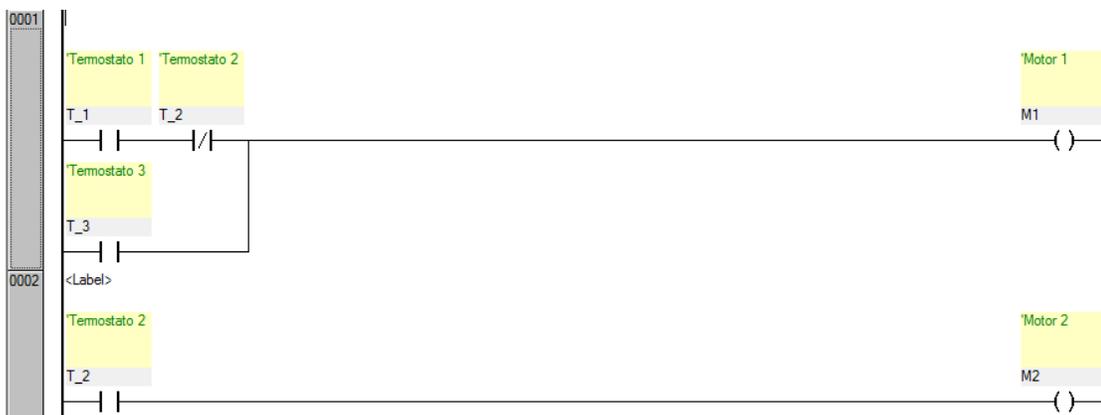


Figura 4.7 Programa de encendido de motores

#### **4.4.3 Compilación y descarga del programa al PLC**

La compilación del programa antes de ser descargado sirve para confirmar que no existen errores en el programa que se grabara en el PLC

#### **4.4.4 Conexión de entradas y salidas**

Se realizaron la conexión de entradas y salidas de señales definidas del PLC y módulo de simulación como muestra la Figura (4.1).

#### **4.4.5 Ejecución en línea para verificación de errores**

Se realizó una práctica en línea con el programa descargado en el PLC, para la verificación del correcto funcionamiento, mediante una tabla de verdad como muestra el Cuadro (4.5).El resultado obtenido de este programa sirvió para la simulación mediante una combinación de puertas lógicas, el cual basado en la tabla de verdad y diferentes tipos de situaciones mostradas en el Cuadro (4.7)

La simulación del programa, sirvió para crear una interfaz entre el PLC y los tres termostatos electrónicos, en el Anexo 7.7, se muestra el detalle del encendido de motores con PLC

## V CONCLUSIONES Y RECOMENDACIONES

### Conclusiones:

- ✚ Se logró un entendimiento de la programación de PICs y PLCs.
- ✚ Se elaboraron cuatro programas que permitieron comprobar el funcionamiento de los PLCs, que incluyo compuertas y ecuaciones lógicas.
- ✚ Se logró una verificación del funcionamiento entre los programas del encendido y apagado de motores empleando tanto PLCs como circuitos lógicos con PICs

### Recomendaciones:

- ✚ Se requiere enriquecer el tablero de automatización con otros actuadores como son los actuadores lineales, motores paso a paso, entre otros.
- ✚ Se sugiere elaborar un banco de sensores digitales que complemente a los sensores de temperatura.
- ✚ Se requiere enriquecer las prácticas con la variable de tiempo así como implementar la automatización en situaciones reales como el control de variables en invernaderos.

## **VI BIBLIOGRAFÍAS**

- ACEDO, J.** 2006. Instrumentación. Ediciones Díaz de Santos. España.
- AGUILERA, P.** 2002. Programación de PLC, Tesis de maestro en ciencias de la ingeniería de manufactura con especialidad en automatización de la Universidad Autónoma de Nuevo León, facultad de ingeniería mecánica y eléctrica, San Nicolás de los Garza, NL.
- CAVADA, D.** 1998. Autosintonización de lazos de control PID en controladores programables, Tesis de maestro en ciencias de la ingeniería eléctrica con especialidad en electrónica de la Universidad Autónoma de Nuevo León, facultada de ingeniería mecánica y eléctrica, San Nicolás de los Garza, NL.
- DOMINGO J, GAMIZ J, GRAU A, MARTÍNEZ H.** 2003. Introducción a los Automatas programables, Editorial VOC, España Barcelona
- KALPAKJIAN S. Y SCHIMID S.** 2002. Manufacturing Engineering and Technology, New Jersey, USA
- LÓPEZ M.** 2010. Control automático de temperatura. tesis de licenciatura de ingeniero mecánico agrícola. Departamento de maquinaria agrícola. Universidad Autónoma Agraria Antonio Narro, Saltillo, Coahuila
- MANDADO E., MENÉNDEZ L., FERNÁNDEZ L., LÓPEZ E.** 2007. Microcontroladores PIC, Sistema integrado para el aprendizaje, Marcobo Ediciones Técnicas, Barcelona España.
- OGATA, K.** 2003. Ingeniería de control moderna. Pearson Educación, S. A. Madrid España
- PALLÁS, R.** 1993. Adquisición y distribución de señales. Marcombo S.A. Barcelona, España.

**PALLÁS, R.** 2003. Sensores y acondicionadores de señal 4º Edición  
Marcombo S.A. Barcelona, España.

**NAVARRO, R.** 2004. Ingeniería de control analógica digital. Ediciones  
McGraw-Hill. México D.F.

**PONSA P. Y VILANOVA R.** 2005. Automatización de procesos mediante la  
guía GEMMA, ediciones UPC, Barcelona, España.

**REYES C.** 2006. Microcontroladores PIC Programación en Basic, segunda  
edición, Quito, Ecuador.

**SCHEY J.** 2002. Introduction to manufacturing processes, By the McGraw-Hill  
Companies, INC. University of Waterloo, Ontario.

**VÁZQUEZ Y.** 2011. Manual de práctica de un sistema de control de  
temperatura, monografía de licenciatura de ingeniero mecánico agrícola.  
Departamento de maquinaria agrícola. Universidad Autónoma Agraria  
Antonio Narro, Saltillo, Coahuila

**VILANOVA I LA G.** 1998. Teoría del control electrónico. Ediciones UPC.  
México D.F

### **Documentos en línea**

Abarca P. 2013. Documento en línea, Mayo de 2013  
<http://www.aie.cl/files/file/comites/ca/abc/sistemas-de-control-automatico.pdf>

International Electrotechnical Commission (IEC), Norma IEC-1131, 1992.  
Documento en línea Abril de 2013  
[http://isa.uniovi.es/docencia/ra\\_marina/cuatrim2/Temas/IEC61131.pdf](http://isa.uniovi.es/docencia/ra_marina/cuatrim2/Temas/IEC61131.pdf)

Instituto Tecnológico Superior de Calkiní en el Estado de Campeche,  
Documento en línea Mayo de 2013  
<http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r69367.PDF>

Instituto Nacional de Tecnologías Educativas y de Profesorado, Documento en  
línea Mayo de 2013  
<http://recursostic.educacion.es/observatorio/version/v2/ca/equipamiento-tecnologico/redes/502-monografico-lenguajes-de-programacion>

Autómatas Industriales, Documento en línea Junio de 2013.  
<http://www.automatas.org/software.htm>

Wikipedia, Abril de 2013, <http://es.wikipedia.org/wiki/Sensor>

Festo AG&Co.KG, documento en línea, Abril de 2013,  
[https://xdki.festo.com/xdki/data/doc\\_es/PDF/ES/FST\\_ES.PDF](https://xdki.festo.com/xdki/data/doc_es/PDF/ES/FST_ES.PDF)

## VII ANEXOS

## ANEXO 7.1

### **FEC FC34, características técnicas, y descripción física como se muestra la Figura (7.1) y Figura (7.2)**

- Dos puertos para expansión de E/S o dos conexiones para RS232 (con adaptador)
- Memoria: 120 KB disponible para programas y controladores.
- Seis salidas por transistor (0.6 Amps)
- Dos salidas por relevador (2 Amps)
- 12 entradas (configurables a NPN o PNP).
- 8 salidas (relevador, transistor).
- 256 Contadores
- 256 Temporizadores (On/Off-delay o Intervalos)
- 256 Registros de 16-bit.
- 160,000 banderas (bobinas internas).
- Un Potenciómetro Analógico para procesos de ajuste.
- Funciones matemáticas de 32 bits.
- Dos contadores rápidos de 4 KHz.
- Montaje por riel DIN y conexiones por tornillo.

## Descripción física PLC Festo FC34, Figura (7.1)

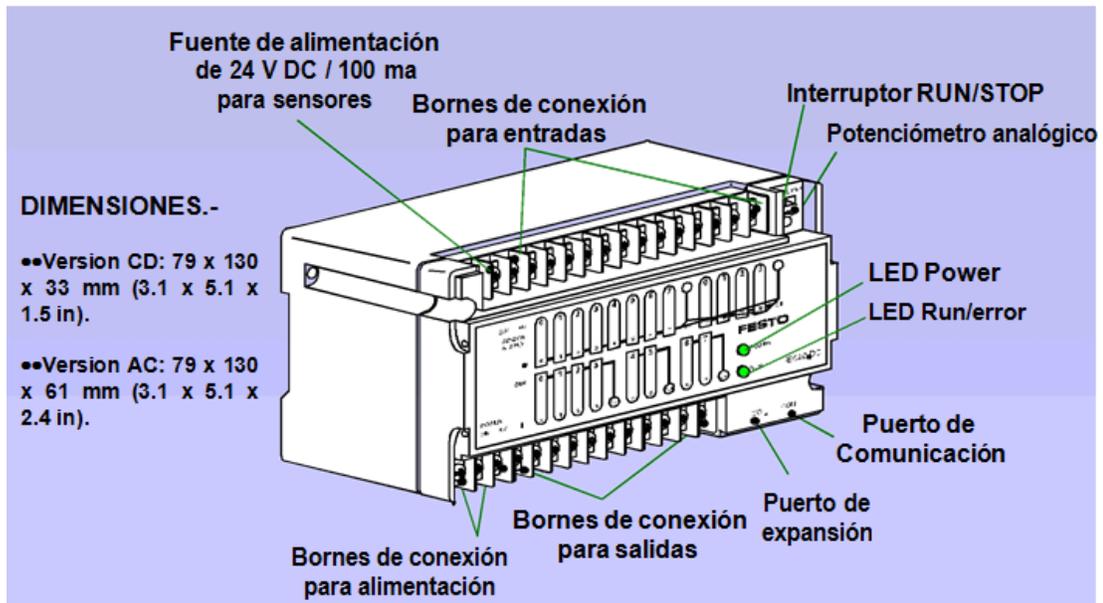


Figura 7.1 Descripción física

## DISTRIBUCION FEC 34, Figura (7.2)

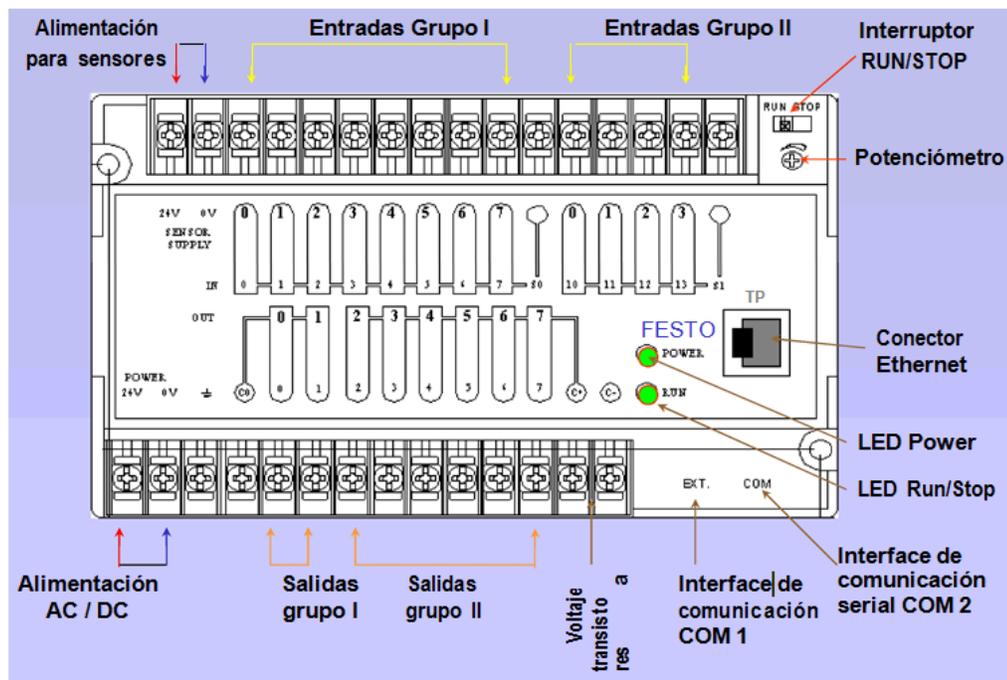


Figura 7.2 Distribución de E/S y demás especificaciones

## LED DE RUN

El PLC cuenta con un led denominado RUN, el cual puede mostrar tres estados posibles como se muestra en la Figura (7.3)

● Verde	Programa ejecutándose.
● Naranja	Programa en paro, almacenamiento de programa.
● Rojo	Error, sin programa.

Figura 7.3 Estados de led

## CONEXIÓN DE ENTRADAS (Conexión de botones) Figura (7.4)

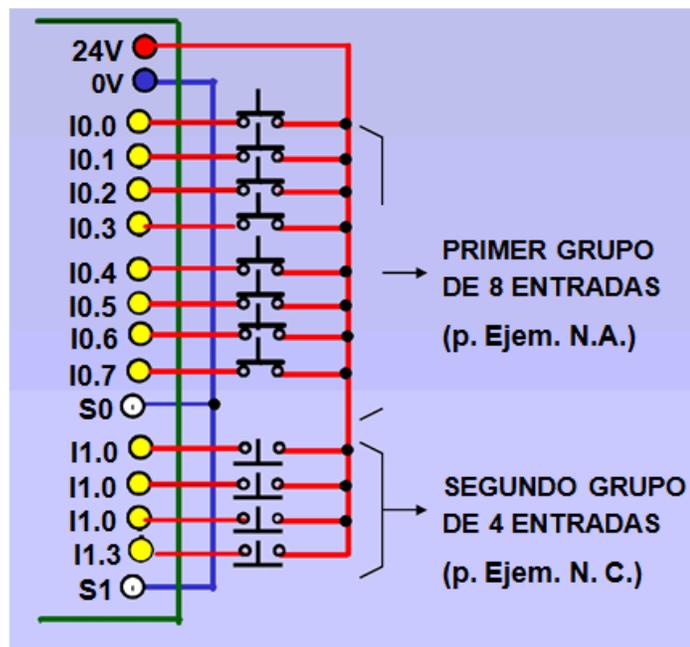


Figura 7.4 Conexión de entradas

## CONEXIÓN DE ENTRADAS (Sensores) Figura (7.5)

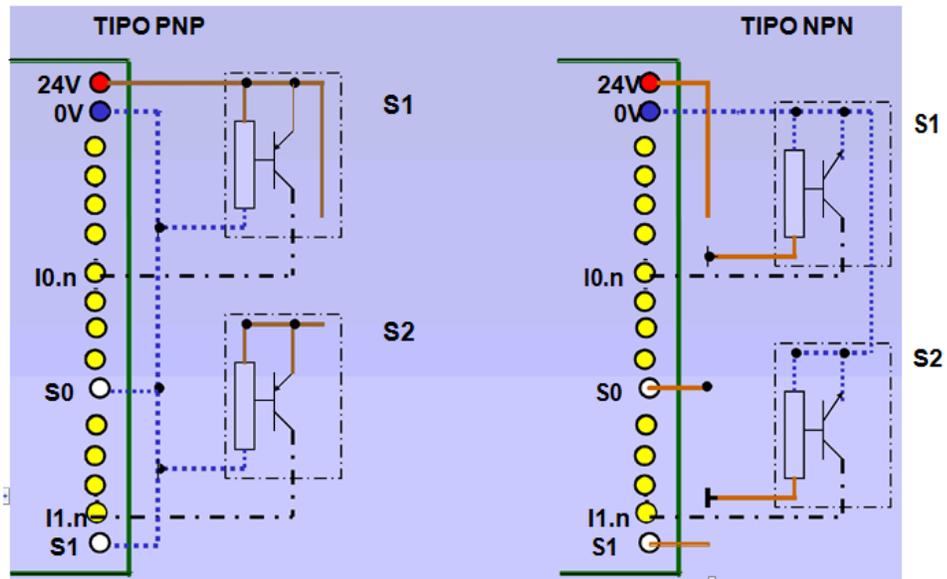


Figura 7.5 Conexión de entradas de señal (sensores)

## CONEXIÓN DE SALIDA, Figura (7.6)

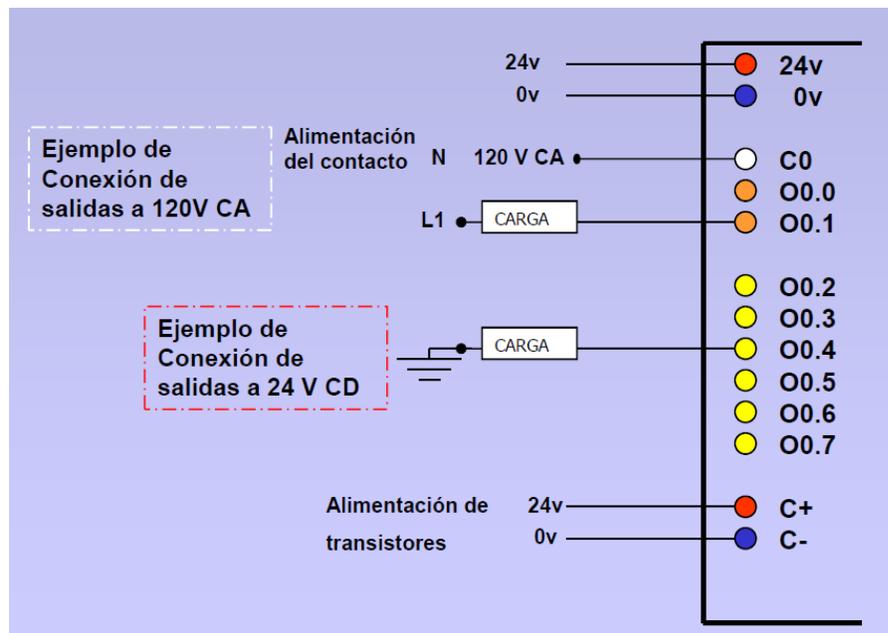


Figura 7.6 Conexión de salida

## ANEXO 7.2

### 7.2.1 Funciones de la placa de control

Para comenzar a explicar las funciones de la placa nos vamos directamente al programa que se utilizó en el microcontrolador.

El programa de control para un termómetro y termostato digital. Utiliza el sensor de temperatura DS18B20 que transmite la información vía serie a través de un bus de una sola línea según un protocolo del fabricante de semiconductores Dallas Semiconductors.

El ajuste de la temperatura a la que conmuta el termostato se logra mediante tres pulsadores como se ve en la Figura (7.7) y son: "MODO" e "INCREMENTAR", además un pulsador de "RESET" para reinicio del programa, que se conectan a pines del Puerto B y cuyo funcionamiento se basa en interrupción por cambio en la línea del Puerto B.

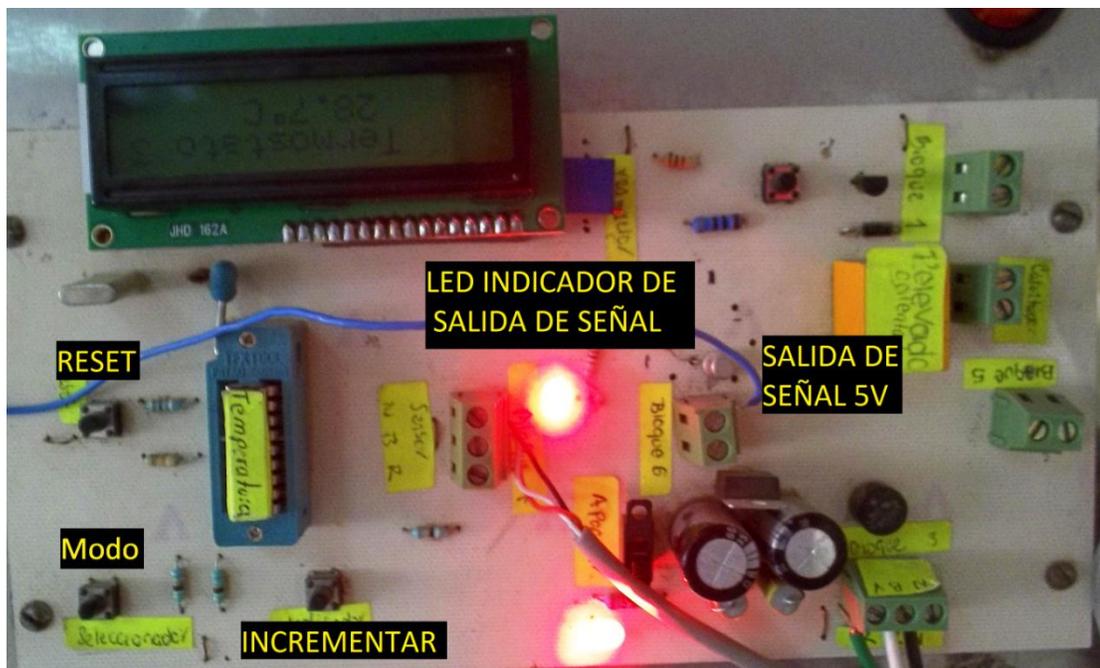


Figura 7.7 Posición de los pulsadores en las placas de control

## 7.2.2 Pulsador Modo

En esta sección se especifica el funcionamiento del pulsador modo, este se basa en tres funciones, con subrutinas mostrada en la Figura (7.8).

**7.2.2.1 Modo "Termostato\_OFF"**, donde funciona como termómetro normal sin termostato. Se reconoce por el flag Termostato\_OFF.

**7.2.2.2 Modo "Termostato\_Ajuste"**, donde se ajusta la temperatura deseada cuando funcione como termostato. Se reconoce por el flag Termostato\_Ajuste.

**7.2.2.3 Modo "Termostato\_ON"**, donde funciona como termómetro normal con termostato. Se reconoce por el flag Termostato\_ON.

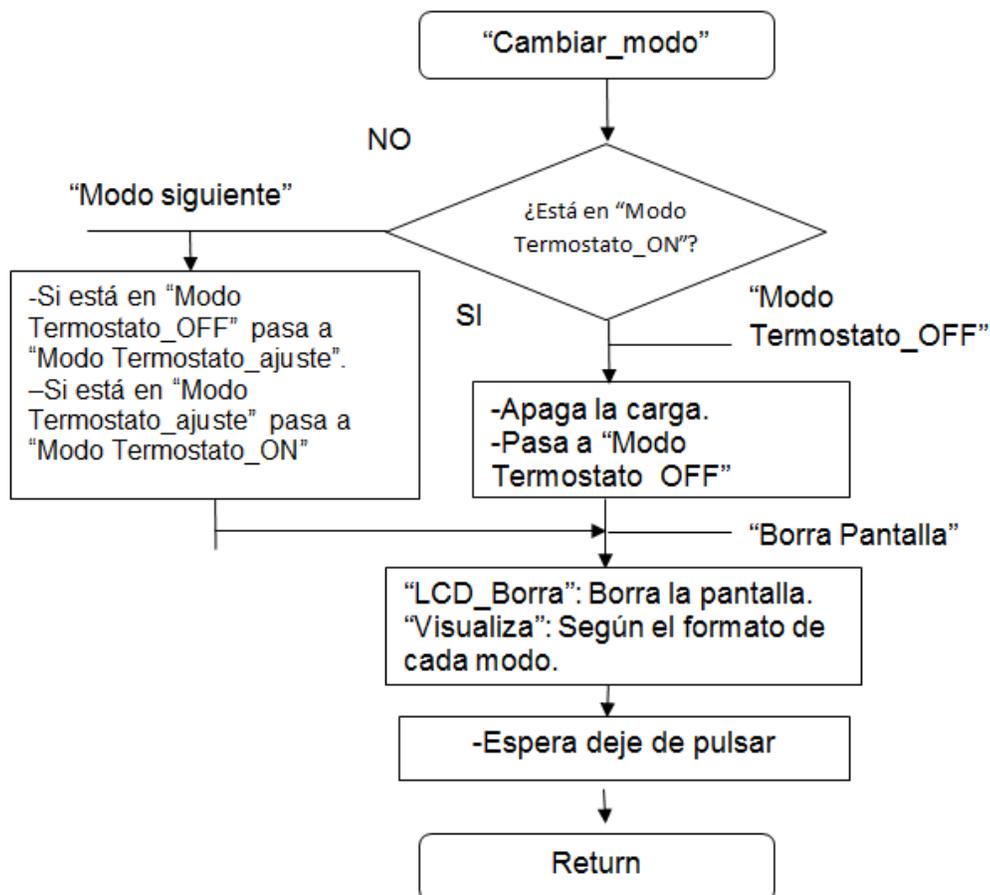


Figura 7.8 Diagrama de flujo de la subrutina Cambiar\_Modo

**7.2.3 Interrupciones:** Esta subrutina lee y visualiza el termómetro cada 2 segundos aproximadamente, como se observa en el diagrama de la Figura (7.9), se ejecuta debido a la petición de interrupción del Timer 0, cada 50 milisegundo. Para conseguir una temporización de 2 s, habrá que repetir 40 veces el lazo de 50 milisegundo ( $40 \times 50\text{ms} = 2000\text{ms} = 2\text{s}$ ).

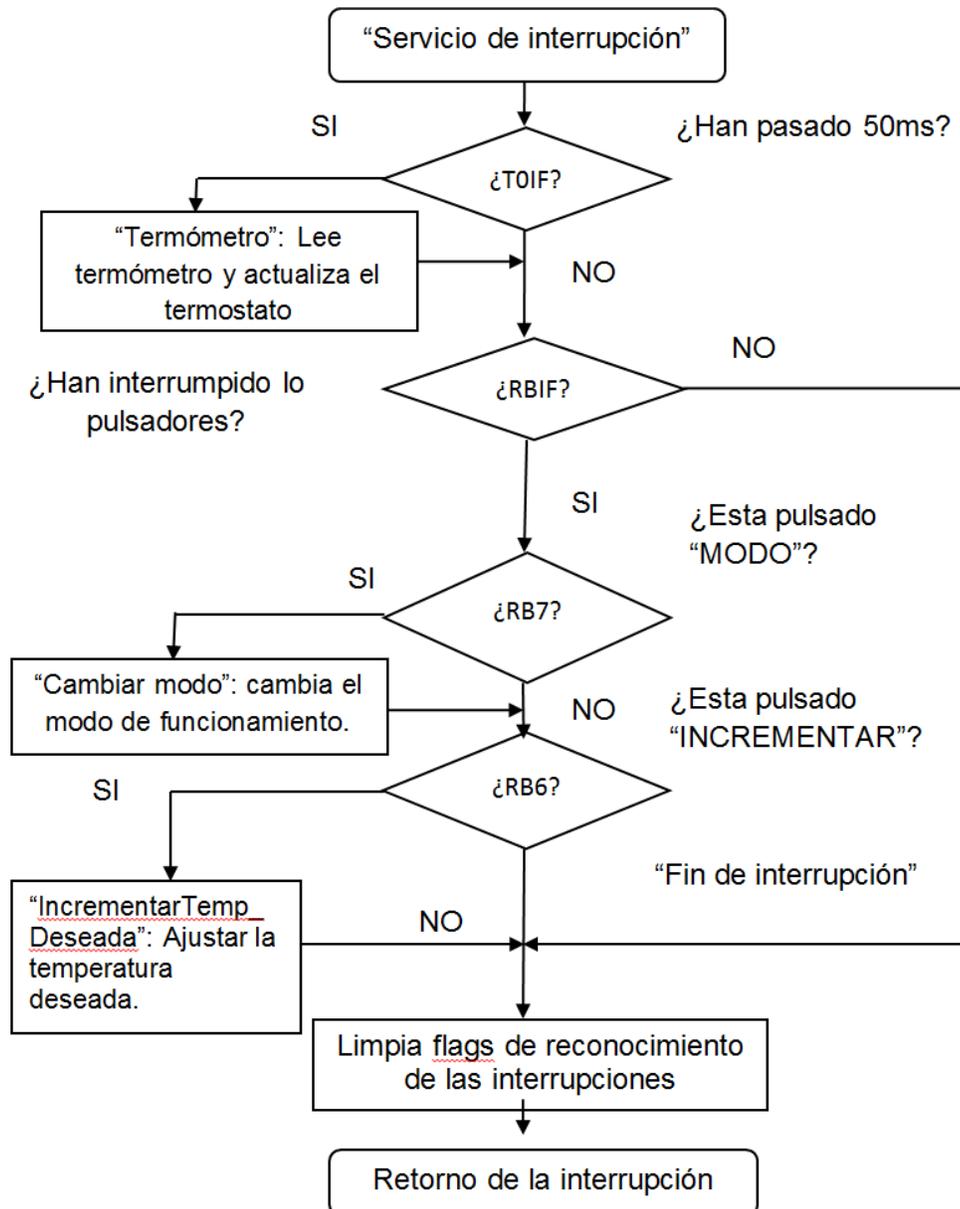
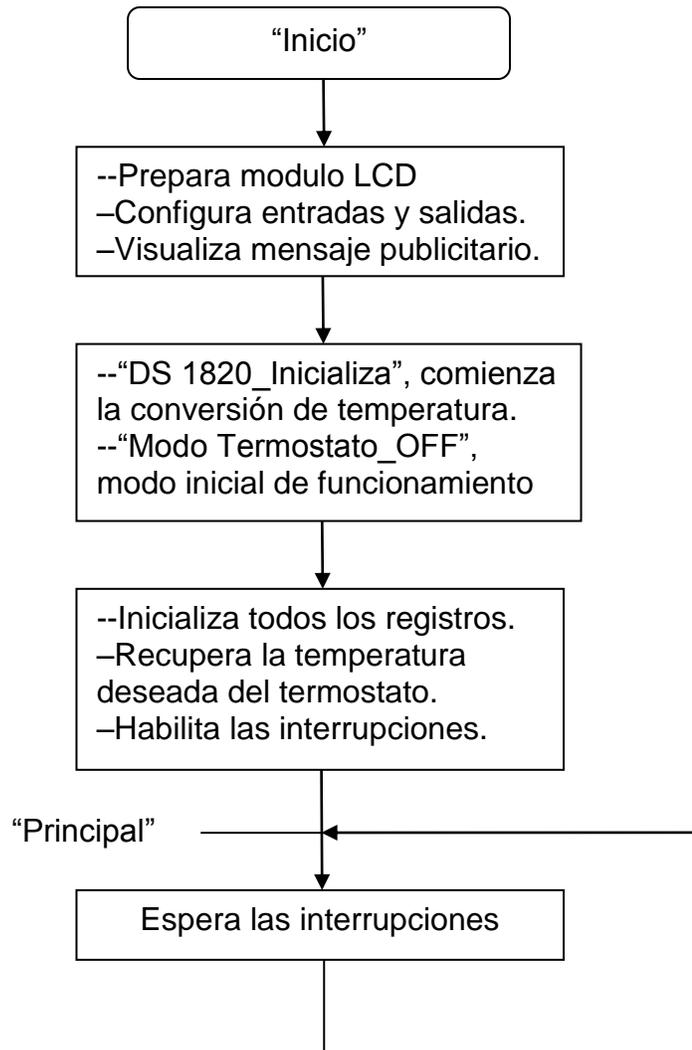


Figura 7.9 Diagrama de flujo de la subrutina de atención a las interrupciones

**7.2.4 Funciones del termómetro DS1820**, en el siguiente diagrama mostrado en la Figura (7.10), se resume su funcionamiento, conjuntamente con el servicio de interrupciones (Figura 7.9).



*Figura 7.10 Diagrama de flujo principal del termostato digital.*

El programa de manera resumida podría decirse que los controles principales son los del MODO e INCREMENTO, por el tipo de programación, que a base de ellos podía regularse los diferentes arreglos para las pruebas, aunque el sistema no tiene un pulsador de "REDUCCION".

**7.2.5 Resumen del programa**, el autor lo resume en 4 pasos:

- 1 En estado de reposo funciona sólo como termómetro. Aparece la temperatura en pantalla del módulo LCD. La salida del termostato está apagada.
- 2 Pulsa "MODO" y se ajusta la temperatura deseada mediante el pulsador "INCREMENTAR".
- 3 Vuelve a pulsar "MODO", se activa el termostato. Si la temperatura medida es menor que la deseada enciende la carga, que puede ser un calefactor. Si la temperatura medida es mayor que la deseada, apaga la carga.

Si se vuelve a pulsar "MODO", apaga la carga y pasa a funcionar sólo como termómetro.

## ANEXO 7.3

### 7.3.1 Uso del MPLAB y grabado del microcontrolador

El MPLAB es un entorno de desarrollo integrado que le permite escribir y codificar los microcontroladores PIC de Microchip para ejecutarlos. El MPLAB incluye un editor de texto, funciones para el manejo de proyectos, un simulador interno y una variedad de herramientas que lo ayudarán a mantener y ejecutar su aplicación. También provee una interface de usuario para todos los productos con lenguaje Microchip, programadores de dispositivos, sistemas emuladores y herramientas de tercer orden.

El MPLAB está diseñado para ser ejecutado bajo Windows 3.11, y puede operar con Windows 95 y 98. Asume que el usuario ya conoce el entorno de Windows y sabe manejarlo.

El código del programa puede estar escrito el blog de notas, tomando en consideración que nuestro archivo debe tener siempre la extensión “.asm”, ya que nuestro programa los reconoce de esta manera, para su posterior ejecución.

Es recomendable que nuestros archivos en “asm” estén guardados en **C** “es el disco local que tiene la computadora” para que el programa evite una búsqueda por todos los sectores; lo cual nos ocasionara un error cuando se esté ensamblando el programa.

Nuestro siguiente paso es ejecutar el archivo “asm”, de la siguiente manera: nos dirigimos al menú de herramientas damos click en File, y seleccionamos Open buscamos nuestro archivo deseado y lo abrimos, Figura (7.11)

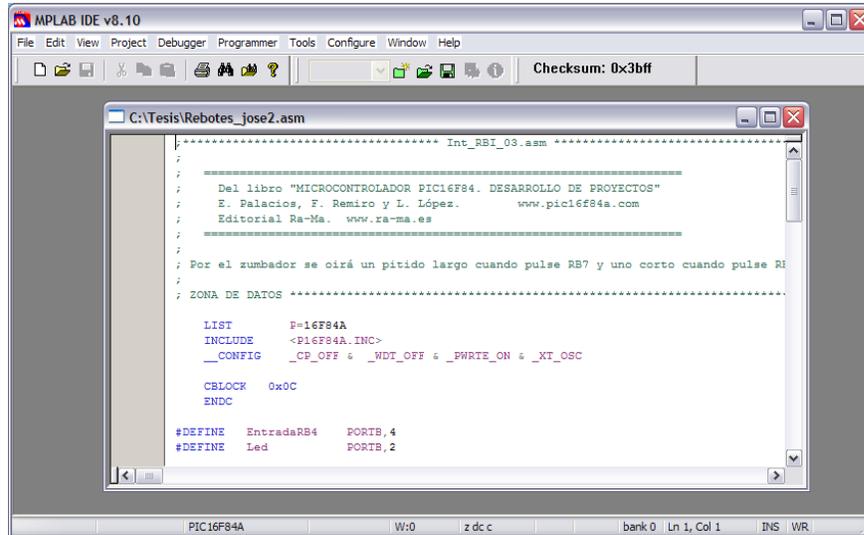


Figura 7.11 Selección del programa “asm”

Una vez que hemos abierto el archivo, el siguiente paso es ensamblarlo, es decir, correr el programa para verificar su correcta escritura y ejecución, esto se hace como lo indica la Figura (7.12)

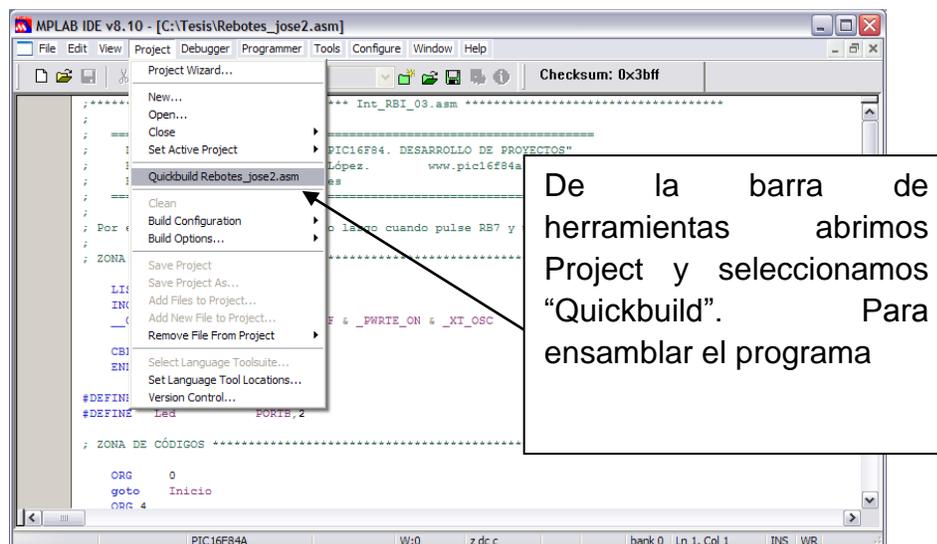


Figura. 7.12 Procedimiento para el ensamblado

Una vez realizado esto, el programa nos enviara un mensaje como este: **BUILD SUCCEEDED**, lo que significa que ha sido ejecutado correctamente, lo que nos indica que podemos continuar con el grabado del micro controlador. De lo contrario nos enviara este otro mensaje: **BUILD FAILED**, lo cual nos indica que existe algún error y debemos verificar nuevamente nuestro programa.

### 7.3.2 Conexión del microcontrolador.

Para realizar este procedimiento es necesario contar con el siguiente equipo:

1. High-Speed USB Serial Adapter Keyspan by Tripp Lite, Figura (7.13)



*Figura 7.13 USB Adapter Keyspan*

### 7.3.3 El quemador o grabador PICSTART Plus, Figura (7.14)

El grabador o programador es el equipo físico donde se procede a grabar la memoria del microcontrolador con las instrucciones del programa de control.

Al instalar un chip en el programador, hay que cerciorarse de que todos los pines o patillas del dispositivo estén rectos y de que entren bien en el zócalo. Hay que tener mucho cuidado porque estos pines se doblan y se rompen con extrema facilidad.

El microcontrolador PIC16F84A tiene 68 registros disponibles de modo que los diseños que utilicen como elemento de control deben ser actualizados. También dispone de un contador de programa de 13 bits constituidos por dos registros que son PLC y PCH

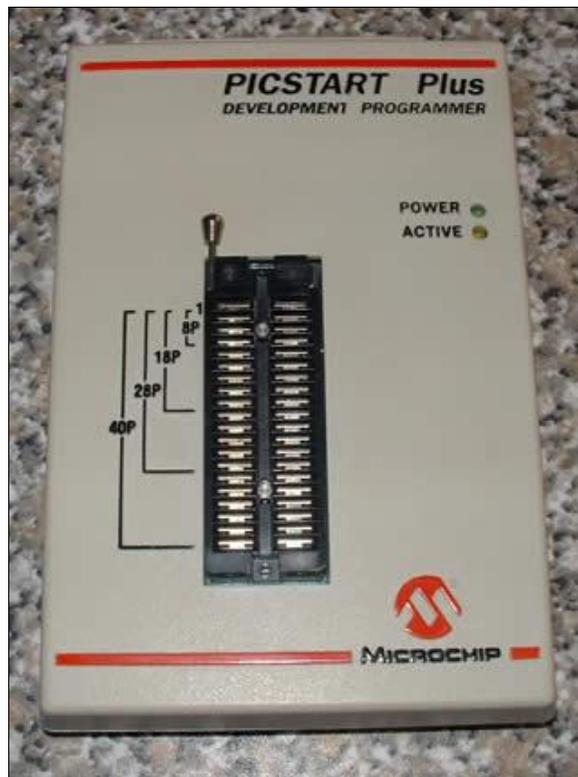


Figura 7.14 PICSTART Plus.

Cuando hemos verificado que contamos con el equipo necesario volvemos a la parte donde se quedó ensamblado y verificado el buen funcionamiento del programa. Lo que prosigue es habilitar el PICSTART Plus desde el MPLAB lo cual se hace de la siguiente manera:

Después verificamos la memoria del PIC para constatar que se encuentra vacío o bien que contenga información, Figura (7.15)

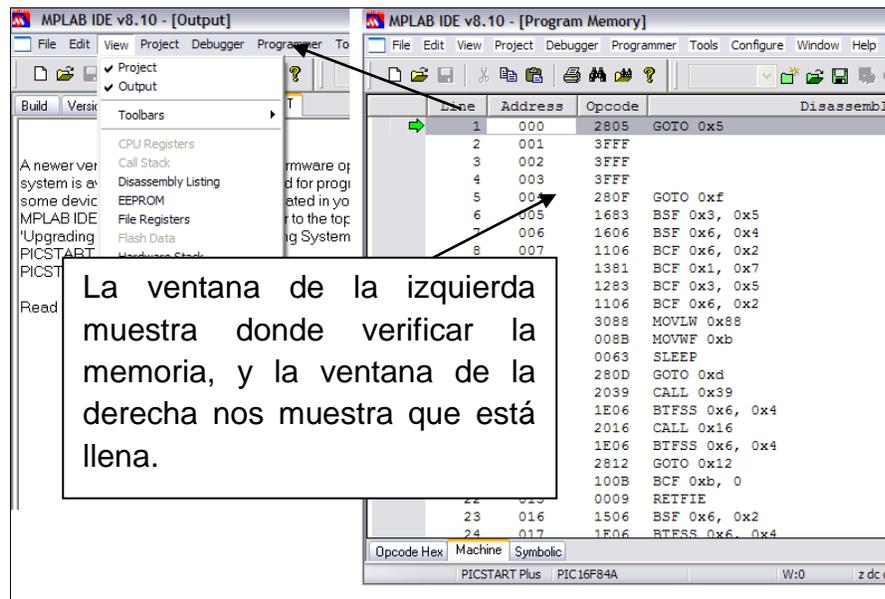


Figura 7.15 Verificación de memoria

Si encontramos que la memoria está llena la borramos haciendo clip en Era Flas Device como lo muestra en la Figura (7.16)

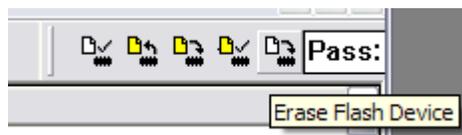


Figura 7.16 Borrar memoria del micro controlador.

Después, del menú de herramientas seleccionamos Debugger, cuando abre la venta seleccionamos Clear Memory y All Memory, y después volvemos a verificar la Memoria del PIC, como se explicó antes. Figura (7.17)

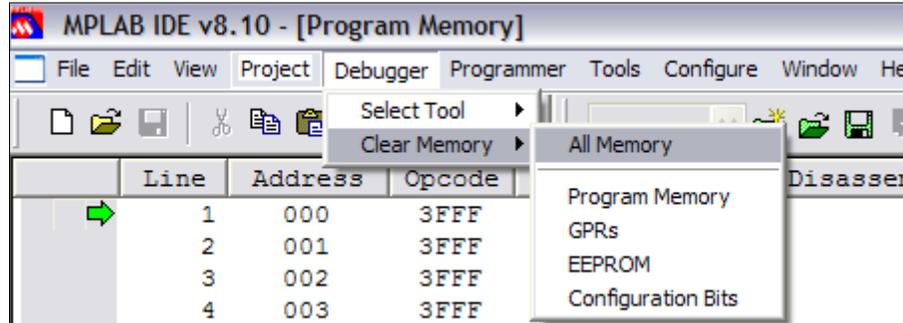


Figura 7.17 Borrar la memoria del microcontrolador

Finalmente cuando hemos verificado que la memoria del microcontrolador está vacía, procedemos a grabar el micro controlador, seleccionamos Program para grabar en el micro controlador como se muestra en la Figura (7.18)

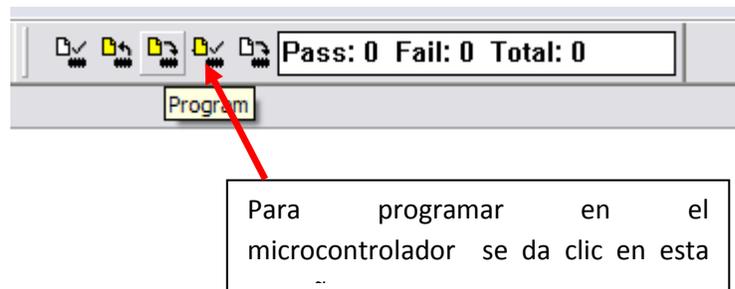


Figura 7.18 Programación directa

## ANEXO 7.4

### 7.4.1 Programación en FST 4.10

La creación de programas para PLC facilita la automatización de funciones que ayudara al individuo en la ejecución de tareas en que se requiera.

### 7.4.2 Instalación del PLC y módulo de simulación de sensores

El PLC funciona con un voltaje de 24 VCD, conectara la fuente de poder, Figura (7.19) al PLC, Figura (7.23) para alimentarlo, y con un par de cables conectar en la entrada de señal, Figura (7.20), 0.0Input, una de las salidas del módulo de simulación de sensores.



*Figura 7.19 Fuente de poder de 24 V*

Para la salida, Figura (7.21), conectar un cable en 0.0 Output, esto hacia el módulo de simulación, Figura (7.22), en los conectores negros.

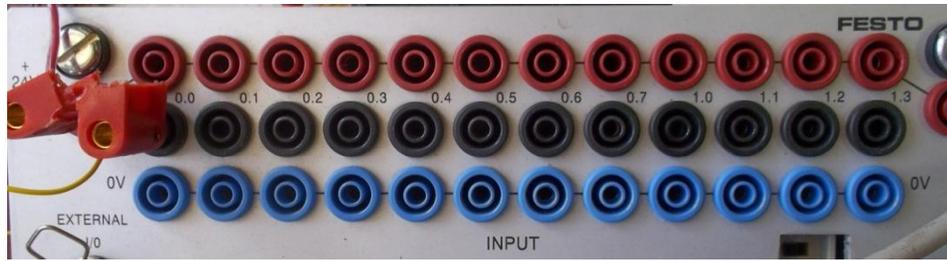


Figura 7.20 Entradas de Señal de PLC

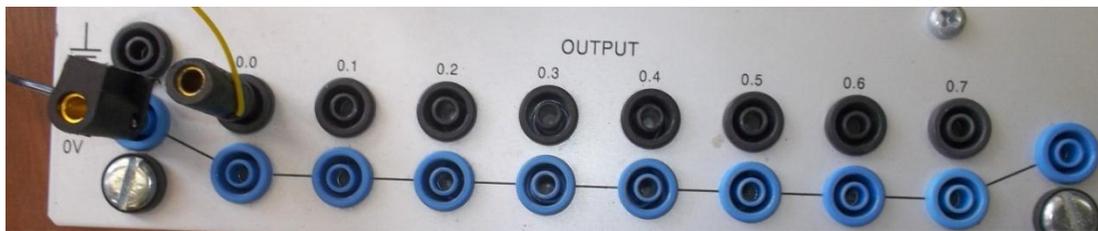


Figura 7.21 Salida de Señal



Figura 7.22 Modulo de simulación de sensores

La conexión entre el PLC y el Modulo de simulación, corresponde a las mismas entradas y salidas nombradas.

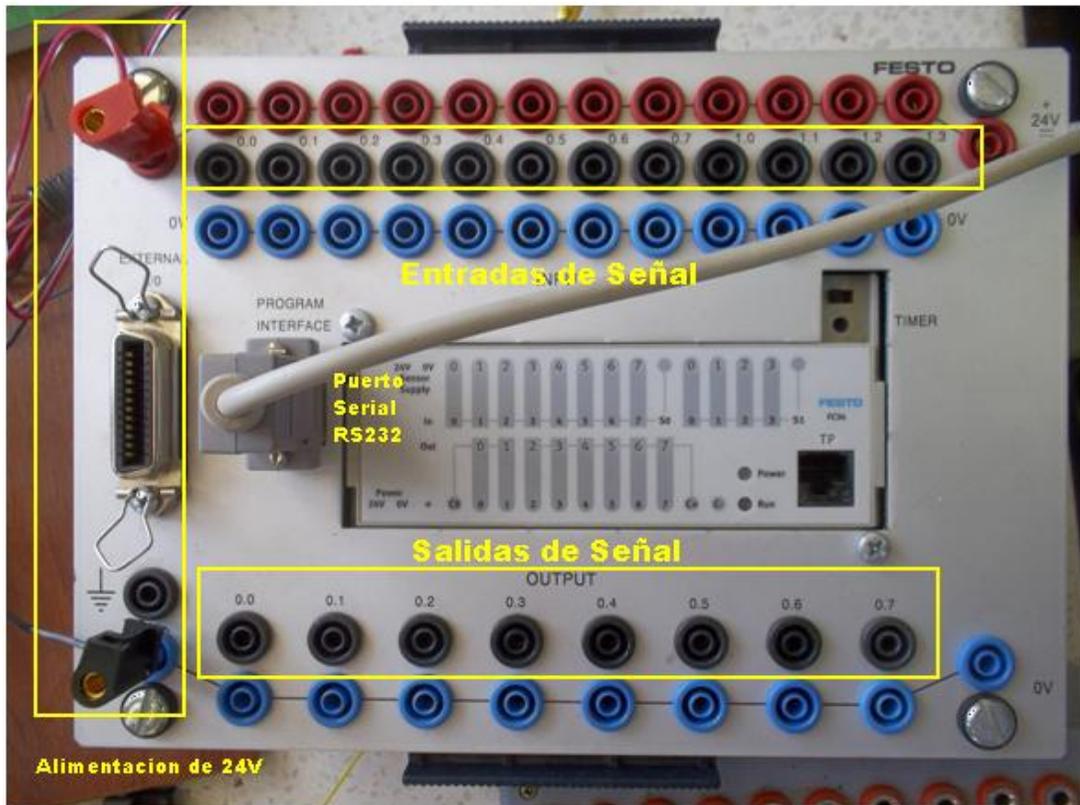


Figura 7.23 PLC Festo FC34

### 7.4.3 PLC Festo FC34

El proceso para crear programas, compilar, descargar y simular en el PLC se realiza mediante el software FST 4.10., Figura (7.23)



Figura 7.23 FST 4.10

#### 7.4.4 Instalación de Software FST 4.10

La instalación del programa FST 4.10 se realizara mediante el siguiente proceso: dirigirse a MI PC y abrir la carpeta donde esté ubicado los archivos de instalación, Figura (7.24)

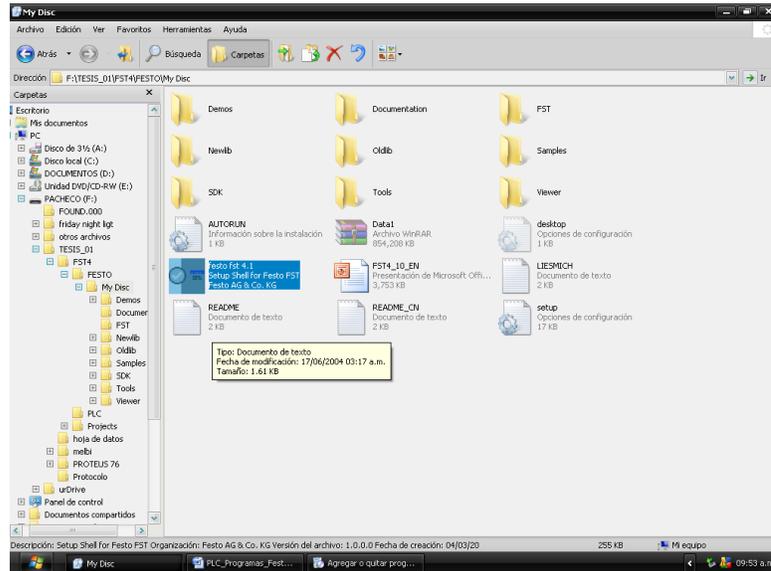


Figura 7.24 Explorador de archivos

Seleccionar “Festo FST 4.1.exe” ´realizar doble clic o presionar enter para la ejecución del archivo, en seguida abrirá una ventana como la Figura (7.25)



Figura 7.26 Selección de opciones a instalar

Seleccionar con un clic install FST 4.0, y cargara una nueva ventana Figura (7.27) la cual, indicara que iniciara la instalación de FST 4.10. Y seleccionar el botón Next.

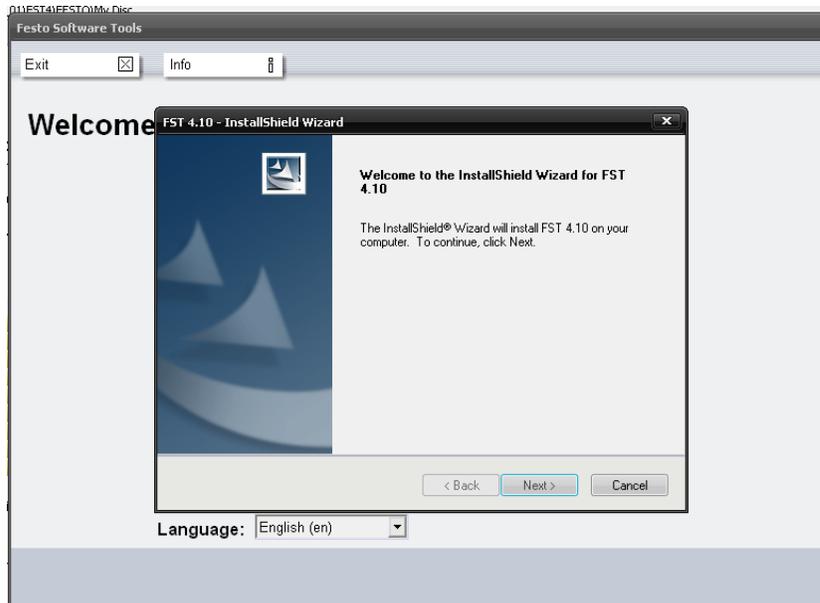


Figura 7.27 Inicio de instalación de FST4.10

Dar clic en el botón next para continuar con la instalación del programa y aceptar términos y condiciones, Figura (7.28)

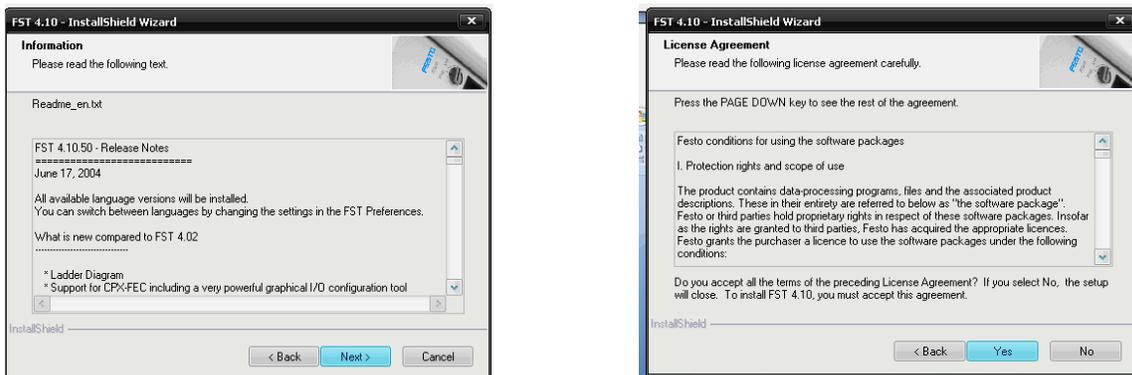
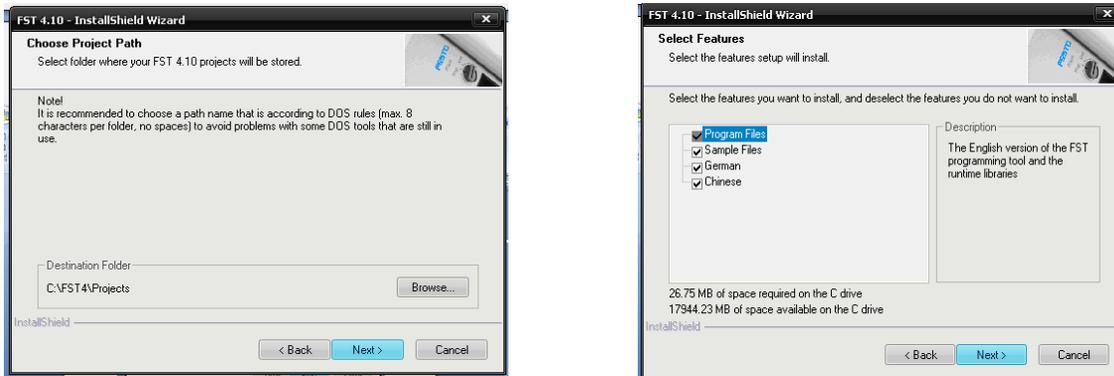


Figura 7.28 Términos y condiciones



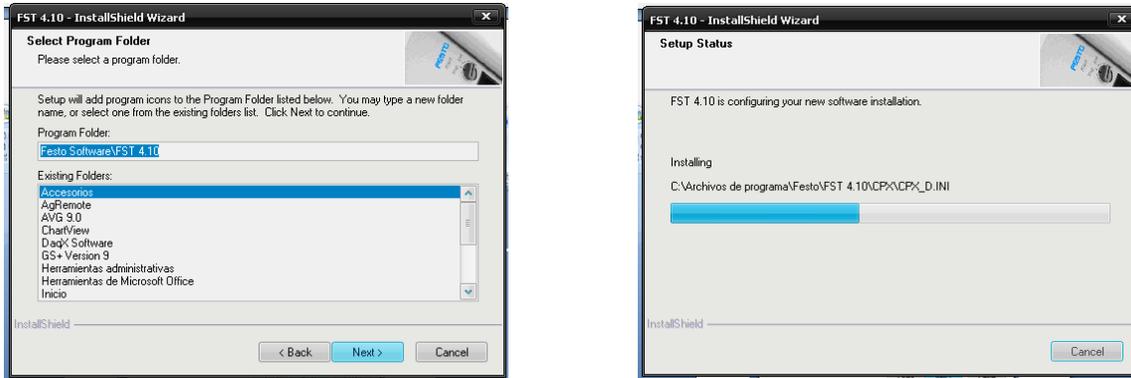
*Figura 7.29 Ventanas en el proceso de instalación*

Seleccionar la ubicación de instalación, Figura (7.29) la cual indicara por default C:\Archivos de programa\Festo\FST 4.10



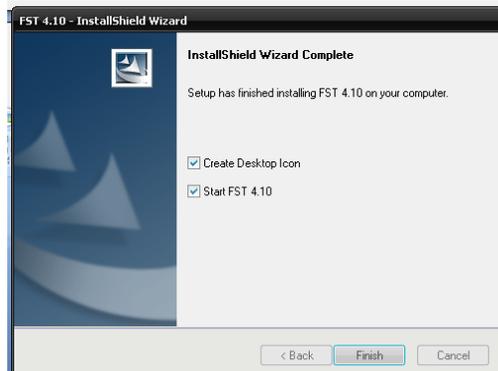
*Figura 7.30 Ventanas con opciones de ubicación e idiomas a instalar*

Tenemos la opción de seleccionar la instalación de los idiomas que tenemos disponibles para el programa, Figura (7.30)



*Figura 7.31 Opciones y copiado de archivos*

Esperar a que cargue los archivos copiados, Figura (7.31)



*Figura 7.32 Finalización de instalación*

Finalizar y verificar el funcionamiento del programa, Figura (7.32) al finalizar la instalación la ventana que se ejecuta al inicio, se cierra con un clic en el botón de Exit como se muestra en la Figura (7.26)

Ahora ejecutado el programa, arrojará una ventana como la que se muestra la Figura (7.33)

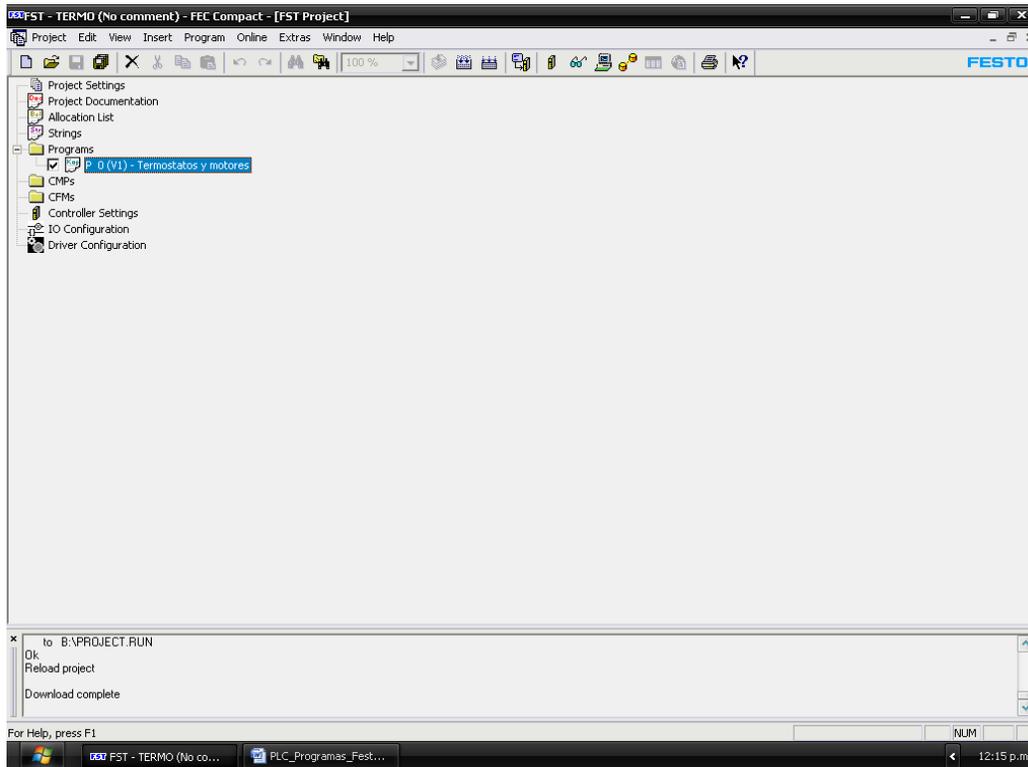


Figura 7.33 Ventana del programa FST 4.10

Para descargar el programa diseñado, se utilizara el puerto Serial y con un cable RS-232 se realiza la conexión entre la PC y el PLC, verificar la velocidad de comunicación la cual corresponde a 9600 baudios, tanto el software como en la configuración del puerto en el Administración de dispositivo.

Esto se puede consultar en el menú principal / Extras / Preferences... (Figura 7.34)

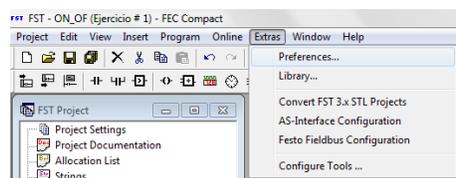


Figura 7.34 Pestaña de Preferences

Esta opción ejecutara un cuadro como la Figura (7.35)

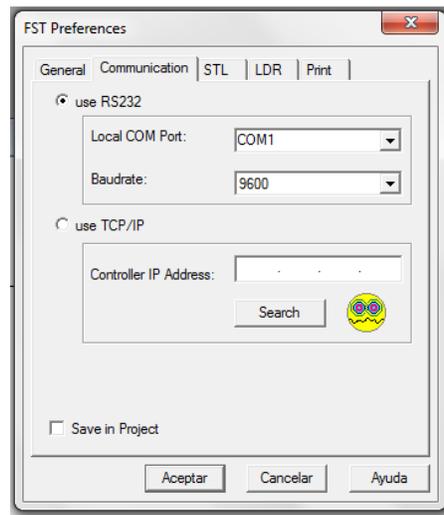


Figura 7.36 FST Preferences

Selecciona *Use RS232*, y en *Local COM Port*, seleccionar *COM1*, y un Baurate de 9600, que en general es el puerto y la velocidad de comunicación con la que se trabaja por default.

Así también verificar el puerto Serial en el administrador de dispositivos de la PC, Figura (7.37) acceder al Panel de control, y accesar a propiedades del sistema/Hardware/Administrador de dispositivos



Figura 7.37 Propiedades del sistema

Seleccionar Puertos COM & LPT, doble clic en Puertos de comunicaciones (COM1), Figura (7.38)

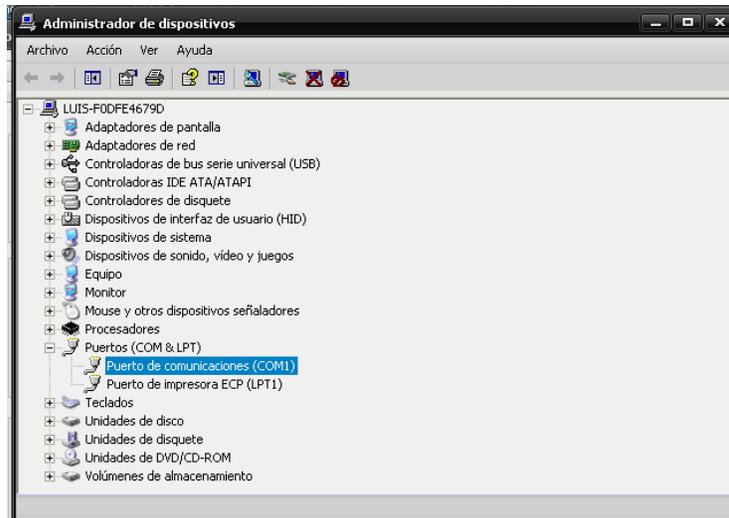


Figura 7.38 Administrador de dispositivos

Dirigirse a Configuración de Puerto, y tener la misma configuración de la Figura (7.39)

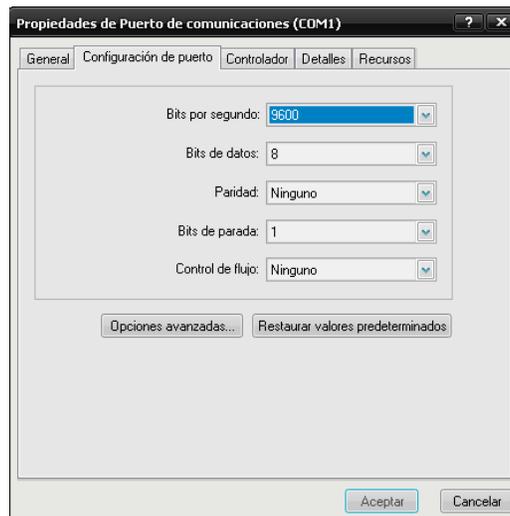
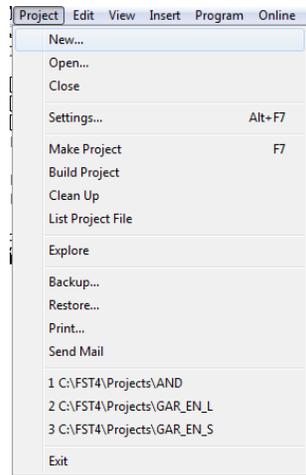


Figura 7.39 Propiedades de puerto de comunicaciones (COM1)

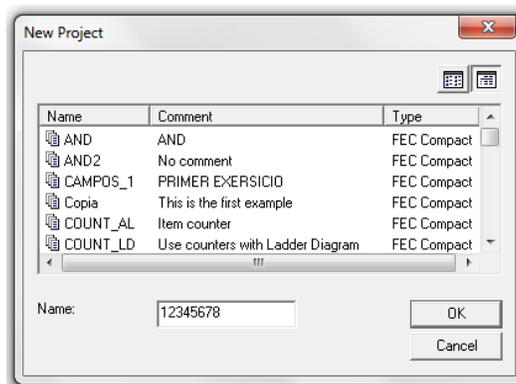
El siguiente paso es la compilación y descarga del programa.

Para iniciar la creación de un programa, Figura (7.40) dirigirse a Project/New



*Figura 7.40 Opciones de pestaña Project*

Se abrirá una ventana en la cual se definirá con el nombre del programa con una restricción de no mayor a 8 caracteres, Figura (7.41)



*Figura 7.41 New Project*

Al momento de designar un nombre al programa, y de haber sido guardado, no se podrá repetir ni sobrescribir el nuevo archivo.

Una vez asignado el nombre, es necesario ingresar la referencia del controlador PLC y seleccionar "COMPACT", agregar algún comentario si se desea y aceptar.

Ahora dirigirse a AllocationList, es donde se muestran las señales de entrada (I0.0, I0.1, I0.2...) y salida (O0.0, O0.1, O0.2...).

Dirigirse a FST Project, que es un menú del programa creado y ahí seleccionar AllocationList / Clic derecho y seleccionar open, Figura (4.23) (A)

Para poder insertar una señal dar clic derecho en la ventana que se abrió de AllocationList, Figura (7.42) (B)

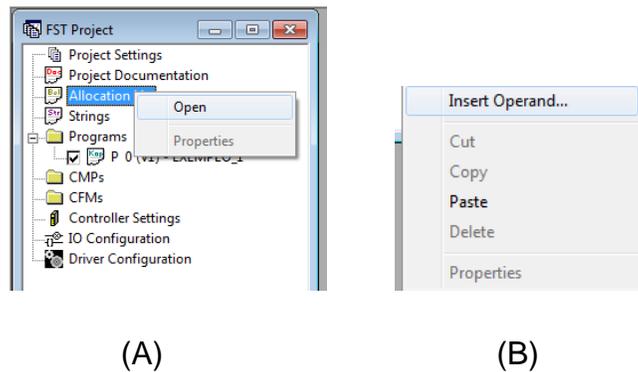


Figura 7.42 Opciones

Ahora se abrirá un cuadro como el que se muestra en la Figura (7.43)

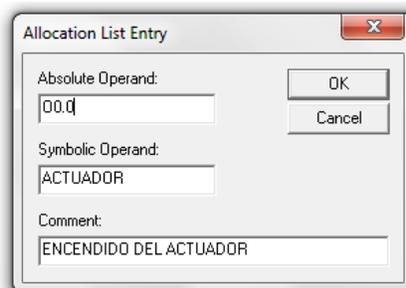


Figura 7.43 Allocation list entry

AllocationListEntry muestra 3 espacios en los cuales se definirán:

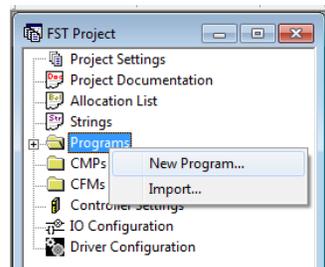
- Absolute Operand.
- Symbolic Operand.
- Comment.

Absolute Operand, es en donde se declara como entrada o salida de señal por ejemplo con la siguiente nomenclatura (Entrada con una “I” seguido de “0.X”, “X” representa cualquier número del canal a utilizar, y para salida señal con una “O” seguido de “0.X” O0.0)

Para el siguiente recuadro “SymbolicOperand”, es donde se asigna el nombre a la variable por ejemplo Star, Switch, A1, X\_1, etc.

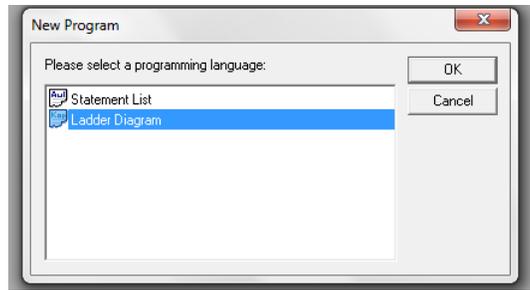
Y al final tendremos un espacio llamado “Comment” es donde se agrega algún comentario que pueda servir referente a la señal de entrada o salida.

Con las señales declaradas con las que se trabajara, cerrar la ventana de AllocationList y dirigirse en la sección de FST Project y dar clic derecho al icono de Program, Figura (7.44) y seleccionar InsertProgram... para poder realizar las funciones a simular.



*Figura 7.44 Insertar nuevo programa*

En este caso seleccionar LadderDiagram, Figura (7.45) ahora dar clic en ok sin modificar nada y mostrara una línea, que es donde diseñaremos el programa.



*Figura 7.45 New program*

Inicialmente crearemos un programa sencillo para relacionarse con los iconos, comandos, etc. de FST 4.10.

El programa consta de una sola entrada y una sola salida, el procedimiento realizado fue el siguiente.

#### **4.5 Ejercicio 1. Encendido y apagado de un led.**

Definir la salida O0.0 como Actuador, la entrada será I0.0 como Switch, será quien activara o desactivara la señal, Figura (7.46)

Operand	Symbol	Comment
O0.0	ACTUADOR	ENCENDIDO DEL ACTUADOR
I0.0	SWITCH	ON_OF....DE UN ACTUADOR

*Figura 7.46 Allocation list*

Cerrar la ventana de AllocationList, y dirigirse a la ventana FST Project después a la carpeta "PROGRAMS" y dar clic derecho/ New Program/ seleccionar Ladderdiagram.

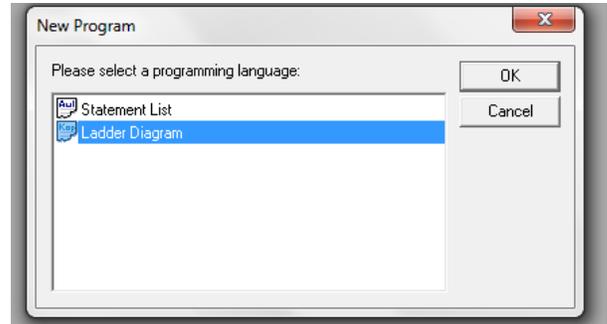
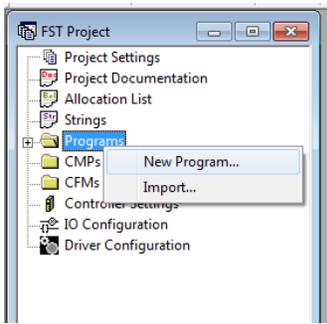


Figura 7.47 Secuencia para un nuevo programa

Y seleccionar OK, abrirá una nueva ventana

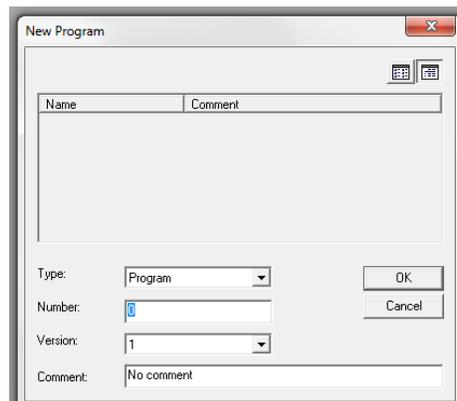


Figura 7.48 Características del programa

Seleccionar Ok, sin mover ningún dato, o en su defecto agregar algún comentario sobre el funcionamiento del programa ah diseñar, Figura (7.48). El espacio donde desarrollamos el programa, es como se muestra en la Figura (7.49)

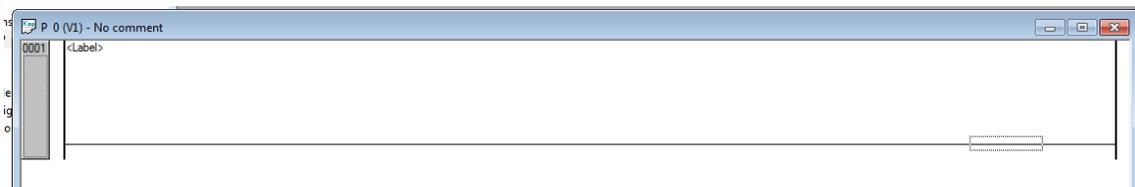


Figura 7.49 Programa en blanco

La siguiente barra de herramientas, Figura (7.50), se activara para la creación y edición del programa.



Figura 7.50 Barra de herramientas

Las opciones que principalmente se utilizara, son:

Contactos



Líneas de operación o conducción.

Bobinas



Seleccionar contactos y bobinas, para contactos le asignaremos la entrada que es I0.0 (Switch), para la bobina será O0.0 (Actuador).

Seleccionamos el icono de contacto  y automáticamente pedirá el nombre de Symbol que asignamos a nuestras entradas y salidas, Figura (7.51)

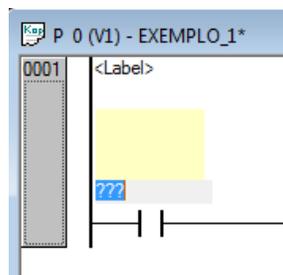


Figura 7.51 Contacto en FST4.10

Entonces le asignamos el nombre de “Switch” y Enter, para que automáticamente se nombre y así también muestre los comentarios que se le ha definido en AllocationList, Figura (7.52)

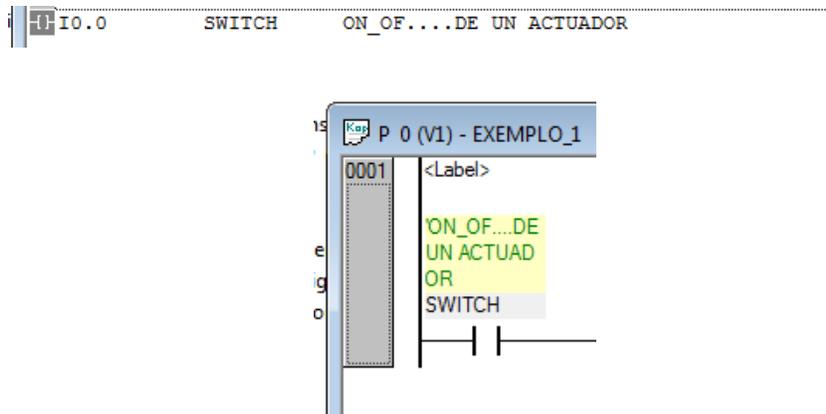


Figura 7.52 Determinación de contacto

Ahora Seleccionar la bobina  que es la salida del programa, y al igual que en el contacto, asignar el nombre de Actuador, Figura (7.53)

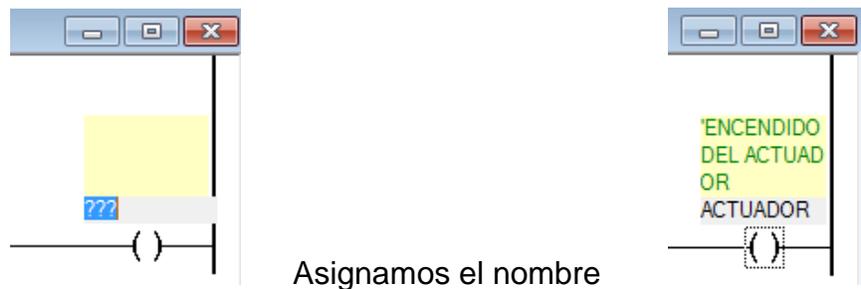


Figura 7.53 Asignación de nombre de la bobina

**\*Nota:** la bobina por default estará al final del diagrama ya que es quien da la respuesta de salida del programa del PLC

El programa finalizado se mostrara como en la Figura (7.54) y las respuestas que mostrara el programa como se muestra el Cuadro (7.1)

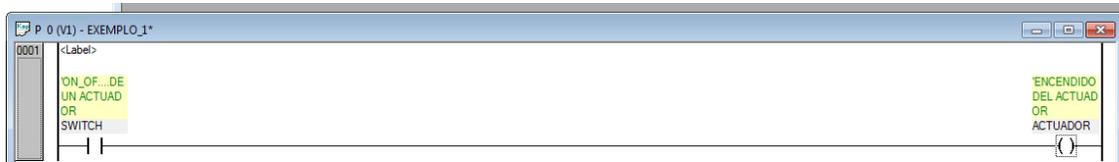


Figura 7.54 Programa de encendido y apagado de un led

Cuadro 7.1 Tabla de verdad

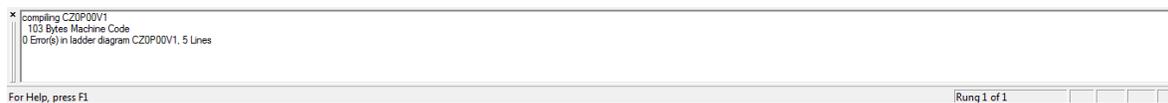
Switch (F)	Actuador (A)
	$F = A$
<b>0</b>	<b>0</b>
<b>1</b>	<b>1</b>

#### 7.4.6 Descarga de programa en el PLC

##### ¿Cómo hacer la compilación y la descarga del programa?

Al haber finalizado el diagrama y verificado la conexión del puerto RS232, dar clic en cada uno de los iconos mostrados en la Figura (7.55)

(a) Compilar la activación del modulo



(b) Compilar programas cuando se cambia y se preparó el proyecto para su descarga



(c) Compilar todos los programas y se prepara el proyecto para su descarga



Figura 7.55 Compilación del programa (a), (b), (c)

Al no mostrar errores en la compilación del programa, como lo muestran las ventanas anteriores, realizar la descarga del programa, dar clic en el botón de

descarga .

**\*Nota:** no se podrá realizar la descarga del programa si existen errores, en su defecto el PLC mostrara con el Led en Run de color rojo, Figura (4.40) lo que indica que hay errores en el programa o no hay programa y no podrá ejecutarse la operación.

Al momento de descargar el programa en el PLC, se ejecutara un cuadro de dialogo, Figura (7.56) que deberemos aceptar, para que inicie el proceso.

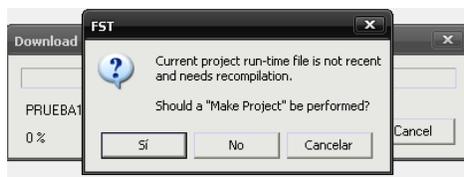


Figura 7.56 Cuadro de dialogo

Iniciará la descarga del programa en el PLC, Figura (7.57)

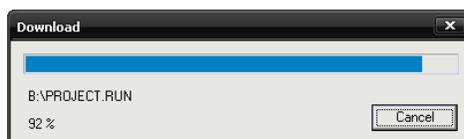


Figura 7.57 Descarga del programa en PLC

Después de haber cargado el programa en el PLC dar clic derecho / Online y después Run, como muestra la Figura (7.58)

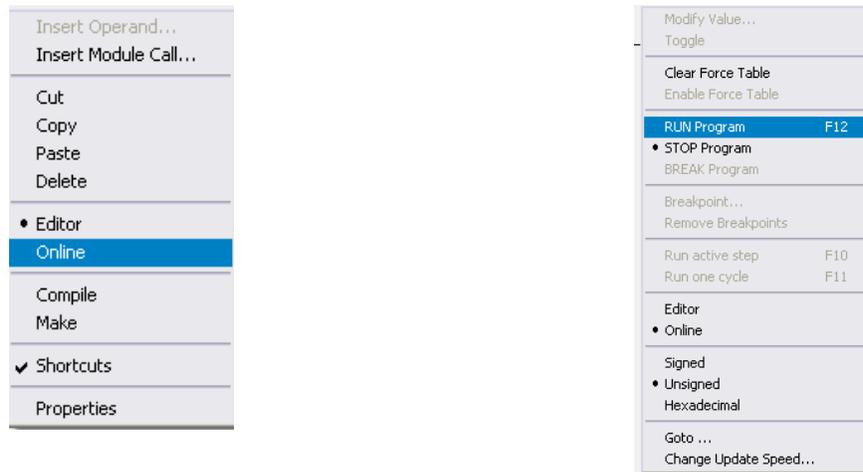


Figura 7.58 Ejecución de programa Online

**\*NOTA:** El PLC Festo FC34 cuenta con un led “Run” que dependiendo del color, mostrara el estado del programa cargado, Figura (7.59)

**Rojo** - Error en el programa

**Amarillo**—programa pausado o en Stop

**Verde** – Programa listo para ejecutarse.

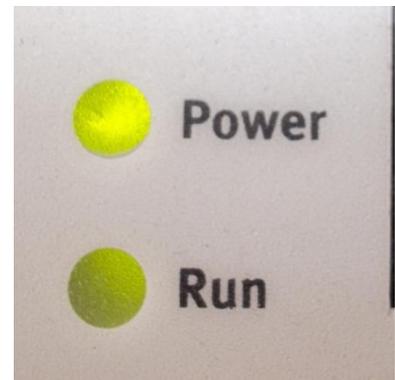
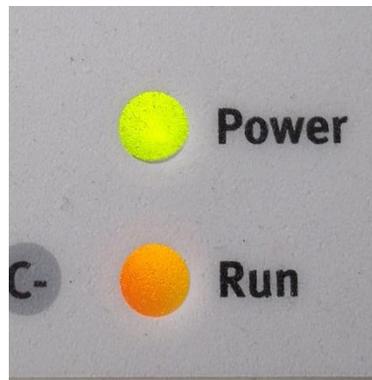
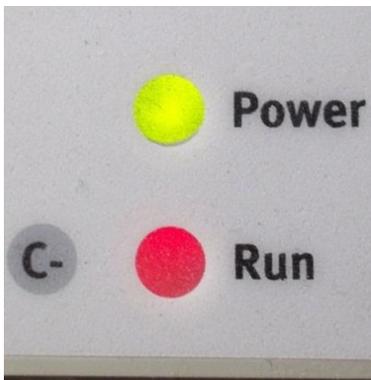


Figura 7.59 Estados posibles del PLC

La ejecución del programa se realizara mediante un módulo de simulación de sensores, Figura (7.22), que comprende de 12 salidas y 8 entradas, cada una con

sus respectivos led para verificar el comportamiento de la señal, el modulo estará funcionando con un voltaje de 24 V, al igual que el PLC FC34, Figura (7.23)

El funcionamiento del programa cargado mostrado en la Figura (7.60), es abrir el switch del módulo de simulación que está conectado al PLC en una de sus entradas en este caso será el I0.0, y en la salida O0.0.



*Figura 7.60 Funcionamiento y ejecución del programa cargado en el PLC*

## ANEXO 7.5

### 7.5.1 Puertas Logicas

Una puerta o compuerta lógica, es un dispositivo electrónico con una función booleana. Suman, multiplican, niegan o afirman, incluyen o excluyen según sus propiedades lógicas. Se pueden aplicar a tecnología electrónica, eléctrica, mecánica, hidráulica y neumática. Son circuitos de conmutación integrados en un chip.

La tabla de verdad se muestra en el Cuadro (7.2)

*Cuadro 7.2 Tabla de verdad de compuertas logicas*

<b>A</b>	<b>B</b>	AND $A * B$	OR $A + B$	NOR $\bar{A} * \bar{B}$	NAND $\bar{A} + \bar{B}$
0	0	0	0	1	1
1	0	0	1	0	1
0	1	0	1	0	1
1	1	1	1	0	0

El PLC, funciona enviando un 1 o un 0, lo que significa Verdadero (1) o Falso (0), la creación de programas se realizó siguiendo las mismas indicaciones que se han ido haciendo anteriormente.

## 7.5.2 Practica AND y OR



Figura 7.61 Programa AND y OR

La creación de este programa Figura (7.61), es el mismo procedimiento que el primero que se realizó anteriormente, la diferencia que aquí se incluyó dos tipos de configuración de contactos ya sea en serie o en paralelo, el funcionamiento de este programa es la respuesta según su fórmula.

AND

$$F = (A) * (B)$$

OR

$$F = A + B$$

Los resultados obtenidos son los mostrados en el Cuadro (7.3)

Cuadro 7.3 Tabla de verdad puertas lógicas AND y OR

A	B	AND	OR
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	1

### 7.5.3 Variables

Cuadro 7.4 Variables de entradas y salidas de señal

Salida	Variable	Operación
O0.0	Salida	Respuesta de operación para compuerta AND
O0.1	Salida1	Respuesta de operación para la compuerta OR
<b>Entradas</b>		
I0.0	A1	Valor <b>A</b> para compuerta AND
I0.1	A2	Valor <b>B</b> para compuerta AND
I0.2	A3	Valor <b>A</b> para compuerta OR
I0.3	A4	Valor <b>B</b> para compuerta OR

Una compuerta AND, se compone de 2 o mas contactos abiertos en serie con las literales (A1 y A2, Figura (7.62) ya que un contacto cerrado es lo inverso a este, esto quiere decir que si un contacto abierto envia un 1, el cerrado enviara un 0, la diferencia entre estos se muestra en la Figura (7.63)

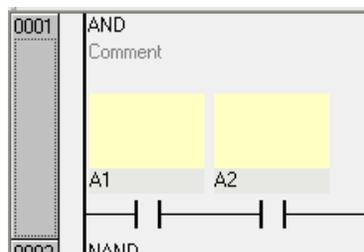


Figura 7.62 Contactos en serie, compuerta AND

Para activar el inverso seleccionar el contacto abierto y se activara el inversor, Figura (5.3) dar clic en el inversor para se vuelva un contacto cerrado.

Contacto abierto



Contacto cerrado



Inversor

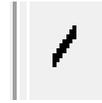


Figura 7.63 Tipos de contactos, e inversor

Para las literales (A3 y A4), el diagrama corresponde a una puerta logica OR, Figura (7.64)

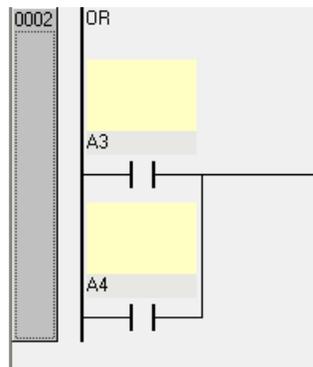


Figura 7.64 Contactos en paralelo, compuerta OR

La salida de la respuesta sera mediante dos bobinas, Figura (7.65), las cuales estan definidas como Salida(O0.0) y Salida1 (O0.1).

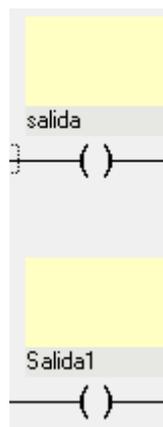


Figura 7.65 Salidas de las diferentes compuertas

#### 7.5.4 Compilacion del programa

El proceso de compilacion es para corroborar que el progama no cuente con errores, los cuales podrian descargarse en el PLC sin tener la operación deseada.

La realizacion de compilacion, dar clic en cada uno de los iconos de la barra de herramientas, Figura (7.66)

(A) (B) (C)



*Figura 7.66 Iconos de compilacion*

Para cada icono corresponde la compilacion de la siguiente forma:

- (a) Compilar la activación del modulo
- (b) Compilar programas cuando se cambia y se preparó el proyecto para su descarga
- (c) Compilar todos los programas y se prepara el proyecto para su descarga

Al finalizar la compilación y no haber mostrado error alguno, continuar con la descarga del programa diseñado de la PC al PLC mediante una comunicación del puerto serial.

#### 7.5.5 Descarga de la PCc al PLC

Dar clic en el icono de descarga Figura (7.67)



*Figura 7.67 Botón de descarga de programa al PLC*

Al momento de descargar el programa en el PLC, se ejecutara un cuadro de dialogo, Figura (7.68) que deberemos aceptar, para que inicie el proceso.

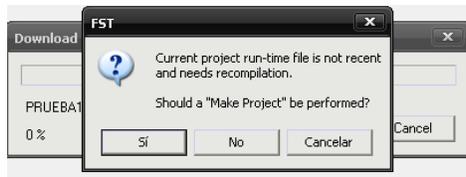


Figura 7.68 Cuadro de dialogo

Iniciará la descarga del programa en el PLC, Figura (7.69)

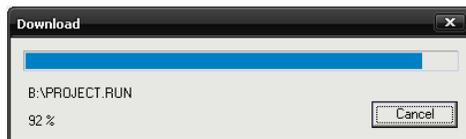


Figura 7.69 Descarga del programa en PLC

Después de haber cargado el programa en el PLC dar clic derecho / Online y después Run, como muestra la Figura (7.70)

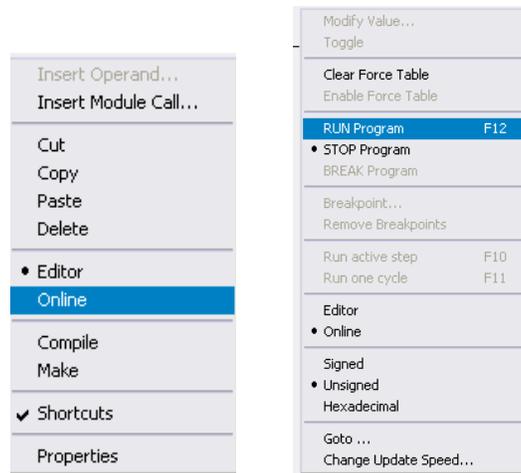


Figura 7.70 Ejecución de programa Online

## 7.5.6 Ejecucion de programa

### 7.5.6.1 Primer caso

Dar clic derecho, seleccionar Online y Run, Figura (7.70) para correr el programa descargado en el PLC. Inicialmente el programa, mostro un comportamiento como

muestra en la Figura (7.71) el programa muestra en la figura que se encuentra corriendo, por que la linea vertical que conecta a los relevadores se encuentra de color azul.



Figura 7.71 programa AND y OR en linea primer caso



Figura 7.72 Respuesta al primer caso de las compuertas AND y OR

**\*Nota: S = Salida y S1= Salida**

Para el comportamiento del primer caso, las respuestas para las dos compuertas se muestran falsas (0), los resultados mostrados en el modulo de simulacion de sensores, Figura (7.72) se muestran en el Cuadro (7.5)

Cuadro 7.5 Respuesta primer caso

A1 Y A3	A2 Y A4	SALIDA	SALIDA1
A	B	AND	OR
0	0	0	0

Todos los Switchs estan abiertos (OFF) y la señal de A y B es Falsa (0), lo que significa que AND y OR tendran una respuesta Falsa (0) en ambos casos.

### 7.5.6.2 Segundo caso

En siguiente caso para el programa activamos A1 y A3, enviando un 1, el comportamiento del programa se muestra en la Figura (7.73), se activo el relevador A1 y A3 , y se muestra ademas la linea de comunicación en color azul para la compuerta OR, encendiendo la bobina (salida1).

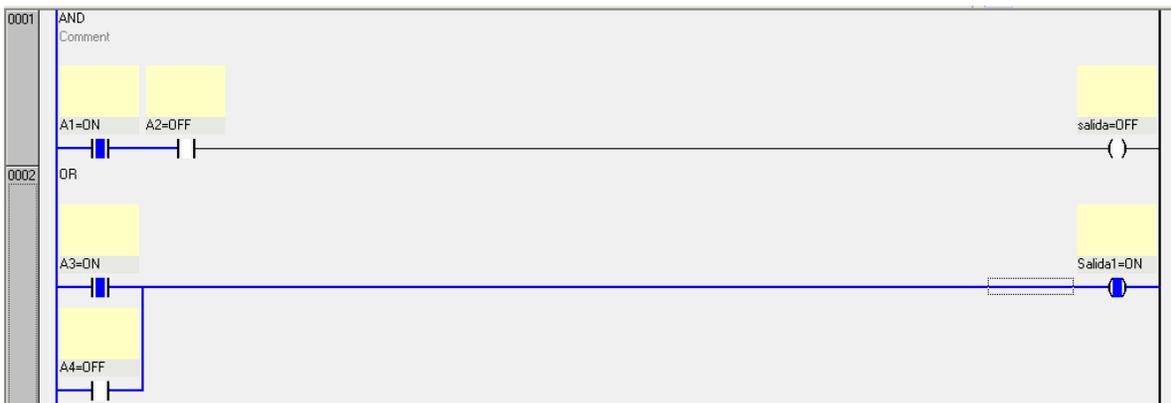


Figura 7.73 Programa en línea segundo caso



Figura 7.74 Respuesta al segundo caso de las compuertas AND y OR

**\*Nota: S = Salida y S1= Salida 1**

Para AND la respuesta continuo siendo Falsa(0) y para OR sera verdadera(1), lo que define la diferencia de AND y OR en base a su configuracion, los resultados se mostraron en ell modulo de simulacion de sensores, Figura (7.74), en base a la tabla de verdad del Cuadro (7.6)

Cuadro 7.6 Respuestas segundo caso

A1 Y A3	A2 Y A4	SALIDA	SALIDA1
A	B	AND	OR
1	0	0	1

### 7.5.6.3 Tercer caso

Para el tercer caso de las puertas logicas se activo A2 y A4 enviando un 1, y desactivar las anteriores (A1 y A3), el programa en linea se mostro con una linea

de color azul como en la Figura (7.75) y para el modulo de simulacion se definio como la Figura (7.76)

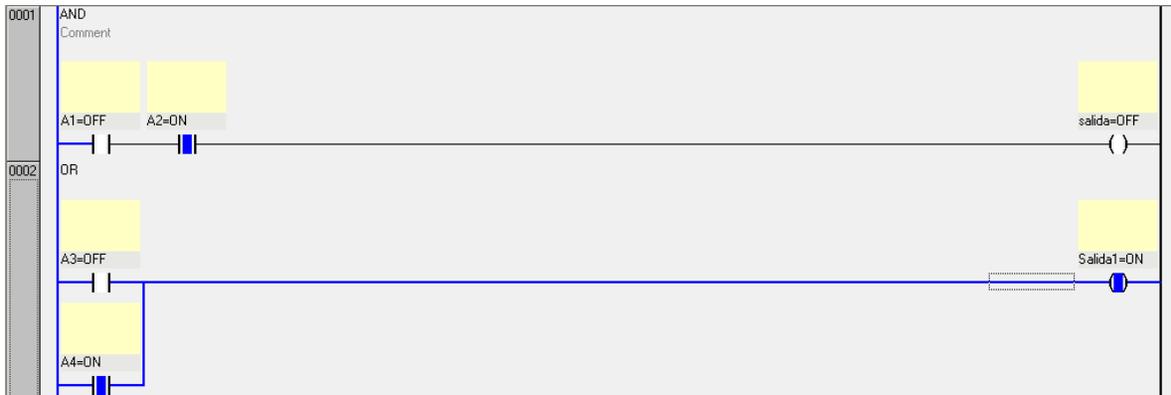


Figura 7.75 Programa en linea tercer caso

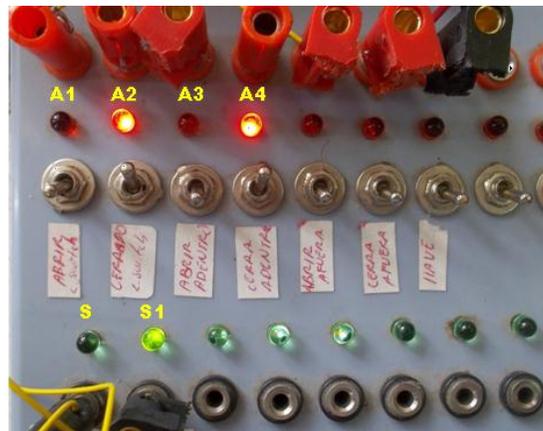


Figura 7.76 Respuesta al tercer caso de las compuertas AND y OR

La puerta OR arrojará un 1, siempre y cuando A3 o A4 envíe la misma señal, y para el caso de AND, la salida de respuesta estará mostrando un 0, ya que la ecuación de cada compuerta es diferente, y los resultados se muestran en el módulo de simulación de sensores, Figura (7.76) y en el Cuadro (7.7)

Cuadro 7.7 Respuestas tercer caso

A1 Y A3	A2 Y A4	SALIDA	SALIDA1
A	B	AND	OR
0	1	0	1

\*Nota: S = Salida y S1= Salida 1

#### 7.5.6.4 Cuarto caso

Para el ultimo caso el programa se activo A1 y A2 para AND, y A3 y A4 para OR, la respuesta sera Verdadera(1) para ambas puertas logicas, las lineas de comunicaci3n se muestran de color azul, Figura (7.77)

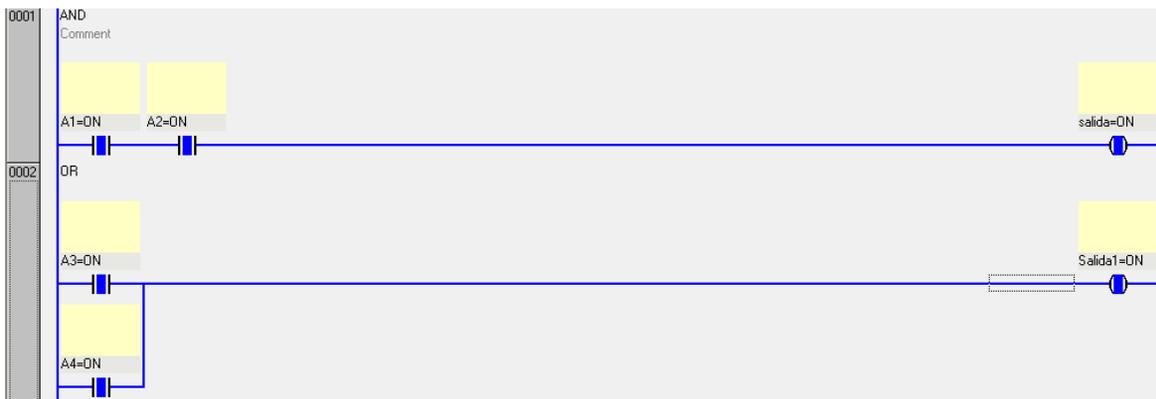


Figura 7.77 Programa en linea cuarto caso

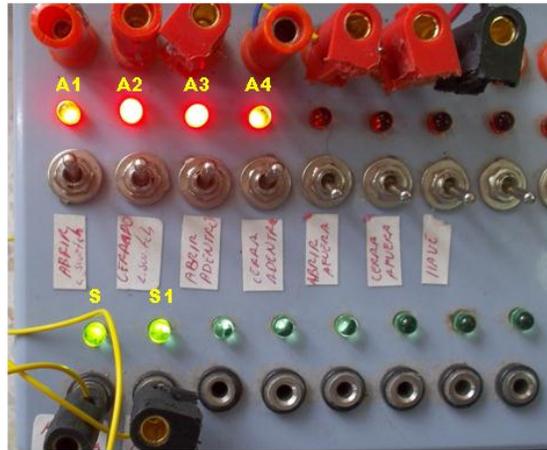


Figura 7.78 Respuesta al cuarto caso de las compuertas AND y OR

**\*Nota: S = Salida y S1= Salida 1**

El modulo de simulacion mostro el resultado del accionamiento de las variables, las cuales se mostraron como en la Figura (7.78), en base a la Tabla de verdad del Cuadro (7.8)

Cuadro 7.8 Respuestas cuarto caso

A1 Y A3	A2 Y A4	SALIDA	SALIDA1
A	B	AND	OR
1	1	1	1

Las dos puertas logicas se diferencian por el modo de ejecucion del programa, ya que la puerta AND, depende de A1 y A2 para dar una respuesta verdadera (1), y por lo mismo el diagrama es en serie, en cambio la puerta OR, el comportamiento al ser en paralelo, la salida es independiente de A3 o de A4, mostrando una respuesta verdadera (1).

Para las compuertas logicas NAND y NOR, es lo inverso a AND y OR.

### 7.5.7 Practica NAN y NOR

La creación de este programa es similar al programa de NAND y NOR, Figura (7.79), solo que la respuesta de este programa mostro lo inverso a AND y OR.

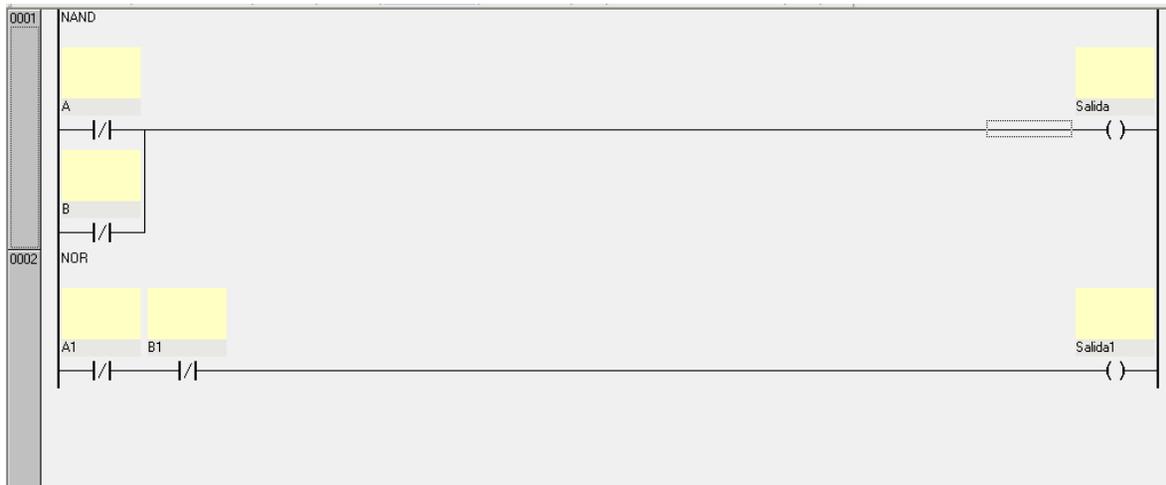


Figura 7.79 Programas NAND y NOR

La diferencia mostrada es la combinación de contactos inversos, por lo cual la compuerta NAND y NOR se define por las siguientes formulas:

NAND

$$F = \overline{AB} = \bar{A} + \bar{B}$$

NOR

$$F = \overline{A + B} = \bar{A} * \bar{B}$$

Los resultados obtenidos son los mostrados en el Cuadro (7.9)

Cuadro 7.9 Tabla de verdad puertas logicas NAND y NOR

A	B	NAND	NOR
0	0	1	1
1	0	1	0
0	1	1	0
1	1	0	0

### 7.5.8 Variables

Cuadro 7.10 Variables de entradas y salidas de señal

Salida	Variable	Operación
O0.0	Salida	Respuesta de operación para compuerta NAND
O0.1	Salida1	Respuesta de operación para la compuerta NOR
<b>Entradas</b>		
I0.0	A	Valor <b>A</b> para compuerta NAND
I0.1	B	Valor <b>B</b> para compuerta NAND
I0.2	A1	Valor <b>A</b> para compuerta NOR
I0.3	B2	Valor <b>B</b> para compuerta NOR

Una compuerta NAND se compone de 2 contactos con inversor en paralelo; las literales (A y B) se configuraran como muestra la Figura (7.80), en base a la formula nos indica para que los valores corresponda a un resultado inverso al de la compuerta AND.

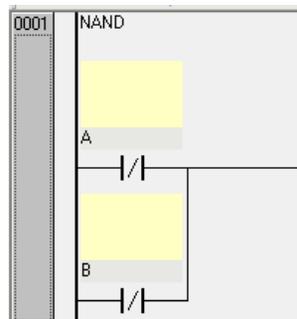
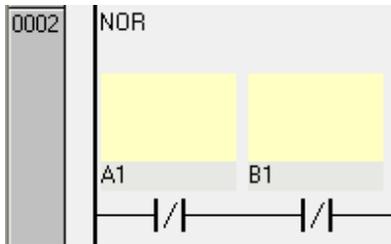


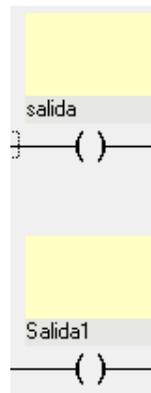
Figura 7.80 Contactos con inversor en paralelo, compuerta NAND

Las compuertas NOR se componen de contactos con inversor, las literales (A1 y B1) se configuran como muestra la Figura (7.81)



*Figura 7.81 Contactos con inversor en serie, compuerta NOR*

La salida de la respuesta sera mediante dos bobina Figura (7.82), las cuales estan definidas como Salida(O0.0) y Salida1 (O0.1).



*Figura 7.82 Salidas de las diferentes compuertas*

### **7.5.9 Compilacion del programa**

El proceso de compilacion es para corroborar que el progama no cuente con errores, los cuales podrian descargarse en el PLC sin tener la operaci3n deseada.

La realizacion de compilacion, dar clic en cada uno de los iconos de la barra de herramientas, Figura (7.83)

(A) (B) (C)



*Figura 7.83 Iconos de compilacion*

Para cada icono corresponde la compilacion de la siguiente forma:

- (a) Compilar la activación del modulo
- (b) Compilar programas cuando se cambia y se preparó el proyecto para su descarga
- (c) Compilar todos los programas y se prepara el proyecto para su descarga

Al finalizar la compilación y no haber mostrado error alguno, continuar con la descarga del programa diseñado de la PC al PLC mediante una comunicación del puerto serial.

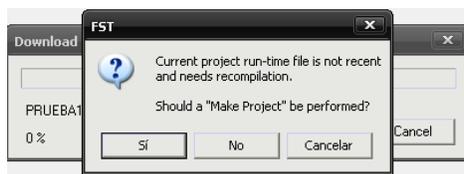
#### **7.5.10 Descarga de la PC al PLC**

Dar clic en el icono de descarga Figura (7.84)



*Figura 7.84 Botón de descarga de programa al PLC*

Al momento de descargar el programa en el PLC, se ejecutara un cuadro de dialogo, Figura (7.85) que deberemos aceptar, para que inicie el proceso.



*Figura 7.85 Cuadro de dialogo*

Iniciará la descarga del programa en el PLC, Figura (7.86)

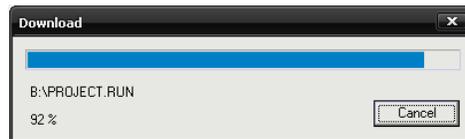


Figura 7.86 Descarga del programa en PLC

Después de haber cargado el programa en el PLC dar clic derecho / Online y después Run, como muestra la Figura (7.87)

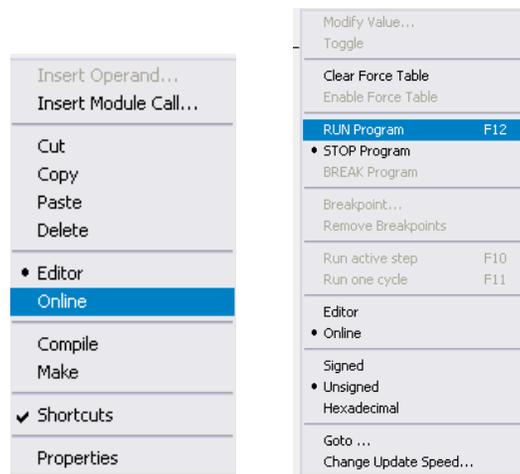


Figura 7.87 Ejecución de programa Online

### 7.5.11 Ejecucion de programa

Dar clic derecho, seleccionar Online y Run, Figura (7.87) para correr el programa descargado en el PLC. Inicialmente el programa de las compuertas NAND y NOR mostro un comportamiento en línea como muestra la Figura (7.88) indicando con una línea de color azul.

#### 7.5.11.1 Primer caso

Al iniciar la conexión del PLC, el módulo de simulación de sensores los leds se comportaran a la tabla de verdad del Cuadro (7.11)

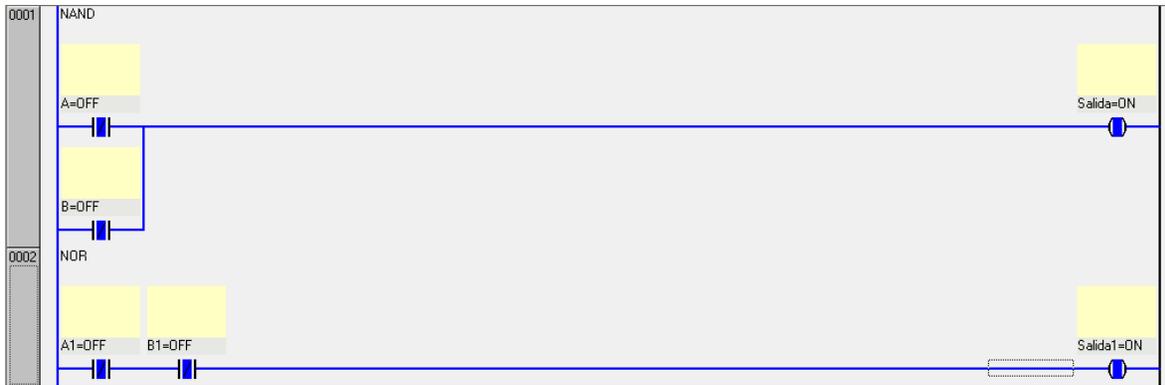


Figura 7.88 Programa NAND y NOR en línea primer caso

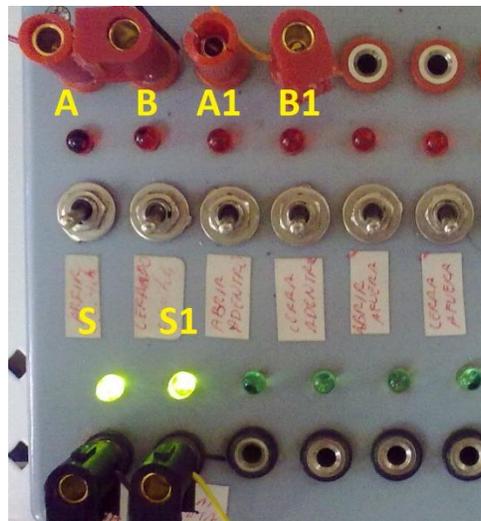


Figura 7.89 Respuesta al primer caso de las compuertas NAND y NOR

**\*Nota: S = Salida y S1= Salida**

Las respuestas del programa en el modulo de simulacion se muestran en la Figura (7.89), el comportamiento de las compuertas NAND y NOR se vera reflejado en el Cuadro (7.11), siendo las respuestas inversa de las compuertas AND y OR.

Cuadro 7.11 Respuesta primer caso

A y A1	B Y B1	SALIDA	SALIDA1
A	B	NAND	NOR
0	0	1	1

Todos los Switchs se encuentran abiertos (OFF) y la señal de A y B es Falsa (0), lo que significa que NAND y NOR tendran una respuesta verdadera (1) en ambos casos.

### 7.5.11.2 Segundo caso

En siguiente caso para el programa se activo A y A1, enviando una respuesta verdadera (1) en cada compuerta, el comportamiento del programa es como se muestra en la Figura (7.90), la activacion y desactivacion de la linea señal en color azul dentro del programa en linea.

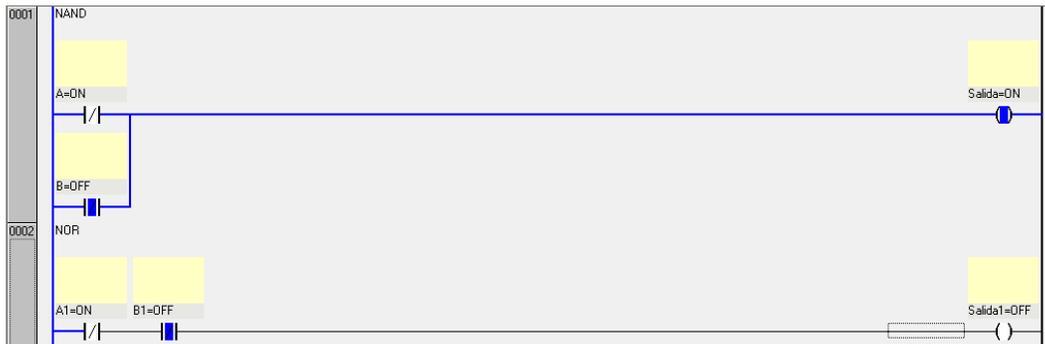


Figura 7.90 Programa en línea segundo caso

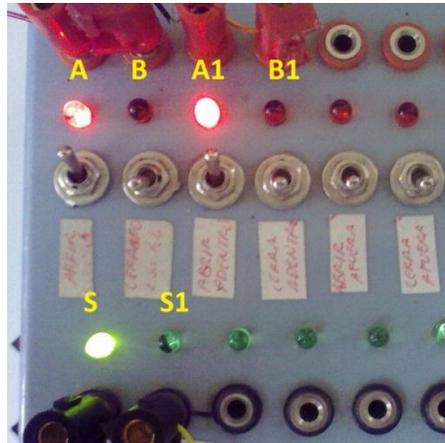


Figura 7.91 Respuesta al segundo caso de las compuertas NAND y NOR

**\*Nota: S = Salida y S1= Salida 1**

Para NAND la respuesta es verdadera (1) y para NOR sera falsa (0), el resultado se muestra en la Figura (7.91), la respuesta se ve reflejada como en Cuadro (7.12)

Cuadro 7.12 Respuestas segundo caso

A Y A1	B Y B1	SALIDA	SALIDA1
A	B	NAND	NOR
1	0	1	0

### 7.5.11.3Tercer caso

Para el tercer caso de las puertas logicas se activo B y B2 enviando una respuesta verdadera (1), se desactivo A y A1, el programa en linea mostro como en la Figura (7.92)

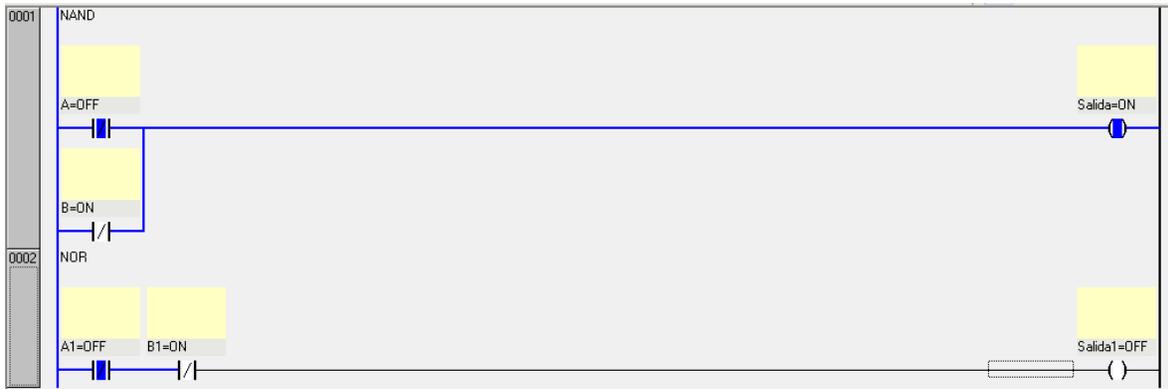


Figura 7.92 Programa en línea tercer caso

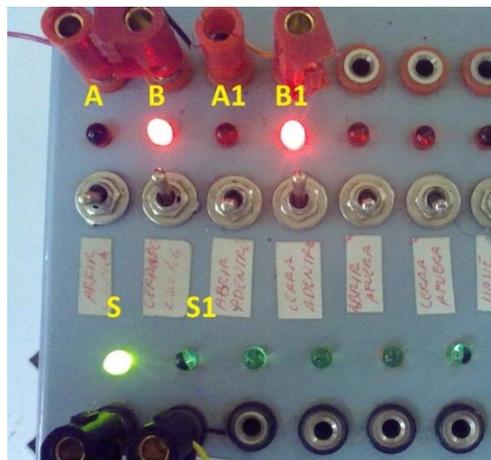


Figura 7.93 Respuesta al tercer caso de las compuertas NAND y NOR

**\*Nota: S = Salida y S1= Salida 1**

Las respuestas mostradas son igual al segundo caso y el resultado se mostrara como en la Figura (7.93), el unico cambio para este caso es la desactivacion de entrada de señal A y A1, y se turnara por la activacion de B y B1.

La compuerta NOR muestra con el mismo valor Falso (0) al igual que la compuerta NAND con un valor verdadero (1). Las respuestas corresponden al Cuadro (7.13)

Cuadro 7.13 Respuestas tercer caso

A Y A1	B Y B1	SALIDA	SALIDA1
A	B	NAND	NOR
0	1	1	0

#### 7.5.11.4 Cuarto caso

Para el ultimo caso el programa se activo A y B para NAND, asi tambien para A1 y B1 para NOR, la respuesta sera verdadera(1) para ambas puertas logicas, se mostro la desactivacion de linea de color azul que representa la señal enviada hacia las bobinas (Salida y Salida1) Figura (7.94)

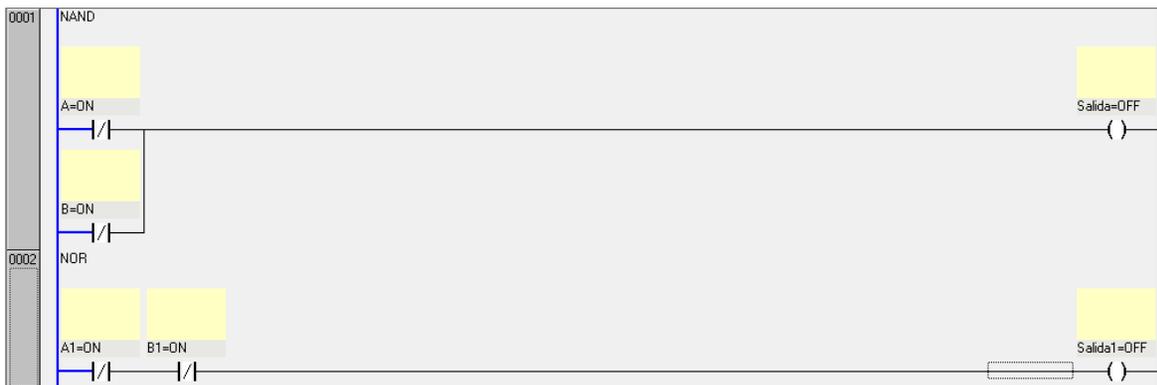


Figura 7.94 Programa en línea cuarto caso

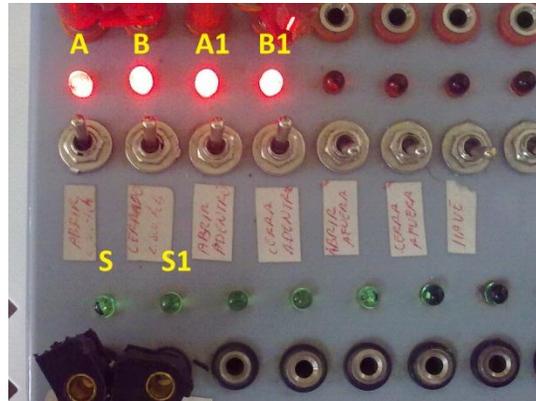


Figura 7.95 Respuesta al cuarto caso de las compuertas AND y OR

**\*Nota: S = Salida y S1= Salida 1**

El comportamiento de las variables Figura (7.95), en el ultimo caso muestra las respuesta opuesta a las compuertas

AND y OR, el resultado es falso como se muestra en el Cuadro (7.14), lo que significa el comportamiento del inversor en los contactos del programa.

Cuadro 7.14 Respuestas cuarto caso

A Y A1	B Y B1	SALIDA	SALIDA1
A	B	NAND	NOR
1	1	0	0

La practica realizada, espuso el comportamiento de las compuertas asi tambien como en el diseño en FST 4.10 el cual se baso en cada una de las formulas de cada compuerta logica, las cuales se diseñaron en serie, paralelo contactos abiertos e inversos.

## ANEXO 7.6

### 7.6.1 Puerta de garage

Al paso del tiempo ha surgido ciertas necesidades y una de ellas ha sido automatizar, para poder facilitar el trabajo del ser humano, con un mínimo esfuerzo mediante la aplicación de las compuertas digitales, un ejemplo con estos principios es la automatización de una puerta de garage, el abrir y cerrar la puerta, ya sea desde adentro o desde afuera, esto se logrará mediante la combinación de compuertas AND y OR.

El proceso en el programa con un PLC, es enfocar al abrir y cerrar una puerta, la cual mediante compuertas lógicas se logrará activar relevadores para el accionamiento de la misma.

Las posibles situaciones que se presentan se muestra en el Cuadro (7.15)

*Cuadro 7.15 Operación del programa puerta de garage*

Situación	Respuesta
Abrir desde afuera	Verdadero (1)
Abrir desde adentro	Verdadero (1)
Cerrar desde afuera	Verdadero (1)
Cerrar desde adentro	Verdadero (1)

### 7.6.2 Variables

*Cuadro 7.16 Variables del programa puerta de garage*

<b>Salidas</b>	<b>Variable</b>	<b>Operación</b>
O0.0	Relay_op	Relay para abrir la puerta
O0.1	Relay_clo	Relay para cerrar la puerta
<b>Entradas</b>		

I0.0	Open	Limit switch cuando se abre la puerta
I0.1	Closed	Limit switch cuando se cierra la puerta
I0.2	Open_in	Abrir puerta desde adentro
I0.3	Close_in	Cerrar puerta desde adentro
I0.4	Open_out	Abrir puerta desde afuera
I0.5	Close_out	Cerrar puerta desde afuera
I0.6	Key	Llave para abrir desde afuera

Programa descargado en el PLC Festo FC34, Figura (7.96)

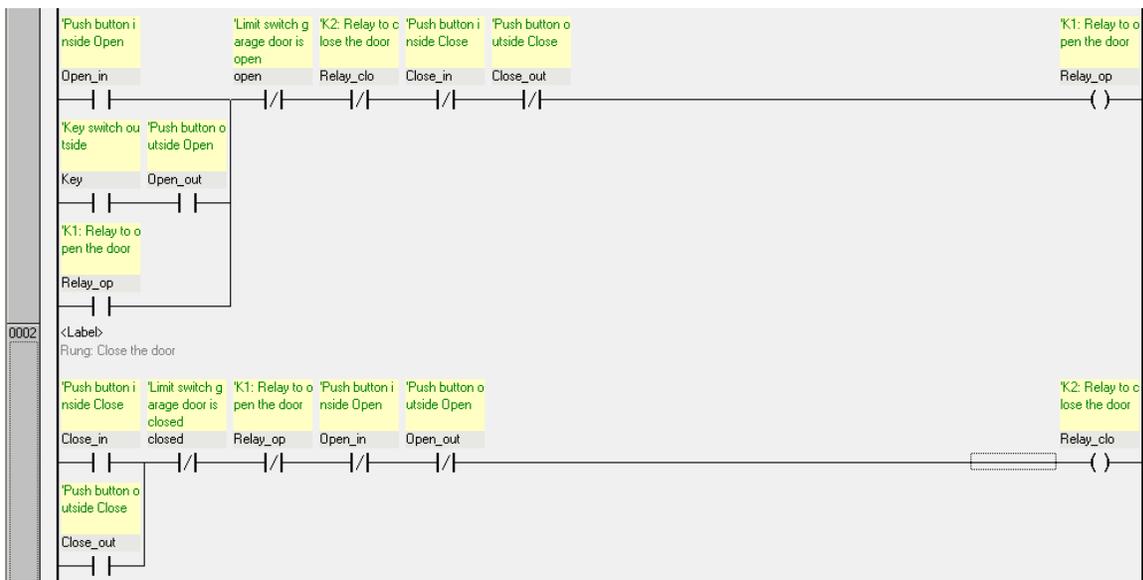


Figura 7.96 Programa de puerta de garaje

### 7.6.3 Compilacion del programa

El proceso de compilacion es para corroborar que el progama no cuente con errores, los cuales podrian descargarse en el PLC sin tener la operación deseada.

La realizacion de compilacion, dar clic en cada uno de los iconos de la barra de herramientas, Figura (7.97)

(A) (B) (C)



*Figura 7.97 Iconos de compilacion*

Para cada icono corresponde la compilacion de la siguiente forma:

- (a) Compilar la activación del modulo
- (b) Compilar programas cuando se cambia y se preparó el proyecto para su descarga
- (c) Compilar todos los programas y se prepara el proyecto para su descarga

Al finalizar la compilación y no haber mostrado error alguno, continuar con la descarga del programa diseñado de la PC al PLC mediante una comunicación del puerto serial.

### 7.6.4 Descarga de la PC al PLC

Dar clic en el icono de descarga Figura (7.98)



*Figura 7.98 Botón de descarga de programa al PLC*

Al momento de descargar el programa en el PLC, se ejecutara un cuadro de dialogo, Figura (7.99) que deberemos aceptar, para que inicie el proceso.

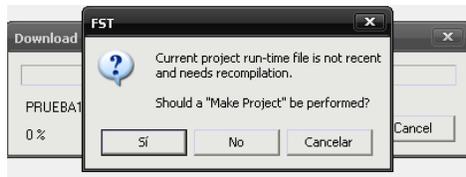


Figura 7.99 Cuadro de dialogo

Iniciará la descarga del programa en el PLC, Figura (8.0)

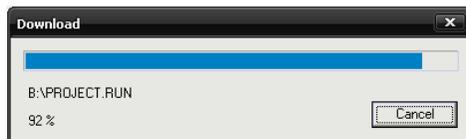


Figura 8.0 Descarga del programa en PLC

Después de haber cargado el programa en el PLC dar clic derecho / Online y después Run, como muestra la Figura (8.1)

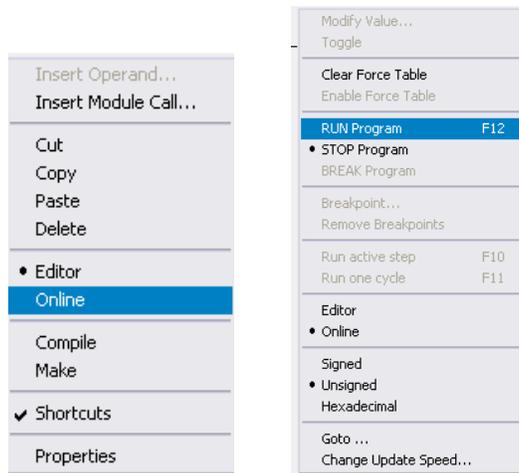


Figura 8.1 Ejecución de programa Online

## 7.6.5 Ejecucion de programa

Dar clic derecho, seleccionar Online y Run, Figura (8.1) para correr el programa descargado en el PLC. El programa descargado al ejecutarlo en línea entre PLC y PC atravez de puerto serial RS232, se muestra como en la Figura (8.2)



Figura 8.2 Programa en linea puerta de garaje

### 7.6.5.1 Abrir puerta desde afuera.

Si la persona viene de la calle, y necesita entrar a su domicilio, se necesita activar con una **llave (abrir I 0.6)** el acceso a la puerta de lo contrario no sucedera nada si se desea abrir, ahora activar **abrir desde afuera (abrir IO.4)**, y se activara **abriendo puerta(OO.0)**,Figura (8.3) el resultado en el modulo de simulacion de se muestra en la Figura (8.5)(A)

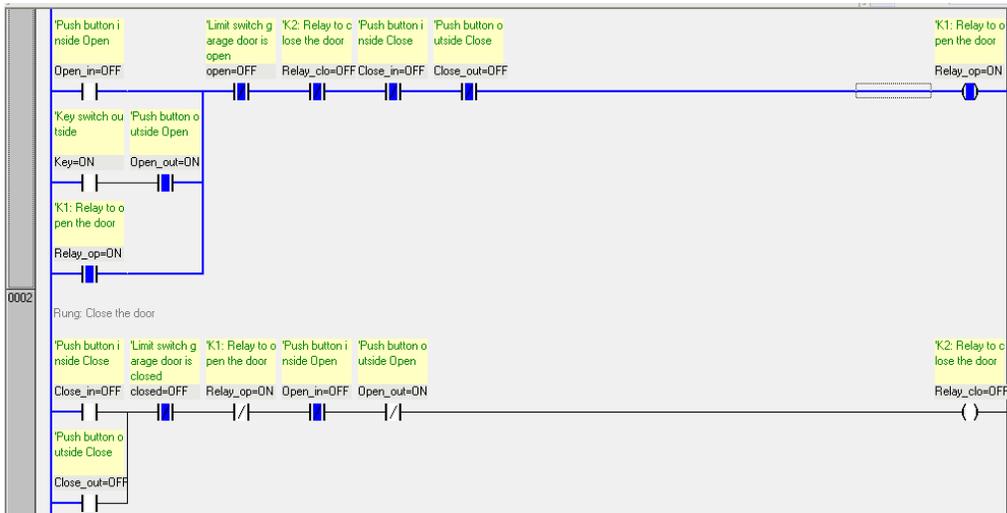


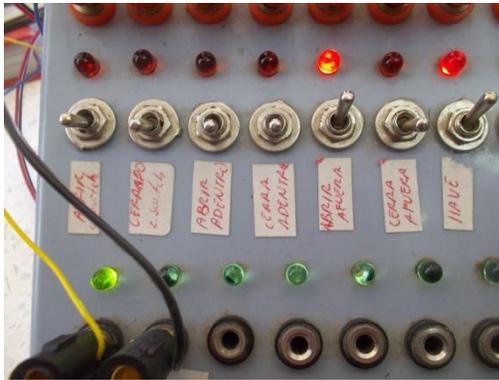
Figura 8.3 Puerta abierta desde afuera

la puerta se abra hasta cierto limite, Figura (8.4), el cual lo determinara **limit switch abierto (abrir IO.0)**. Figura (8.5)(B)

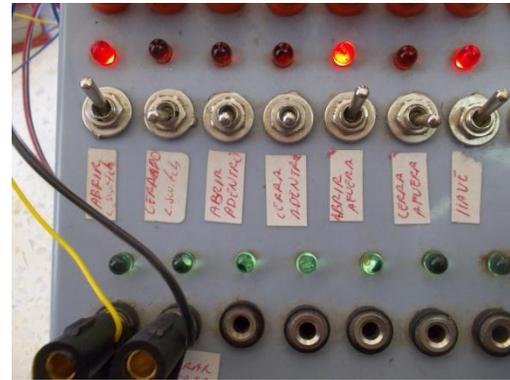


Figura 8.4 Puerta abierta desde afuera con limit switch activo

El funcionamiento de Limit Switch detiene el accionamiento cuando la puerta se abre o cierra, las respuestas se muestran en el modulo de simulacion de la Figura (8.5)



(A)



(B)

Figura 8.5 Respuesta en modulo de simulacion de sensores

### 7.6.5.2 Cerrar puerta desde adentro.

Cuando la puerta esta abierta, se debe **cerrar desde adentro**(abrir I0.3) del Garage, lo que significa que se debe desactivar la llave(Cerrar I.0.6), **limit Swicht**(Cerrar I0.0) y **Abrir desde afuera**(I0.4) entonces se activara **cerrando puerta**(O0.1), en este caso el programa se mostrara como la Figura(8.6)

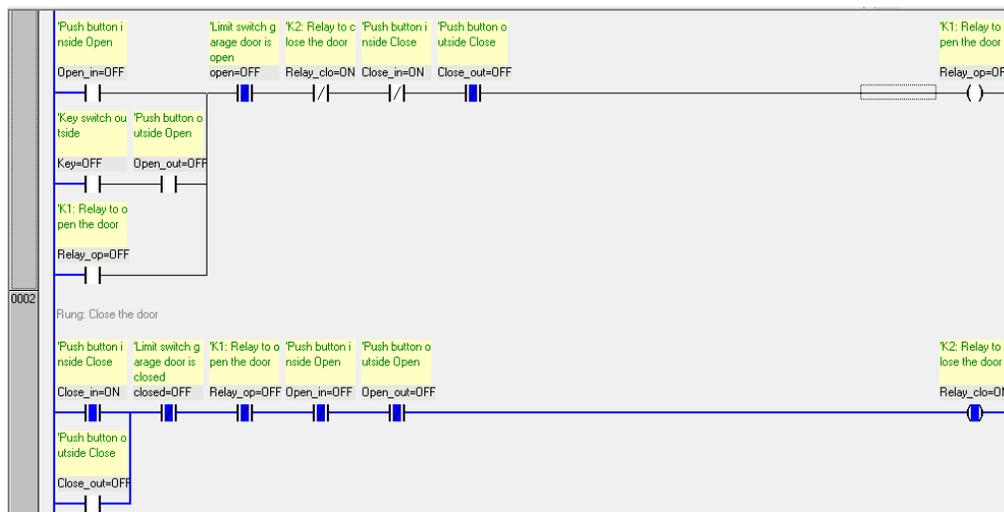


Figura 8.6 Puerta cerrada desde adentro

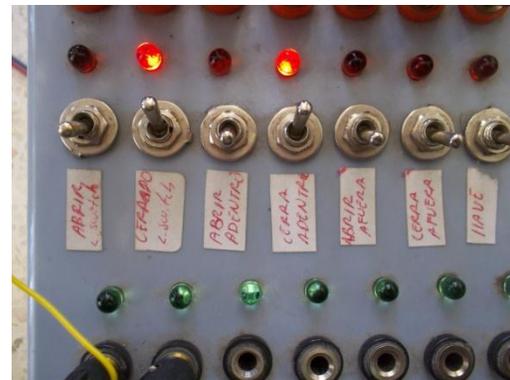
La puerta comenzara a cerrarse y activar **limit swicht cerrado** (abrir I0.1), para detener la puerta que se esta cerrando. Figura (8.7)



Figura 8.7 Puerta cerrada desde adentro con limit switch activo



(A)



(B)

Figura 8.8 Respuesta en el modulo de simulacion

Las respuestas obtenidas se muestran en modulo de simulacion al activar cuando se cierra desde afuera hasta cuando se desactiva con limit switch, Figura (8.8)

### 7.6.5.3 Abrir desde adentro.

Entonces, al momento de que la persona quisiera salir del garage desactivar las señales de entrada activadas (10.3 y 10.3), para continuar a la activacion de **abrir**

desde adentro(abrir IO.2) de garage, la puerta activara la funcion de **abriendo puerta(O0.0)**, Figura (8.9)

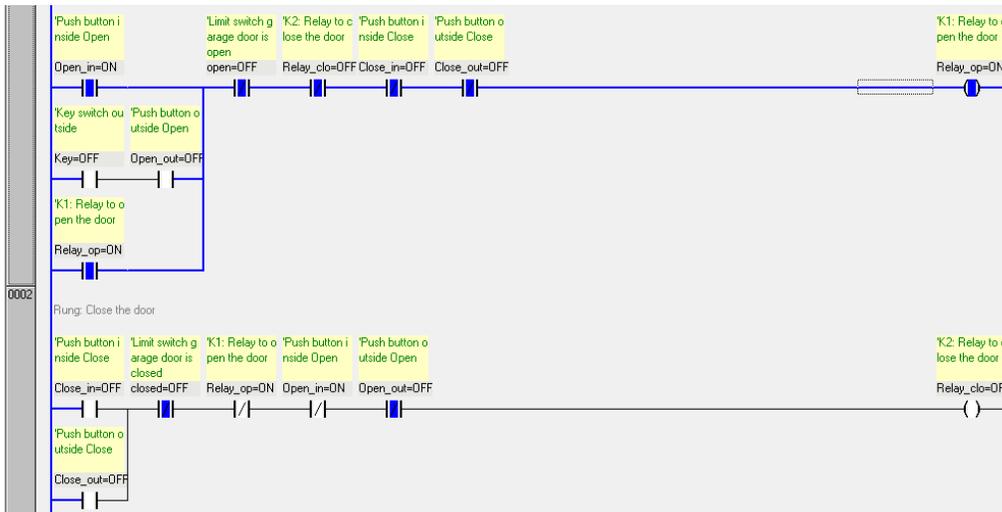


Figura 8.9 Puerta abierta desde adentro

hasta que **limit swicht abierto (abrir IO.0)** se active y la puerta se detenga.Figura (8.10)

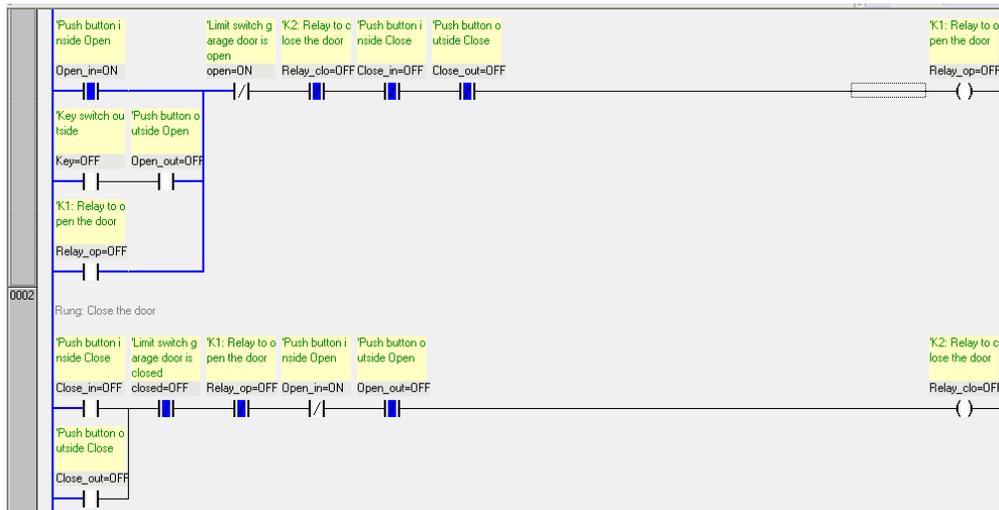


Figura 8.10 Puerta abierta desde adentro con limit swicht activo

La respuesta obtenida en este caso se muestra en la Figura (8.11), donde (A) se observa activa la respuesta y para (B), se observa una respuesta falsa (0)



(A)



(B)

Figura 8.11 Respuesta en el modulo de simulacion

#### 7.6.5.4 Cerrar puerta desde afuera

(abrir I0.5) del Garage, abrir el switch (I0.05) para activar cerrando puerta (O0.1), el funcionamiento del programa en linea, Figura (8.12)

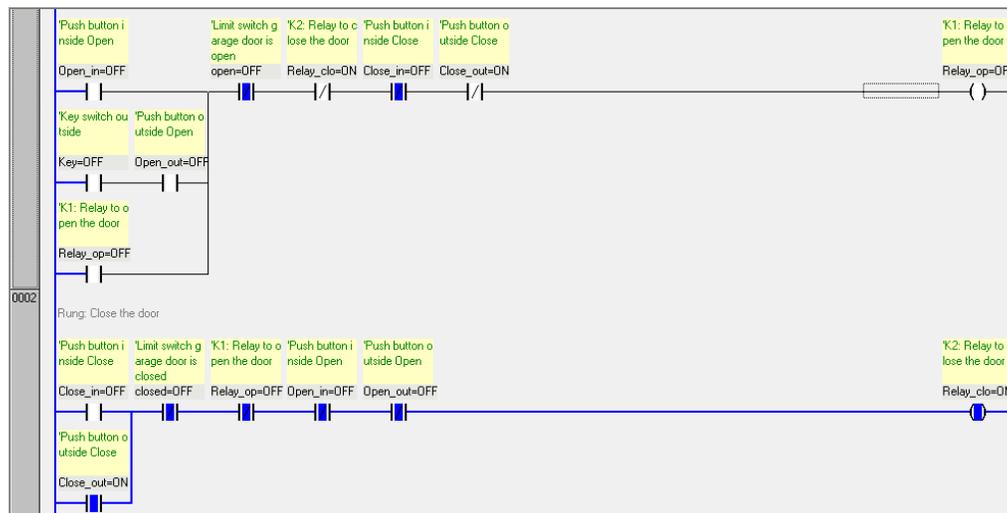


Figura 8.12 Cerrar puerta desde afuera

activar su limite con **limit swicht cerrado(abrir I0.1)**, Figura (8.13)



Figura 8.13 Cerrar puerta desde afuera con limit switch activo

El resultado obtenido en el modulo de simulacion se muestra en la Figura (8.14), en base las situaciones al estar activo o desactivo el limit switch



(A)



(B)

Figura 8.14 Respuesta en el modulo de simulacion

**\*Nota:** Para abrir la puerta (O0.0)desde afuera(I0.4) solo se podra activando la llave (abrir I0.6).

Los resultados mostrados son favorables ya que mediante la combinacion de compuertas logicas se obtuvieron las respuestas esperadas.

Abrir y cerrar una puerta de garage es un ejemplo cotidiano, la aplicación de estas funciones es buscar mejores alternativas para la automatizacion en el campo Mexicano.

## ANEXO 7.7

### 7.7.1 Activación de motores.

Los cambios climáticos de hoy en día a llevado a la necesidad de considerar la automatización del campo, la agricultura protegida es un claro ejemplo que necesitamos buscar y dar las mejores condiciones a los cultivos, como humedad, temperatura, fertirriegos tecnificados, etc.

El control de temperatura es un factor importante ya que se debe tener un clima favorable para que las plantas aprovechen las mejores condiciones, en su desarrollo.

A continuación presentamos un control de temperatura mediante 3 termostatos, y dos motores que simulan la ventilación dentro de un invernadero el cual, consideraremos una temperatura para la activación y drenado de calor, encontrado una estabilidad de un clima en el cual es favorable para el máximo aprovechamiento y desarrollo de los cultivos, el programa diseñado en FST 4.10 se muestra en la Figura (8.15)

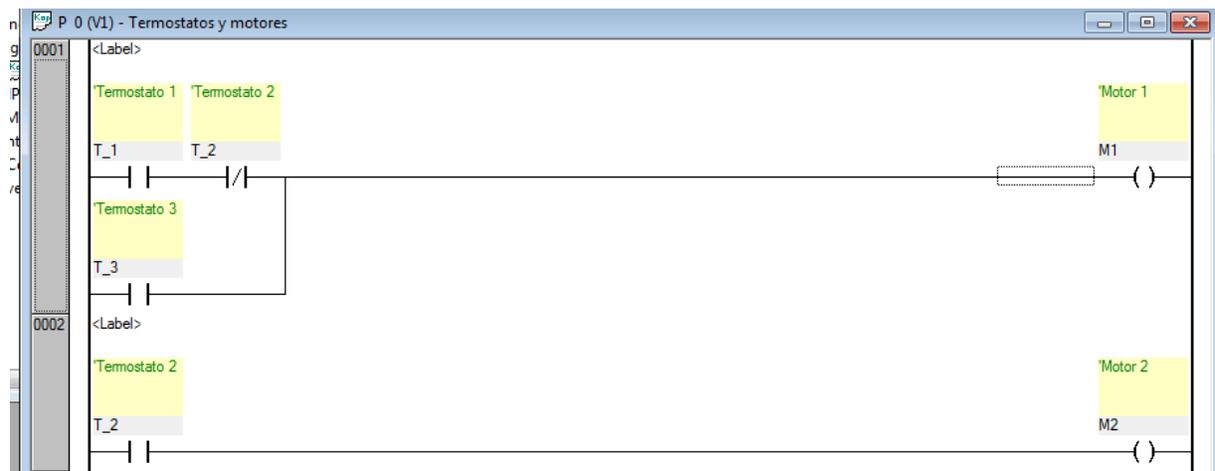


Figura 8.15 Programa de encendido de motores

## 7.7.2 Variables

Cuadro 7.17 Variables de encendido de motores

Salida	Variable	Operación
O0.0	M1	Motor 1
O0.1	M2	Motor 2
<b>Entradas</b>		
I0.0	T_1	Termostato 1
I0.1	T_2	Termostato 2
I0.2	T_3	Termostato 3

El programa consiste en que cada termostato indicara la temperatura actual dentro de un invernadero, y enviara una señal cuando la temperatura sea critica, esto significa que si la temperatura deseada es de 25° C. y el termostato muestra una temperatura de 26°C. el termostato enviara una señal la cual sera simulada mediante el modulo de simulacion de sensores que se ha ido utilizando anteriormente.

La tabla de verdad del programa, Cuadro (7.18), es en base a las ecuaciones:

$$M_1 = T_1 * \bar{T}_2 + T_3 \qquad M_2 = T_2$$

Cuadro 7.18 Tabla de verdad del programa encendido de motores

Termostato 1	Termostato 2	Termostato 3	Motor 1	Motor 2
0	0	0	0	0
1	0	0	1	0
1	1	0	0	1
1	1	1	1	1

### 7.7.3 Compilacion del programa

El proceso de compilacion es para corroborar que el progama no cuente con errores, los cuales podrian descargarse en el PLC sin tener la operación deseada.

La realizacion de compilacion, dar clic en cada uno de los iconos de la barra de herramientas, Figura (8.16)

(A) (B) (C)



*Figura 8.16 Iconos de compilacion*

Para cada icono corresponde la compilacion de la siguiente forma:

- (a) Compilar la activación del modulo
- (b) Compilar programas cuando se cambia y se preparó el proyecto para su descarga
- (c) Compilar todos los programas y se prepara el proyecto para su descarga

Al finalizar la compilación y no haber mostrado error alguno, continuar con la descarga del programa diseñado de la PC al PLC mediante una comunicación del puerto serial.

### 7.6.4 Descarga de la PC al PLC

Dar clic en el icono de descarga Figura (8.17)



*Figura 8.17 Botón de descarga de programa al PLC*

Al momento de descargar el programa en el PLC, se ejecutara un cuadro de dialogo, Figura (8.18) que deberemos aceptar, para que inicie el proceso.

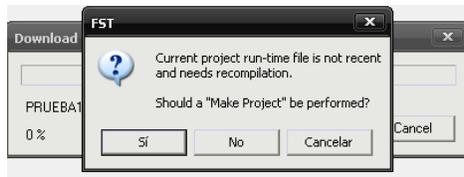


Figura 8.18 Cuadro de dialogo

Iniciará la descarga del programa en el PLC, Figura (8.19)

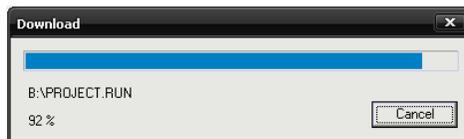


Figura 8.19 Descarga del programa en PLC

Después de haber cargado el programa en el PLC dar clic derecho / Online y después Run, como muestra la Figura (8.20)

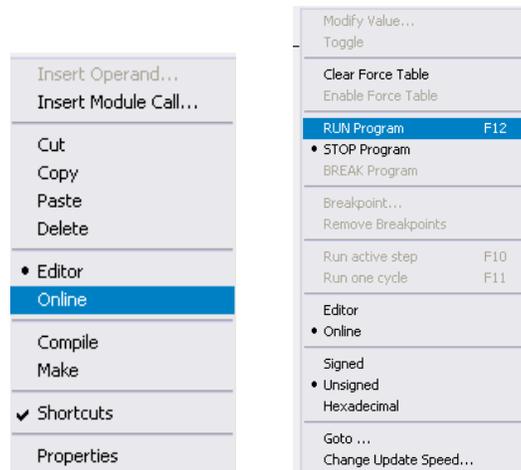


Figura 8.20 Ejecución de programa Online

Al haber realizado la correspondiente descarga del programa de la PLC al PLC, se procedió a la realización de la práctica mediante el módulo de simulación de sensores.

## 7.7.5 Ejecución del programa

### 7.7.5.1 Primer caso

El programa descargado en el PLC se ejecutara en línea, Figura (8.21) mostrando el primer comportamiento como se muestra en el Cuadro (7.19)

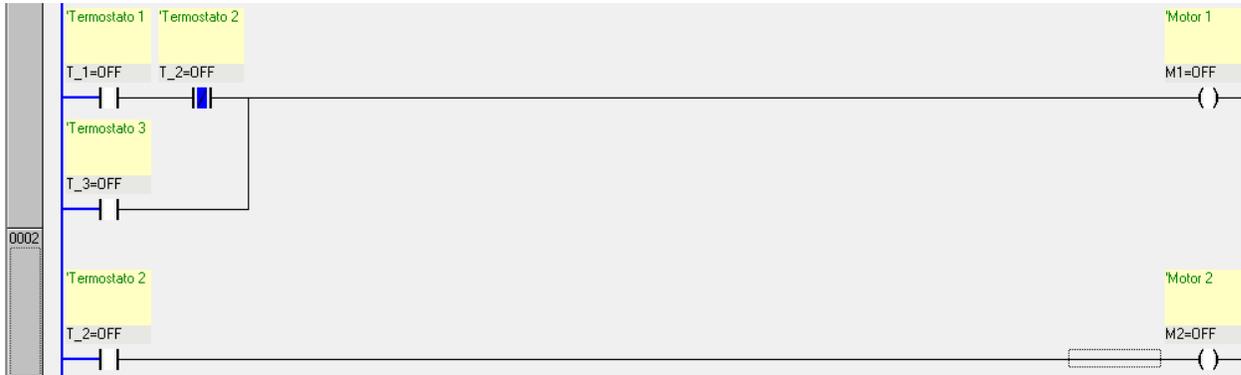


Figura 8.21 Programa de encendido de motores en línea

En el modulo de simulacion de sensores como en el programa se muestra una respuesta falsa(0), ya que los estados de las temperaturas son favorales y la activacion se mostrara cuando el primer termostato muestre una elevacion de temperatura, Figura (8.22)



Figura 8.22 Modulo de simulacion de sensores con variables definidas

Cuadro 7.19 Tabla de verdad primer caso

Termostato 1	Termostato 2	Termostato 3	Motor 1	Motor 2
0	0	0	0	0

### 7.7.5.2 Segundo caso

El termostato 1 incrementa su temperatura y envia una señal **T\_1(abrir I0.0)**, entonces el motor 1 se activara automaticamente motor 1, **M1(O0.0)**. Figura (8.23) mientras que los dos termostatos muestran una temperatura por debajo de la indicada **T\_2(Cerrado I0.0)** y **T\_3(Cerrado I0.1)**, **M2(Cerrado O0.1)**, las respuestas a este caso se muestran en el Cuadro (7.20)

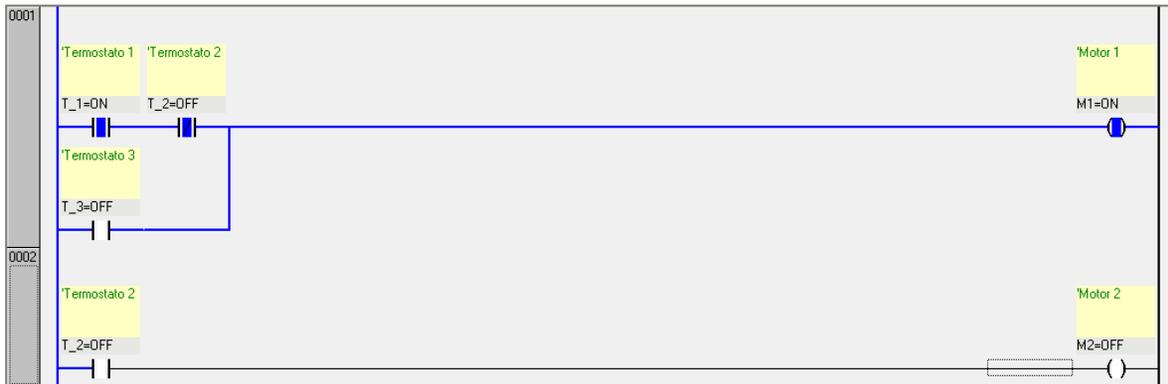


Figura 8.23 Programa en linea temperatura 1 activa



Figura 8.24 Respuesta del modulo de simulacion temperatura 1

las respuestas del programa se muestran en el modulo de simulacion mostrando una actividad en el termostato 1 y en el motor 1, Figura (8.24)

### Tabla de verdad

Cuadro 7.20 Tabla de verdad temperatura 1 activa

Termostato 1	Termostato 2	Termostato 3	Motor 1	Motor 2
1	0	0	1	0

### 7.7.5.3 Tercer caso

Para el tercer caso el termostato 1 continua activado y el termostato 2, Cuadro (7.21) incrementa incrementa su temperatura activando la señal de la temperatura 2, **T\_2(Abrir IO.1)**. Entonces el motor 1 se detiene y se activa el motor 2, **M2(O0.1)**, Figura (8.25)

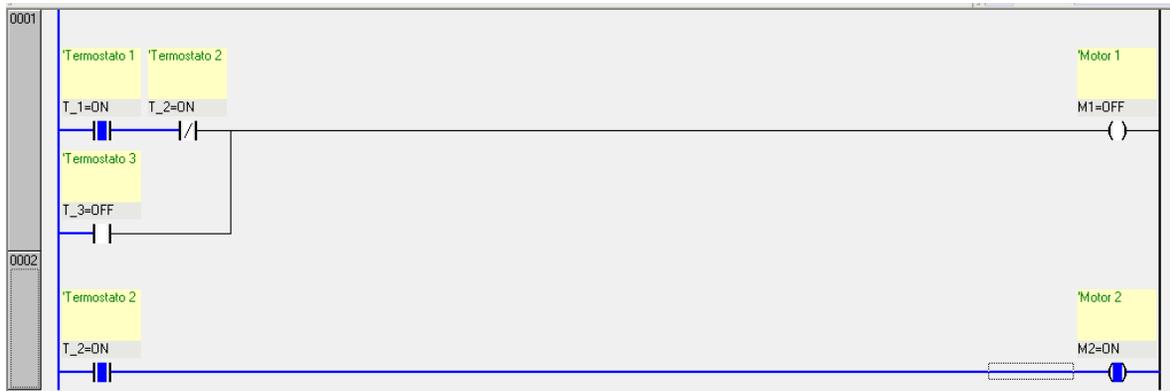


Figura 8.25 Programa en linea temperatura 1 y 2 activa



Figura 8.26 Respuesta de modulo de simulacion temperatura 1 y 2

Las respuestas obtenidas en el modulo de simulacion muestra actividad en las temperaturas 1 y 2, se apaga el motor 1 automaticamente y se enciende el motor 2, Figura (8.26).

Cuadro 7.21 Tabla de verdad temperatura 1 y 2 activas

Termostato 1	Termostato 2	Termostato 3	Motor 1	Motor 2
1	1	0	0	1

#### 7.7.5.4 Cuarto caso

Para el ultimo caso, los tres termostatos muestran una temperatura critica lo que significa que el termostato 3 **T\_3(abrir I0.2)**, activara los dos motores **M1(O0.0)** y **M2(O0.1)**, Figura (8.27)

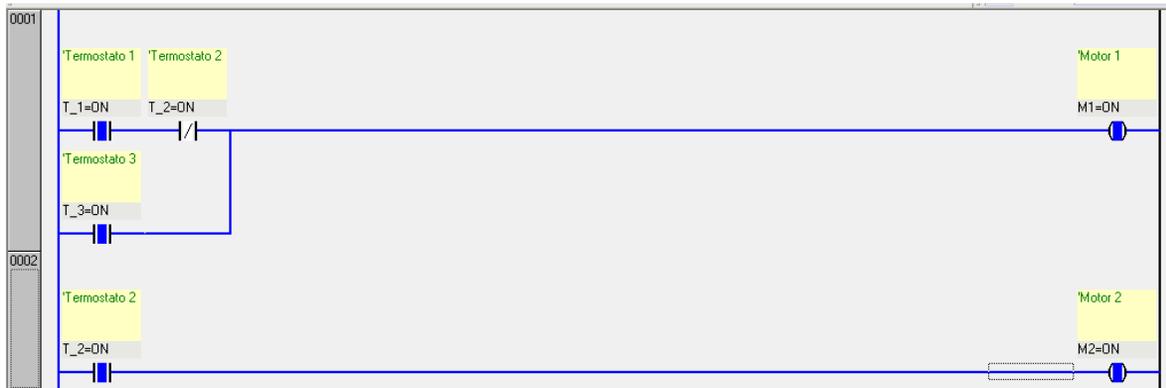


Figura 8.27 Programa en linea temperatura 1,2 y 3 activo



Figura 8.28 Respuesta de modulo de simulacion temperatura 1,2 y 3

Para este ultimo caso las respuestas mostradas en el modulo de simulacion, Figura (8.28) es la activacion de los tres termostatos indicando una temperatura critica lo que dara como respuesta la activacion de dos motores en base a la tabla

del Cuadro (7.22), los cual acondicionaran la temperatura deseada para una estabilidad.

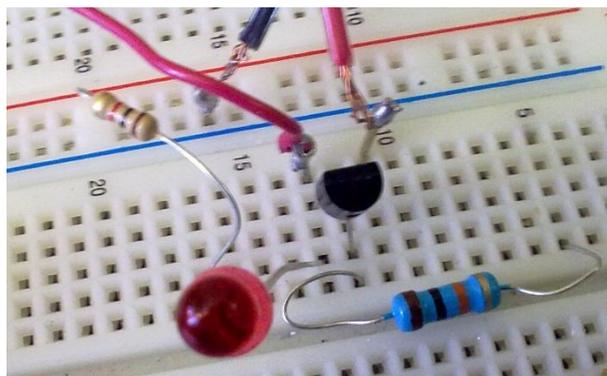
### Tabla de verdad

*Cuadro 7.22 Temperaturas 1,2 y 3 activas*

Termostato 1	Termostato 2	Termostato 3	Motor 1	Motor 2
1	1	1	1	1

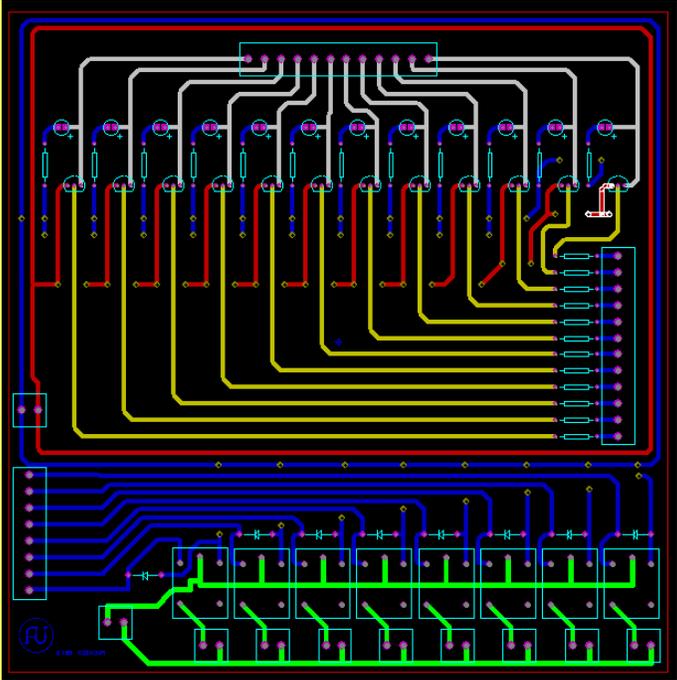
La creacion de programas en PLC,resulta practica al haber adquirido las bases y conocimientos de la compuertas logicas asi como la familiarizacion del software, se pueden realizar combinaciones de las mismas para la creacion de funciones como los ejemplos mostrados, para la aplicación y facilitacion de la operación en el area que se desea automatizar.

Al finalizar esta practica se diseño una tarjeta con relevadores y transistores entre el PLC y los tres termostatos electronicos, con la finalidad de llevar a cabo una practica en el tablero de automatizacion. El circuito construido, se diseño inicialmente en el protoboard, Figura (8.29) para verificar conexiones.

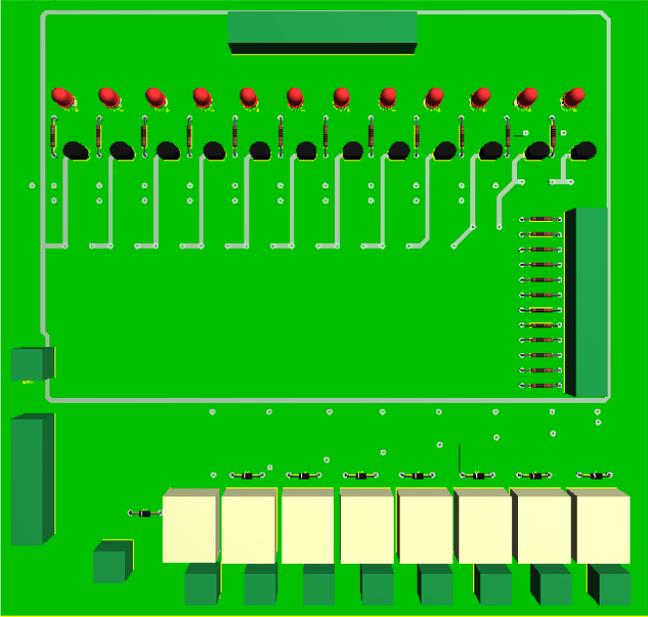


*Figura 8.29 Circuito en Protoboard*

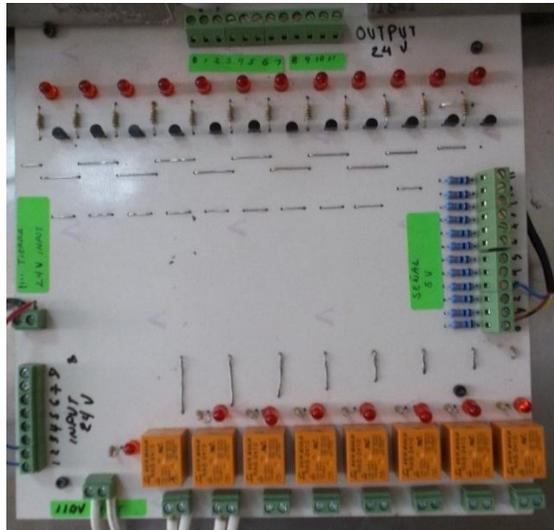
En base al circuito creado en el protoboard, se creó un circuito en ambiente de Ares 7 Profesional (Proteus), el cual cuenta con una entrada de alimentación de 24V y entradas de señal de 5V, y mediante un transistor PNP, el cual permite el accionamiento de salida de señal hacia el PLC, Figura (8.30)



(a) Circuito diseñado en Ares 7 Profesional



(b) circuito en 3D



(c) Circuito montado en placa fenólica

Figura 8.30 (a) Circuito diseñado en Ares 7 Profesional, (b) circuito en 3D, (c) circuito montado en placa fenólica.

El PLC con el programa previamente cargado activa o desactiva la salida de señal para que esta a la vez accione relevadores RAS-2410, dando paso al voltaje de 110V para el funcionamiento de los motores, Figura (8.31)

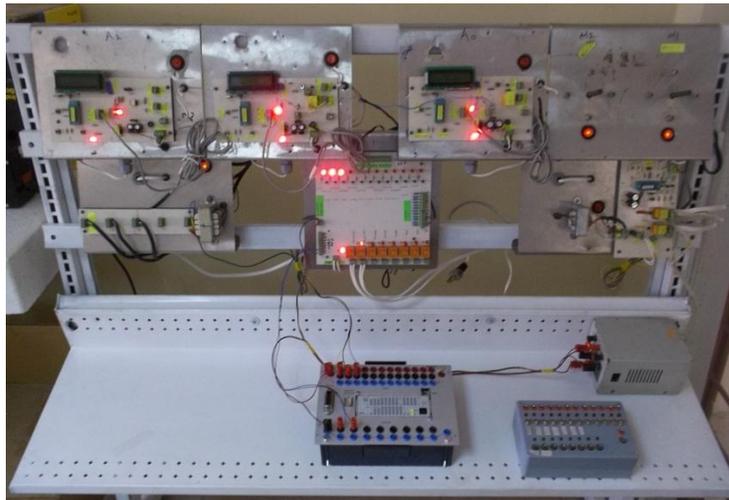


Figura 8.31 Tablero de Automatización del Departamento de Maquinaria Agrícola