

**RED DE SENSORES DE PROXIMIDAD PARA EL LABORATORIO DE
INSTRUMENTACIÓN INDUSTRIAL
ANEXOS**

NILSON ADIEL GUTIÉRREZ ASTUDILLO

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
POPAYÁN
2008**

**RED DE SENSORES DE PROXIMIDAD PARA EL LABORATORIO DE
INSTRUMENTACIÓN INDUSTRIAL
ANEXOS**



NILSON ADIEL GUTIÉRREZ ASTUDILLO

Director

I. E. Juan Fernando Flórez Marulanda

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
POPAYÁN
2008**

TABLA DE CONTENIDO

ANEXO A. CONFIGURACIÓN DE LOS PARÁMETROS DEL NODO	1
A.1. INICIALIZACIÓN DEL DISPOSITIVO.....	1
A.1.1. MODO DE INICIALIZACIÓN NORMAL	1
A.1.2 INICIALIZACIÓN STAND ALONE	2
A.2 MODOS DE INICIALIZACIÓN E IMPLEMENTACIONES POSIBLES	2
A.2.1 INICIALIZACIÓN NORMAL	2
A.2.2 INICIALIZACIÓN AUTOMÁTICA	4
A.3 CONFIGURACIÓN DE LOS PARÁMETROS DEL CONTROLADOR AB6001.....	6
A.3.1 CONFIGURACIÓN DEL HYPER TERMINAL DE WINDOWS	6
A.3.2 MENÚ PRINCIPAL EN UNA COMUNICACIÓN VÍA MIF.....	6
A.3.3 INFORMACIÓN DEL MENÚ DEL MODULO (MENÚ MODULE INFORMATION)	7
A.3.4 MENÚ PARAMETERS	7
A.3.5. MENÚ MONITOR	7
A.4. PARÁMETROS DEL CONTROLADOR.....	8
A.4.1 MODULE MODE. (#1).....	9
A.4.2 MODULE STATUS. (#2).....	9
A.4.3 MODULE. TYPE (#3)	10
A.4.4 LED STATE. (#7).....	11
A.4.5 CONFIGURATIONS BITS (#8)	11
A.4.6 SWITCH CODING (#9).....	12
A.4.7 OFFLINE ACTION CONFIG (#10).....	13
A.4.8 IDLE ACTION CONFIG (#11)	13
A.4.9 INTERRUPT CONFIG (#12)	13
A.4.10 INTERRUPT CAUSE (#13).....	15
A.4.11 SCI RATE CONFIG (#14)	16
A.4.12 SCI RATE ACTUAL (#15).....	16
A.4.13 SCI SETTINGS CONFIG (#16)	16
A.4.14 SCI SETTINGS ACTUAL (#17).....	17
A.4.15 MIF RATE CONFIG (#18)	18
A.4.16 MIF RATE ACTUAL (#19).....	18

A.4.17	MIF SETTINGS CONFIG (#20)	18
A.4.18	MIF SETTINGS ACTUAL (#21)	19
A.5	I/O PARAMETERS	20
A.5.1	FB BYTE ORDER (#40)	20
A.5.2	FB OUT CONFIG (#41)	21
A.5.3	FB OUT ACTUAL (#42)	21
A.5.4	FB IN ACTUAL (#43)	21
A.5.5	FB IN SSC OFFSET (#44)	21
A.5.6	FB IN SSC SIZE (#45)	21
A.5.7	FB IN SCI OFFSET (#46)	22
A.5.8	FB IN SCI SIZE (#47)	22
A.5.9	SSC BYTE ORDER (#50)	22
A.5.10	SSC IN CONFIG (#51)	22
A.5.11	SSC IN AUTO (#52)	22
A.5.12	SSC IN ACTUAL (#53)	22
A.5.13	SSC OUT CONFIG (#54)	23
A.5.14	SSC OUT AUTO (#55)	23
A.5.15	SSC OUT ACTUAL (#56)	23
A.5.16	SSC OUT FB OFFSET (#57)	23
A.5.17	SSC OUT FB SIZE (#58)	23
A.5.18	SSC OUT SCI OFFSET (#59)	23
A.5.19	SSC OUT SCI SIZE (#60)	23
A.5.20	SCI BYTE ORDER (#63)	23
A.5.21	SCI IN CONFIG (#64)	24
A.5.22	SCI IN ACTUAL (#65)	24
A.5.23	SCI OUT ACTUAL (#66)	24
A.5.24	SCI OUT FB OFFSET (#67)	24
A.5.25	SCI OUT FB SIZE (#68)	24
A.5.26	SCI OUT SSC OFFSET (#69)	24
A.5.27	SCI OUT SSC SIZE (#70)	25
A.6	PARÁMETROS ESPECÍFICOS DEL CONTROLADOR AB6001	25
A.6.1	FB STATUS (#100)	25
A.6.2	FB PASSWORD (#102)	26
A.6.3	FB NODE ADDRESS CONFIG (#103)	26

A.6.4.	SCC FB NODE ADDRESS (#104).....	26
A.6.5.	FB ACTUAL NODE ADDRESS (#105)	26
A.6.6.	DEVICENET BAUD RATE CONFIG (#106)	26
A.6.7	DEVICENET BAUD RATE ACTUAL (#107).....	27
A.6.8	DEVICENET VENDOR ID (#108).....	27
A.6.9	DEVICENET PRODUCT TYPE (#109).....	27
A.6.10	DEVICENET PRODUCT CODE (#110).....	27
A.6.11	DEVICENET PRODUCT NAME (#111)	27
A.6.12	CONNECTION STATUS POLLED (#113).....	28
A.6.13	CONNECTION STATUS COS (#114).....	28
A.6.14	CONNECTION STATUS BIT STROBED (#115)	28
A.6.15	CONNECTION STATUS CYCLIC (#116)	29
A.6.16	FB AUTO BAUD CFG (#117)	29
A.6.17	FB QUICK CONNECT CFG (#118)	29
A.7	SECUENCIA DE AUTO/NET DEL NODO DEVICENET	30
A.8	SECUENCIA DE AUTO TEST DEL NODO DEVICENET	30
A.8.1	INFORMACIÓN GENERAL DE AUTO TEST DEL NODO	30
A.8.2	PROCEDIMIENTO BÁSICO PARA GENERAR UN AUTO TEST	31
A.8.3	EVALUACIÓN DEL TEST DEL NODO DEVICENET.	31
	ANEXO B CONSIDERACIONES PARA REALIZAR EL CABLEADO	32
B.1	INTRODUCCIÓN	32
B.2	COMPONENTES DE LA RED DEVICENET	32
B.2.1	LÍNEA TRONCAL DEVICENET	33
B.2.2	LÍNEA DE DERIVACIÓN DEVICENET	34
B.2.3	CONECTORES DEVICENET.	35
B.2.4	TERMINACIONES DE RED DEVICENET.....	36
B.3	TAREAS EN LA EJECUCIÓN DEL CABLEADO DEVICENET	37
B3.1	INSTALACIÓN DE UN CONECTOR TIPO ABIERTO KWIKLINK A UN CABLE DE DERIVACIÓN	37
B.3.2	INSTRUCCIONES PARA LA INSTALACIÓN DEL CABLE KWIKLINK.....	39
B.3.3	PASOS PARA INSTALAR CORRECTAMENTE EL CABLE KWIKLINK EN UN CONECTOR IDC	39
B.3.4.	INSTALACIÓN DE LA TAPA TERMINAL DEL CONECTOR IDC.....	44
B.4	RECOMENDACIONES PARA CONECTAR FUENTES DE ALIMENTACIÓN A MEDIOS FÍSICOS PLANOS KWIKLINK.....	43
B.4.1.	PASOS PARA INSTALAR LA FUENTE DE ALIMENTACIÓN DEVICENET	44

B.5	MÉTODOS PARA DETERMINAR SI SE DISPONE DE LA ALIMENTACIÓN ADECUADA	44
B.5.1	MÉTODO DE BÚSQUEDA	44
B.5.1	MÉTODO DE CÁLCULO TOTAL.....	47
ANEXO C CONFIGURACIÓN DE LA RED DE BUS DE CAMPO DEVICENET		55
C.1	SOFTWARE DE CONFIGURACIÓN DE LA RED DEVICENET.....	55
C.2	INTERFACES REQUERIDAS PARA CONFIGURAR LA RED DEVICENET	55
C.3	PASOS PARA CONFIGURAR LOS DRIVER DE RSNETWORX	56
C.4	CONFIGURACIÓN DEL MODULO 1769SDN	58
C.5	DIRECCIONAMIENTO DE LOS NODOS DEVICENET	60
C.6	CONFIGURACIÓN DE DISPOSITIVOS DEVICENET	63
C.6.1	GENERAL (MENÚ)	63
C.6.2	MODULE (MENÚ)	64
C.6.3	SCANLIST (MENÚ).....	65
C.6.4	AVAILABLE DEVICES (MENÚ)	66
C.6.5	AUTOMAP (MENÚ).....	66
C.6.6	EDIT I/O PARAMETERS (MENÚ)	67
C.6.7	INPUT (MENÚ).....	67
C.6.8	OUTPUT (MENÚ)	68
C.6.9	AUTO DEVICE REPLACEMENT (ADR)	69
C.6.10	CONSIDERACIONES IMPORTANTES CON ADR	69
C.6.11	SUMMARY (MENÚ)	70
C.6.12	DESCARGAR Y GUARDAR LA CONFIGURACIÓN AL SCANNER	70
ANEXO D CONSIDERACIONES IMPORTANTES DEL NODO DEVICENET		72
D.1.	CONSIDERACIONES CON RESPECTO A LOS REGISTROS I/O DEL NODO	72
D.2.	CARACTERÍSTICAS DE LAS FAMILIAS LÓGICAS RECOMENDADAS.	74
D.3.	DISPOSITIVOS USADOS COMO ENTRADA AL NODO.....	75
D.3.1.	CIRCUITO DE ENTRADAS AL NODO DEVICENET	76
D.3.2.	SENSOR DE PROXIMIDAD USADO PARA ACOPLAR A LA TARJETA DE ENTRADAS.	79
D.3.3.	SENSOR DE PROXIMIDAD SHARP P2Y0D02YK.....	80
D.2.3.	CIRCUITO DE SALIDAS DEL NODO DEVICENET UNICAUCA	81
D.4.	CONFIGURACIÓN DEL BAUD RATE Y MAC ID DEL NODO DEVICENET	88
D.5.	CONFIGURACIÓN DEL BAUD RATE Y MAC ID DEL NODO DEVICENET	89
ANEXO E, MANUAL DE CONFIGURACIÓN LA RED DEVICENET DE LOS LABORATORIOS DEL PIAI		95
E.1.	CONFIGURACIÓN DE LA RED.....	96

E.1.1.	CONFIGURACIÓN DE DRIVERS DE COMUNICACIÓN A TRAVÉS DE RSLINX.....	96
E.2.	CONFIGURACIÓN DE LA RED DEVICENET	102
E.2.1.	DIRECCIONAMIENTO DE LOS NODOS DEVICENET	105
E.2.2.	CONFIGURACIÓN DE DISPOSITIVOS DEVICENET	107
E.2.3.	CREACIÓN DE UN PROGRAMA EN ESCALERA PARA REALIZAN CONTROL A TRAVÉS DE LA RED DEVICENET	111
	ANEXO F MANUAL DE CONFIGURACIÓN DE LA RED DH485	118
F.1.	LISTADO DE ELEMENTOS NECESARIOS.....	119
F.1.1.	CONFIGURACIÓN DE LA RED DH 485	119
F.2.	NOTAS ADICIONALES.....	128
F.2.1.	CONSIDERACIONES HARDWARE.....	129
F.2.1.	CONSIDERACIONES SOFTWARE	129
	ANEXO G ESTÁNDAR DEVICENET.....	131
G.1.	INTRODUCCIÓN	131
G.1.2.	MODELO DE OBJETOS: CLASES Y OBJETOS.....	131
G.1.3.	MODELO DE OBJETOS: PERFILES DE DISPOSITIVO.....	135
G.1.4.	MODELO DE COMUNICACIONES	139
G.1.5.	CAPA DE TRANSPORTE.	143
	ANEXO J MANUAL DE CUSUARIO DEL NODO DEVICENET UNICAUCA	159
	BIBLIOGRAFÍA	159

LISTA DE FIGURAS

	PÁG.
Fig. 1. Inicialización Normal indicadores y Switch sobre los registros cero.....	3
Fig. 2. Inicialización Normal indicadores, Switch y datos vía SCI.....	4
Fig. 3. Inicialización Automática	5
Fig. 4. Inicialización con el controlador pre configurado.....	5
Fig. 5. Disposición de los bits del parámetro (#2).....	9
Fig. 6. Distribución de bits del parámetro (#7)	11
Fig. 7. Distribución de bits del parámetro (#8)	11
Fig. 8. Distribución de bits del parámetro (#12)	14
Fig. 9. Mapa de bits del parámetro (#13)	15
Fig. 10. Mapa de bits del parámetro (#16)	17
Fig. 11. Mapa de bits del parámetro (#17)	17
Fig. 12. Mapa de bits del parámetro (#20)	18
Fig. 13. Mapa de bits del parámetro (#21)	19
Fig. 14. Organización de los bits del parámetro (#100)	26
Fig. 15. Secuencia de auto net del Nodo	30
Fig. 16. Secuencia de Auto test del controlador AB6001	31
Fig. 17. Arquitectura física de una red DeviceNet con cable plano	33
Fig. 18. Cable plano DeviceNet	33
Fig. 19. Cable de derivación 1485K.....	34
Fig. 20. Líneas de derivación para dispositivos sellados.....	34
Fig. 21. Multitomas DeviceNet	35
Fig. 22. Tipos de conectores para cable plano Kwiklink.....	35
Fig. 23. Conectores desplazables	36
Fig. 24. Tipos de terminación DeviceNet.....	36
Fig. 25. Acople de las líneas de derivación al conector abierto KwikLink	37
Fig. 26. Adecuación del cable de derivación.....	38
Fig. 27. Recubrimiento del cable de derivación	38
Fig. 28. Adecuación de los conductores del cable de derivación.	38
Fig. 29. Conectores tipo abierto	39
Fig. 30. Cable plano para la línea troncal.....	39
Fig. 31. Ubicación del conector en el cable plano DeviceNet.....	40
Fig. 32. Ajuste del conector IDC.....	40
Fig. 33. Verificación de la posición correcta del conector IDC	41
Fig. 34. Ajuste de la base del conector IDC	41
Fig. 35. Fijación de los contactos del conector en el cable DeviceNet.....	41
Fig. 36. Inserción del micromódulo al conector IDC.....	42
Fig. 37. Ubicación de la tapa terminal en el conector IDC.....	42
Fig. 38. Forma en que se debe fijar la base del IDC	43
Fig. 39. Posición de lo receptáculos guía en el conector Kwiklink	43
Fig. 40 Fuente de alimentación en un extremo del cable plano.....	45
Fig. 41. Un par de fuentes de alimentación aplicadas al sistema	45
Fig. 42. Ejemplo del método de búsqueda en un sistema con cuatro dispositivos	46
Fig. 43. Fuente de alimentación en un extremo del cable plano.....	47
Fig. 44 Una fuente de alimentación eléctrica conectada en el extremo.....	49
Fig. 45. Una fuente de alimentación conectada en el centro	50

Fig. 46. Moviendo la fuente de alimentación	52
Fig. 47. Dos fuentes de alimentación	53
Fig. 48. Dos fuentes de alimentación para duplicar la corriente del sistema.	54
Fig. 49. Ventana inicial de RSLinx	56
Fig. 50. Drivers DeviceNet	56
Fig. 51. Agregar el nuevo driver	57
Fig. 52. Configuración del modulo 1770KFD.....	57
Fig. 53. Ventana principal de RSNetworkx	58
Fig. 54. Opciones de selección de archivos del tipo de red.....	59
Fig. 55. Redes disponibles.....	59
Fig. 56. Confirmación de búsqueda de dispositivos	59
Fig. 57. Dispositivos disponibles en la red DeviceNet.....	60
Fig. 58. Comisionar dispositivos DeviceNet.....	61
Fig. 59. Selección de dispositivos DeviceNet	61
Fig. 60. Valores por defecto del scanner Devicenet	62
Fig. 61. Configuración del scanner 1769SDN.....	63
Fig. 62. Características generales del scanner en RsNetworkx	64
Fig. 63. Module configuración	64
Fig. 64. El escáner como dispositivo esclavo.....	65
Fig. 65. Dispositivos en scanlist.....	66
Fig. 66. AutoMap.....	66
Fig. 67. Datos de los dispositivos mapeados en las imágenes de entrada del controlador	67
Fig. 68 datos de los dispositivos en las imágenes de salida del controlador	68
Fig. 69.resumen del estado del sistema	70
Fig. 70. Como descargar y guardar la configuración	70
Fig. 71. Ventana que indica donde y el tipo de archivo a guardar de RSNetWorx	71
Fig. 72. Niveles de voltaje de salida del controlador versus voltaje de entrada de los registros.	73
Fig. 73. Niveles de voltaje de entrada del controlador versus voltaje de salida de los registros.	73
Fig. 74.área de entradas del nodo DeviceNet.....	76
Fig. 75. Nodo DeviceNet y tarjeta de entradas.....	76
Fig. 76. Divisor de tención formado por las resistencias de entradas (Rx) y las resistencias de la Main board (Ra)	77
Fig. 77. Circuito de entradas de señal de 12V	78
Fig. 78. Boar de entrada de sensores	78
Fig. 79. Sensor Riko DU50KP2	79
Fig. 80 opciones de conexión del sensor DU50KP2	79
Fig. 81. Diagrama de bloques del nodo, tarjetas de entradas, salidas y el sensor de proximidad Riko	80
Fig. 82. Sensor infrarrojo Sharp IR Sensor	80
Fig. 83.Aplicación típica del sensor sharp	81
Fig. 84 diagrama esquemático de la tarjeta de salidas de relevo	82
Fig. 85. Board de salidas de relevo	82
Fig. 86 Relevos usados en el circuito de salidas.	83
Fig. 87 Relevos usados en el circuito de salidas.	84
Fig. 88 Relevos usados en el circuito de salidas.	85
Fig. 89 circuito esquemático de la board de salidas con TRIAC.....	87
Fig. 90 PCB de salidas con TRIAC.....	87
Fig. 87. Mainboar del nodo DeviceNet	88
Fig. 92. Diagrama esquemático de la tarjeta principal del nodo DeviceNet Unicauca. Parte (a)	90
Fig. 93. PCB de la tarjeta principal del nodo DeviceNet Unicauca.....	92
Fig. 94 PCB de la tarjeta acondicionadora de entradas discretas	92
Fig. 95 Diagrama del circuito impreso de la tarjeta acondicionadora de salidas tipo relé.....	92
Fig. 96. Nodo DeviceNet desensamblado con las tarjetas de entradas y salidas	93
Fig. 97 nodo DeviceNet desarrollad en la Universidad del Cauca.....	94
Fig. 98 Nodo DeviceNet Unicauca.....	94

Fig. 99 arquitectura física de la red Unicauca	95
Fig. 100 ventana inicial de RSLinx	96
Fig. 101. Ventana menú comunicaciones	97
Fig. 102. Ventana de selección de drivers	97
Fig. 103 Seleccionando el driver	98
Fig. 104 driver DeviceNet	98
Fig. 105. Configuración del puerto COM	99
Fig. 106 inicializando la interfaz DeviceNet	100
Fig. 107. Nombre del driver	100
Fig. 108. Driver en modo running	101
Fig. 109. Nodos activos	101
Fig. 110. Interfaces conectadas al nodo esclavo 1734ADN	102
Fig. 111. Ventana principal de RSNetwork para DeviceNet	102
Fig. 112. Seleccione el Path	103
Fig. 113. Confirmación de búsqueda de dispositivos	103
Fig. 114. Proceso de Autobrowse	104
Fig. 115. Nodos activos en la red	104
Fig. 116. Comisionar dispositivos DeviceNet	105
Fig. 117. Selección de dispositivos DeviceNet	106
Fig. 118. Valores por defecto del scanner Devicenet	107
Fig. 119. Pestaña General de la ventana de propiedades del scanner	108
Fig. 120. Pestaña Module	108
Fig. 121. Pestaña Scanlist	109
Fig. 122. Advertencia ADN	109
Fig. 123. Configuración de parámetros I/O del ADN	110
Fig. 124. Posición de I/O del ADN mapeadas en el scanner	111
Fig. 125. Selección del procesador MicroLogix 1500	112
Fig. 126. Configuración de entradas y salidas dispuestas en el PLC	113
Fig. 127. Selección del Path para detectar los módulos de salidas y entradas en el PLC	113
Fig. 128. Ventana de selección del dispositivo de comunicaciones	114
Fig. 129. Módulos conectados en las ranuras del PLC	114
Fig. 130. Configuración general del modulo scanner (Word I/O)	115
Fig. 131. Imágenes de entrada y salida mapeadas en el controlador	116
Fig. 132. Inicializando el Scanner	116
Fig. 133. Ladder Logic	117
Fig. 134. Arquitectura física de la red DH485	118
Fig. 135. Opciones de conexión punto a punto entre los PLCs y el PC. a) Empleando un AIC b) Conexión directa c) Conexión mediante PIC 1747	120
Fig. 136. Ventana principal de RSLinx	120
Fig. 137. Drivers disponibles	121
Fig. 138. Seleccionando el driver	121
Fig. 139. Configurando el dispositivo RS232	122
Fig. 140. Driver configurado correctamente	122
Fig. 141. Driver configurado correctamente	123
Fig. 142. Ventana principal de RSLogix 500	123
Fig. 143. Asignando un Link ID	124
Fig. 144. Asignando un Link ID	125
Fig. 145. Configurando devics	125
Fig. 146. Seleccionando Devices	126
Fig. 147. Nombre del driver	126
Fig. 148. Configuración de dispositivos	127
Fig. 149. Dispositivos en línea	127
Fig. 150. Dispositivos en línea	128
Fig. 151. Modelo de comunicaciones en un dispositivo DeviceNet	143

LISTA DE TABLAS

	PÁG.
Tabla 1. Estado de los parámetros (Switch sobre los registros cero)	3
Tabla 2. Estado de los parámetros (Switch y datos vía SCI)	4
Tabla 3. Parámetros de la Inicialización Automática	5
Tabla 4. Valor de los parámetros (controlador pre configurado)	5
Tabla 5. Parámetros generales del controlador AB6001	8
Tabla 6. Parámetros I/O	20
Tabla 7. Parámetros específicos del controlador AB6001	35
Tabla 8. Características de los conductores para cable plano	32
Tabla 9. Cables compatibles con conectores KwikLink	36
Tabla 10. Versiones de software requerido para la configuración de la red DeviceNet	55
Tabla 11. Niveles de voltaje E/S de los registros.	75
Tabla 12. Corrientes de E/S	75
Tabla 13. Descripción de pines del sensor sharp	81
Tabla 14. Características del Optotriac.	83
Tabla 15. Características del TRIAC.	85
Tabla 16. Switch de baud rate del nodo DeviceNet Unicauca	88
Tabla 17. Switch de MAC ID del nodo DeviceNet Unicauca	89
Tabla 18. Indicadores de estado del nodo DeviceNet (señalización)	89
Tabla 19. Mapeo de nodos en la memoria del scanner	111
Tabla 20. Elementos hardware de la red	119
Tabla 21. Elementos Software	119
Tabla 22. Service Codes de los DeviceNet Common Services	133
Tabla 23. Perfiles estándar de dispositivo definidos por DeviceNet	136
Tabla 24. Rangos de numeración de perfiles de dispositivo	137
Tabla 25. Ejemplo de assembly de datos de I/O	138
Tabla 26. Mapeo entre assembly de datos de I/O y objeto	138
Tabla 27. Grupos de mensajes en DeviceNet	141
Tabla 28. Utilización de valores de arbitración en DeviceNet	142
Tabla 29. Request Message Body Format en Open Connection Request	150
Tabla 30. Posibles códigos de error en mensajes Error Response de DeviceNet	153
Tabla 31. Reglas de codificación de Compact Encoding	156

PREFACIO

El documento presentado a continuación, contiene varios apartados que lejos de ser una producción autónoma e inédita, es una recopilación de datos Sobre DeviceNet, nodos esclavos DeviceNet. Etc. Datos difundidos en Internet y en los Datasheet de los diferentes dispositivos usados (AnybusIC, registros, Sensores Etc.). Se exponen conceptos de diversos autores, algunos sin mayores modificaciones. Sin embargo se han adicionado algunos Conceptos y sugerencias como resultado de conocimientos adquiridos en el desarrollo de este trabajo de grado. El anexo digital E es una copia del estándar DeviceNet entregado por la ODVA con fines educativos y con el compromiso de confidencialidad es por eso que esta copia se entrega como anexo con el compromiso de confidencialidad y uso únicamente con fines académicos.

Los Autores.

ANEXO A. CONFIGURACIÓN DE LOS PARÁMETROS DEL NODO

El nodo desarrollado en este proyecto de grado se configura vía interfaz MIF usando el puerto serial de un Computador convencional, hay dos opciones de configuración una es usando el Hyper Terminal de Windows y la otra es usando software desarrollado por el fabricante del integrado ABICDEV. Por facilidad y disposición se opta por utilizar el método de configuración vía RS232 a través del Hyper Terminal de Windows, para configurar el nodo DeviceNet implementado en este proyecto.

A.1. INICIALIZACIÓN DEL DISPOSITIVO

La comunicación con dispositivos externos está fuertemente ligado a la forma de inicialización del modulo. El AB6001 tiene dos modos de inicializar que son: inicialización normal e inicialización automática.

A.1.1. Modo de Inicialización normal

Este modo de inicialización es usado con aplicaciones inteligentes donde el AB6001 será controlado por un micro Controlador o UART (*Universal Asynchronous Receiver-Transmitter*) a través del canal SCI.

Pasos para la inicialización normal.

1. si es necesario ejecute el auto Baud Rate del canal SCI en el Hyper Terminal
2. Configure el tamaño de los datos de I/O usando los parámetros I/O.
3. Configure el parámetro #1 ("*Module Mode*") al valor 0x0001 que corresponde a Modo normal
4. Revise si la inicialización fue satisfactoria y finalícela.

A.1.2 inicialización *Stand Alone*

La inicialización *Stand Alone* (automática) es el método mas apropiado y simple para aplicaciones no inteligentes que utilicen el canal SSC, hay que tener en cuenta que en el modo *stand alone* es deshabilitado automáticamente el canal SCI, pero la configuración de parámetros se puede realizar a través de la interfaz MIF.

Al conectar el pin 29 del integrado AB6001 a tierra (GND), el dispositivo inicializa automáticamente. Una de las ventajas de este modo de inicialización es que el controlador detecta automáticamente el número de registros y mapea los datos entre el canal SSC y el bus de campo. El parámetro #8 debe estar configurado a cero (0000h), para que el inicio automático opere correctamente. Para evitar inconvenientes se puede reinicializar el modulo a la configuración por defecto de fabrica configurando el parámetro #1("Module Mode") a 0004h.

A.2 MODOS DE INICIALIZACIÓN E IMPLEMENTACIONES POSIBLES

A.2.1 Inicialización normal

La iniciación normal tiene dos opciones de inicialización que son:

Opción uno. Switch y LEDs sobre Registros, datos vía SCI. Las características de este modo de inicialización son:

- El canal SCI está interfazado a un micro Controlador.
- El SCI usa la detección automática de Baud Rate.
- Las I/O son mapeado entre el SCI y el canal del bus de campo (FB).(8 bytes en cada dirección)
- LEDs en el Registro (0).
- switch BCD sobre el Registro (0).

La figura 1 representa la configuración del modo de inicialización normal descrito en la sección A.2.1, mientras que la tabla 1 lista el estado de los parámetros inherentes a la inicialización normal.

Fig. 1. Inicialización Normal indicadores y Switch sobre los registros cero

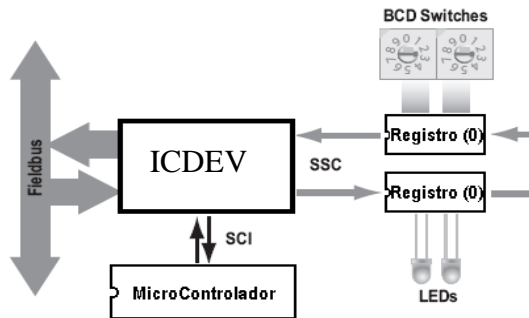


Tabla 1. Estado de los parámetros (Switch sobre los registros cero)

Parámetro	Configuración
Configuración Bits (#8)	SSCI=1, SSC0=1
Switch Coding (#9)	0x00
SCI Rate Config (#14)	0x00 (default)
FB Out Config (#41)	0x0008
FB In SSC Size (#45)	0x0000 (default)
FB In SCI Offset (#46)	0x0000 (default)
FB In SCI Size (#47)	0x0008
SSC In Config (#51)	0x0000 (default)
SSC Out Config (#54)	0x0000 (default)
SCI In Config (#64)	0x0008
SCI Out FB Offset (#67)	0x0000 (default)
SCI Out FB Size (#68)	0x0008
SCI Out SSC Size (#70)	0x0000 (default)

Opción dos. Switch, LEDs e intercambio de Datos vía SCI. Las características de de este modo de inicialización son:

- El canal SCI está interfazado a un micro Controlador.
- Las I/O son mapeado entre el SCI y el canal del bus de campo (FB).
- LEDs y salidas directamente del micro Controlador.
- Switch y entradas directamente al micro Controlador.

En la figura 2 se puede observar la configuración del ABICDEV en una segunda opción de de inicialización, la tabla 3 describe el estado de los parámetros en este modo de inicialización.

Fig. 2. Inicialización Normal indicadores, Switch y datos vía SCI

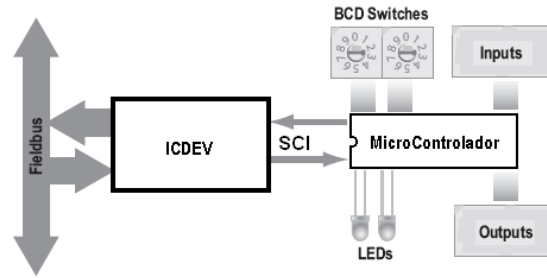


Tabla 2. Estado de los parámetros (Switch y datos vía SCI)

Parámetro	Configuración
Configuration Bits (#8)	FBNP=1, FBLP =1, SSCI=1, SSCO=1
Switch Coding (#9)	0x00
SCI Rate Config (#14)	0x00 (default)
FB Out Config (#41)	0x0008
FB In SSC Size (#45)	0x0000 (default)
FB In SCI Offset (#46)	0x0000 (default)
FB In SCI Size (#47)	0x0008
SSC In Config (#51)	0x0000 (default)
SSC Out Config (#54)	0x0000 (default)
SCI In Config (#64)	0x0008
SCI Out FB Offset (#67)	0x0000 (default)
SCI Out FB Size (#68)	0x0008
SCI Out SSC Size (#70)	0x0000 (default)

A.2.2 Inicialización automática

Switch y LEDs sobre el canal SSC. Las características de este modo de inicialización son:

- Operación *Stand Alone* (inicialización Automática)
- Todos los datos son mapeados entre el SSC y el FB
- LEDs sobre el registro (0), datos sobre los registros
- switch BCD y salidas sobre los registros

La figura 3 indica la configuración del controlador en modo de inicialización automática, este modo de inicialización se adopto en la implementación del nodo DeviceNet de este proyecto de grado, al igual que en las secciones anteriores se usa una tabla para indicar el estado de los parámetros.

Fig. 3. Inicialización Automática

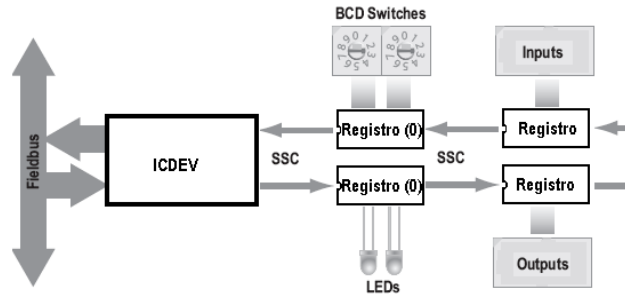


Tabla 3. Parámetros de la Inicialización Automática

Parámetro	Configuración
Configuración Bits (#8)	00h (default)
Switch Coding (#9)	00h

Inicialización con el controlador Pre configurado. Otra forma de inicialización del controlador es haciendo una configuración previa en un dispositivo funcional habilitado con una interfaz MIF, se usa en el caso de que se diseñe un dispositivo sin esta interfaz de comunicación. Las Características de este modo son:

- operación *Stand Alone* (inicialización Automática).
- Todos los datos son mapeados entre el SSC y el FB.
- No LEDs indicadores, no switch de configuración.

La figura 4 y la tabla 4 describen las características de este modo de inicialización.

Fig. 4. Inicialización con el controlador pre configurado

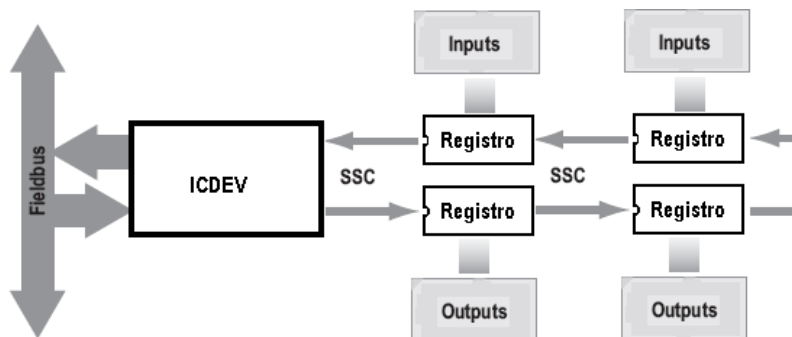


Tabla 4. Valor de los parámetros (controlador pre configurado)

Parámetro	Configuración
Configuración Bits (#8)	00h (default)
Switch Coding (#9)	00h

A.3 CONFIGURACIÓN DE LOS PARÁMETROS DEL CONTROLADOR AB6001

Interfaz MIF es una vía o canal muy importante para monitorear datos y acceder a los parámetros del controlador a través del Hyper Terminal de Windows, generalmente se usa solo en las tareas iniciales de desarrollo del nodo DeviceNet. El nodo no necesita la comunicación con esta interfaz para operar como nodo esclavo DeviceNet.

A.3.1 Configuración del Hyper Terminal de Windows

Procedimiento de configuración de los parámetros del controlador usando el Hyper Terminal de Windows.

- 1 inicie el Hyper Terminal de Windows.
- 2 seleccione nueva conexión, se debe asignar un nombre a la conexión.
- 3 especifique el puerto COM a usar.
- 4 especifique el baudrate a 38400Kbps.

Al conectar el nodo al puerto COM (serial) del computador e iniciar la comunicación, posterior a la configuración del puerto COM, aparecerá en pantalla un sub menú de los diferentes parámetros del nodo. Para acceder a un sub menú o parámetro se debe seleccionar la opción deseada digitando el número que antecede la opción y presionar <Enter>, Para entrar el valor de un parámetro introducir el nuevo valor y presionar <Enter> y Para retornar a un menú determinado o cancelar la entrada de un parámetro presionar <ESC>.

A.3.2 Menú Principal en una comunicación vía MIF

Menú que aparece en pantalla cuando se ha iniciado una comunicación a través del canal MIF del Nodo, es un menú general que provee acceso a varios submenús de configuración y monitoreo.

A continuación se presenta un esquema del menú que aparece en pantalla en una comunicación activa entre el computador y el nodo DeviceNet.

Anybus-IC - Main Menu
DeviceNet

- 1 - Module Information
 - 2 - Parameters
 - 3 - Monitor
 - 4 - Firmware Upgrade
-

A.3.3 Información del Menú del Modulo (*Menú Module Information*)

Este menú está dividido en dos submenús; '*Software Versions*', que suministra información del firmware, y '*Product Information*', encargado de mostrar información del producto como el número de serie y la fecha de producción. En las siguientes líneas se muestra el menú en pantalla de de este sub menú.

Anybus-IC - Information

- 1 - Software Versions
 - 2 - Product Information
-

A.3.4 Menú Parameters

Este menú permite el acceso a todos los parámetros del controlador. A través del Hyper Terminal de Windows se puede observar la siguiente ventana.

Anybus-IC - Parameters

- 1 - Anybus-IC
 - 2 - FB I/O Settings
 - 3 - SSC I/O Settings
 - 4 - SCI I/O Settings
 - 5 - Fieldbus Specific
-

A.3.5. Menú Monitor

Todos los buffers de I/O pueden ser monitoreados a través de este menú. Los buffer susceptibles de monitoreo se describen en la siguiente línea tal cual como se ven a través del Hyper Terminal de Windows.

Anybus-IC - I/O Areas

-
- 1 - Fieldbus Out
 - 2 - SSC In
 - 3 - SCI In
 - 4 - Fieldbus In
 - 5 - SSC Out
 - 6 - SCI Out
-

A.4. PARÁMETROS DEL CONTROLADOR

Este controlador se puede configurar a través de ciertos parámetros accesibles usando el canal de comunicaciones MIF. Estos parámetros se dividen en: Parámetros Generales, parámetros I/O y parámetros específicos. La tabla 5 lista los parámetros generales del controlador AB6001 de HMS en la tabla se puede observar los valores por defecto de los parámetros el tamaño, el tipo de si es un parámetro de lectura, escritura, lectura escritura, etc.

Tabla 5. Parámetros generales del controlador AB6001

#	Address	Name	Size	Default	Access
1	0x500	Module Mode	2 bytes		R/W
2	0x5002	Module Status	2 bytes		R
3	0x5003 -	Module Type	2 bytes		R
4	0x5004	Fieldbus Type	2 bytes		R
7	0x5007	LED State	2 bytes		R
8	0x5008	Config Bits	2 bytes	0x0000	R/W
9	0x5009	Switch Coding	1 byte		R/W
10	0x500A	Offline Action	1 byte	0x00	R/W
11	0x500B	Idle Action	1 byte	0x00	R/W
12	0x500C	Interrupt Config	2 bytes	0x0001	R/W
13	0x500D	Interrupt Cause	2 bytes		R
14	0x500E	SCI Rate Config	1 byte	0x00	R/W
15	0x500F	SCI Rate Actual Config	1 byte		R
16	0x5010	SCI Settings	1 byte	0x00	R/W
17	0x5011	SCI Settings Actual -	1 byte		R
18	0x5012	MIF Rate Config	1 byte	0x04	R/W
19	0x5013	MIF Rate Actual	1 byte		R
20	0x5014	MIF Settings Config	1 byte	0x00	R/W
21	0x5015	MIF Settings Actua	1 byte		R
22	0x5016	Modbus RTU Address	1 byte	0x01	R/W
23	0x5017	Modbus CRC Disable	1 byte	0x00	R/W
27	0x501B...0x5032 FB	Fault Values	48 bytes	0x00	R/W

A continuación se hace una descripción de cada parámetro de la tabla 5.

A.4.1 Module Mode. (#1)

Es un parámetro R/W (*read/write*) de lectura configurable usado para determinar la operación del Nodo, más específicamente el modo de operación del controlador AB6001. Este parámetro tiene 5 opciones de configuración que son:

- **0x0000 Start-up Mode.** Este es el valor inicial del parámetro, después de aplicar alimentación del circuito, excepto cuando el modulo está en modo automático.
- **0x0001 Normal Operation Mode.** Indica el modo normal de operación.
- **0x0003 Reset Module.** Indica un estado de reinicio del controlador AB6001.
- **0x0004 Set Default.** Todos los parámetros son cambiados al valor por defecto de fábrica. (rest).
- **0x0005 Self Test.** Este parámetro habilita la función de un auto test del integrado, y al función principal es hacer un auto test (auto examen) al controlador.

A.4.2 Module Status. (#2)

Este parámetro permite mantener información del estado del modulo, se usa también para mostrar el estado del modulo después de un test. La figura 5 muestra el estado de los bits de este parámetro

Fig. 5. Disposición de los bits del parámetro (#2)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RCF	FCF	ECF	-	-	SCIC	SSCC	FBC	-	-	-	-	-	-	SCI	SSC

- **SSC. (estado de los bits).** Esta sección describe el estado de los bits del canal SSC.

1: SSC. Canal en funcionamiento, los *Shift registers* (registros) son detectados en el canal SSC y el inicio automático es correcto.
0: SSC. El canal está parado, fuera de uso, o los *shift register* no han sido detectados en el canal SSC.

- **SCI.** Esta sección describe el estado de los bits del canal SCI.
 - 1: SCI canal funcionando correctamente.
 - 0: SCI el canal no está en uso o no funciona correctamente, error de configuración en el Baud Rate o Baud Rate no detectado.

- **FBC.**
 - 1: FB I/O. Falla en la configuración, error en la configuración de mapeo de datos I/O.
 - 0: FB I/O. Configuración OK.

- **SSC.**
 - 1: SSC I/O. Falla de configuración, error de configuración de mapeo de datos.
 - 0: SSC I/O. Configuración OK.

- **SCI.**
 - 1: SCI I/O. Falla de configuración, error de configuración de mapeo de datos.
 - 0: SCI I/O. Configuración OK.

- **ECF.**
 - 1: EEPROM. Chequeo de memoria fallido.
 - 0: EEPROM. Chequeo de memoria OK.

- **FCF.**
 - 1: FLASH. Chequeo de memoria fallido.
 - 0: FLASH. Chequeo de memoria OK.

- **RCF.**
 - 1: RAM. Chequeo fallido.
 - 0: RAM. Chequeo OK.

A.4.3 Module. Type (#3)

Este parámetro describe el tipo de integrado que se está usando.

A.4.4 LED State. (#7)

El estado de los LEDs ubicados en el canal SSC del registro 0 puede ser leído usando este parámetro. Este parámetro indica el estado de los LED, aun que el canal SSC no se use.

La figura 6 indica el estado de los LEDs del registro cero (0) de salidas.

Fig. 6. Distribución de bits del parámetro (#7)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LED 8		LED 7		LED 6		LED 5		LED 4		LED 3		LED 2		LED 1	

A.4.5 Configurations Bits (#8)

Indica el valor para los diferentes parámetros de inicialización. La figura 7 indica la distribución de bits de este parámetro.

Fig. 7. Distribución de bits del parámetro (#8)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	FBNA	BR	NA	FBLP	FBNP	SSCO	SSCI

- **SSCI.**
 - 1: configura el tamaño de entrada SSC usada.
 - 0: inicialización Automática del tamaño de SSC de entrada.
- **SSCO.**
 - 1: configura el tamaño de salida SSC usada.
 - 0: inicializa automáticamente el tamaño de salida SSC.
- **FBNP.** Si no hay registros de entrada conectados este bit es automáticamente configurado a 1.
 - 1: no hay registros de entrada conectados al canal SSC
 - 0: registros presentes en el canal SSC.

- **FBLP.** Si no hay registros de salida conectados este bit es automáticamente configurado a 1.
1: omitido el uso de registros de salida en el canal SSC.
0: registros de salida presentes en el canal SSC.
- **NA1.** Si no se usan registros en el canal SSC este bit es automáticamente configurado a 1.
1: el direccionamiento del nodo es determinado por los parámetros del *specific fieldbus*.
0: el direccionamiento del nodo es determinado por los registros del *specific fieldbus*.
- **BR.** Si no hay registros de entrada conectados este bit es automáticamente configurado a 1.
1: si el baud rate se configura a través del bus (*specific fieldbus*).
0: si el baud rate se configura vía fieldbus o vía switch sobre el registro 0 de entradas.
- **FBNA.** Este bit determina el comportamiento sobre sistemas donde la dirección de los nodos es recibida desde el fieldbus. En sistemas que no tienen esta funcionalidad este bit no se tiene en cuenta.
1: el direccionamiento del nodo es recibido desde el fieldbus. El valor del bit NA será ignorado.
0: el direccionamiento del nodo es determinado por el bit.

A.4.6 Switch Coding (#9)

Si los registros de entrada están habilitados, este parámetro determina como, el valor de los switch son interpretados por el nodo. Los valores que puede tomar este parámetro son:

- **0x00 BCD-coded switches.** Uso de switch en código BCD para configuración de la dirección dl nodo y el baudrate. Ejemplo: la posición 63 (decimal) del nodo equivale a 0110 0011 en código BCD.

- **0x01 - Binary switches.** Switches en código binario. Ejemplo: la posición 63 del nodo equivale a 0011 1111 en código binario.

A.4.7 Offline Action Config (#10)

Cuando el nodo pasa de *on line* (en línea) a *off line* (fuera de línea) o de modo inactivo a *off line* las salidas se pueden configurar de diferentes formas según los valores de los parámetros:

- **0x00 Clear.** Las salidas son clareadas cuando el bus pasa a modo *off line*.
- **0x01 Freeze.** Las salidas son congeladas cuando el nodo pasa a *off line*.
- **0x02 Fault values.** Usando el parámetro #27 se pueden configurar valores por defecto, cuando el modulo cambia de modo *on line* a *off line* los valores configurados en el parámetro 27 son copiados a las salidas del nodo.

A.4.8 Idle Action Config (#11)

Cuando el fieldbus va de modo *on line* a modo *idle* (inactivo) o de *offline* a *idle* las salidas del dispositivo pueden ser configuradas de diversas formas según el valor de este parámetro, las posibles opciones son:

- **0x00 Clear.** Las salidas del nodo son clareadas cuando pasa a modo *idle*.
- **0x01 Freeze.** El estado de las salidas son congeladas cuando el nodo pasa a modo *idle*.
- **0x02 Fault values.** Los valores configurados en el parámetro-# 27 son copiados a las salidas del nodo cuando pasa a modo *idle*.

A.4.9 Interrupt Config (#12)

Este parámetro define que eventos son permitidos para iniciar una interrupción. La figura 8 muestra la distribución de los bits configurables en este parámetro.

Fig. 8. Distribución de bits del parámetro (#12)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	IDLE	RES	DEF	ACYC	FBOFF	FBON	START

- **START1.**
 1: genera una interrupción cuando el nodo ha iniciado y está listo para comunicarse.
 0: este evento no causa ninguna interrupción.
- **FBON.**
 1: se genera una interrupción cuando el modulo pasa de *off line* a *on line*.
 0: no se produce ninguna interrupción.
- **FBOFF.**
 1: origina una interrupción cuando el nodo pasa de *on line* a *off line*.
 0: no se causa interrupción.
- **ACYC2.**
 1: interrupción cuando el nodo recibe un nuevo dato a cíclico del *fiel bus*.
 0: no se produce interrupción.
- **DEF2.**
 1: genera una interrupción, si se recibe una configuración por defecto desde el *fieldbus*.
 0: no se produce interrupción.
- **RES2.**
 1: producirá una interrupción cuando se recibe un Reset desde el FB
 0: no se produce interrupción.
- **IDLE2.**
 1: origina una interrupción cuando el nodo pasa de *online* a *idle* o de *off line* a *idle*
 0: no se produce interrupción.

A.4.10 Interrupt Cause (#13)

Este parámetro indica el evento que ha causado una interrupción configurada en el parámetro #12. La figura 9 indica la distribución de bits de este parámetro.

Fig. 9. Mapa de bits del parámetro (#13)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	IDLE	RES	DEF	ACYC	FBOFF	FBON	START

- **START.**
 - 1: el modulo inició y está listo para comunicarse.
 - 0: no se ha causado ninguna interrupción.
- **FBON.**
 - 1: ha ocurrido una transición de *off line* a *on line*.
 - 0: este evento no ha causado interrupción.
- **FBOFF.**
 - 1: ha ocurrido una transición de *on line* a *off line*.
 - 0: el evento no origino interrupción.
- **ACYC.**
 - 1: nuevo dato a cíclico recibido desde el FB.
 - 0: el evento no genero ninguna interrupción.
- **DEF (Fieldbus Dependant).**
 - 1: se ha recibido una configuración por defecto desde el FB.
 - 0: no se produjo interrupción.
- **RES (Fieldbus Dependant).**
 - 1: un *Reset* ha sido recibido desde el FB.
 - 0: no se genero interrupción.

- **IDLE (Fieldbus Dependant).**

1: el nodo ha pasado de modo *on line* a *idle* o de *off line* a *idle*.

0: no se origino interrupción.

A.4.11 SCI Rate Config (#14)

Este parámetro permite configurar Baud Rate del canal SCI es necesario reinicializar el modulo para que los cambios tengan efecto. los valores de los parámetros son:

- 0x00 – detección automática de Baud Rate (configuración por defecto).
- 0x01 - 4,8 kbit/s
- 0x02 - 9,6 kbit/s
- 0x03 - 19,2 kbit/s
- 0x04 - 38,4 kbit/s
- 0x05 - 57,6 kbit/s

A.4.12 SCI Rate Actual (#15)

Este parámetro retorna el valor actual del Baud Rate del canal SCI. Los posibles valores son:

- 0x00 – Baud Rate no configurado.
- 0x01 - 4,8 kbit/s
- 0x02 - 9,6 kbit/s
- 0x03 - 19,2 kbit/s
- 0x04 - 38,4 kbit/s
- 0x05 - 57,6 kbit/s

A.4.13 SCI Settings Config (#16)

Este parámetro permite cambiar la configuración del canal SCI permite habilitar o deshabilitar la paridad, se debe reiniciar el nodo para que los cambios tengan efecto. No

aplica cuando está configurado en detección automática de Baud Rate. La figura 10 indica la distribución de bits de este parámetro.

Fig. 10. Mapa de bits del parámetro (#16)

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	PAR1	PAR2

- **PAR2.**
1: paridad habilitada.
0: paridad deshabilitada.
- **PAR1.** (Este parámetro no tiene efecto si está deshabilitada la paridad)
1: paridad impar.
0: paridad par.

A.4.14 SCI Settings Actual (#17)

Este parámetro retorna la configuración actual del canal SCI si la detección automática de Baud Rate está habilitado, la configuración detectada por el puerto es mostrada en este parámetro.

Fig. 11. Mapa de bits del parámetro (#17)

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	PAR1	PAR2

- **PAR2.**
1: paridad habilitada.
0: paridad deshabilitada.
- **PAR1.** (Este parámetro no tiene efecto si está deshabilitada la paridad).
1: paridad impar.
0: paridad par.

A.4.15 MIF Rate Config (#18)

Este parámetro es usado para configurar el Baud Rate de la interfaz MIF. Esta interfaz no tiene detección automática de Baud Rate. Los valores disponibles son:

- 0x01 - 4,8 kbit/s.
- 0x02 - 9,6 kbit/s.
- 0x03 - 19,2 kbit/s.
- 0x04 - 38,4 kbit/s (por defecto).
- 0x05 - 57,6 kbit/s.

A.4.16 MIF Rate Actual (#19)

Este parámetro retorna el valor de Baud Rate configurado en la interfaz MIF. Las opciones disponibles son:

- 0x01 - 4,8 kbit/s.
- 0x02 - 9,6 kbit/s.
- 0x03 - 19,2 kbit/s.
- 0x04 - 38,4 kbit/s.
- 0x05 - 57,6 kbit/s.

A.4.17 MIF Settings Config (#20)

El siguiente parámetro se usa para cambiar la configuración del puerto de la interfaz MIF. Se puede habilitar y deshabilitar la paridad des de este parámetro. La figura 12 muestra la distribución de bits de este parámetro.

Fig. 12. Mapa de bits del parámetro (#20)

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	STOP	PAR1	PAR2

- **PAR2.**
1: paridad.
0: sin paridad.
- **PAR1.** (Si la paridad de este parámetro está deshabilitada (PAR2=0) este bits no tendrá efectos).
1: paridad impar.
0: paridad par.
- **STOP.**
1: dos bits de stop son usados.
0: se usa un solo bit de stop.

A.4.18 MIF Settings Actual (#21)

Este parámetro notifica la configuración actual de la interfaz, muestra opciones relacionadas con al paridad. La figura 13 describe la distribución de bits de este parámetro.

Fig. 13. Mapa de bits del parámetro (#21)

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	STOP	PAR1	PAR2

- **PAR2.**
1: paridad habilitada.
0: paridad deshabilitada.
- **PAR1.** (Si la paridad está deshabilitada (PAR2=0) este bits no afectara).
1: paridad impar.
0: paridad par.
- **STOP.**
1: se usan dos bits.
0: solo se usa un bit.

A.5 I/O PARAMETERS

Cada canal de comunicación tiene sus propias características de configuración de los parámetros, el cual determinará como los datos producidos serán mapeados a otros canales. La tabla 6 muestra los diferentes parámetros de entrada/salida del controlador ABICDEV.

Tabla 6. Parámetros I/O

#	Address	Name	Size	Default	Access
40	0x6000	FB Byte Order	1 byte	0x00	R/W
41	0x6001	FB Out Config	2 bytes	0x0000	R/W
42	0x6002	FB Out Actual	2 bytes		R
43	0x6003	FB In Actual	2 bytes		R
44	0x6004	FB In SSC Offset	2 bytes	0x0000	R/W
45	0x6005	FB In SSC Size	2 bytes	0x0000	R/W
46	0x6006	FB In SCI Offset	2 bytes	0x0000	R/W
47	0x6007	FB In SCI Size	2 bytes	0x0000	R/W
50	0x600A	SSC Byte Order	1 byte	0x00	R/W
51	0x600B	SSC In Config	2 bytes	0x0000	R/W
52	0x600C	SSC In Auto	2 bytes		R
53	0x600D	SSC In Actual	2 bytes		R
54	0x600E	SSC Out Config	2 bytes	0x0000	R/W
55	0x600F	SSC Out Auto	2 bytes		R
56	0x6010	SSC Out Actual	2 bytes		R
57	0x6011	SSC Out FB Offset	2 bytes	0x0000	R/W
58	0x6012	SSC Out FB Size	2 bytes	0x0000	R/W
59	0x6013	SSC Out SCI Offset	2 bytes	0x0000	R/W
60	0x6014	SSC Out SCI Size	2 bytes	0x0000	R/W
63	0x6017	SCI Byte Order	1 byte	0x00	R/W
64	0x6018	SCI In Config	2 bytes	0x0000	R/W
65	0x6019	SCI In Actual	2 bytes		R
66	0x601A	SCI Out Actual	2 bytes		R
67	0x601B	SCI Out FB Offset	2 bytes	0x0000	R/W
68	0x601C	SCI Out FB Size	2 bytes	0x0000	R/W
69	0x601D	SCI Out SSC Offset	2 bytes	0x0000	R/W
70	0x601E	SCI Out SSC Size	2 bytes	0x0000	R/W

A.5.1 FB Byte Order (#40)

Este parámetro determina si los bytes en las I/O del FB (*fieldbus*) deben ser re direccionados a otras áreas de I/O del nodo. Para que esta función trabaje adecuadamente la longitud de I/O, debe ser múltiplo de 16 bits.

Ejemplo: Si el dato es mapeado de “FB Output” a “SSC Output” el parámetro #40 “FB Byte Order” es 0x00 y el parámetro #50 “SSC Byte Order” es 0x01 los datos escritos al FB Output” serán intercambiados cuando estos alcancen el “SSC Output” si el parámetro byte order tiene el mismo valor de ambas áreas, el intercambio no será llevado a cabo. Este parámetro afecta tanto salidas de FB como entradas. Los valores de los parámetros son:

- **0x00:** no se intercambian bytes en el área de I/O del FB.
- **0x01:** intercambio de bytes en el área de I/O del *Fieldbus*.

A.5.2 FB Out Config (#41)

Este parámetro configura el tamaño del área de salida del FB. El tamaño se especifica en bytes.

A.5.3 FB Out Actual (#42)

Este parámetro guarda el tamaño actual de salida del FB después de la inicialización. El tamaño se especifica en bytes.

A.5.4 FB In Actual (#43)

Permite guardar el tamaño actual de entrada del FB después de la inicialización. El valor de este parámetro se calcula usando el parámetro #45 “FB In SSC Size” y el parámetro #47 “FB In SCI Size”. El tamaño está especificado en bytes.

A.5.5 FB In SSC Offset (#44)

Este parámetro se usa para configurar la localización del SSC input.

A.5.6 FB In SSC Size (#45)

El anterior parámetro permite especificar cómo serán mapeados las entradas SSC desde el área de entradas del FB.

A.5.7 FB In SCI Offset (#46)

Este parámetro se utiliza para configurar la localización de las entradas del canal SCI.

A.5.8 FB In SCI Size (#47)

El anterior parámetro permite especificar cómo serán mapeados las entradas SCI desde el área de entradas del FB.

A.5.9 SSC Byte Order (#50)

Este parámetro determina si los bytes de I/O en el canal SSC serán intercambiados a otras áreas de I/O.

Ejemplo Si el dato mapeado desde “SSC Input” a “FB Input”, el parámetro #50 “SSC Byte Order” es 0x00, y el parámetro #40 “FB Byte Order” es 0x01, el dato escrito en el área “SSC Input” será intercambiado cuando este alcance el área de el “FB Input”. Los valores que puede tomar este parámetro son:

- **0x00**: no se intercambiaran bytes en el área SSC I/O.
- **0x01**: se Intercambiaron datos en el canal SSC I/O.

A.5.10 SSC In Config (#51)

Este parámetro se utiliza para configurar el tamaño total del canal SSC.

A.5.11 SSC In Auto (#52)

Muestra la autoconfiguración del tamaño del área SSC el tamaño es especificado en bytes.

A.5.12 SSC In Actual (#53)

Este parámetro entrega el tamaño actual del área de SSC después de la inicialización.

A.5.13 SSC Out Config (#54)

Se usa para configurar el tamaño total del área SSC de salida.

A.5.14 SSC Out Auto (#55)

Entrega la configuración automática del tamaño de salida de SSC.

A.5.15 SSC Out Actual (#56)

Parámetro que retorna el tamaño actual de de salida del SSC después de la inicialización.

A.5.16 SSC Out FB Offset (#57)

Este parámetro se usa para configurar la localización de la fuente de salidas del FB.

A.5.17 SSC Out FB Size (#58)

Este parámetro es usado para especificar cómo serán mapeados los bytes de salida del FB al área de salida del canal SSC.

A.5.18 SSC Out SCI Offset (#59)

Este parámetro se usa para configurar la localización de la fuente del canal SSC.

A.5.19 SSC Out SCI Size (#60)

Se usa para especificar como los bytes son mapeados desde el área de entrada del canal SCI al área de salida del canal SSC.

A.5.20 SCI Byte Order (#63)

Parámetro que determina si los bytes en el área de I/O del canal SCI serán intercambiados a otras áreas de I/O.

Ejemplo: Si el dato es mapeado desde el “SCI Input área” al “FB Input área el parámetro #63 “SCI Byte Order” es 0x00 y el #40 “FB Byte Order” es 0x01 el dato escrito en el canal SCI será intercambiado cuando este alcance el área del “FB Input área”. Los valores de este parámetro son:

- **0x00:** no intercambio de bytes en el área I/O.
- **0x01:** Intercambio de bytes en el área de I/O SCI.

A.5.21 SCI In Config (#64)

Este parámetro configura el tamaño del área de entrada del canal SCI (bytes).

A.5.22 SCI In Actual (#65)

Este parámetro guarda el tamaño actual del SCI In después de la inicialización (bytes).

A.5.23 SCI Out Actual (#66)

Este parámetro guarda el tamaño actual del SCI Out después de la inicialización (bytes).

A.5.24 SCI Out FB Offset (#67)

Se usa para modificar la localización de la fuente FB Out (SCI Out mapeo).

A.5.25 SCI Out FB Size (#68)

Este parámetro se usa para especificar cómo serán mapeados los bytes desde el área del FB Out al área del SCI Output.

A.5.26 SCI Out SSC Offset (#69)

Este parámetro se usa para configurar la localización de la fuente de datos para el SSC Input. (SCI output mapeo).

A.5.27 SCI Out SSC Size (#70)

Parámetro usado para especificar cómo se mapearan los bytes desde el área SSC input al área SCI out.

A.6 PARÁMETROS ESPECÍFICOS DEL CONTROLADOR AB6001

Ésta sección está orientada a describir los parámetros específicos del controlador AB6001, describe las características de cada parámetro, define las opciones de configuración, etc. La tabla 7 indica los parámetros generales del controlador ABIC6001.

Tabla 7. Parámetros específicos del controlador AB6001

#	Dirección	Nombre	tamaño	V defecto	Acceso
100	0x7000	FB Status	2 byte		R
102	0x7022	FB Password	2 byte		W
103	0x7002	FB Node Address Config	1 byte	0x3F	R/W
104	0x7003	SSC FB Node Address	1 byte		R
105	0x7004	FB Actual Node Address	1 byte		R
106	0x7005	DeviceNet baudrate config	1 byte	0x00	R/W
107	0x7006	DeviceNet baudrate actual	1 byte		R
108	0x7007	DeviceNet Vendor ID 2 byte	2 byte	0x005A	R(W)
109	0x7008	DeviceNet Product Type	2 byte	0x000C	R(W)
110	0x7009	DeviceNet Product Code	2 byte	0x003D	R(W)
111	0x700A 0x7019	DeviceNet Product Name	32 byte a	AB6001	R(W)
113	0x701C	Connection Status Polled	1 byte		R
114	0x701D	Connection Status COS	1 byte		R
115	0x701E	Connection Status Bit Strobed	1 byte		R
116	0x0x701F 701E	Connection Status Cyclic	1 byte		R
117	0x7020	FB Auto Baud Cfg	1 byte	0x00	R/W
118	0x7021	FB Quick Connect Cfg	1 byte	0x00	R/W
200	0x8000	Application Parameter	1	b	R/W
201	b	Application Parameter	2	b	R/W
202	b	Application Parameter	3	b	R/W
..
c	bc	Application Parameter 50c	b		

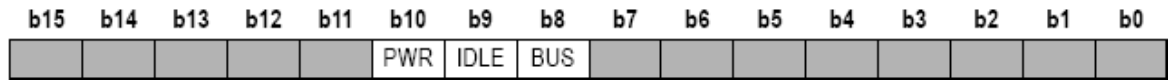
Algunas características de estos parámetros son:

- Se pueden leer y escribir los 32 byte.
- Parámetros dependientes.
- Es posible registrar hasta 50 parámetros de con un máximo de un byte.

A.6.1 FB Status (#100)

Este parámetro guarda información acerca del estado actual del bus, si está en línea o si está fuera de línea. La figura 14 describe la distribución de bits de este parámetro.

Fig. 14. Organización de los bits del parámetro (#100)



- **BUS.**
1: el Bus está *on line*
0: el Bus está *off line*.
- **IDLE1.**
1: el Bus está en modo *idle*
0: el Bus no está en modo *Idle*.
- **PWR.**
1: la fuente de alimentación de 24V está conectada al bus
0: no hay fuente de alimentación activa.

A.6.2 FB Password (#102)

Este parámetro se usa para desbloquear los siguientes parámetros:

- DeviceNet Vendor ID (#108).
- DeviceNet Product Type (#109).
- DeviceNet Product Code (#110).
- DeviceNet Product Name (#111).

A.6.3 FB Node Address Config (#103)

Mantiene la configuración manual de direccionamiento del nodo DeviceNet.

A.6.4. SCC FB Node Address (#104)

Mantiene la autoconfiguración de direccionamiento del nodo desde el la interfaz SSC. Para que este valor sea valido se debe modificar el bit 5 en el parámetro #8.

A.6.5. FB Actual Node Address (#105)

Este parámetro mantiene el valor actual de la dirección del nodo.

A.6.6. DeviceNet Baud rate Config (#106)

Se usa para cambiar el Baud Rate del nodo DeviceNet. Para que este valor sea válido el bit 5 en el parámetro #8 (configuración bits) debe ser modificado. Los valores de baudrate que maneja DeviceNet son:

0x00: 125kbps.

0x01: 250kbps.

0x02: 500kbps.

A.6.7 DeviceNet Baud rate Actual (#107).

Este parámetro guarda el valor actual del baud rate. Estos valores son son:

0x00: 125kbps.

0x01: 250kbps.

0x02: 500kbps.

A.6.8 DeviceNet Vendor ID (#108)

Es usado para cambiar el *Vendor ID* (identificador vendedor). Este parámetro está bloqueado desde el parámetro #102 (*FB Password*). Es importante saber que si este parámetro se altera se debe también cambiar los archivos EDS acorde a la modificación realizada.

A.6.9 DeviceNet Product Type (#109)

Se usa para cambiar el tipo de producto (*Product Type*) el nodo se debe desbloquear usando el parámetro #102. Si se modifica este parámetro también se debe modificar los EDS.

A.6.10 DeviceNet Product Code (#110)

Este parámetro se utiliza para cambiar el código del producto. Si se modifica este parámetro también se debe modificar los archivos EDS.

A.6.11 DeviceNet Product Name (#111)

Parámetro usado para cambiar el nombre del producto. El modulo se debe desbloquear usando el parametro#102. Si se modifica este parámetro también se debe modificar los archivos EDS.

A.6.12 Connection Status Polled (#113)

Este parámetro retorna el estado de la conexión, para conexiones Polled I/O. Los valores que puede tomar este parámetro son:

- 0x00: no existe conexión.
- 0x01: conexión en configuración.
- 0x03: conexión estabilizada.
- 0x04: Conexión fuera de tiempo (*time out*).

A.6.13 Connection Status COS (#114)

Este parámetro retorna el estado de la conexión, para la conexión cambio de estado. Los valores que puede tomar el parámetro son:

- 0x00: no existe conexión.
- 0x01: conexión en configuración.
- 0x03: conexión estabilizada.
- 0x04: Conexión fuera de tiempo (*time out*).

A.6.14 Connection Status Bit Strobed (#115)

Este parámetro retorna el estado de la conexión, para conexiones bit strobed. Los valores que puede tomar el parámetro son:

- 0x00: no existe conexión.
- 0x01: conexión en configuración.
- 0x03: conexión estabilizada.
- 0x04: Conexión fuera de tiempo (time out).

A.6.15 Connection Status Cyclic (#116)

Este parámetro retorna el estado de la conexión para conexiones cyclic. los valores que puede tomar el parámetro son:

- 0x00: no existe conexión.
- 0x01: conexión en configuración.
- 0x03: conexión estabilizada.
- 0x04: Conexión fuera de tiempo (time out).

A.6.16 FB Auto Baud Cfg (#117)

Este parámetro habilita y deshabilita la característica de auto Baud Rate. Los valores que puede tomar este parámetro son:

- 0x00: Auto Baud deshabilitado (valor por defecto).
- 0x01: Auto Baud habilitado.

A.6.17 FB Quick Connect Cfg (#118)

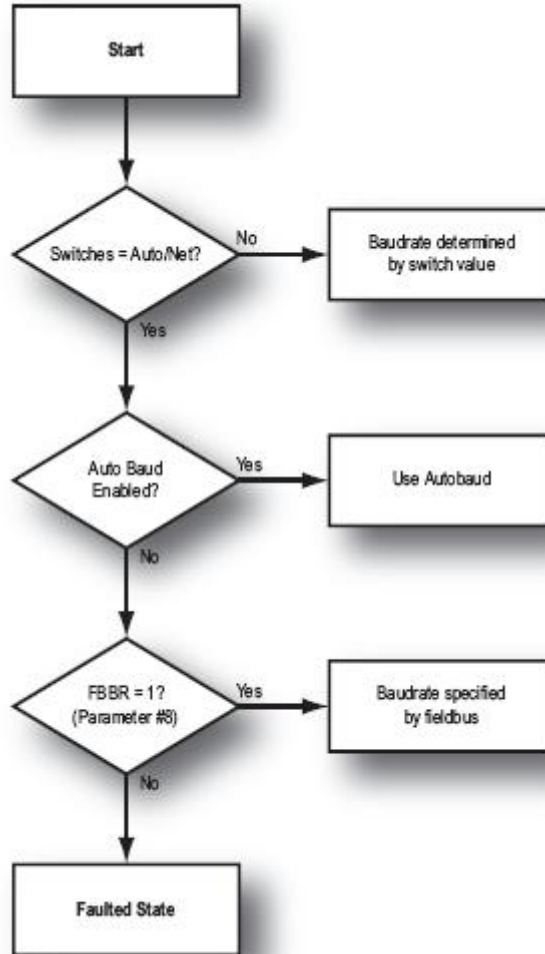
Este parámetro habilita y deshabilita la característica de conexión rápida (*Quick Connect*). Los valores que puede tomar este parámetro son:

- 0x00: conexión rápida deshabilitada (default).
- 0x01: conexión rápida habilitada.

A.7 SECUENCIA DE AUTO/NET DEL NODO DEVICENET

En esta sección se explica la secuencia de auto Baud Rate del controlador. Cuando los switch (b6 y b7)...véase la sección A.6.16...están configurados en *Auto/Net*, el *Baud Rate* se establece acorde al diagrama de flujo de la figura 15:

Fig. 15. Secuencia de auto net del Nodo



A.8 SECUENCIA DE AUTO TEST DEL NODO DEVICENET

A.8.1 Información General de auto test del nodo

El nodo se puede configurar para que efectúe una secuencia de auto chequeo usando el parámetro #1 "*Module Mode*", los resultados del test son mostrados en el parámetro #2 "*Module Status*". El test únicamente se puede realizar antes de inicializar el nodo.

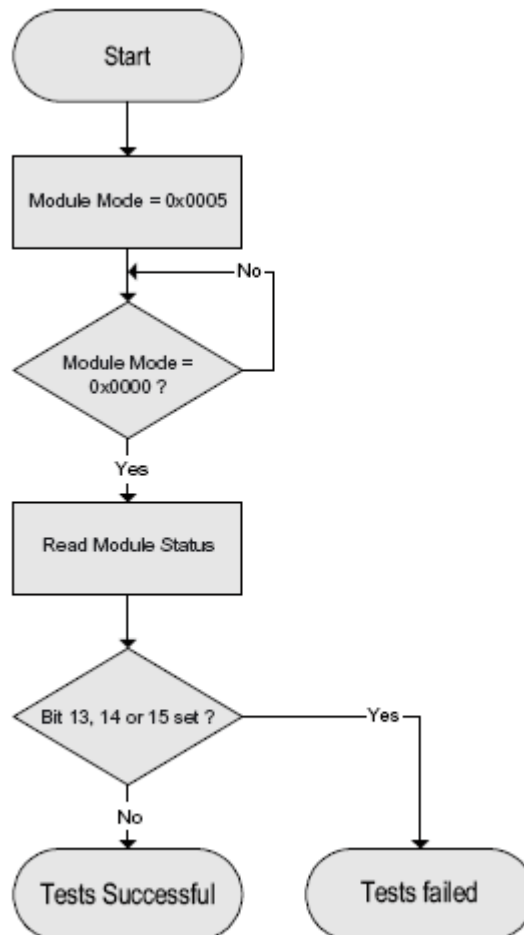
A.8.2 Procedimiento básico para generar un auto test

1. iniciar el testeo del hardware escribiendo 0x0005 en el parámetro #1 ("Module Mode").
2. durante el test el parámetro #1 ("Module Mode") tiene el valor de 0x0005. Cuando termina la operación de escaneo el valor del parámetro cambia a 0x0000.
3. la información del test se puede leer en el parámetro #2 ("Module Status").

A.8.3 Evaluación del test del nodo DeviceNet.

Si el bit 13, 14 o 15 es activo o si está presente, al final del procedimiento de test ha encontrado fallas si los bits están clareados el test es satisfactorio.

Fig. 16. Secuencia de Auto test del controlador AB6001



ANEXO B CONSIDERACIONES PARA REALIZAR EL CABLEADO

B.1 INTRODUCCIÓN

Considerando que en la aplicación del sistema de comunicaciones DeviceNet en los laboratorios del PIAI, se optó por la utilización de cable plano para la línea troncal, este anexo tomando como referencia el estándar DeviceNet solo expone temas referentes al uso de cable plano, los interesados en conocer las características de los medios físicos empleados por DeviceNet pueden consultar la especificación DeviceNet a través de la ODVA. La elección de cable plano Clase 1 se dio teniendo en cuenta las ventajas que presta (conectores desplazables, capacidad de corriente alta, facilidad para adicionar nuevos nodos, etc.).

El cable plano contiene cuatro conductores: un par rojo y negro para alimentación de corriente continua de 24 V; un par azul y blanco para datos. El cable plano Clase 1 diseñado para sistemas de 8 A, el cable plano de Clase 2 no debe superar los 4 A. la tabla 8 de esta sección describe las función de los conductores del cable DeviceNet.

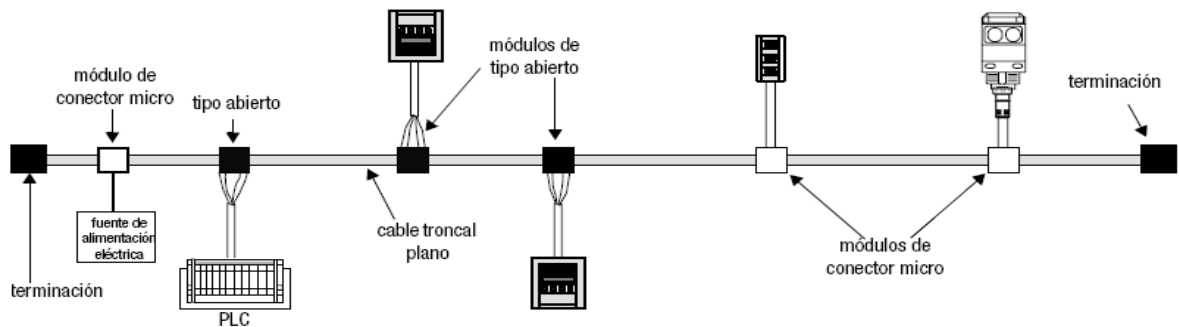
Tabla 8. Características de los conductores para cable plano

Color del cable	Identidad del cable	Uso del Cable plano
blanco	CAN_H	Señal
azul	CAN_L	Señal
sin forro	tierra	n/a
negro	V-	alimentación
rojo	V+	alimentación

B.2 COMPONENTES DE LA RED DEVICENET

A continuación se citan los componentes de la red DeviceNet del diseño en consideración, además se exponen elementos adicionales con el propósito de facilitar las etapas futuras de expansión y crecimiento del sistema de comunicaciones del PIAI. La figura 17 describe la arquitectura física de una red DeviceNet.

Fig. 17. Arquitectura física de una red DeviceNet con cable plano

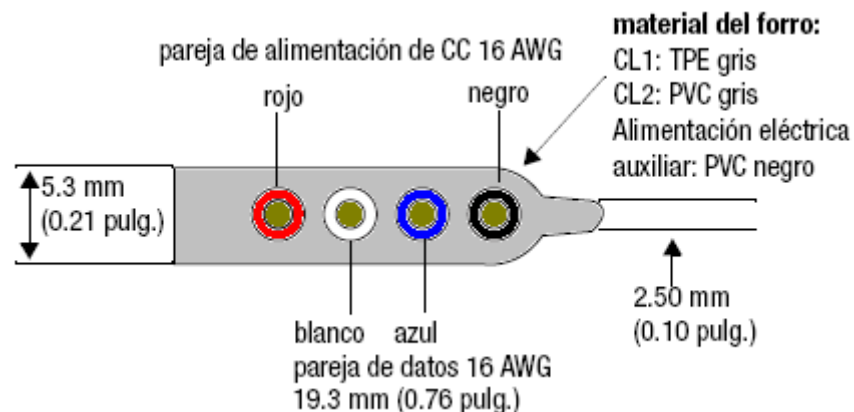


B.2.1 Línea troncal DeviceNet

- **Características del cable plano, KwikLink 1485C-P1E75.**

El cable plano KwikLink tiene una guía física para evitar fallos de conexión, las tres variaciones del cable KwikLink no disponen de blindaje y contienen cuatro conductores (rojo blanco azul y negro). El cable plano se usa exclusivamente para la línea troncal. La figura 18 representa un extremo del cable plano DeviceNet.

Fig. 18. Cable plano DeviceNet



- **Cable Clase 1 (CL1).** Según las especificaciones NEC correspondiente a un circuito Clase 1 dice que: la fuente de alimentación eléctrica debe tener una salida nominal de menos de 30 V y 1000 VA, en función del tamaño de los conductores del cable plano, la corriente máxima a través de la red no debe superar los 8 A.

- **Cable Clase 2 (CL2).** Más flexible físicamente que el cable CL1, cumple con el Artículo 725 del NEC, que especifica que para un circuito de Clase 2, la fuente de alimentación eléctrica debe tener una salida nominal de menos de 30 V y 100 VA. En el caso de DeviceNet, funcionando a 24 V, la corriente máxima permitida es de 4 A.

B.2.2 Línea de derivación DeviceNet

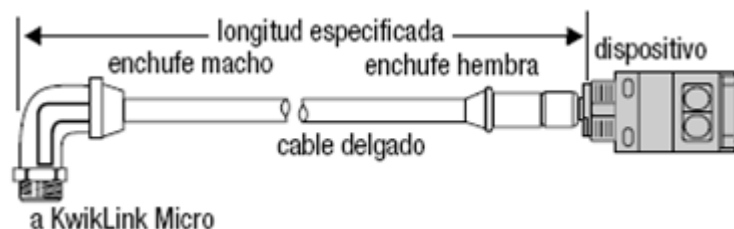
La Línea de derivación conecta tomas ubicadas en la línea troncal a nodos en la red. Los conjuntos de cable delgado predeterminados para utilizar como línea de derivación están disponibles con diversos tipos de conectores y con longitudes de 1, 2, 3 y 5 metros. El cable de derivación KwikLink es un cable de 4 conductores no blindado.

Información sobre los cables de derivación KwikLink. Estos cables de derivación de PVC de cuatro conductores sin blindar diseñados específicamente para utilizarlos con conectores KwikLink tiene algunas variantes que son: conectores micro macho de 90° a hembra recta, micro hembra o conductores al dispositivo. Los cables de derivación (1485K) son para uso exclusivo con el sistema de cable plano KwikLink. La figura 19 representa la conexión a una toma KwikLink desde un dispositivo abierto, mientras que la figura 20 muestra un ejemplo de una conexión a una toma KwikLink desde un dispositivo sellado.

Fig. 19. Cable de derivación 1485K



Fig. 20. Líneas de derivación para dispositivos sellados



B.2.3 Conectores DeviceNet.

DeviceNet ofrece una amplia variedad de conectores, entre estos tenemos: tomas deviceport, devicebox, conectores abiertos, cerrados, etc. El lector interesado en conocer mas a fondo este tema puede consultar la especificación DeviceNet a través de la ODVA.

Fig. 21. Multitomas DeviceNet

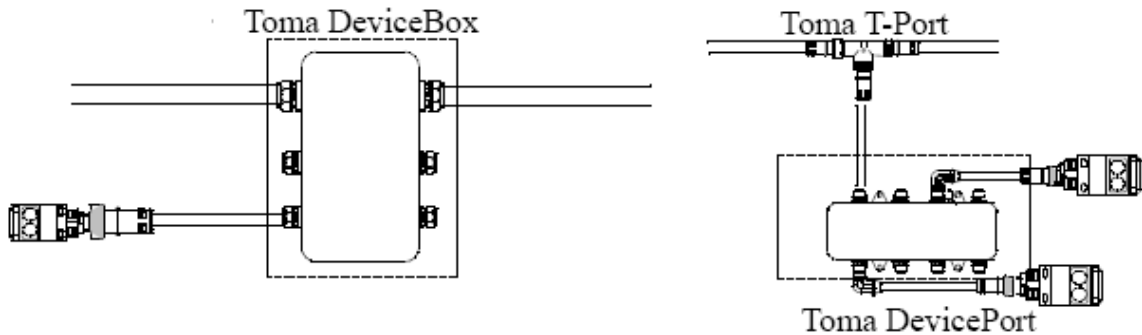
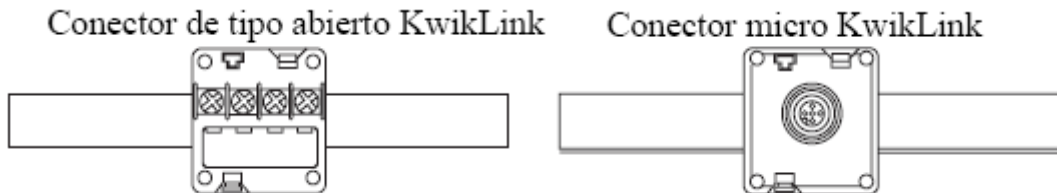


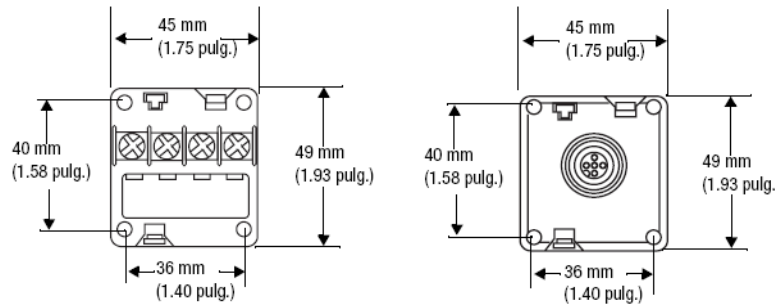
Fig. 22. Tipos de conectores para cable plano Kwiklink



- **Los conectores con aislamiento desplazable (IDC) KwikLink.**

Los conectores con aislamiento desplazable (IDC) KwikLink interconectan cables de derivación y dispositivos a la línea troncal de cable plano. Cada conector está conformado por una base integrada por dos piezas con bisagras que encajan alrededor del cable plano en cualquier punto de la línea troncal, los contactos con los conductores del cable se hacen apretando dos tornillos que empujan los contactos contra el forro del cable y dentro de los conductores. La figura 23 expone los dos tipos de conectores empleados en este trabajo de grado.

Fig. 23. Conectores desplazables



Los conectores KwikLink de Allen-Bradley sólo están aprobados para ser usados con los cables planos DeviceNet listados en la tabla 9.

Tabla 9. Cables compatibles con conectores KwikLink.

Número de parte	Descripción	Suministrador	Color
1485C-P1E	CL1	Allen-Bradley	Gris
1485C-P1G	CL2	Allen-Bradley	Gris

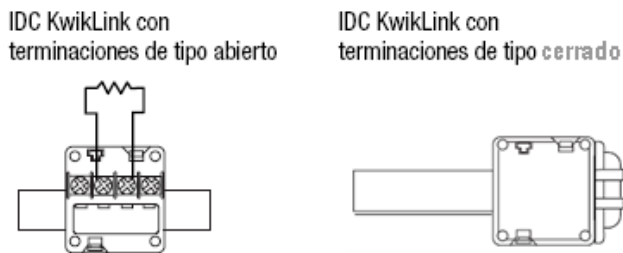
B.2.4 Terminaciones de red DeviceNet

Las resistencias de terminación sirven para estabilizar eléctricamente la comunicación DeviceNet, deben estar ubicadas en cada extremo de la línea troncal, con una resistencia de 121 ohmios, 1 %, 1/4 W.

- **Terminación de tipo abierto.**

Estas resistencias deberán conectarse directamente a través de los conductores azul y blanco del cable DeviceNet como indica la figura 24.

Fig. 24. Tipos de terminación DeviceNet



B.3 TAREAS EN LA EJECUCIÓN DEL CABLEADO DEVICENET

Se recomienda seguir atentamente los pasos definidos por la especificación DeviceNet expuestos en este documento, para todas las operaciones y procedimiento para realizar el cableado de la red DeviceNet.

Antes de comenzar, se debe asegurar que:

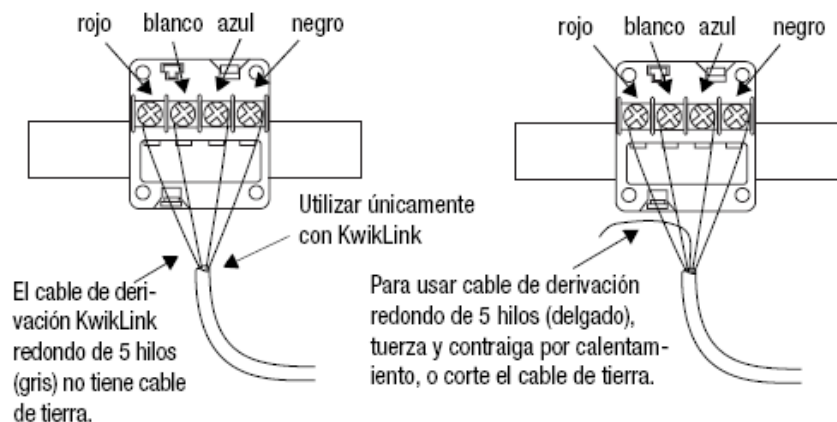
- el sistema de cables DeviceNet esté inactivo.
- todos los dispositivos conectados estén apagados.
- cualquier fuente de alimentación conectada esté desactivada.
- seguir las instrucciones del fabricante para pelar, acoplar y/o apretar los diferentes componentes del sistema de cableado.

B3.1 Instalación de un conector tipo abierto KwikLink a un cable de derivación

La instalación del cable de derivación es una operación muy simple pero se aconseja seguir los pasos detalladamente para evitar fallas de cableado y daños en los componentes.

Las líneas de derivación se pueden implementar con cable redondo serie 1485R de 5 hilos o con cable KwikLink serie 1485K de 4 hilos. La figura 25 exhibe la forma física de conectar el cable de derivación a un conector abierto.

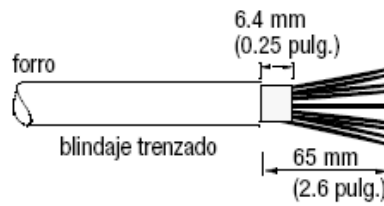
Fig. 25. Acople de las líneas de derivación al conector abierto KwikLink



Para acoplar un conector de tipo abierto “enchufable” a una línea de derivación se debe seguir los siguientes pasos:

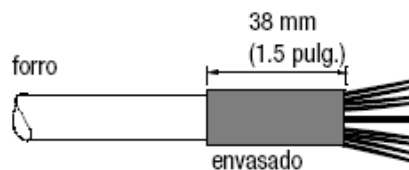
1. Pelar 65 mm del forro exterior de un extremo del cable, dejando expuesto no más de 6.4 mm del blindaje trenzado como se presenta en la figura 26.

Fig. 26. Adecuación del cable de derivación



2. Ajustar el extremo del cable con 38 mm de cinta retráctil, cubriendo la parte de los conductores expuestos y la parte del aislamiento de la línea de derivación, como indica la figura 27.

Fig. 27. Recubrimiento del cable de derivación



3. Pelar 8.1 mm del aislamiento del extremo de cada uno de los conductores aislados, como se indica en la figura 28.

Fig. 28. Adecuación de los conductores del cable de derivación.

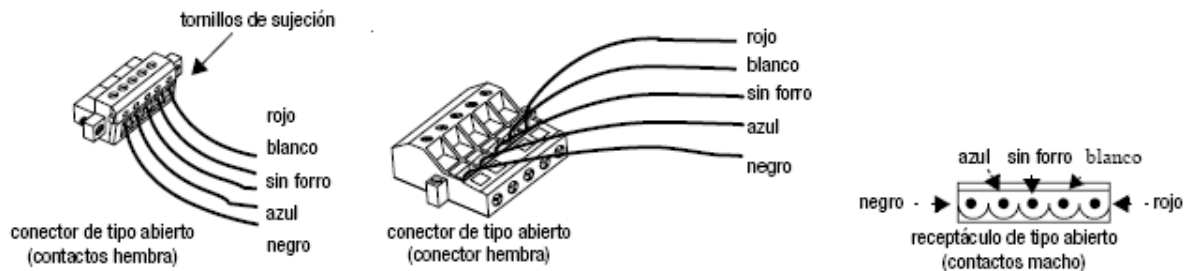


4. Estañar los últimos 6.5 mm de los conductores sin forro de modo que la dimensión exterior no exceda los 0.17 mm.

5. Insertar cada conductor en la cavidad de fijación apropiada del conector de tipo abierto o el terminal de tornillos del dispositivo, según el color del cable.

6. Apretar los tornillos de sujeción para asegurar cada conductor. Los contactos macho del conector del dispositivo deben coincidir con los contactos hembra del conector. La figura 29 indica la forma de insertar los cables en el conector.

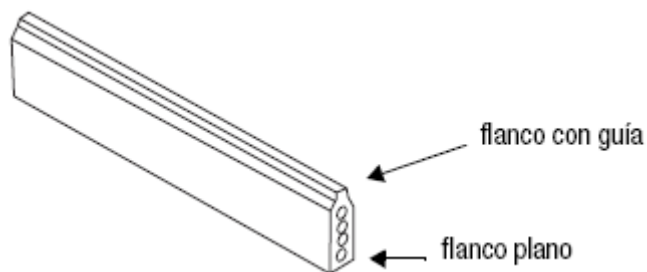
Fig. 29. Conectores tipo abierto



B.3.2 Instrucciones para la instalación del cable KwikLink

Instale el cable KwikLink con el flanco plano más ancho del cable en la parte inferior, como indica la figura 30.

Fig. 30. Cable plano para la línea troncal

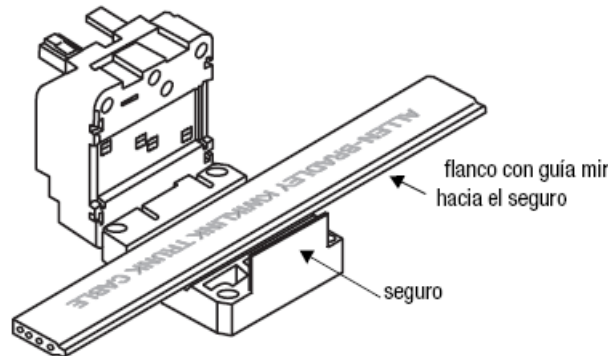


B.3.3 Pasos para instalar correctamente el cable KwikLink en un conector IDC

1. Colocar el cable en la base con bisagras, poniendo atención al perfil guía; el flanco que no tiene guía está más cerca de la bisagra, el flanco con guía, mira hacia el seguro para obtener información mas detallada ver la figura 31.

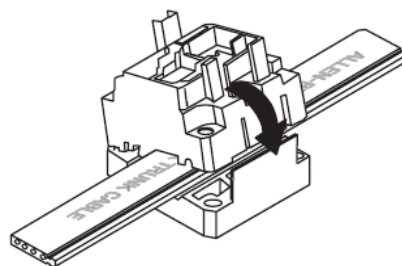
Importante: Antes de cerrar el conector, asegurarse que las hojas del IDC no sobresalen del envoltorio. Si las hojas quedan expuestas, introdúzcalas suavemente en la base. En caso de que las hojas no entren con facilidad o sólo entren parcialmente, compruebe que los tornillos del IDC no estén parcialmente apretados.

Fig. 31. Ubicación del conector en el cable plano DeviceNet



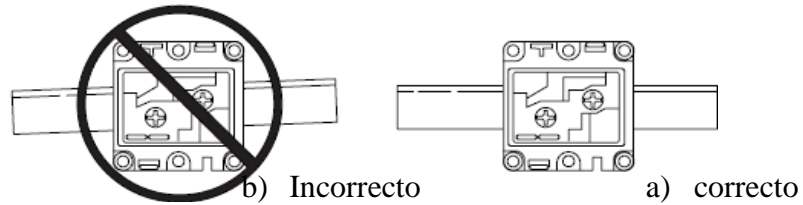
2. Cerrar el conjunto de bisagras haciendo presión hasta que el seguro cierre en su sitio, como indica la figura 32. El seguro tiene dos pestillos, el primer pestillo sujeta suavemente el conector en el cable, el segundo pestillo necesita más presión para cerrar el conector, si el cable no está en la posición correcta, el conector no cerrará.

Fig. 32. Ajuste del conector IDC



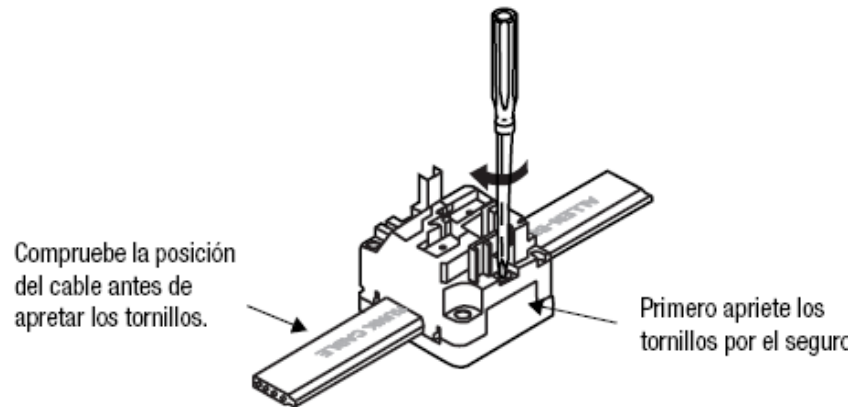
3. Comprobar que el cable está recto antes de pasar al paso cuatro. Antes de apretar los tornillos debe asegurarse que el cable esté recto, la ubicación inadecuada del cable puede causar un mal funcionamiento del dispositivo, daños en el cable y en el conector. La figura 33 a) muestra una posición incorrecta del cable plano mientras que la figura 33 b) indica la posición correcta del mismo

Fig. 33. Verificación de la posición correcta del conector IDC



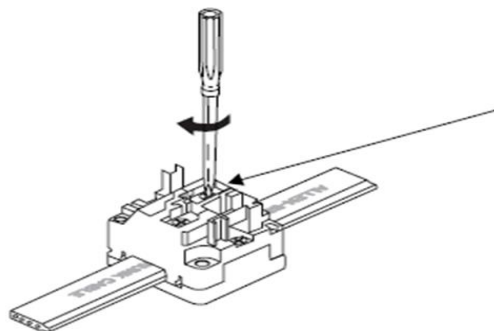
4. Apretar los dos tornillos ubicados en los puntos centrales de la bisagra y en los laterales del seguro de la base; apretar primero el lado del seguro. Tener cuidado para evitar peladuras, un par suficiente es 5.56 N (15 pulg.-lbs). La figura 34 indica la forma de ajustar los tornillos de la base del conector.

Fig. 34. Ajuste de la base del conector IDC



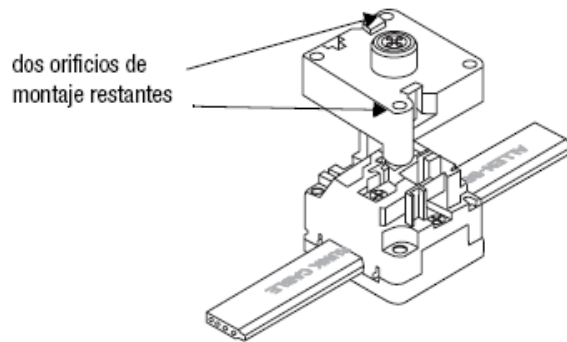
5. Introducir los contactos del IDC en el cable, apretando los dos tornillos en el centro del conjunto de la base como indica la figura 35, aplicar un par suficiente de 5.56 N (15 pulg.-lbs).

Fig. 35. Fijación de los contactos del conector en el cable DeviceNet



6. Alinear los orificios rectangulares con la guía de la interface de conexión con los puntos correspondientes de la base y encajar el micro módulo en su sitio como indica la figura 36. Asegurar el módulo apretando los tornillos en los dos orificios de montaje restantes.

Fig. 36. Inserción del micromódulo al conector IDC

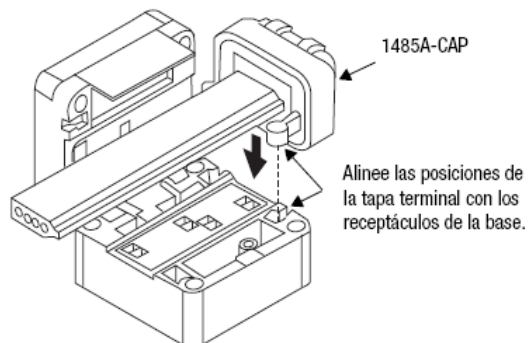


B.3.4. Instalación de la tapa terminal del conector IDC

Cada módulo de terminación de KwikLink se suministra con una tapa terminal diseñada para cubrir el extremo expuesto del cable. Pasos para instalar la tapa terminal:

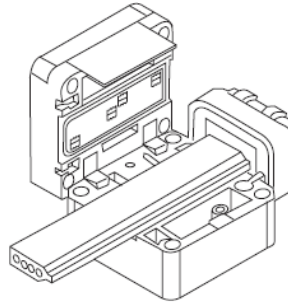
1. Colocar la tapa terminal (1485A-CAP) en el cable como indica la pestaña guía. Alinear las posiciones de la tapa terminal con los receptáculos de la base IDC inferior y empujar hacia abajo hasta que la tapa terminal esté firmemente encajada. La figura 37 indica la forma correcta de ubicar la tapa terminal.

Fig. 37. Ubicación de la tapa terminal en el conector IDC



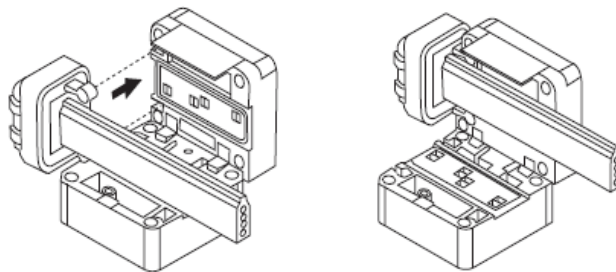
2. Cerrar la base del IDC y continuar con el proceso de conexión tal como se muestra en la figura 38.

Fig. 38. Forma en que se debe fijar la base del IDC



Cuando se instale la tapa terminal en el otro extremo del cable, se debe observar que los receptáculos guía estén en la parte superior de la base del IDC. Repetir el proceso de instalación de la tapa terminal, tal como se ha indicado anteriormente en la sección previa. Cerrar la base del IDC y continuar con la conexión tal como se indica en las instrucciones de instalación. La figura 39 indica la posición de los receptáculos guía.

Fig. 39. Posición de los receptáculos guía en el conector Kwiklink



B.4 RECOMENDACIONES PARA CONECTAR FUENTES DE ALIMENTACIÓN A MEDIOS FÍSICOS PLANOS KWIKLINK

Sistema 8 A, Clase 1. Para un Sistema de 8 A, Clase 1, la alimentación sólo puede ser interconectada a través de un conector de tipo abierto KwikLink. DeviceNet especifica que la fuente de alimentación tenga un máximo regulado de 24 VCC y que el circuito de alimentación esté limitado a 8 A, según la especificación DeviceNet los conductores que transportan alimentación eléctrica del cable están dimensionados para una carga máxima de 8A.

Sistema 4 A, Clase 2. Para un Sistema de 4 A, Clase 2, la alimentación puede aplicarse a la red utilizando conectores KwikLink micro o de tipo abierto. DeviceNet especifica que la fuente de alimentación sea de un máximo de 24 VCC, y la corriente máxima permitida debe ser de 4A.

B.4.1. Pasos para instalar la fuente de alimentación DeviceNet

En el momento de la instalación, asegúrese de que la fuente de alimentación de CA permanezca desactivada.

1. Montar la fuente de alimentación, en un lugar con buena ventilación.
2. Conectar la fuente de alimentación usando un cable con un par de conductores calibre 12 o el equivalente, o dos pares de conductores calibre 15 con una longitud máxima de cable de 3 m a la toma PowerTap o al conector kwikLink destinado para incorporar la fuente de alimentación a la red.

B.5 MÉTODOS PARA DETERMINAR SI SE DISPONE DE LA ALIMENTACIÓN ADECUADA

Existen dos métodos para determinar si la red dispone de la alimentación requerida por lo dispositivos DeviceNet estos métodos son: Método de cálculo total y Método de búsqueda. A continuación se presentan algunos ejemplos de cada método, tomados de la especificación DeviceNet.

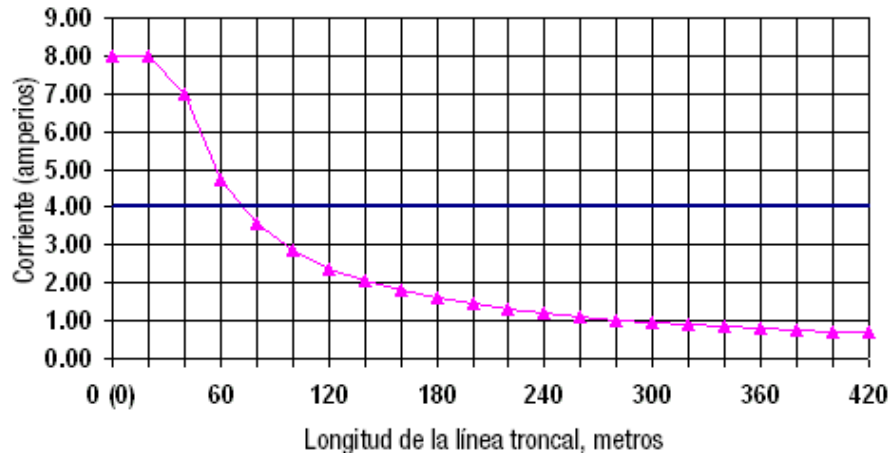
B.5.1 Método de búsqueda

Es un método gráfico que consiste en hallar la corriente total de todos los dispositivos, conociendo la longitud de la red se busca el siguiente valor más grande en la figura correspondiente al sistema o en una tabla que determina la corriente máxima permitida con relación a la longitud del cable. El sistema tendrá la apropiada alimentación eléctrica si la carga total no supera el valor mostrado por la curva o la tabla. El peor de los casos es donde todos los nodos están juntos en el extremo opuesto de la fuente de alimentación.

- **Una fuente de alimentación eléctrica en segmento final del cable KwikLink.**

Se supone que todos los nodos están en el extremo opuesto del cable, a la fuente de alimentación. La figura 40 es el grafico correspondiente a esta descripción donde la fuente está ubicada en el extremo izquierdo de la figura y los nodos en el extremo derecho.

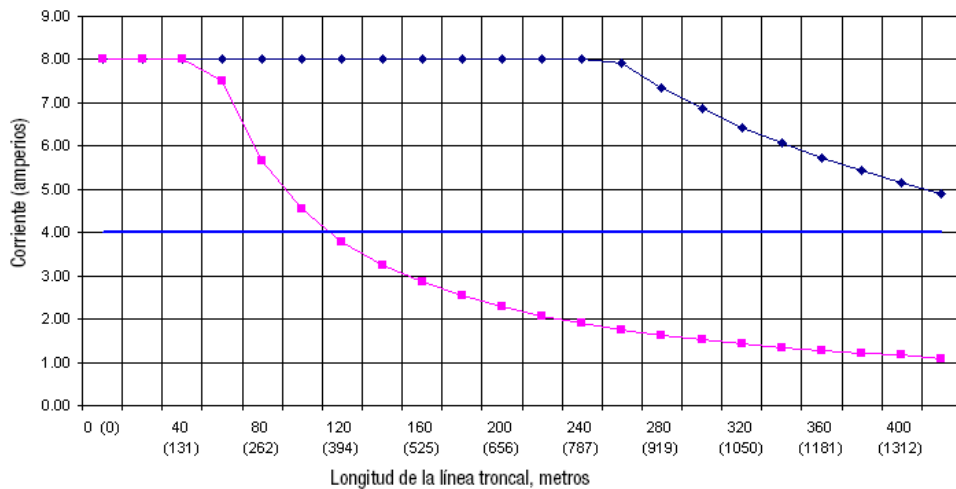
Fig. 40 Fuente de alimentación en un extremo del cable plano



- **Dos fuentes de alimentación eléctrica (una conectada al final, y otra conectada a la mitad del cable plano).**

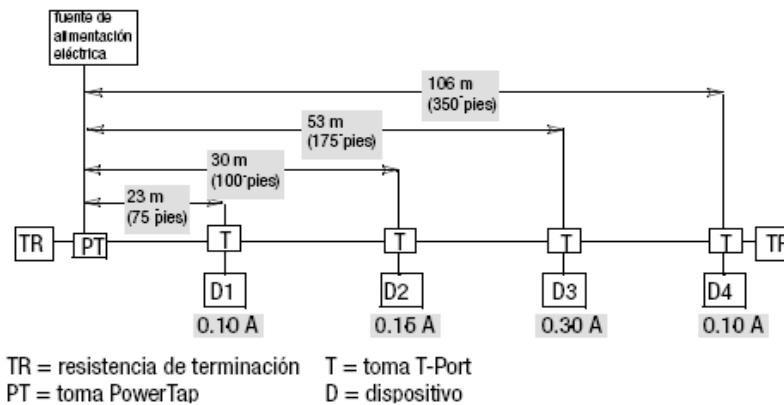
La figura 41 describe 2 fuentes de alimentación conectadas en el mismo sistema, se puede evidenciar el aumento de la capacidad de corriente del sistema con respecto al anterior sistema.

Fig. 41. Un par de fuentes de alimentación aplicadas al sistema



Ejemplo: de la figura 42 es un sistema con cuatro dispositivos al cual se le aplicara el método de búsqueda para determinar la capacidad de corriente de la fuente de alimentación a través de la sección del cable. El objetivo principal de este ejemplo es comprobar si el sistema funciona adecuadamente, teniendo en cuenta los requerimientos de corriente, la longitud del cable y la fuente de alimentación.

Fig. 42. Ejemplo del método de búsqueda en un sistema con cuatro dispositivos



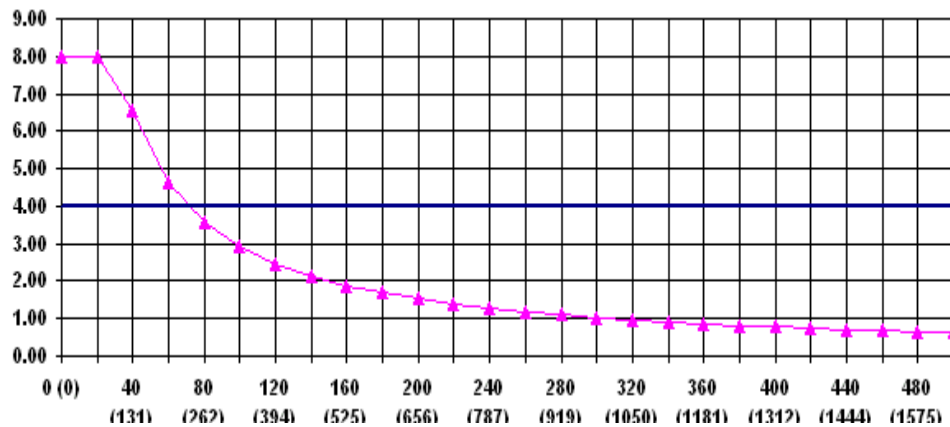
Pasos para aplicar el método de búsqueda a un sistema con fuente de alimentación en el extremo del cable.

1. Determinar la longitud total de la red. – 106 m.
2. Sumar la corriente de cada dispositivo para hallar el total de corriente. – $0.10 + 0.15 + 0.30 + 0.10$.
3. Hallar el valor siguiente más grande de la longitud de red mediante la figura 43 para determinar la corriente máxima permitida para el sistema. Como el valor siguiente a 106m es 120m, en 120 m la corriente es (2.47 A) y el consumo de los nodos es 0.65 A.

Puesto que la corriente total (0.65 A) no excede la corriente máxima permitida (2.47 A), el sistema funcionará correctamente. Si el sistema no cumple con los requerimientos de corriente del método de búsqueda se debe:

- utilizar el método de cálculo total descrito en la sección B.5.1 de este capítulo.
- mover la fuente de alimentación a algún lugar en el centro del sistema de cables y vuelva a hacer una evaluación.

Fig. 43. Fuente de alimentación en un extremo del cable plano



B.5.1 Método de cálculo total

Se usa el método de cálculo total si la evaluación inicial indica que una sección está sobrecargada o si los requisitos de configuración no pueden cumplirse usando el método de búsqueda.

Una fuente de alimentación que no está conectada en un extremo crea dos secciones de línea troncal, se debe evaluar cada sección independientemente.

- **Ecuación del método del cálculo total.**

$$\text{SUM } \{[(Ln \times (Rc)) + (Nt \times (0.005))] \times In\} < 4.65 \text{ V}$$

Donde:

Ln.

L = La distancia (m o pies) entre el dispositivo y la fuente de alimentación, excluyendo la longitud de la línea de derivación.

n = El número de un dispositivo que se va a evaluar, comenzando con uno para el dispositivo más cercano a la fuente de alimentación y aumentando en uno para el siguiente dispositivo. La ecuación suma la línea de derivación calculada para cada dispositivo y la compara con 4.65 V.

Rc. Es una constante que depende del tipo de cable donde:

Cable grueso 0.015 Ω /m.

Cable delgado 0.069 Ω /m.

Cable plano 0.019 Ω /m.

Nt.

El número de tomas entre el dispositivo que está siendo evaluado y la fuente de alimentación. Por ejemplo: cuando un dispositivo es el primero más cercano a la fuente de alimentación, este número es 1, cuando un dispositivo tiene otro dispositivo entre sí mismo y la fuente de alimentación este número es 2, cuando existen 10 dispositivos entre el dispositivo evaluado y la fuente de alimentación, este número es 11.

En el caso de dispositivos conectados a una toma *DeviceBox* o *DevicePort*, se trata la toma como una toma común. La corriente de todos los dispositivos conectados a una de estas tomas debe sumarse y usarse con la ecuación solamente una vez.

(0.005). La resistencia de contacto nominal usada para cada conexión a la línea troncal.

In.

I. El uso de corriente desde el sistema de cables por el dispositivo. En el caso de corrientes que se encuentran dentro del 90 % del máximo, usar la corriente del dispositivo nominal. En cualquier otro caso usar la corriente nominal máxima del dispositivo. Para tomas *DeviceBox* o para tomas *DevicePort*, sumar las corrientes de todos los dispositivos conectados y contar la toma como una toma única.

n. El número de un dispositivo que se están evaluando, comenzando con 1 para el dispositivo más cercano a la fuente de alimentación y aumentando en uno para el dispositivo siguiente.

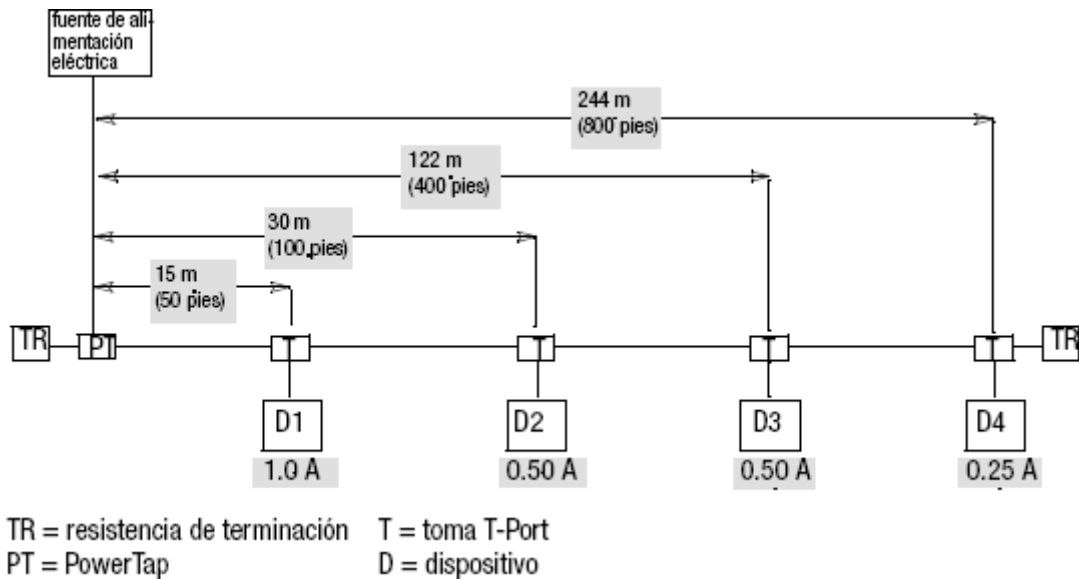
4.65 V. Es la caída máxima de voltaje permitida en la línea troncal DeviceNet. Esta es la caída total de voltaje del sistema de 5.00 V menos 0.35 V reservada para caída de voltaje de línea de derivación.

Ejemplo: Una fuente de alimentación eléctrica (conectada en el extremo).

El siguiente ejemplo usa el método de cálculo total a fin de determinar la configuración para una fuente de alimentación conectada en el extremo de una línea troncal de cable grueso.

El dispositivo 1 y el dispositivo 2 causan la misma caída de voltaje pero el dispositivo 2 está al doble de distancia de la fuente de alimentación y usa la mitad de corriente, el dispositivo 4 usa la menor cantidad de corriente pero está a la mayor distancia de la fuente de alimentación y causa la mayor caída de voltaje incremental.

Fig. 44 Una fuente de alimentación eléctrica conectada en el extremo



1. Hallar los voltajes de cada dispositivo usando la ecuación de cálculo total para cable grueso. La ecuación del cálculo total se puede aplicar con los tres tipos de cable teniendo en cuenta la constante **Rc**.

Reemplazando en la ecuación $SUM \{[(Ln \times (0.0045)) + (Nt \times (0.005))] \times In\} < 4.65 \text{ V}$, se tiene una ecuación para cada dispositivo como se indica a continuación:

D1 **A.** $[(50 \times (0.0045)) + (1 \times (0.005))] \times 1.00 = 0.23 \text{ V}$

D2 **B.** $[(100 \times (0.0045)) + (2 \times (0.005))] \times 0.50 = 0.23 \text{ V}$

D3 **C.** $[(400 \times (0.0045)) + (3 \times (0.005))] \times 0.50 = 0.91 \text{ V}$

D4 **D.** $[(800 \times (0.0045)) + (4 \times (0.005))] \times 0.25 = 0.91 \text{ V}$

1. Sumar el voltaje de cada dispositivo para hallar el voltaje total.

$0.23 \text{ V} + 0.23 \text{ V} + 0.91 \text{ V} + 0.91 \text{ V} = 2.28 \text{ V}$

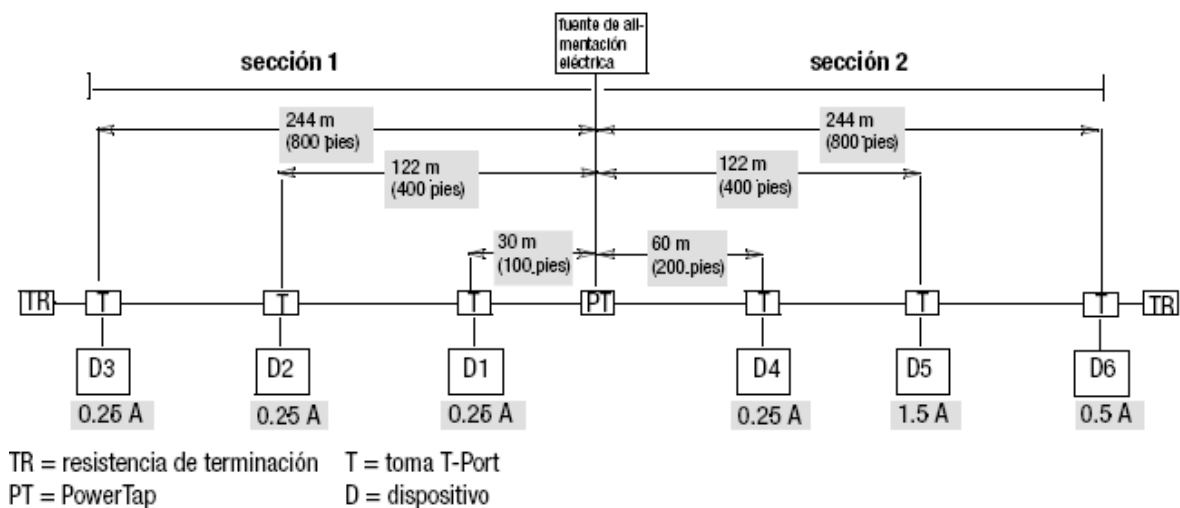
Puesto que el voltaje total no supera los 4.65 V, el sistema funcionará correctamente ($2.28 \text{ V} < 4.65 \text{ V}$). El porcentaje de carga se halla dividiendo el voltaje total por 4.65 V.

$\% \text{ Carga} = 2.28/4.65 = 49 \%$

Ejemplo: Una fuente de alimentación conectada en el centro.

Este ejemplo se usa para verificar la carga a ambos lados de una fuente de alimentación conectada en el centro de una línea troncal de cable grueso.

Fig. 45. Una fuente de alimentación conectada en el centro



1. Hallar los voltajes de cada dispositivo en la sección 1 usando la ecuación para cable grueso.

$$SUM \{[(Ln \times (0.0045)) + (Nt \times (0.005))] \times In\} < 4.65 \text{ V.}$$

D1 A. $[(100 \times (0.0045)) + (1 \times (0.005))] \times 0.25 = 0.12 \text{ V}$

D2 B. $[(400 \times (0.0045)) + (2 \times (0.005))] \times 0.25 = 0.45 \text{ V}$

D3 C. $[(800 \times (0.0045)) + (3 \times (0.005))] \times 0.25 = 0.90 \text{ V}$

2. Sumar el voltaje de cada dispositivo para hallar el voltaje total de la sección 1.

$$0.12 \text{ V} + 0.45 \text{ V} + 0.90 \text{ V} = 1.47 \text{ V.}$$

3. Hallar los voltajes de cada dispositivo en la sección 2.

$$SUM \{[(Ln \times (0.0045)) + (Nt \times (0.005))] \times In\} < 4.65 \text{ V.}$$

D4 A. $[(200 \times (0.0045)) + (1 \times (0.005))] \times 0.25 = 0.23 \text{ V}$

D5 B. $[(400 \times (0.0045)) + (2 \times (0.005))] \times 1.5 = 2.72 \text{ V}$

D6 C. $[(800 \times (0.0045)) + (3 \times (0.005))] \times 0.5 = 1.81 \text{ V}$

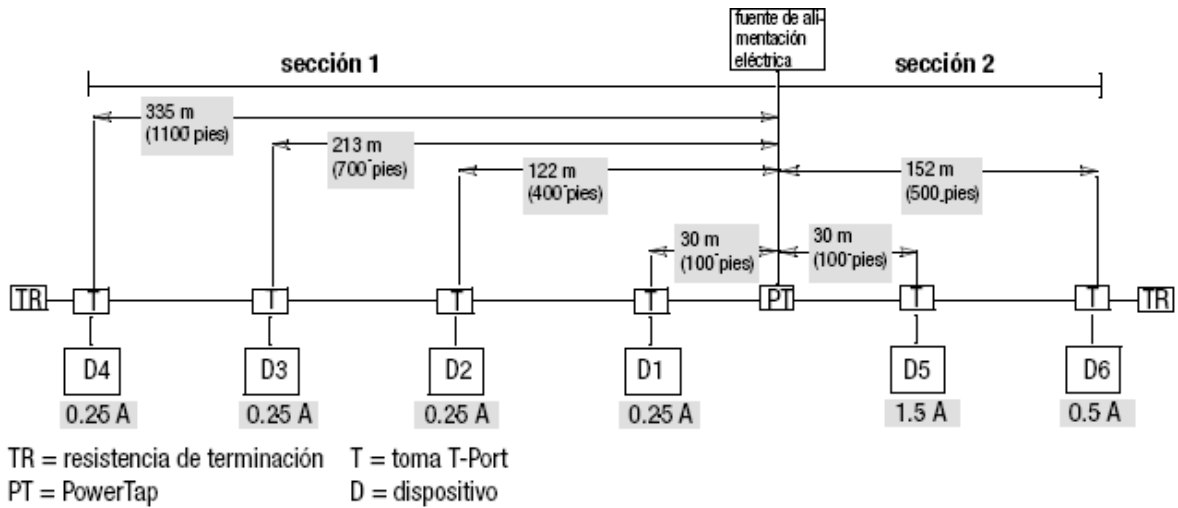
4. Sumar el voltaje de cada dispositivo para hallar el voltaje total de la sección 2.

$$0.23 + 2.72 + 1.81 = 4.76 \text{ V.}$$

Puesto que el voltaje total de la sección 2 supera 4.65 V, el sistema no funcionará correctamente ($4.76 \text{ V} > 4.65 \text{ V}$).

Se puede corregir esta sobrecarga moviendo la fuente de alimentación, en este caso se moverá 91 m hacia la sección sobrecargada. Ahora hay cuatro dispositivos en la sección 1 y dos en la sección 2. Una vez que haya movido la fuente de alimentación, se deben hacer los cálculos de nuevo.

Fig. 46. Moviendo la fuente de alimentación



1. Hallar los voltajes de cada dispositivo en la sección 1.

$$SUM \{[(Ln \times (0.0045)) + (Nt \times (0.005))] \times In\} < 4.65 \text{ V.}$$

D1 A. $[(100 \times (0.0045)) + (1 \times (0.005))] \times 0.25 = 0.11 \text{ V}$

D2 B. $[(400 \times (0.0045)) + (2 \times (0.005))] \times 0.25 = 0.45 \text{ V}$

D3 C. $[(700 \times (0.0045)) + (3 \times (0.005))] \times 0.25 = 0.79 \text{ V}$

D4 D. $[(1100 \times (0.0045)) + (4 \times (0.005))] \times 0.25 = 1.24 \text{ V}$

2. Sumar el voltaje de cada dispositivo para hallar el voltaje total de la sección 1.

$$0.11 + 0.45 + 0.79 + 1.24 = 2.59 \text{ V}$$

3. Hallar los voltajes de cada dispositivo de la sección 2.

$$SUM \{[(Ln \times (0.0045)) + (Nt \times (0.005))] \times In\} < 4.65 \text{ V.}$$

D5 A. $[(100 \times (0.0045)) + (1 \times (0.005))] \times 1.5 = 0.68 \text{ V}$

D6 B. $[(500 \times (0.0045)) + (2 \times (0.005))] \times 0.5 = 1.13 \text{ V}$

4. Sumar el voltaje de cada dispositivo para hallar el voltaje total de la sección 2.

$$0.68 + 1.13 = 1.81 \text{ V.}$$

Puesto que el voltaje total no supera 4.65 V en ninguna de las secciones, el sistema funcionará correctamente.

Sección 1 (2.59 V < 4.65 V)

Sección 2 (1.81 V < 4.65 V).

El porcentaje de carga se halla dividiendo el voltaje total entre 4.65 V.

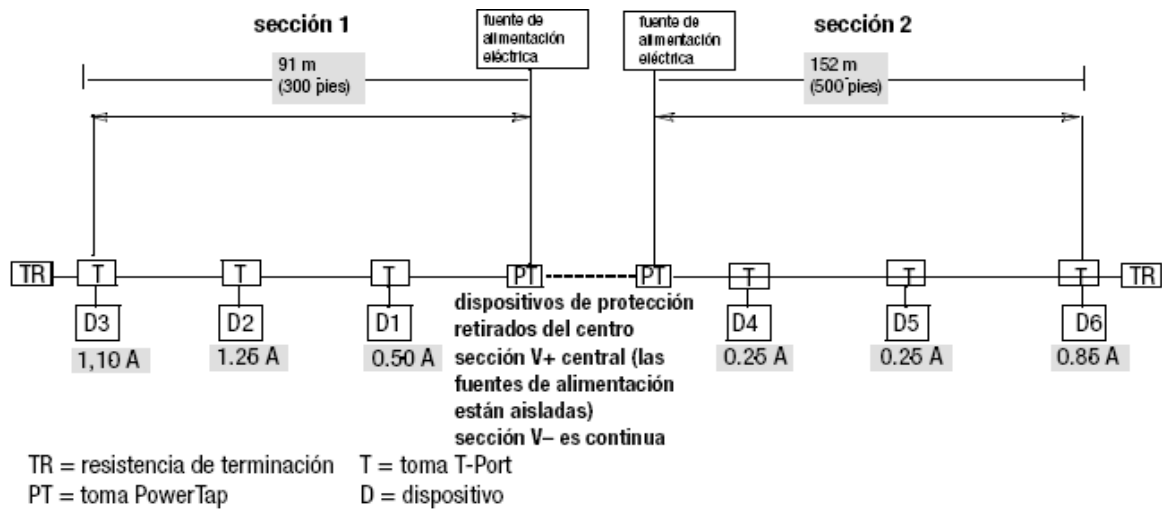
Sección 1 % Carga = $2.59/4.65 = 56 \%$

Sección 2 % Carga = $1.81/4.65 = 39 \%$

Utilización de dos fuentes de alimentación.

Utilizar la siguiente configuración presentada en la figura 47 para sustituir una única fuente de alimentación y obtener una mayor corriente en el sistema.

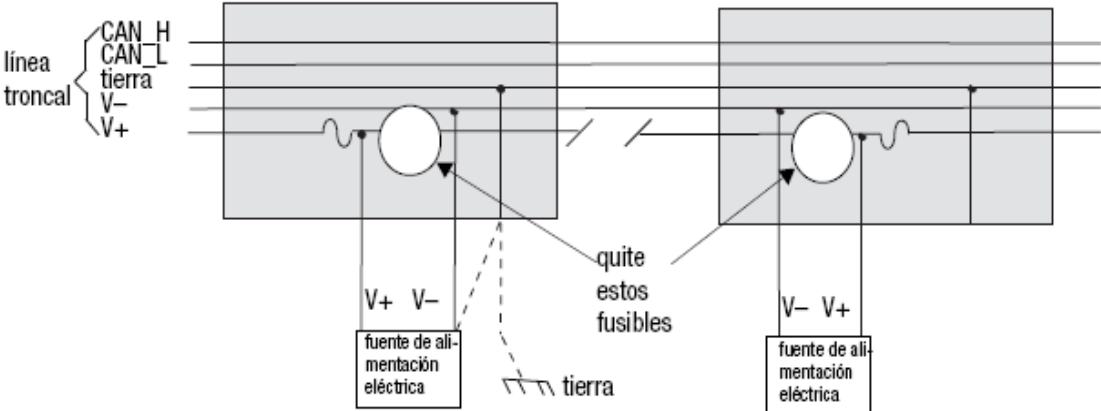
Fig. 47. Dos fuentes de alimentación



Esta configuración duplica eficientemente la corriente disponible a través de la línea troncal. Se debe considerar las siguientes características:

- no se permiten cargas entre las tomas *PowerTap*.
- los fusibles entre las dos tomas *PowerTap* deben quitarse para segmentar el conductor V+ en la línea troncal entre las tomas como indica la figura 48.
- cortar V+ (rojo) al ras con el forro del cable.

Fig. 48. Dos fuentes de alimentación para duplicar la corriente del sistema.



ANEXO C CONFIGURACIÓN DE LA RED DE BUS DE CAMPO DEVICENET

Este anexo trata el tema de configuración de la red DeviceNet usando las herramientas software de Rockwell, RSNetworx y RSLinx, se asume que el usuario está relacionado con estas aplicaciones software.

C.1 SOFTWARE DE CONFIGURACIÓN DE LA RED DEVICENET

La tabla 10 muestra el software y las versiones de software compatibles con el scanner 1769SDN.

Tabla 10. Versiones de software requerido para la configuración de la red DeviceNet.

Función	Programa	Versión
Comunicación	RSLinx	2.30
Configuración DeviceNet	RSNetWorx para DeviceNet	3.00
Software de Programación Ladder Logic	RSLogic 500	5.00

C.2 INTERFACES REQUERIDAS PARA CONFIGURAR LA RED DEVICENET

Se puede usar el modulo 1770-KFD es un adaptador DeviceNet-RS-232, para las tareas iniciales de configuración del sistema, o usar otros dispositivos como: 1784-PCD, 1784-PCID tarjeta de interfaz DeviceNet (interna), etc.

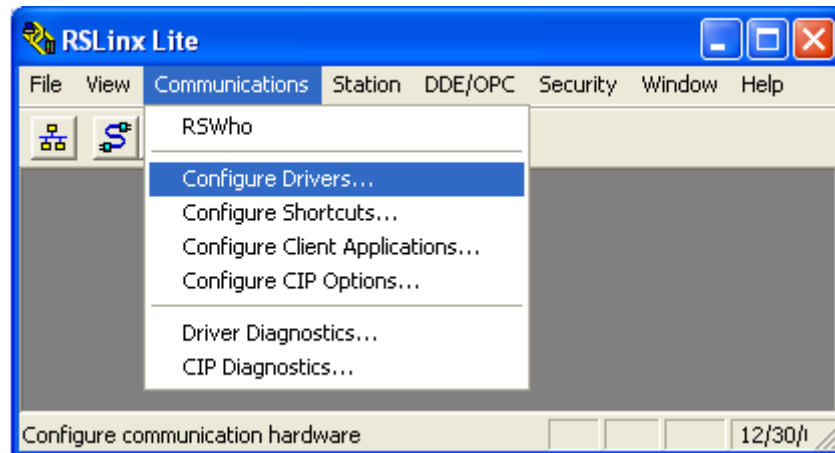
El uso de cualquiera de las interfaces mencionadas anteriormente permite comunicar un computador con la red DeviceNet con fines de configuración y monitoreo. En este trabajo de grado se usa el modulo 1770-KFD.

El computador puede ser conectado en cualquier lugar del cable DeviceNet, y usarse como un nodo permanente o ser retirado y conectado cuando sea pertinente. El modulo puede conectarse a un dispositivo vía cable DeviceNet en una conexión física point to point o conectarse a la red de dispositivos.

C.3 PASOS PARA CONFIGURAR LOS DRIVER DE RSNETWORX

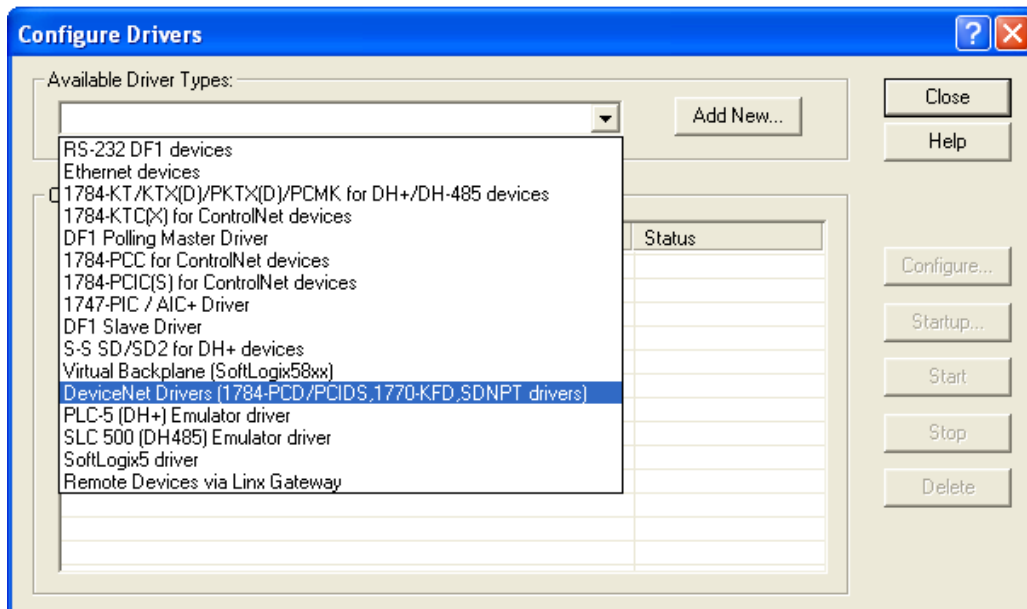
1. Iniciar RSLinx, en el menú comunicaciones dar Click izquierdo, a continuación Click izquierdo en el sub menú “Configure Drivers” como indica la figura 49.

Fig. 49. Ventana inicial de RSLinx



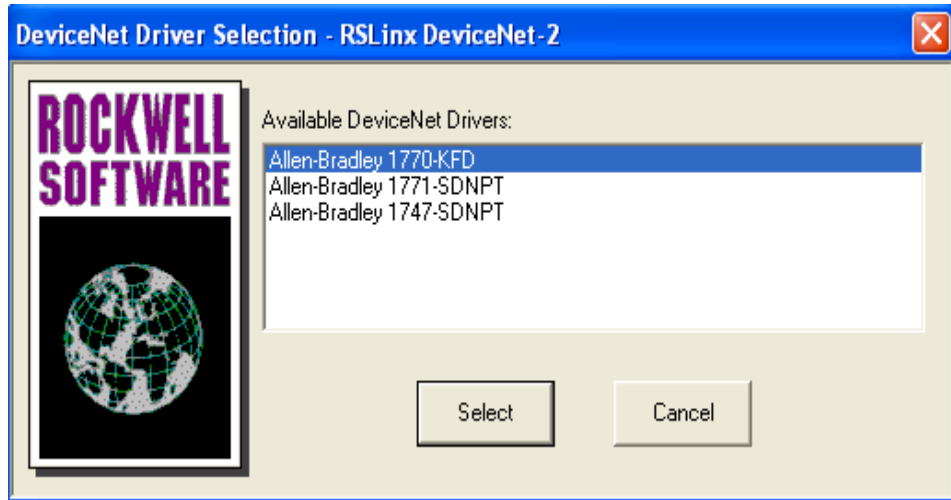
2. Del menú configuración seleccionar “DeviceNet Driver” ver la figura 50.

Fig. 50. Drivers DeviceNet



3. Click en *add new* para adicionar el nuevo driver, aparecerá la ventana de la figura 51

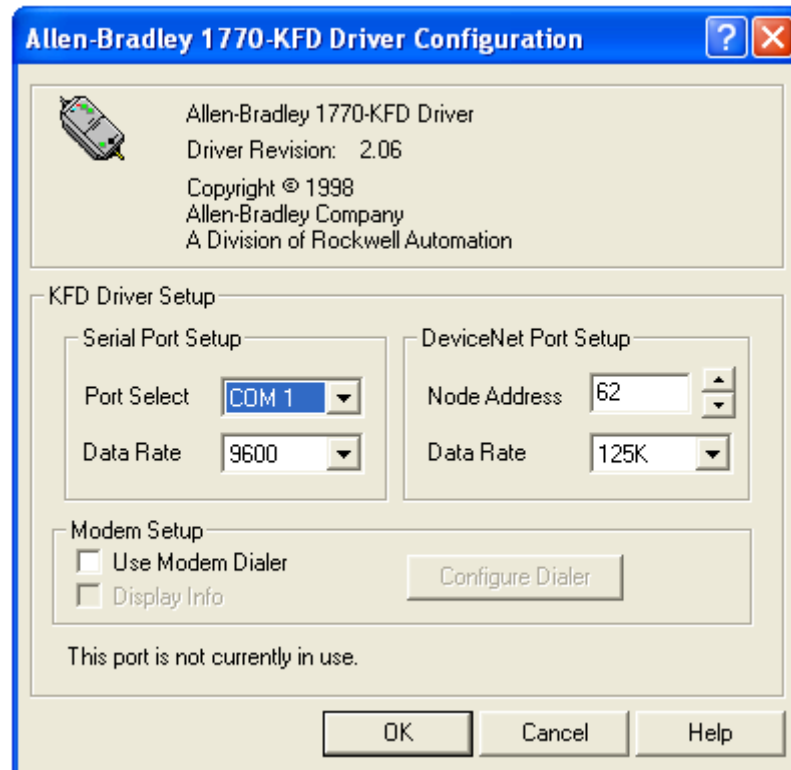
Fig. 51. Agregar el nuevo driver



4. Seleccionar 1770 KFD como indica la figura 51.

La figura 52 representa el menú de configuración del modulo 1770KFD.

Fig. 52. Configuración del modulo 1770KFD



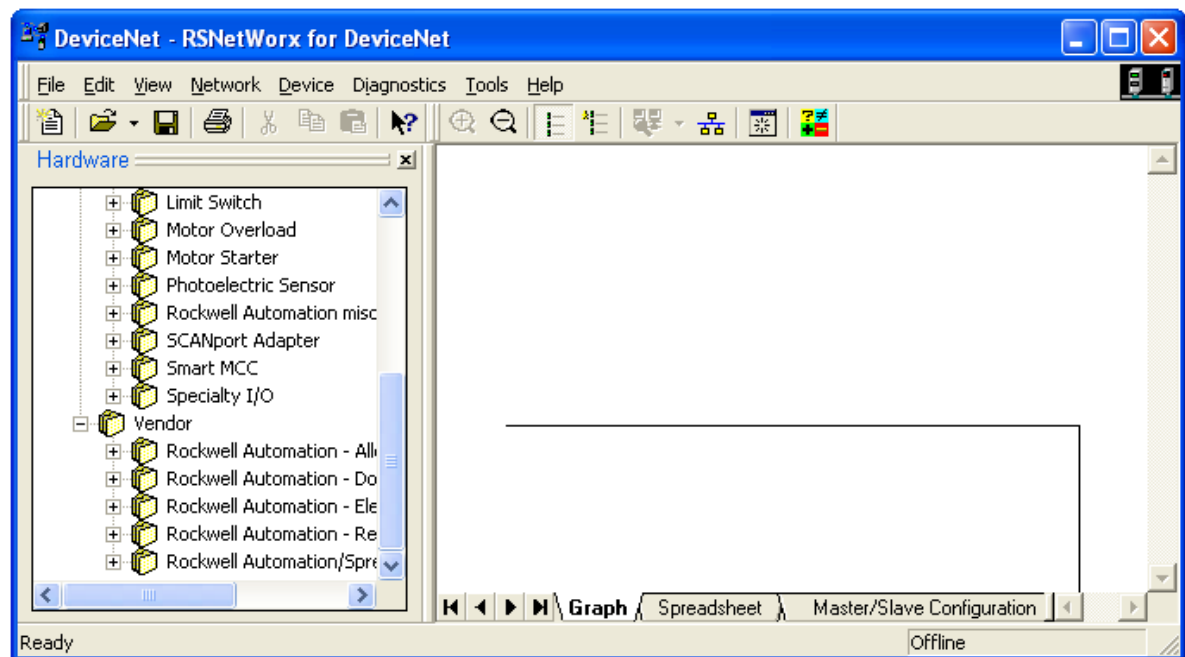
Configurar el puerto serial, tanto la posición del COM y la tasa de transmisión de datos dependiendo de la disponibilidad del PC. En la configuración del puerto DeviceNet se debe tener en cuenta el Baud Rate de la red y la dirección del nodo, es importante configurar una dirección diferente de 63 debido a que los dispositivos inician automáticamente en la posición 63 (configuración de fábrica). Después de configurar el driver minimizar RSLinx y configurar el scanner usando RSNetWorx.

C.4 CONFIGURACIÓN DEL MODULO 1769SDN

El siguiente procedimiento expone los pasos de configuración de la red DeviceNet usando el modulo de comunicaciones 1770-KFD.

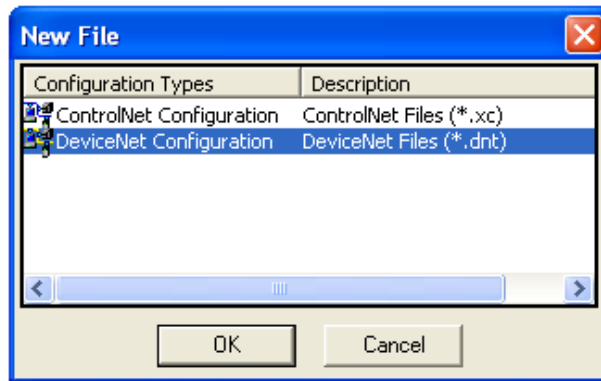
1. Iniciar RSNetWorx. La figura 53 muestra la ventana principal de RSNetworx.

Fig. 53. Ventana principal de RSNetworx



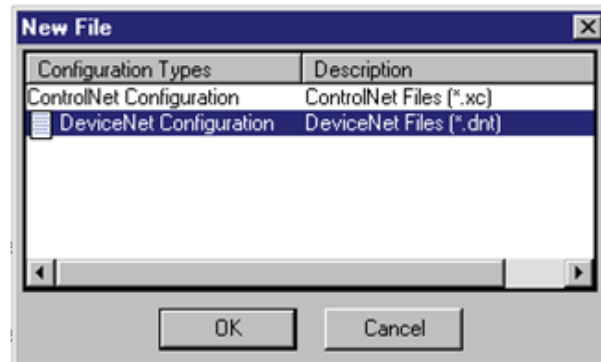
2. En el menú File seleccione New, como indica la figura 54.

Fig. 54. Opciones de selección de archivos del tipo de red.



3. Seleccionar DeviceNet y dar Click en OK.
4. dar Click en el botón online de la barra de herramientas. Entonces aparecerá una lista de las redes disponibles, como indica la figura 55.

Fig. 55. Redes disponibles



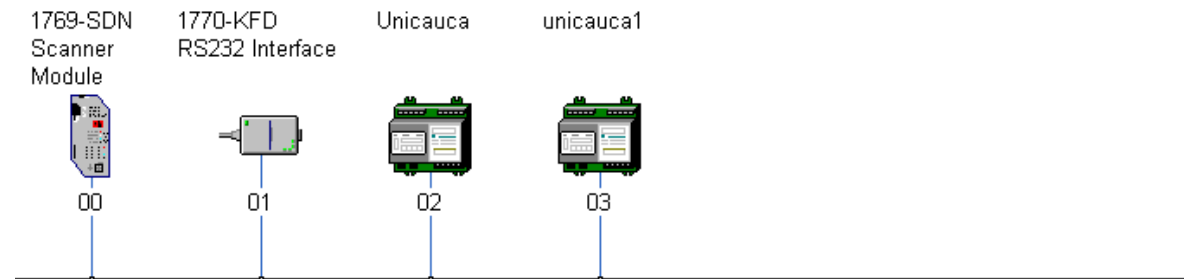
5. Seleccionar el driver DeviceNet y dar Click en OK, entonces aparecerá en pantalla la siguiente ventana mostrada en la figura 56.

Fig. 56. Confirmación de búsqueda de dispositivos



6. Dar click en OK (figura 56) para subir (*Upload*) los dispositivos y pasar a modo online. Durante este proceso RSNetWorx busca los dispositivos disponibles en la red DeviceNet, cuando el software termina la búsqueda, aparece en pantalla los dispositivos en línea del sistema, ver la figura 57.

Fig. 57. Dispositivos disponibles en la red DeviceNet



Si un nodo en línea pasa a estado fuera de línea no se mostrara en la red DeviceNet después del proceso de *scan* (exploración), se puede buscar manualmente el nodo perdido usando la herramienta Browse Network de DeviceNet. Para ejecutar búsqueda manual, presionar el botón browse Network en la barra de herramientas, si la búsqueda falla entonces se debe verificar la conexión física del dispositivo, también de debe verificar si el dispositivo tiene el mismo Baud Rate de la red.

El software de configuración de DeviceNet debe incluir los archivos EDS requeridos para cada dispositivo, para obtenerlos se puede remitir a la página <http://www.ab.com/networks/eds> para descargar los archivos. Posterior a la descarga se deben registrar los archivos, usando EDS wizard en RSNetWorx. Los archivos EDS del nodo DeviceNet Unicauca están ubicados en el anexo digital I...ver la sección 4.9.10...

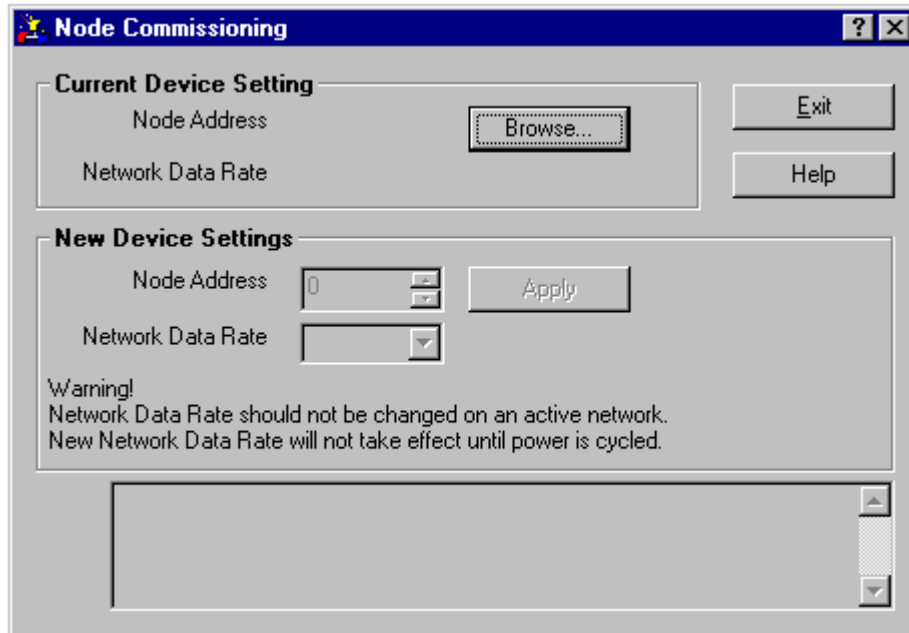
C.5 DIRECCIONAMIENTO DE LOS NODOS DEVICENET

Una vez que finaliza el escaneo de dispositivos, se puede cambiar la dirección de los nodos, si es pertinente, para evitar duplicaciones de MAC ID en la red, también es posible configurar el Baud Rate del dispositivo de acuerdo con la velocidad de la red.

El controlador debe estar en modo Program o el scanner en modo Idle para que el scanner acepte la nueva configuración. La figura 58 muestra la ventana de configuración.

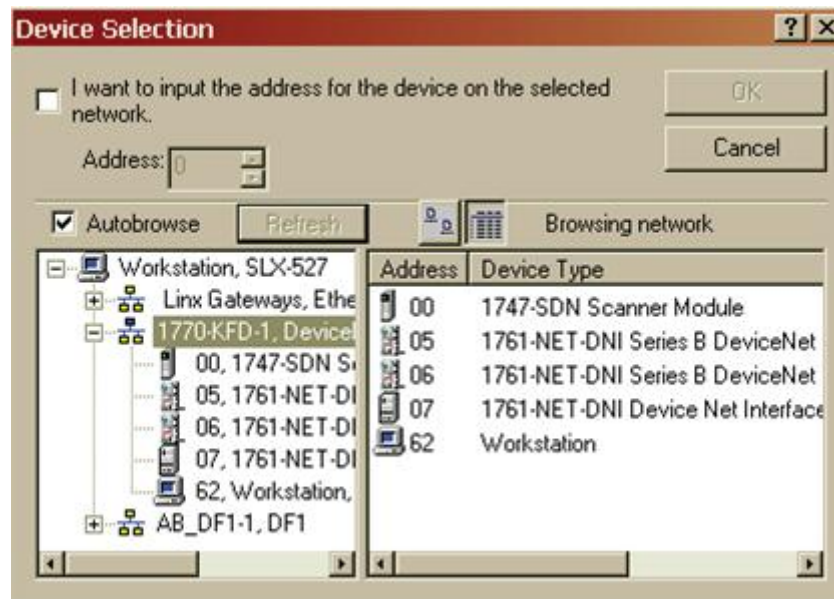
1. En el menú Tools seleccionar Node Commissioning.

Fig. 58. Comisionar dispositivos DeviceNet



2. Dar clic en el botón browse, como indica la figura 58, entonces aparecerá la ventana de selección de dispositivos. Ver la figura 59.

Fig. 59. Selección de dispositivos DeviceNet

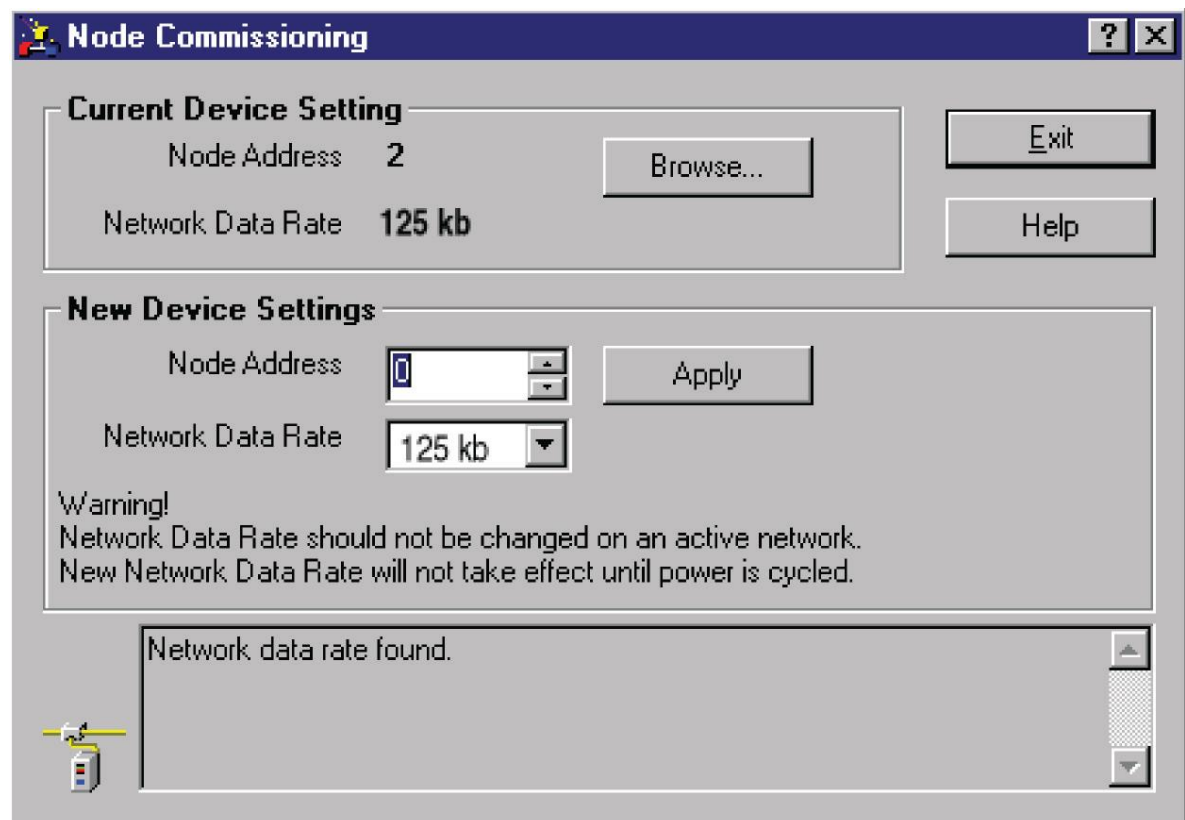


3. Seleccionar DeviceNet Network. Los dispositivos de la red DeviceNet aparecerán en el panel derecho de la ventana.
4. Seleccionar el dispositivo en el panel derecho y dar Click en ok.

Entonces aparece la siguiente ventana mostrada en la figura 60, donde se puede observar los valores por defecto del dispositivo. Por ejemplo los valores por defecto del scanner son: Baud Rate = 125K, Dirección del nodo = 63.

Se recomienda cambiar la dirección del nodo (diferente de 63) para evitar duplicaciones de MAC ID.

Fig. 60. Valores por defecto del scanner Devicenet



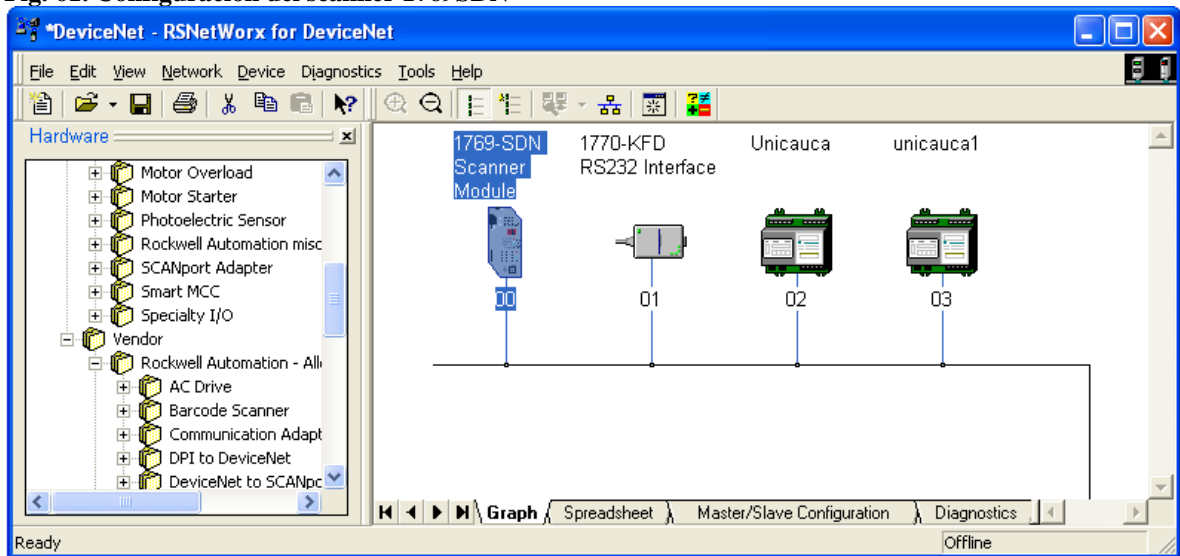
5. En la sección New Device Setting de la ventana, modificar *Node address* y *Network Data Rate* según la conveniencia teniendo en cuenta los parámetros de la red.
6. Dar clic en el botón apply y cerrar la ventana.

C.6 CONFIGURACIÓN DE DISPOSITIVOS DEVICENET

Se puede configurar y cambiar los parámetros de cada dispositivo de la red DeviceNet, en este caso a modo de ejemplo se menciona únicamente la configuración del scanner.

La siguiente ventana (figura 61) muestra una red DeviceNet en RSNetWorx con los diferentes dispositivos en línea. Para ver y modificar los parámetros dar doble *click* en el dispositivo pertinente, en este caso se eligió el scanner 1769SDN.

Fig. 61. Configuración del scanner 1769SDN

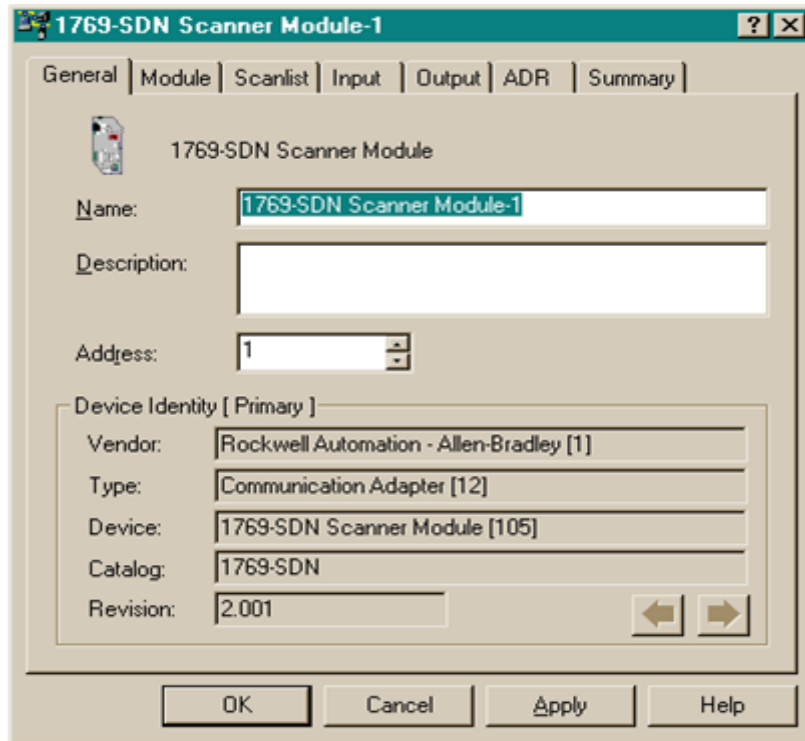


Entonces aparecerá la siguiente ventana (figura 62) donde se muestran los diferentes parámetros configurables del dispositivo. (*General, Module, Scanlist, Input, Output, ADR y Summary*). A continuación se describe cada uno de estos parámetros:

C.6.1 General (menú)

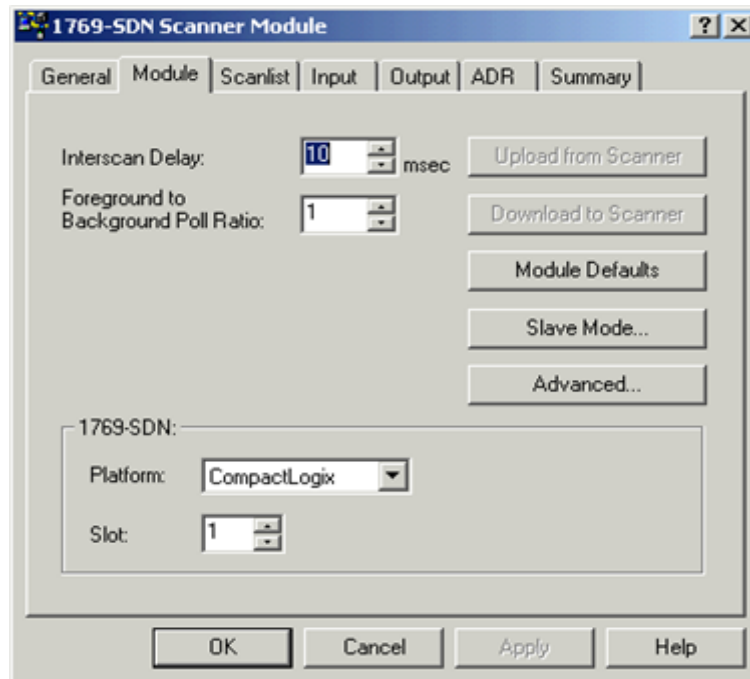
En la siguiente ventana de RSNetWorx mostrada en la figura 62 se presenta las Características generales del scanner 1769SDN como: fabricante, nombre, tipo de dispositivo, etc.

Fig. 62. Características generales del scanner en RsNetworkx



C.6.2 Module (menú)

Fig. 63. Module configuración



La ventana Modulo de la figura 63 abarca los siguientes parámetros: Interscan Delay, Foreground to Background Poll Ratio, Module Defaults, Slave Mode y Advanced.

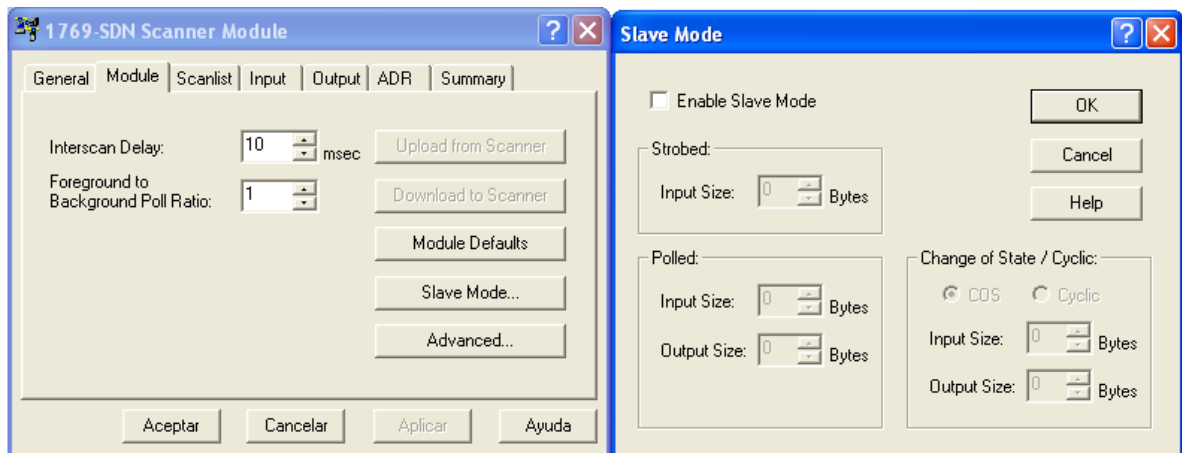
- **Interscan Delay** Este parámetro define el retardo de tiempo que el scanner usa entre cada ciclo de scan de la red DeviceNet.

Si en la red posee dispositivos configurados con la característica Poll, en el scanner 1769SDN el scanlist, Interscan Delay (ISD) define el tiempo que tarda el scanner en actualizar los dispositivos en un nuevo ciclo de scan.

Si se incrementa excesivamente el ISD causa una saturación de datos de entrada y salida de la red, sin embargo si se aumenta el ISD disminuye la prioridad del mensaje. Si el tiempo de respuesta de los dispositivos de la red es grande entonces se puede aumentar el ISD.

- **Slave Mode** el scanner puede usarse también como esclavo, pero en este caso esta opción debe estar deshabilitada porque el scanner actúa como nodo maestro de la red. La figura 64 indica que el scanner es un dispositivo maestro en la red porque está en *enable slave mode* (modo esclavo desactivado).

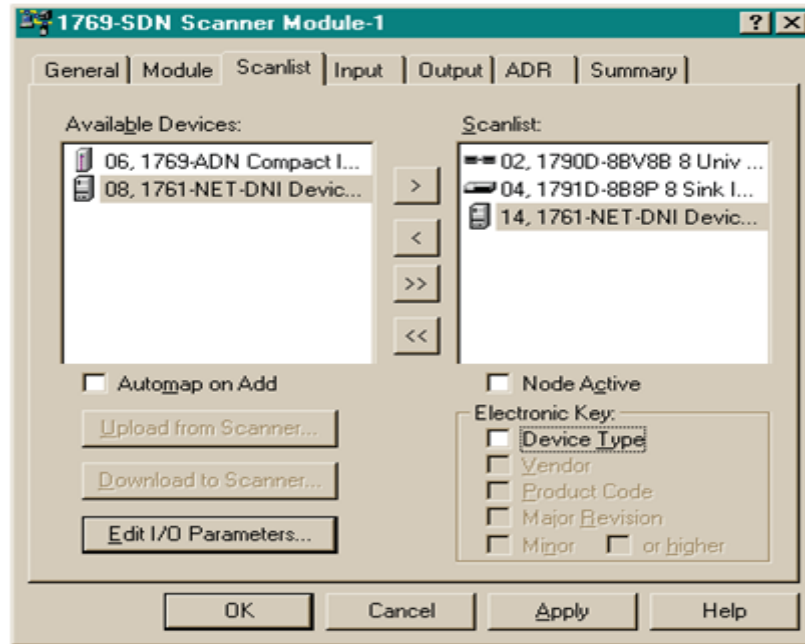
Fig. 64. El escáner como dispositivo esclavo



C.6.3 Scanlist (menú)

Scanlist define los dispositivos disponibles en la red, que están en la lista de dispositivos esclavos adoptados por el scanner. la figura 65 muestra los dispositivos en *Scanlist*

Fig. 65. Dispositivos en scanlist



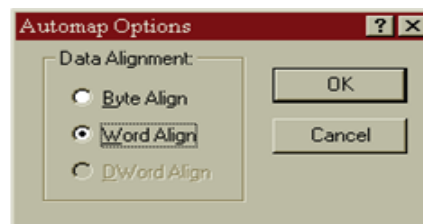
C.6.4 Available Devices (menú)

Son los dispositivos esclavos disponibles en la red, que podrían ser adoptados por el scanner según la conveniencia. Los dispositivos esclavos ubicados en la red que no se incluyan en el *Scanlist*, podrían ser usados por otro scanner de la misma red o podrían tener doble funcionalidad (Maestro o Esclavo).

C.6.5 AutoMap (menú)

Permite mapear automáticamente dentro del scanner la tabla de imagen de entradas y salidas de los dispositivos, cuando son adicionados al scanlist. No seleccionar esta opción si se desea mapear un dispositivo esclavo en una localización particular de memoria. La figura 66 muestra la ventana de AutoMap.

Fig. 66. AutoMap



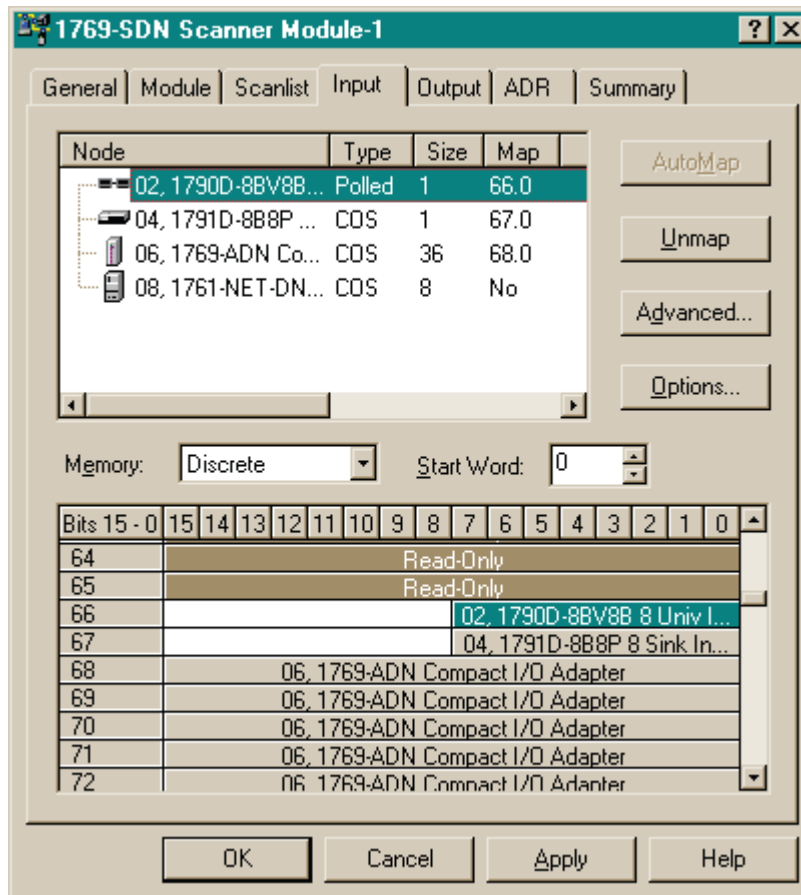
C.6.6 Edit I/O Parameters (menú)

Estos parámetros varían dependiendo de la información del dispositivo esclavo, en este caso se debe recurrir a la documentación del fabricante de cada dispositivo.

C.6.7 Input (menú)

Input permite definir como serán mapeados los datos de los dispositivos del *Scanlist* en las imágenes de entrada del Controlador. La figura 67 indica la ventana Input.

Fig. 67. Datos de los dispositivos mapeados en las imágenes de entrada del controlador



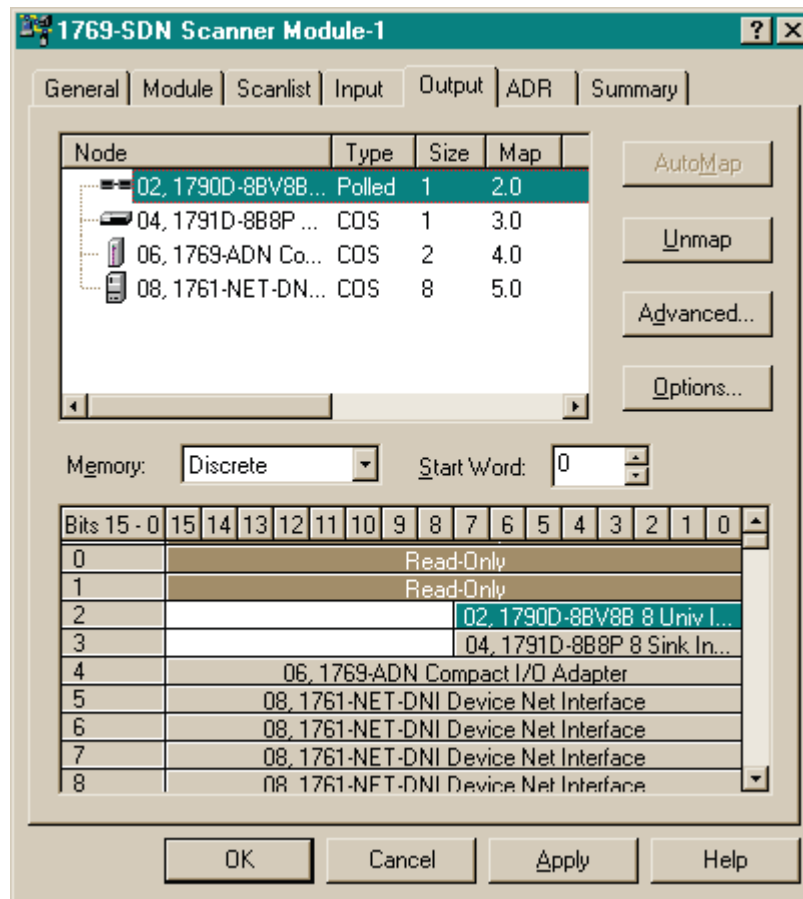
En este ejemplo se tienen cuatro dispositivos que son mapeados dentro de las imágenes de entrada del scanner. Los datos que se muestran de cada dispositivo en este ejemplo son: (ver la figura 67)

- **Node number.** (Numero de nodo).
- **Catalog number.** (Numero de catalogo).
- **Typo.** Conexión usada entre el scanner y el dispositivo esclavo (strobed, polled, cyclic or change of state).
- **Size.** La cantidad de datos que son intercambiados (en bytes).
- **Map.** palabras de imágenes de entrada dentro de los datos del controlador.

C.6.8 Output (menú)

Output permite definir como serán mapeados los datos de los dispositivos del scanlist en las imágenes de salida del controlador. La figura 68 lista los dispositivos que son mapeados dentro de las imágenes de salida del scanner en este ejemplo.

Fig. 68 datos de los dispositivos en las imágenes de salida del controlador



En este ejemplo se tienen cuatro dispositivos que son mapeados dentro de las imágenes de salida del scanner. Los datos que se muestran de cada dispositivo son:

- **Node number.** (numero de nodo).
- **Catalog number.** (numero de catalogo).
- **Typo.** Tipo de conexión usada entre el scanner y el dispositivo esclavo (strobed, polled, cyclic or change of state).
- **Size.** La cantidad de datos que son intercambiados (en bytes).
- **Map.** palabras de imágenes de salida dentro de los datos del controlador.

C.6.9 Auto Device Replacement (ADR)

Auto Device Replacement (reemplazo automático de dispositivo) es el reemplazo automático de un dispositivo esclavo tras una falla del dispositivo en la red DeviceNet, consiste en retornar o llevar el dispositivo a un estado anterior de operación antes de que se presentara la falla. Esta función incluye recuperación de la configuración *Configuration Recovery* (CR) y auto recuperación de la dirección del dispositivo *Auto-Address Recovery* (AAR).

CR permite que un dispositivo esclavo pueda ser removido de la red y reemplazado con un dispositivo idéntico configurado con el mismo Baud Rate y dirección, del dispositivo removido de la red.

AAR permite quitar un dispositivo esclavo de la red y ser reemplazado con otro dispositivo idéntico residente en la red con una dirección de nodo 63 y que no esté en el Scanlist, el dispositivo cambiara automáticamente la dirección de nodo a la del nodo removido.

C.6.10 Consideraciones importantes con ADR

Hay que tener en cuenta las siguientes características cuando se use ADR con RSNetWorx para DeviceNet:

- El AAR únicamente se puede habilitar si el CR es habilitado

- La característica CR no se puede habilitar en dispositivos que no tenga parámetro re escribibles, como los dispositivos que se configuran con switches.

C.6.11 Summary (menú)

Summary suministra información de la configuración del scanner, nodos mapeados, nodos activos y si están recibiendo y transmitiendo información. Como indica la figura 69.

Fig. 69.resumen del estado del sistema

Node	Active	Key	Input	Mapped	Output	Mapped
01, <Slave...	No		0	No	0	No
02, 1790D...	Yes	DV...	1	Yes	1	Yes
04, 1791D...	Yes	DV...	1	Yes	1	Yes
06, 1769-A...	Yes	DV...	36	Yes	2	Yes
08, 1761-...	Yes	DV...	8	Yes	8	Yes

C.6.12 Descargar y guardar la configuración al scanner

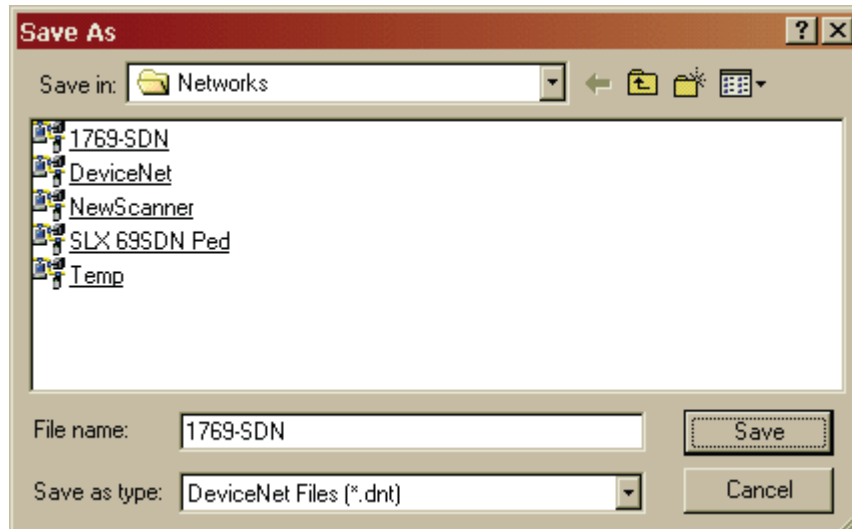
1. dar Click en la opción Scanlist, dar Click en el botón *Download to Scanner* (descargar al escáner), entonces se abre la siguiente ventana. Mostrada en a figura 70.

Fig. 70. Como descargar y guardar la configuración

2. seleccionar All Records.

3. Click en el botón *Download* para descargar la configuración al scanner.
4. Click en OK para completar la configuración del scanner DeviceNet.
5. desde el menú archivo seleccione Save As...Ver la figura 71...

Fig. 71. Ventana que indica donde y el tipo de archivo a guardar de RSNetWorx



6. guardar la configuración como archivo DeviceNet.
7. finalmente si la configuración es satisfactoria se puede cerrar RSNetWorx

ANEXO D CONSIDERACIONES IMPORTANTES DEL NODO DEVICENET

D.1. CONSIDERACIONES CON RESPECTO A LOS REGISTROS I/O DEL NODO

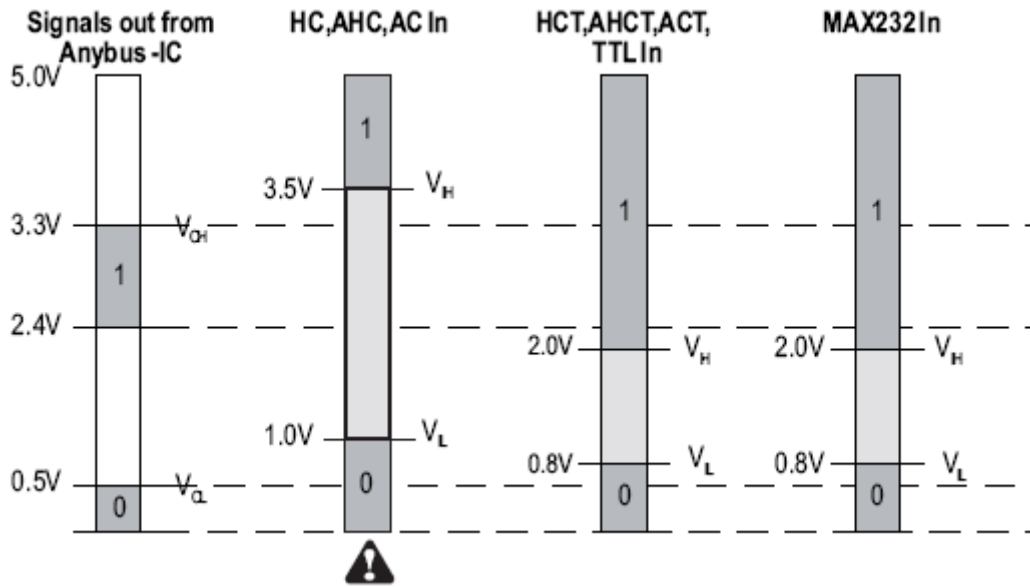
Con respecto a las entradas y salidas, el nodo usa el canal SSC, y los registros empleados son 74HCT165 Y 74HCT594 respectivamente. Para diseños e implementaciones posteriores se debe tener en cuenta que se pueden usar otros registros disponibles, teniendo en cuenta las características de los dispositivos entregadas por el fabricante. A continuación se presentan diferentes opciones para usar otros registros.

Teniendo en cuenta los niveles de señal se recomienda usar: 74HCTxxx / 74ACTxxx. El fabricante del controlador AB6001 no recomienda utilizar 74HCxxx, 74ACxxx, y 74AHCxxx para las entradas debido a que podrían no trabajar adecuadamente puesto que el área indeterminada de estos registros va desde 1voltio a 3.5voltios y el área de señales del AnyBus IC o AB6001 para el nivel alto va desde 2.4 a 3.3 voltios, se puede ver claramente que el uno lógico del controlador máster no coincide con el uno lógico del registro de entradas... ver la figura 72...

Con los registros de salida no pasa esto, pero hay que tener en cuenta la capacidad de corriente de salida de los registros ej. En el caso de que se desee conectar directamente diodos LED a las salidas del registro, los 74ACTxxx tienen suficiente corriente para esto. En cada caso es muy importante consultar los datasheet de cada dispositivo a usar para saber si cumple con las características deseadas por el diseñador.

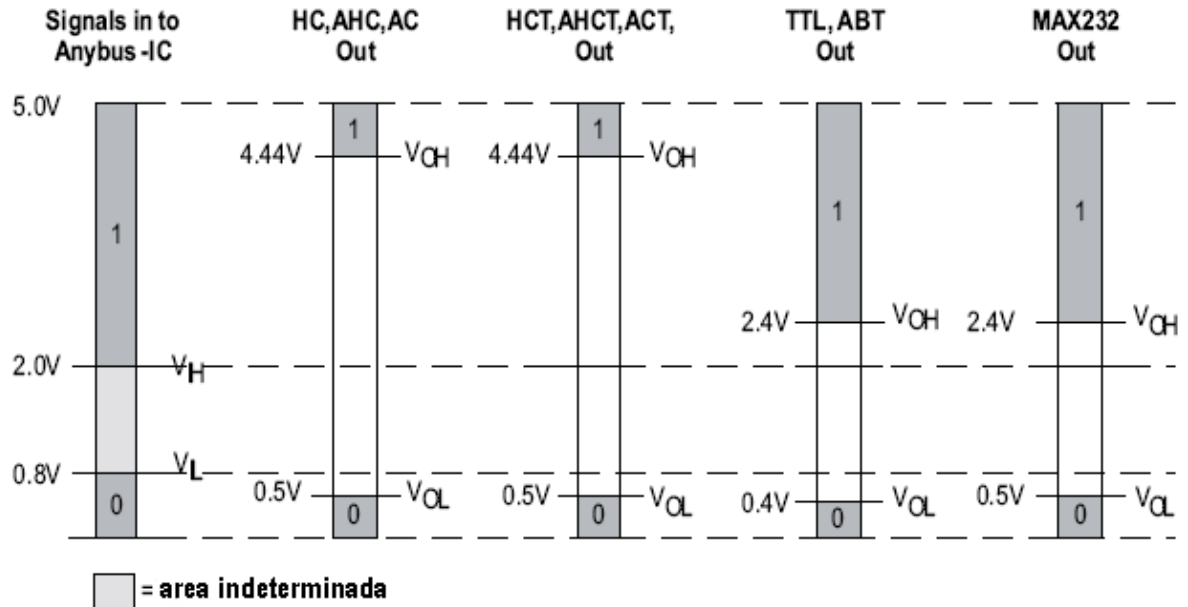
La figura 73 indica los niveles de voltaje de los registros de salida recomendados para trabajar con el AB600.1

Fig. 72. Niveles de voltaje de salida del controlador versus voltaje de entrada de los registros.



⚠️ Uso no recomendado en la implementación de entradas usando el canal SSC.

Fig. 73. Niveles de voltaje de entrada del controlador versus voltaje de salida de los registros.



D.2. CARACTERÍSTICAS DE LAS FAMILIAS LÓGICAS RECOMENDADAS.

74HC/HCT (CMOS de alta velocidad) se trata de la versión mejorada de la serie 74C, que tiene un incremento de 10 veces más la velocidad de conmutación, comparada con la velocidad de conmutación de los 74LS y una capacidad de corriente de salida mucho mayor que la del 74C (Tocci).

Las series 74AC/ACT (CMOS avanzado) esta es una de la serie más nueva de los CMOS es funcionalmente equivalente con la serie TTL pero los terminales no son compatibles la razón es que los terminales en la serie 74AC/74ACT se han seleccionado para mejorar la inmunidad al ruido, con lo cual la entrada a los dispositivos son menos sensibles a los cambios en las señales que ocurren en terminales de otros dispositivos los 74AC no son eléctricamente compatibles con TTL, los 74ACT se pueden conectar directamente con TTL (Tocci).

74HC/HCT y las 74AC/ACT funcionan dentro de un intervalo más bajo de voltaje que las demás series CMOS, que va desde los 3 a 15 Voltios, en la serie 74HC/HCT y las 74AC/ACT va desde 2 a 6 voltios. En el caso de que se use CMOS y TTL se debe usar una fuente de alimentación de 5v de modo que pueda alimentar tanto a V_{DD} para CMOS, como a V_{CC} para TTL (Tocci).

En la tabla 12 se pueden observar los niveles de voltaje de entrada/salida en voltios con $V_{DD} = V_{CC} = 5v$.

Calculo de los márgenes de ruido. Las márgenes de ruido para cada serie también se enumeran en la tabla 11 y se calculan con la ecuación D.1) y D.2

$$V_{NH} = V_{OH(MIN)} - V_{IH(MIN)} \quad (D.1)$$

$$V_{NL} = V_{IL(MAX)} - V_{OL(MAX)} \quad (D.2)$$

Tabla 11. Niveles de voltaje E/S de los registros.

parámetro	CMOS					TTL			
	4000B	74HC	74HCT	74AC	74ACT	74	74LS	74AS	74ALS
$V_{IH}(\text{min})$	3.5	3.5	2.0	3.5	2.0	2.0	2.0	2.0	2.0
$V_{IL}(\text{máx})$	1.5	1.0	0.8	1.5	0.8	0.8	0.8	0.8	0.8
$V_{OH}(\text{min})$	4.95	4.9	4.9	4.9	4.9	2.4	2.7	2.7	2.7
$V_{OL}(\text{máx})$	0.05	0.1	0.1	0.1	0.1	0.4	0.5	0.5	0.4
V_{NH}	1.45	1.4	2.9	1.4	2.9	0.4	0.4	0.4	0.4
V_{NL}	1.45	0.9	0.7	1.4	0.7	0.4	0.3	0.3	0.4

Tomado de: (Tocci)

Corrientes de entrada/salida para dispositivos estándar con voltaje de alimentación De 5V

Tabla 12. Corrientes de E/S.

parámetro	CMOS			TTL			
	4000B	74HC/74HCT	74AC/74ACT	74	74LS	74AS	74ALS
$I_{IH}(\text{min})$	1 μ A	1 μ A	1 μ A	40 μ A	20 μ A	200 μ A	20 μ A
$I_{IL}(\text{máx})$	1 μ A	1 μ A	1 μ A	1.6mA	0.4mA	2mA	100 μ A
$I_{OH}(\text{máx})$	0.4mA	4mA	24mA	0.4mA	0.4mA	2mA	400 μ A
$I_{OL}(\text{máx})$	0.4mA	4mA	24mA	16mA	8mA	20mA	8mA

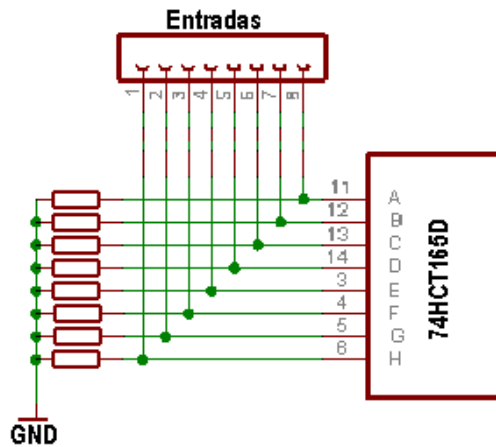
Tomado de: (Tocci)

D.3. DISPOSITIVOS USADOS COMO ENTRADA AL NODO

El área de entradas del nodo está diseñada para señales discretas de 5V. Hay que tener especial cuidado con el tipo de sensor usado ya que voltajes mayores a 5V pueden provocar daños en los registros de entrada o un mal funcionamiento del dispositivo.

En este caso se usa un sensor de 12V, de alimentación y salida de 12V, teniendo en cuenta las características de la tarjeta principal (*main board*) del nodo DeviceNet Unicauca, no se puede incorporar directamente los sensores al nodo DeviceNet, por este motivo se diseñó un circuito adicional para adecuar la señal de salida de los sensores y para su alimentación, ver la figura 74 que describe el área de entradas del nodo DeviceNet desarrollado en este proyecto.

Fig. 74. área de entradas del nodo DeviceNet



D.3.1. Circuito de entradas al nodo DeviceNet

La tarjeta de entradas se implementa con resistencias en serie formando un divisor de tensión que divida el voltaje de entrada de 12V provenientes del sensor y entregue 5V a las entradas del registro. La figura 75 muestra la *board* de entradas acoplada al área de entradas del nodo DeviceNet, se puede observar el divisor de tensión formado por la resistencia Rx y Ra en la figura 75.

Fig. 75. Nodo DeviceNet y tarjeta de entradas.

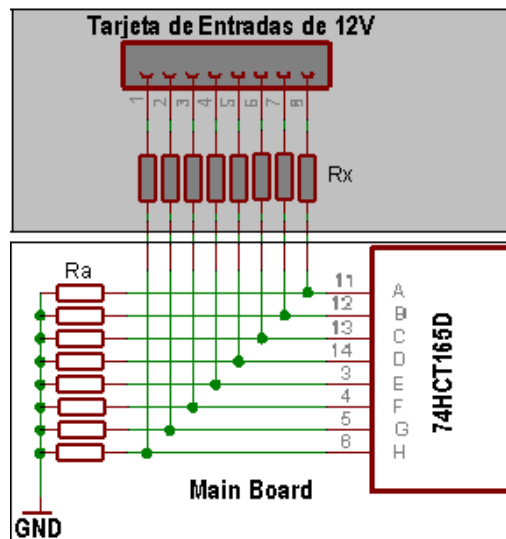
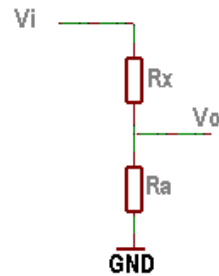


Fig. 76. Divisor de tención formado por las resistencias de entradas (Rx) y las resistencias de la Main board (Ra)



Con $V_i=12V$ (voltaje de salida del sensor), $V_o=5V$ (voltaje de entrada a los registros), y R_a de la mainboard de $10K\Omega$ el valor de R_x se obtiene de la ecuación D.3)

$$V_o = \frac{R_a}{R_a + R_x} V_i \quad (D.3)$$

Entonces reemplazando V_o , V_i y R_a se obtiene que $R_x=14k\Omega$.

La board de entradas tiene un conector de salida de 10 pines que se acopla directamente a la tarjeta principal del nodo, un conector de tornillos de 24 pines para las salidas, GND y el voltaje de alimentación de los sensores. Además de la fuente de alimentación de los sensores que se toma de la *main board* y corresponde a un voltaje *mis wiring*.

El conector OUTPUT transporta las 8 señales de los sensores por los pines 3 a 10 al circuito principal (conector Input pines 2 a 9). El pin 1 del conector input es una salida opcional de 5V para alimentación de los sensores, no usada en esta aplicación debido a que los sensores tienen fuente de 12V incluida en la *board* de entradas.

En el circuito las señales Out1 a Out8 corresponde a las salidas de los sensores, la alimentación y la tierra están también disponibles en el mismo conector, ver la figura 77 donde el pin1, 2 y 3 son: salida del sensor, GND y VCC respectivamente.

Fig. 77. Circuito de entradas de señal de 12V

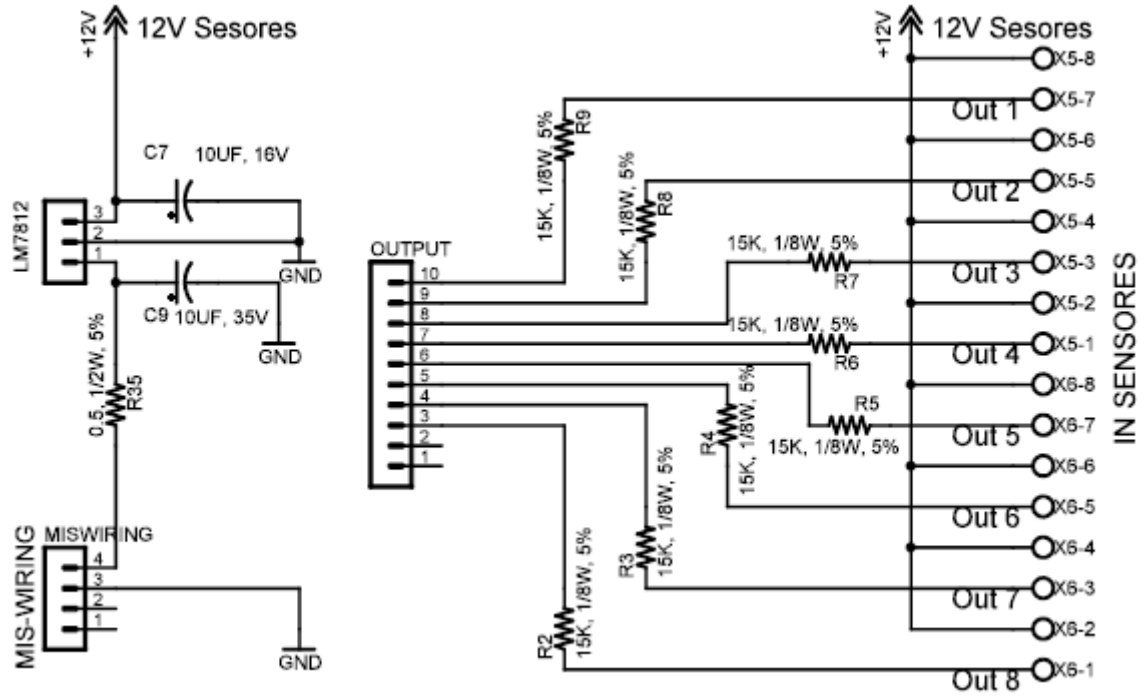
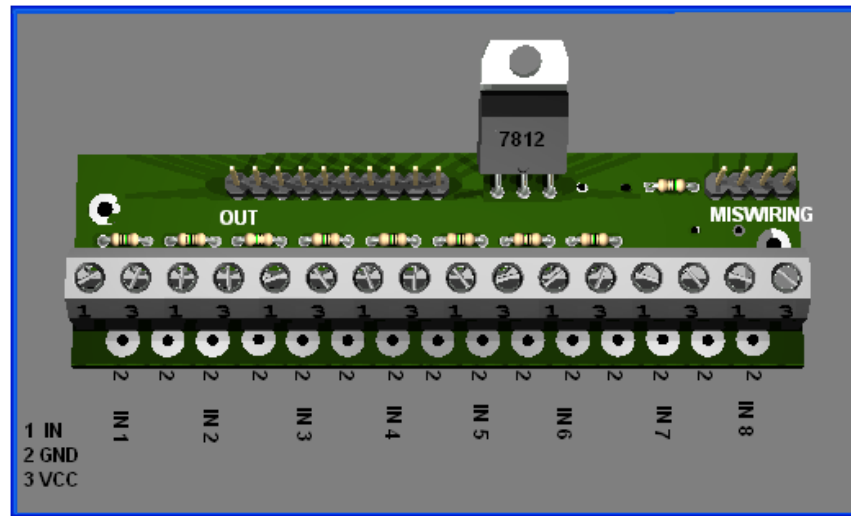


Fig. 78. Boar de entrada de sensores



D.3.2. Sensor de proximidad usado para acoplar a la tarjeta de entradas.

La figura 79 muestra el sensor DU50KP2 usado como entrada al nodo DeviceNet.

Fig. 79. Sensor Riko DU50KP2



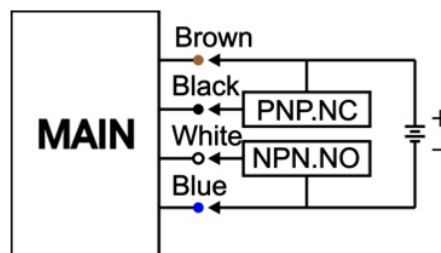
Características del sensor.

- Voltaje de operación DC 12V
- Consumo de corriente 30mA Máximo
- Corriente de salida con alimentación de 24V DC, 100mA Máximo
- Distancia de censado Máximo 50Cm

Conexiones.

Las salidas del sensor se conectan directamente a la tarjeta de entrada del nodo, el terminal o cable café se conecta Vcc, el cable azul se conecta a Gnd y el cable blanco es la señal de salida del sensor que se conecta al terminal In (pin1) en la *board* de entradas. La figura 80 describe los terminales del sensor DU50KP2

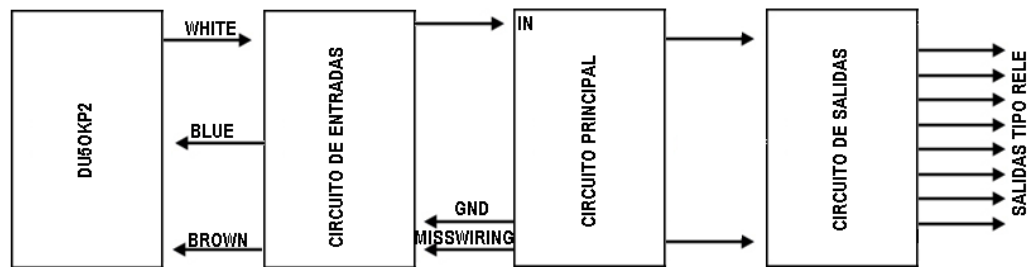
Fig. 80 opciones de conexión del sensor DU50KP2



El sensor Riko DU50KP2 tiene dos opciones de conexión, normalmente abierta y normalmente cerrada como muestra la figura 80. Con el terminal blanco se obtiene una salida normalmente abierta, mientras que usando el terminal negro la salida es normalmente cerrada.

La figura 81 muestra la forma de conectar el sensor DU50KP2 al nodo DeviceNet.

Fig. 81. Diagrama de bloques del nodo, tarjetas de entradas, salidas y el sensor de proximidad Riko



D.3.3. Sensor de proximidad Sharp P2Y0D02YK

Fig. 82. Sensor infrarrojo Sharp IR Sensor



Características del sensor P2Y0D02YK mostrado en la figura 82

- *Trigger point set at factory to 80°.*
- Salida Digital.
- Rango de 20 a 150 cm.
- Tiempo de respuesta 39 ms.
- Consumo promedio de corriente 33 mA
- Voltaje de operación de 4.5 a 5.5 Vcc.

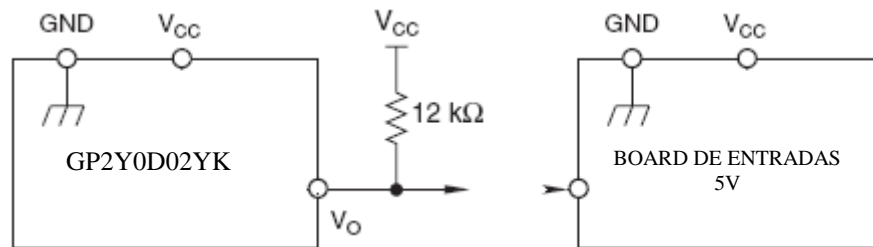
La tabla 13 describe la distribución de pines del sensor P2Y0D02YK

Tabla 13. Descripción de pines del sensor sharp

Pin	Señal
1	V _o
2	GND
3	V _{cc}

La figura 83 muestra la forma de acoplar un sensor P2Y0D02YK a la *board* de entradas del nodo DeviceNet Unicauca.

Fig. 83. Aplicación típica del sensor sharp



Los sensores Sharp se pueden incorporar directamente a las *board* de entradas de 5V, solo requiere que se adicione una resistencia de 12KΩ entre V_o y V_{cc} del sensor Sharp.

D.2.3. Circuito de salidas del nodo DeviceNet Unicauca

El circuito de salidas permite adicionar actuadores al dispositivo, debido a que la corriente de los registros es limitada, no se pueden poner cargas directamente a la salida de los registros, se debe de usar relevos magnéticos, o usar relevos de estado solido que permitan manejar la corriente del actuador a incorporar al nodo, si se usa relevos de estado solido con alimentación externa como 120V o una fuente diferente a la de la red se debe usar un circuito *isolator* (separador) para evitar cortocircuitos y daños permanentes a los dispositivos.

En este diseño las salidas de relevo permiten conectar actuadores al nodo DeviceNet con una corriente máxima de 3A, no se puede sobrepasar los 3A en cada salida de relevo. Los relevos funcionan a 12 voltios, por lo que se dispone de una fuente de alimentación en la tarjeta de salidas tomada del circuito mis wiring de la tarjeta principal. La figura 84 representa el diagrapha esquemático del circuito de salidas del nodo DeviceNet.

Fig. 84 diagrama esquemático de la tarjeta de salidas de relevo

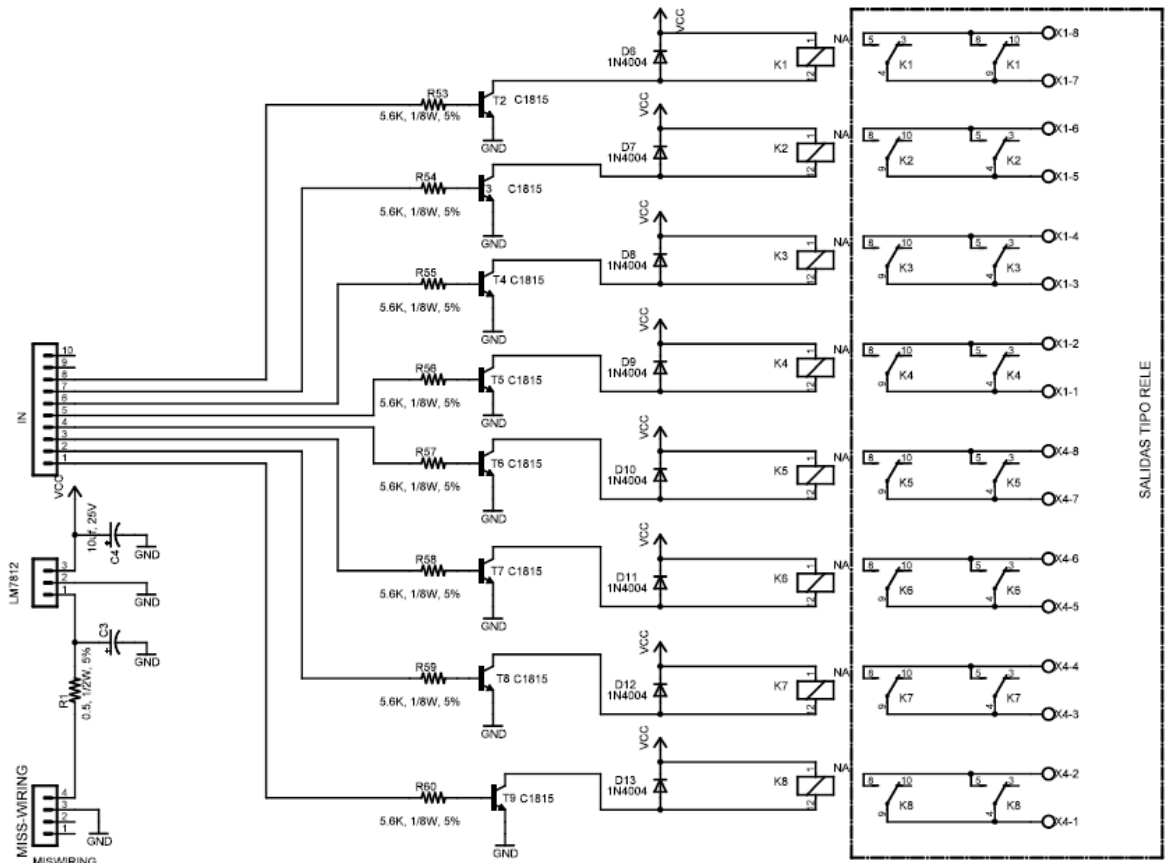
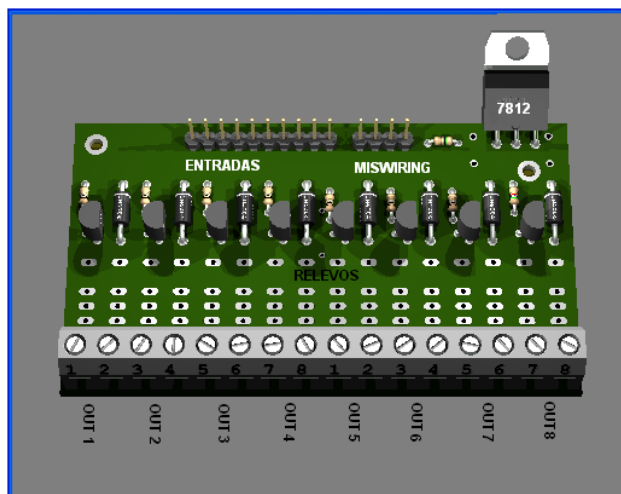


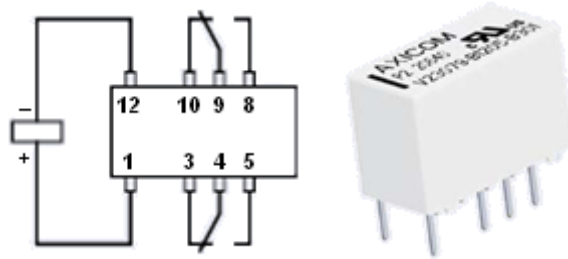
Fig. 85. Board de salidas de relevo



En la figura 85 se observa la ausencia de algunos componentes como los relevos y los condensadores, esta situación se presenta porque la imagen es un archivo generado por

POV-Ray v3.6 un programa que a través de Eagle permite obtener la PCB en tercera dimensión del circuito esquemático. Un relevo es un *switch* magnético que permite conmutar cargas de alta corriente, la figura 86 representa los relevos usados en este proyecto.

Fig. 86 Relevos usados en el circuito de salidas.



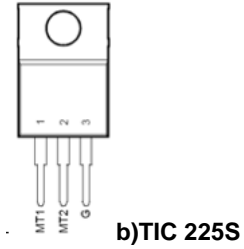
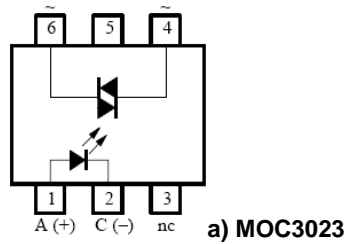
La figura 87 muestra la tarjeta principal o *Main Board* del nodo DeviceNet desarrollado en este proyecto, la imagen está generada con POV-Ray v3.6.

Salidas con TRIACS

Las salidas de 120VAC con TRIACs permite conmutar cargas mayores que las que maneja los relevos del diseño de la sección D.2.3 párrafo 2, además de las ventajas que tiene un circuito de potencia con TRIAC comparado con relevos, algunas de estas ventajas son: la velocidad de conmutación es mucho mayor que a que se puede conseguir con relevos, no sufren de desgaste mecánico, no se generan chispas etc.

Cada salidas de relevo esta conformada por dos elementos básicos mostrados en la figura 87, donde la figura 87 (a) es un OptoTriac usado para aislar galvánicamente las salidas del nodo DeviceNet de la parte Hot del circuito de potencia y evitar daños a los registros de salida del nodo Unicauca, también sirve como drive del TRIAC de potencia. El Triac figura 87 (b) es el elemento de potencia usado para conmutar la carga. En si la funcionalidad del circuito es similar a la del circuito con relevos, de echo el circuito con TRIAC conforman un Relevo de estado solido.

Fig. 87 Relevos usados en el circuito de salidas.



La tabla 14 lista las principales características de Optocouplers, éstas características son básicas para hacer los cálculos de corrientes y voltajes del circuito de Relés de estado solido y sirve para modificarlo (resistencias) en caso de conseguir un Optotriac mostrado en la tabla 14

Tabla 14. Características del Optotriac.

Optocouplers	V_{IOTM}	V_{DRM}	I_{FT}		V_{TM}		I_H	dv/dt	I_{TRMS}
	min.	min.	typ.	max.	typ.	max.	typ.	typ.	max.
	V	V	mA	mA	V	V	μA	V/ μs	mA
K3010P(G)	6000	250	8	< 15	1.5	3	100	10	100
K3011P(G)			5	< 10					
K3012P(G)			2	< 5					
K3020P(G)	6000	400	15	< 30	1.5	3	100	10	100
K3021P(G)			8	< 15					
K3022P(G)			5	< 10					
K3023P(G)			2	< 5					

Tabla 15. Características del TRIAC.

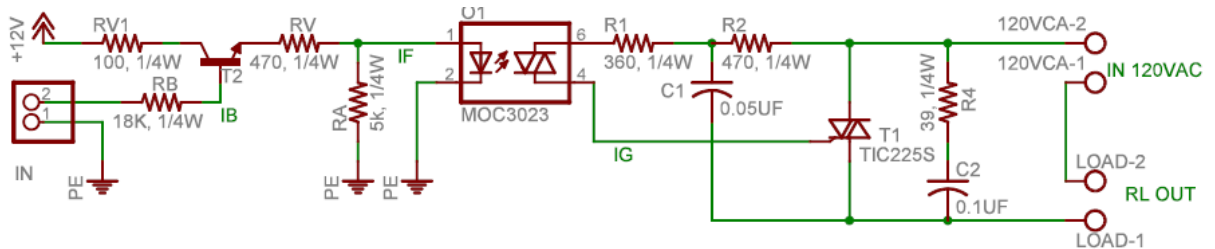
RATING		SYMBOL	VALUE	UNIT
Repetitive peak off-state voltage (see Note 1)	TIC225D	V_{DRM}	400	V
	TIC225M		600	
	TIC225S		700	
	TIC225N		800	
Full-cycle RMS on-state current at (or below) 70°C case temperature (see Note 2)		$I_{T(RMS)}$	8	A
Peak on-state surge current full-sine-wave (see Note 3)		I_{TSM}	70	A
Peak on-state surge current half-sine-wave (see Note 4)		I_{TSM}	80	A
Peak gate current		I_{GM}	± 1	A
Peak gate power dissipation at (or below) 85°C case temperature (pulse width $\leq 200 \mu s$)		P_{GM}	2.2	W
Average gate power dissipation at (or below) 85°C case temperature (see Note 5)		$P_{G(AV)}$	0.9	W
Operating case temperature range		T_C	-40 to +110	°C
Storage temperature range		T_{stg}	-40 to +125	°C
Lead temperature 1.6 mm from case for 10 seconds		T_L	230	°C

La figura 88 muestra el circuito del relé de estado sólido usado para manejar cada salida del nodo DeviceNet. La board de salidas con TRIAC esta conformada por ocho circuitos similares al de la figura 88 mostrados en el diagrama esquemático de la figura 89

Diseño del circuito de potencia con TRIACS

Cada uno de los circuitos desarrollados en este trabajo de grado fueron diseñados por el autor de este documento, no han sido tomados de ninguna parte, para el diseño de solo se hace soporte en cada Datasheet de las partes usadas, es por eso que para el siguiente circuito solo se hace el calculo de las resistencias de la parte del drive del Optotriac. Dado que la corriente máxima de los registros de salidas del nodo DeviceNet es de máximo 4mA al circuito de ejemplo entregado por el fabricante del Optotriac , se adiciono un driver que amplifique la corriente de los registros como indica claramente la figura 88

Fig. 88 Relevos usados en el circuito de salidas.



Calculando las resistencias RV y RB

Los datos que se obtienen del fabricante del MOC 3023 son:

$$I_b = \frac{I_c}{B} \tag{D4}$$

$$B = H_{fe} = 120 \tag{D5}$$

$$V_{fMax} = 1.5V \tag{D6}$$

$$I_{fmax} = 20mA \tag{D7}$$

$$R_b = \frac{V_{bi} - V_{bF}}{I_b} \tag{D8}$$

$$R_v = \frac{V_i - V_{fmax}}{I_{fmax}} \tag{D9}$$

$$R_a = 5k$$

Hallando la corriente de base del transistor

Como la corriente de base I_b es igual a la corriente de colector I_c dividido entre beta B , entonces podemos calcular la corriente de base fácilmente sabiendo que $I_c \approx I_F$ la corriente del diodo del Optotriac véase la figura 88

Reemplazando en D4 tenemos:

$$I_b = \frac{20mA}{120} = 0.17mA \quad (D10)$$

Hallando R_v

Reemplazando en D9 se puede obtener el valor de esta resistencia

$$R_v = \frac{(12 - 1.5)V}{(0.02)A} = 525\Omega \quad (D11)$$

Hallando la resistencia limitadora de corriente de base R_b

$$R_b = \frac{5 - 2}{0.00017} = 17650 \approx 18K \quad (D12)$$

La corriente I_a

Como $R_a = 5k$ entonces la corriente I_a es:

$$I_a = \frac{1.5}{5000} = 0.003A \quad (D13)$$

Observación. Como la corriente I_a es tan pequeña se considera despreciable

Fig. 89 circuito esquemático de la board de salidas con TRIAC.

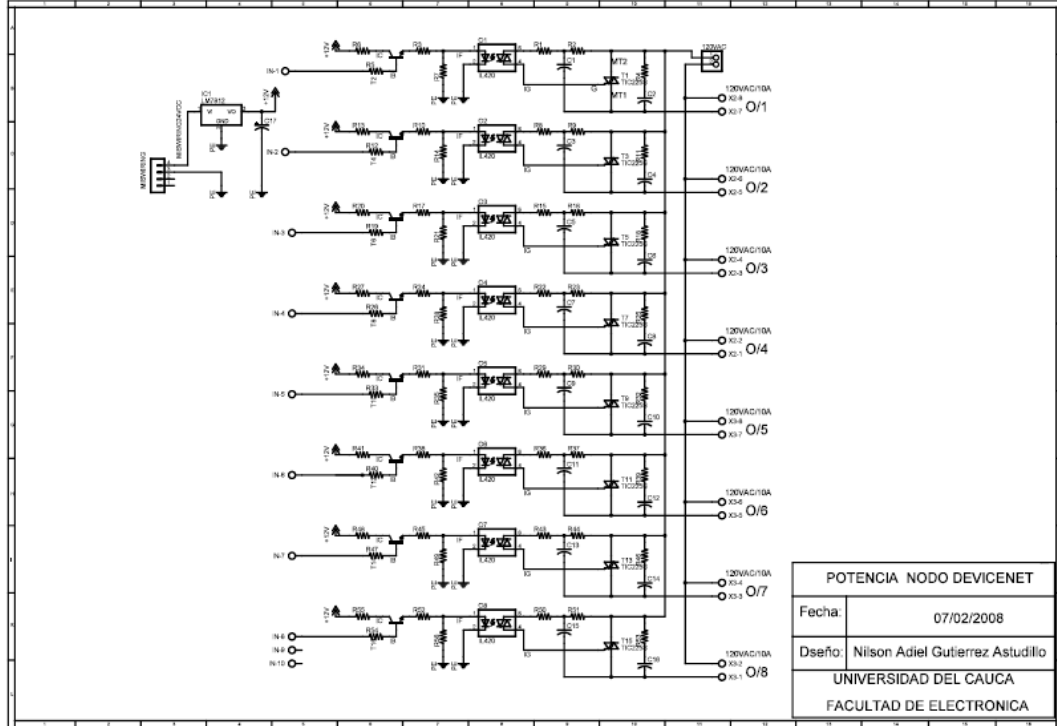


Fig. 90 PCB de salidas con TRIAC.

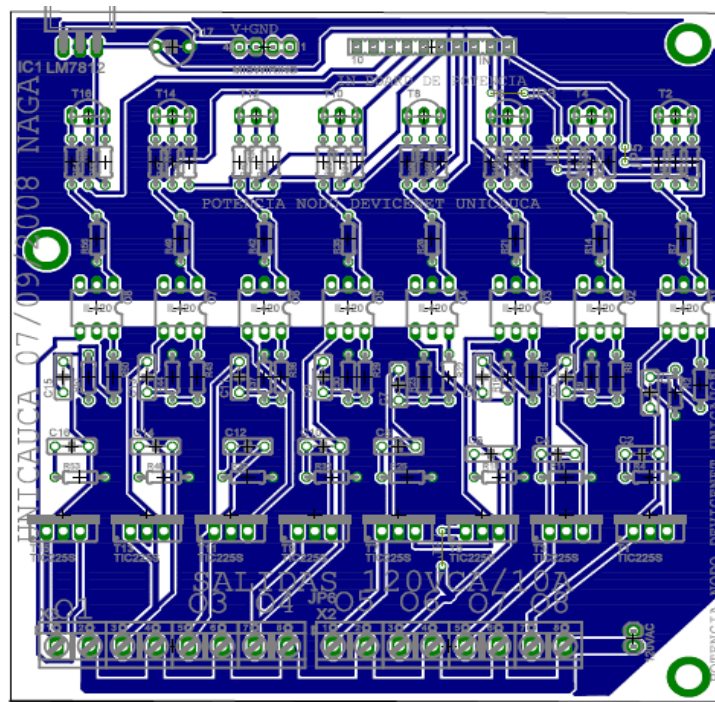
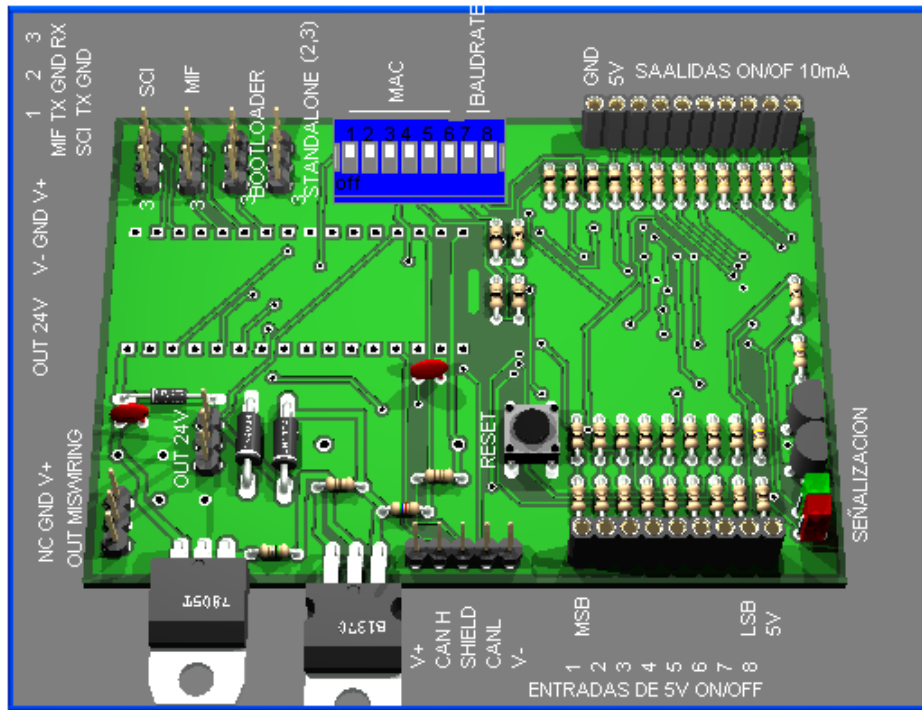


Fig. 91. Mainboard del nodo DeviceNet



D.4. CONFIGURACIÓN DEL BAUD RATE Y MAC ID DEL NODO DEVICENET

Se usa un *switch* binario para configurar el MAC ID y el Baud Rate, los 6 bit menos significativos usados para configurar el MAC ID y los 2 mas significativos modifican el Baud Rate. La tabla 16 lista el valor del Baud Rate en función del estado de los bits 6 y 7 del registro 0 (cero) de entradas...véase la figura 87...

Tabla 16. Switch de baud rate del nodo DeviceNet Unicauca

Bit 7 (MSB)	Bit 6	Baud Rate
0	0	125kbps
0	1	250kbps
1	0	(500kbps
1	1	NC

La tabla 17 lista la dirección del nodo (0-63) en función del estado de los bits 0-5

Tabla 17. Switch de MAC ID del nodo DeviceNet Unicauca

Bit 5	Bit 4	Bit 3	Bit 2	5 Bit 1	Bit0 (LSB)	MAC ID
0	0	0	0	0	0	0
0	0	0	0	0	1	1
0	0	0	0	1	0	2
-	-	-	-	-	-	-
1	1	1	1	0	1	61
1	1	1	1	1	0	62
1	1	1	1	1	1	63

La tabla 16 indica el estado del nodo DeviceNet en función de los bits 0 y 1 del registro cero de salidas.

Tabla 18. Indicadores de estado del nodo DeviceNet (señalización)

Bit	Estado	Indicación
-	(off)	Sin alimentación (Vcc)
0 (LSB)	Verde	Normal
1	rojo	Falla irrecuperable
	Rojo (intermitente)	Falla menor

D.5. CONFIGURACIÓN DEL BAUD RATE Y MAC ID DEL NODO DEVICENET

Esta sección está destinada a mostrar los diferentes diagramas del nodo DeviceNet generados por Eagle, aunque no muestran mucho detalle del trazado sirven para hacerse a una idea general de cada tarjeta del dispositivo. Para obtener información más precisa...véase el anexo digital H...la figura 92 (parte a) y 92(parte b) representa el diagrama esquemático de la tarjeta principal del nodo DeviceNet.

Fig. 92. Diagrama esquemático de la tarjeta principal del nodo DeviceNet Unicauca. Parte (a)

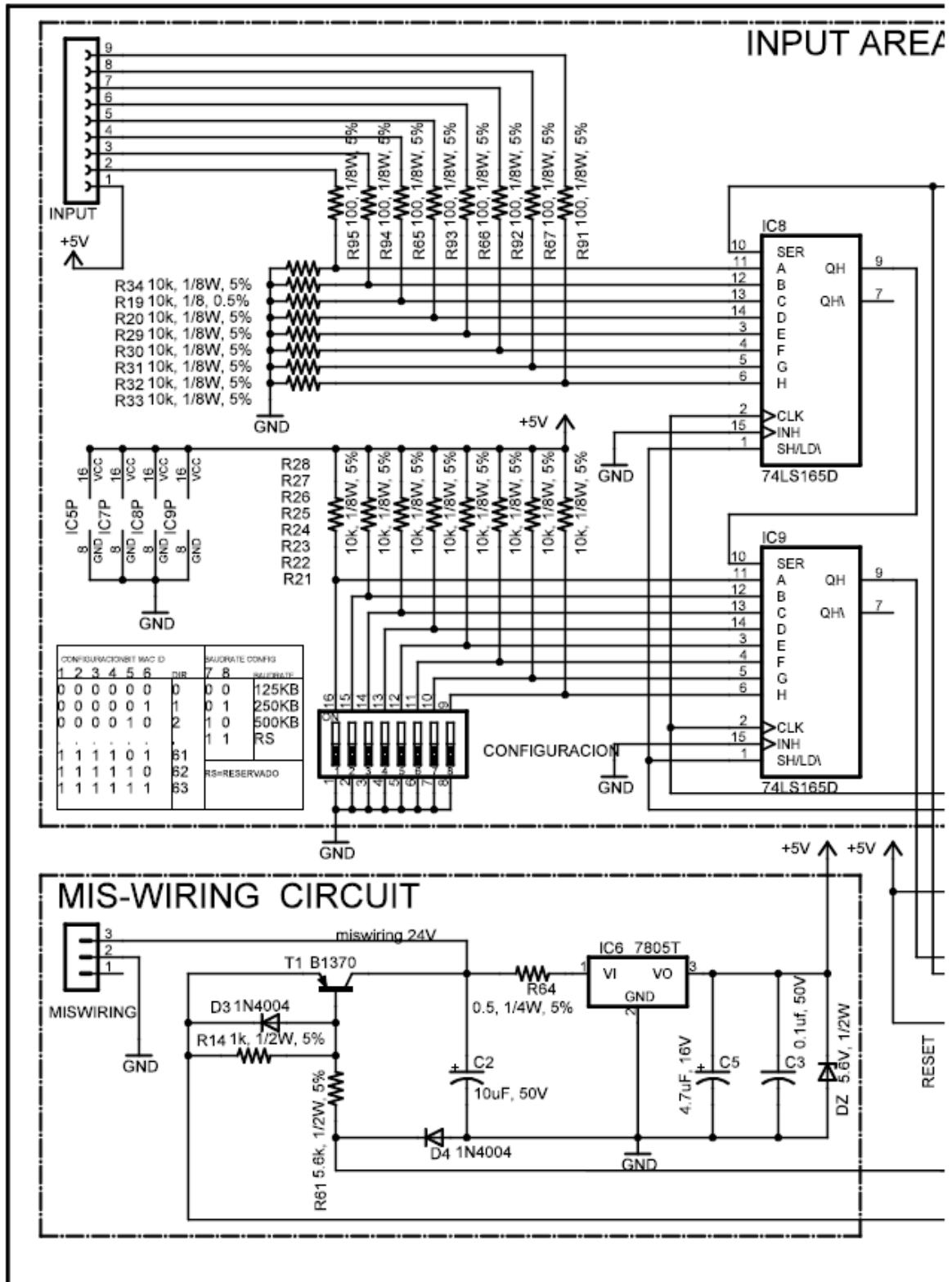
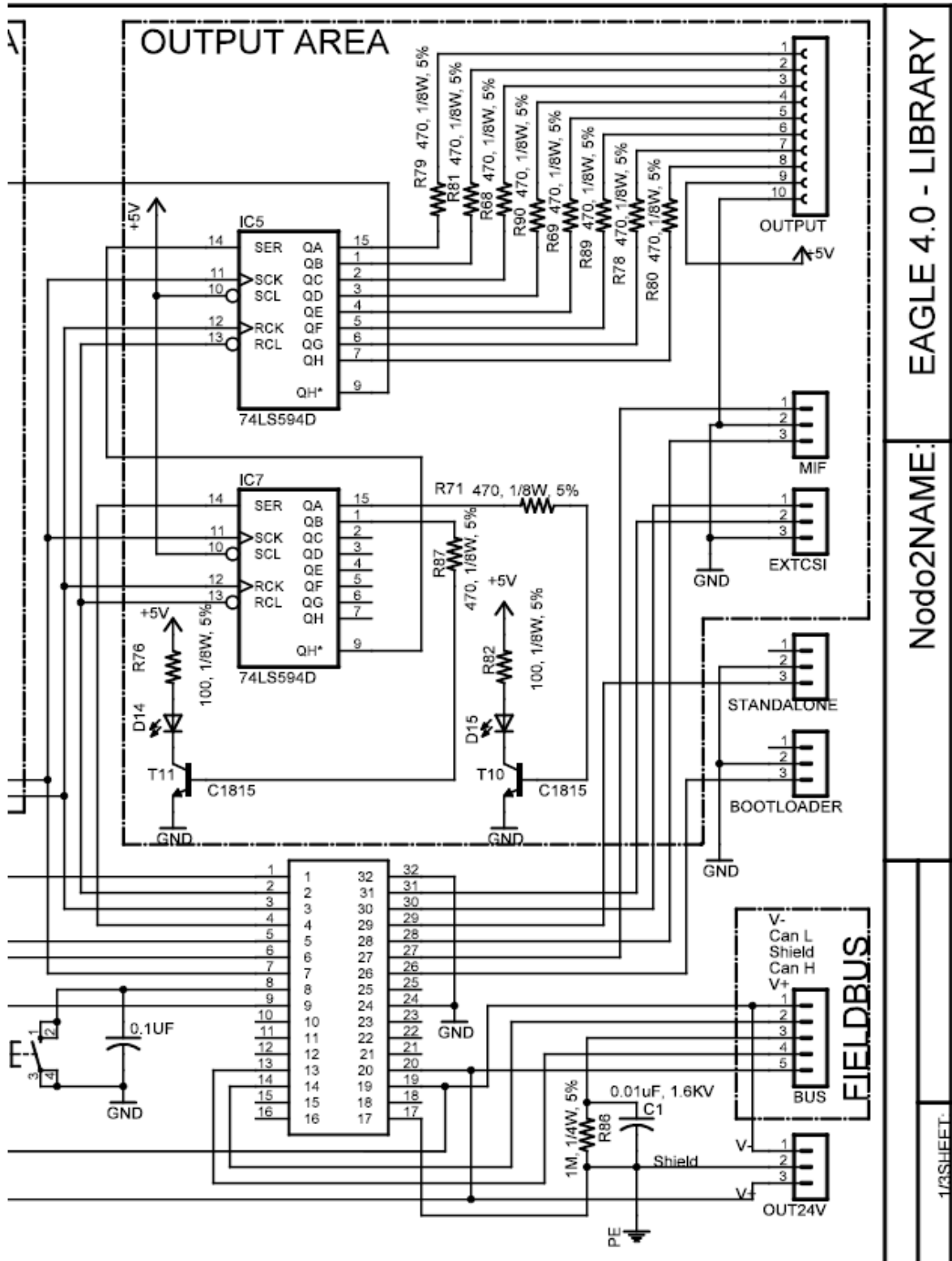


Fig 88. Diagrama esquemático de la tarjeta principal del nodo DeviceNet Unicauca. Parte (b)



La figura 93 muestra los lados bottom y top (lado inferior y superior) de la tarjeta principal, donde la figura 93 (a) corresponde a l lado top y 93 (b) a bottom.

Fig. 93. PCB de la tarjeta principal del nodo DeviceNet Unicauca.



La figura 94 y 95 corresponden a las tarjetas adaptadoras de entradas y salidas discretas respectivamente. Cada una de estas a diferencia de la tarjeta principal son tarjetas de una sola faz, lo que quiere decir que el cobre está en un solo lado (lado de las soldaduras).

Fig. 94 PCB de la tarjeta acondicionadora de entradas discretas



Fig. 95 Diagrama del circuito impreso de la tarjeta acondicionadora de salidas tipo relé

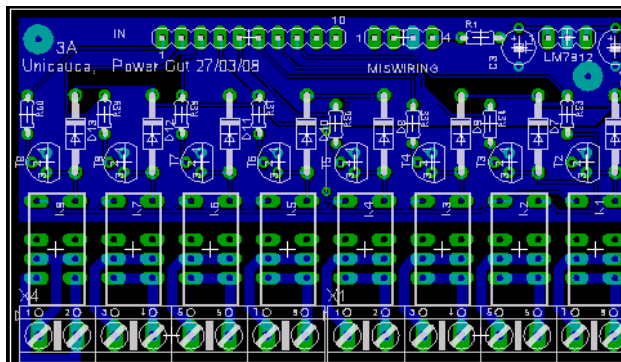
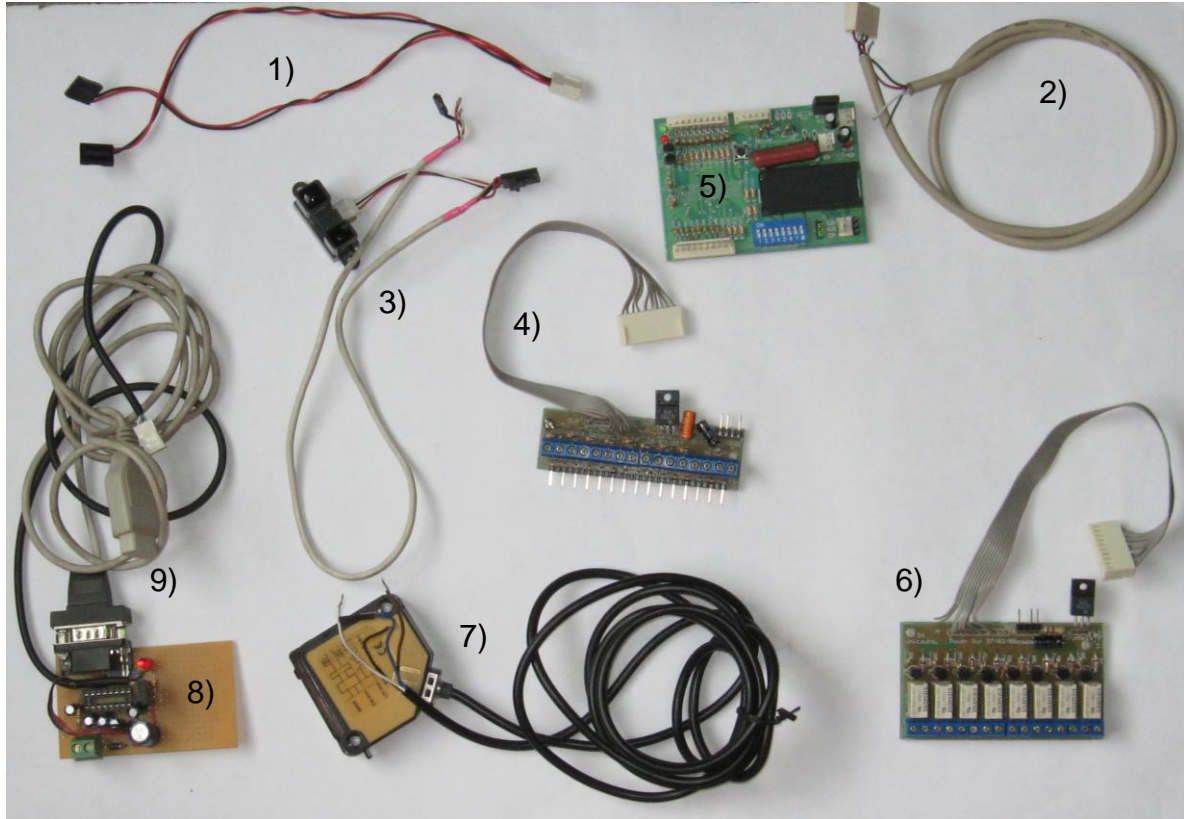


Fig. 96. Nodo DeviceNet desensamblado con las tarjetas de entradas y salidas



- 1) Cable que suministra alimentación del conector *miswiring* ubicado en la main board (MB) a los circuitos acondicionadores de señal (tarjetas de entradas y salidas discretas)
- 2) cable que comunica el nodo con el medio físico DeviceNet
- 3) sensor de proximidad Sharp
- 4) tarjeta acondicionadora de señal de entrada de los sensores
- 5) tarjeta principal del nodo DeviceNet Unicauca
- 6) tarjeta de salidas de relevo
- 7) sensor de proximidad de 12V Riko
- 8) interfaz de comunicación RS232
- 9) cable serial de configuración

Fig. 97 nodo DeviceNet desarrollad en la Universidad del Cauca

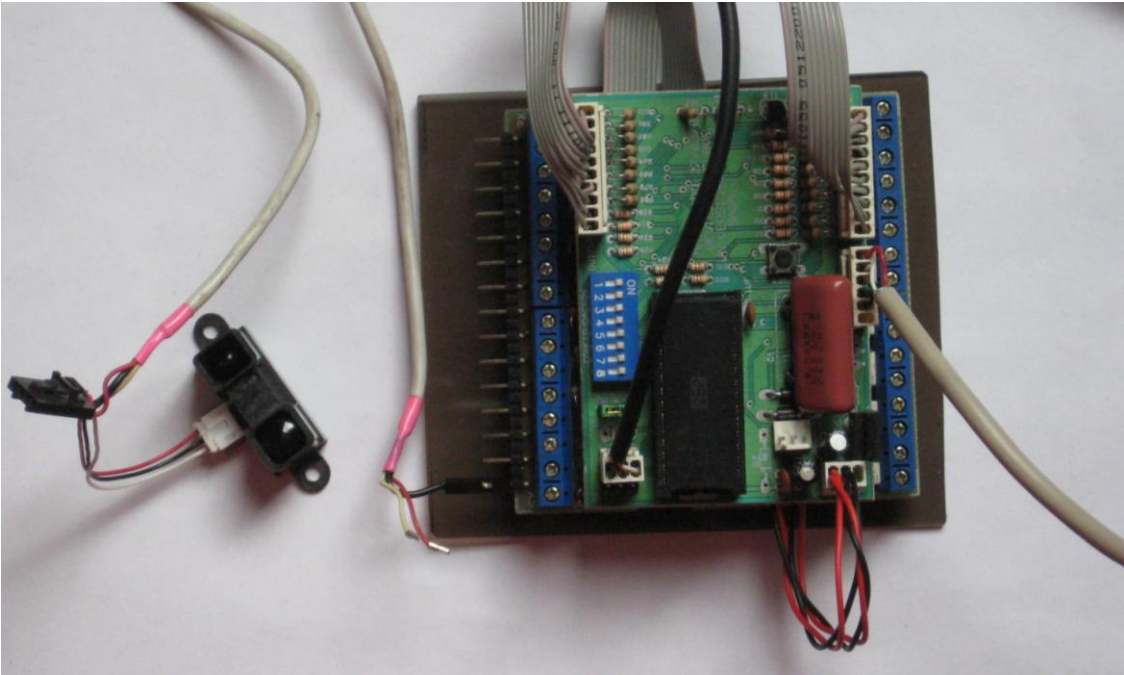
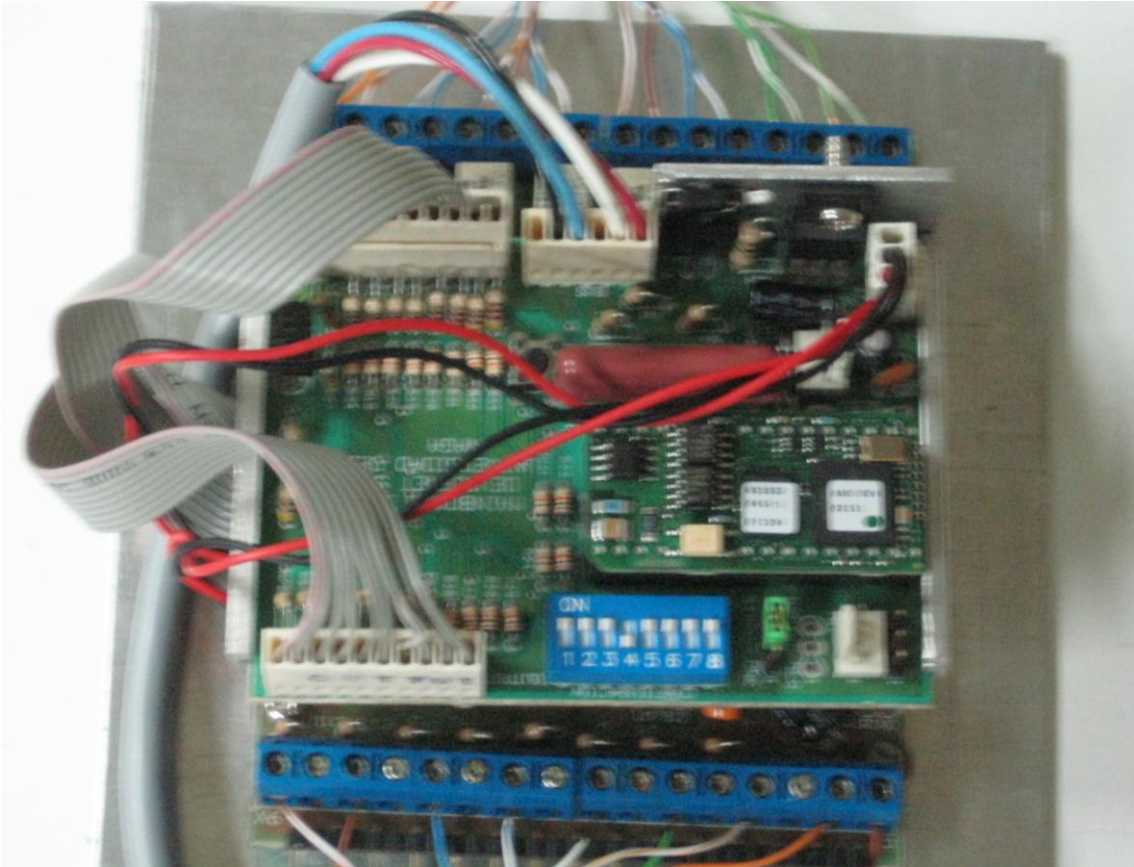


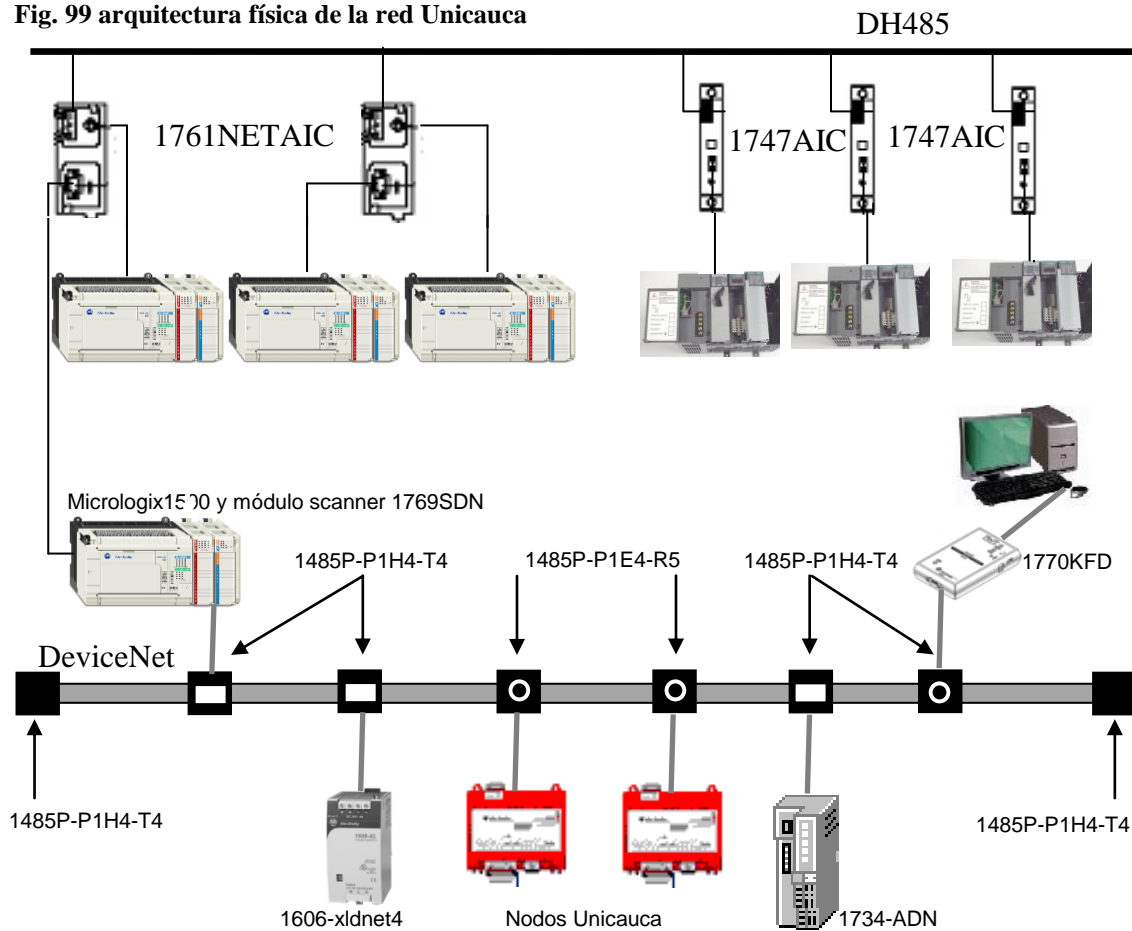
Fig. 98 Nodo DeviceNet Unicauca



ANEXO E, MANUAL DE CONFIGURACIÓN LA RED DEVICENET DE LOS LABORATORIOS DEL PIAI

A continuación se presenta la arquitectura física de la red DeviceNet montada en los laboratorios del PIAI de la Universidad del Cauca.

Fig. 99 arquitectura física de la red Unicauca



Después de verificar que todos los dispositivos en la red estén correctamente instalados conecte el PLC a la línea de alimentación de 120VCA y la fuente de alimentación DeviceNet de 24VDC para inicializar el sistema. Revisar que todo opere correctamente que el PLC y el Scanner enciendan y que el voltaje de 24VDC esté activo en los conductores V+ y V- de la línea troncal y en las diferentes líneas de derivación.

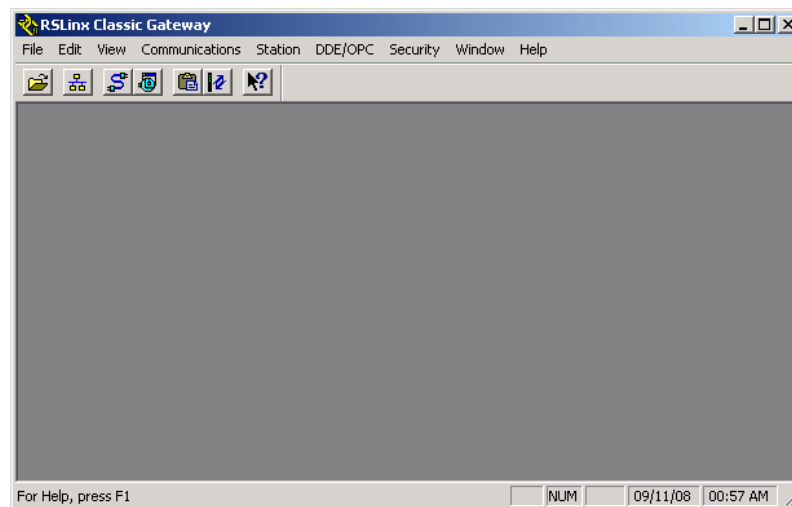
La configuración de la red DeviceNet involucra tres herramientas básicas de Rockwell, RSLinx, RSNetworkx y RSLogix 500, aunque para que el sistema opere solo requiere de RSLinx y RSNetworkx, pero hay que tener en cuenta que una red que no controle o que permita solo ver los dispositivos en línea no serviría de mucho, es por eso que se necesita generar un Ladder Logix o programa en escalera. Después de descargar la configuración al PLC se puede retirar el PC y el modulo de comunicaciones 1770KFD si se considera pertinente

E.1. CONFIGURACIÓN DE LA RED

E.1.1. Configuración de Drivers de comunicación a través de RSLinx.

1. Ejecute el programa RSLinx que se encuentra en: Inicio>Programas>Rockwell Software>RSLinx>RSLinx Classic, se mostrara la ventana indicada en la figura 100.

Fig. 100 ventana inicial de RSLinx



2. En el menú *communications* seleccione configurar drivers como se indica en la figura 101. Al hacer doble Click en configure drivers se abrirá la opción que permite adicionar drivers de comunicación véase la figura 102

Fig. 101. Ventana menú comunicaciones

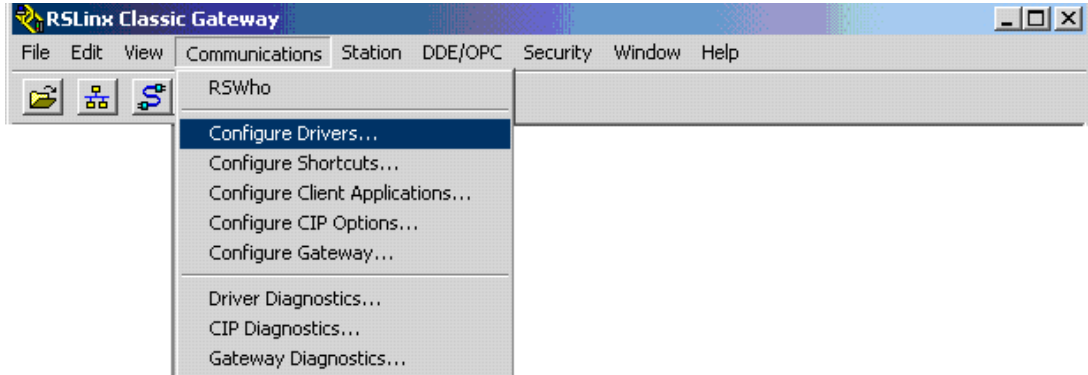
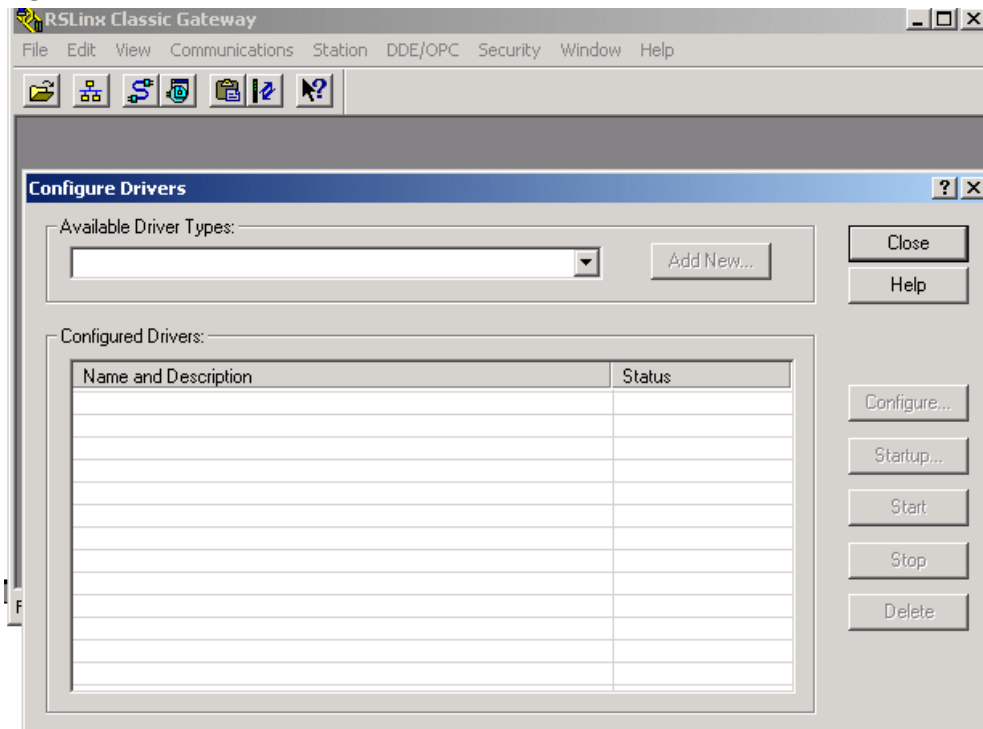
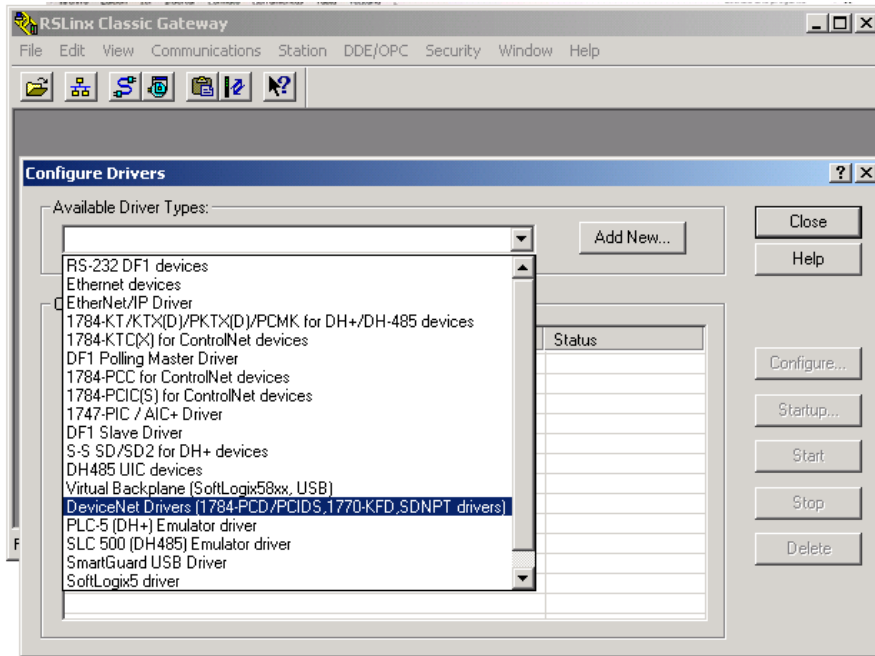


Fig. 102. Ventana de selección de drivers



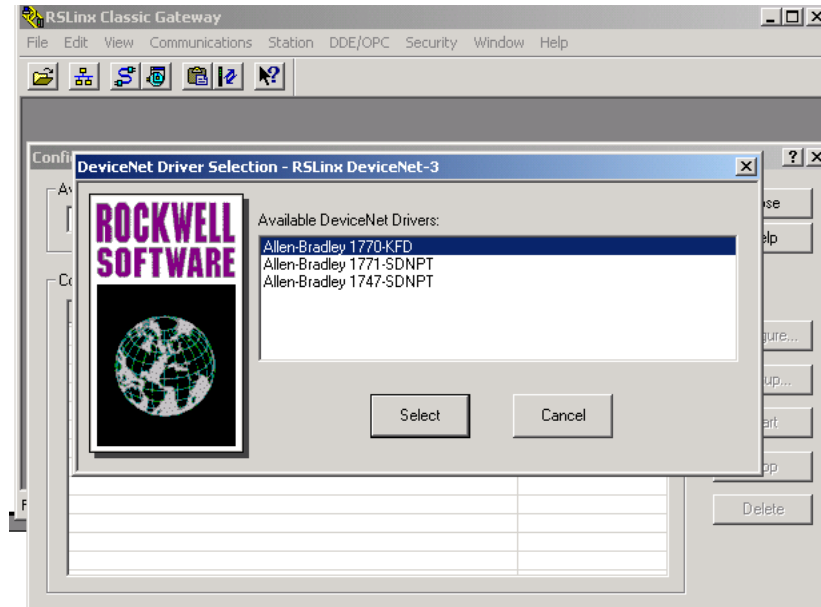
3. En la ventana Configure Drivers seleccione DeviceNet drivers [1784-PCD/PCIDS.1770-KFD. SDNPT drivers] y de Click en el botón Add New, como se indica a continuación en la figura 103.

Fig. 103 Seleccionando el driver



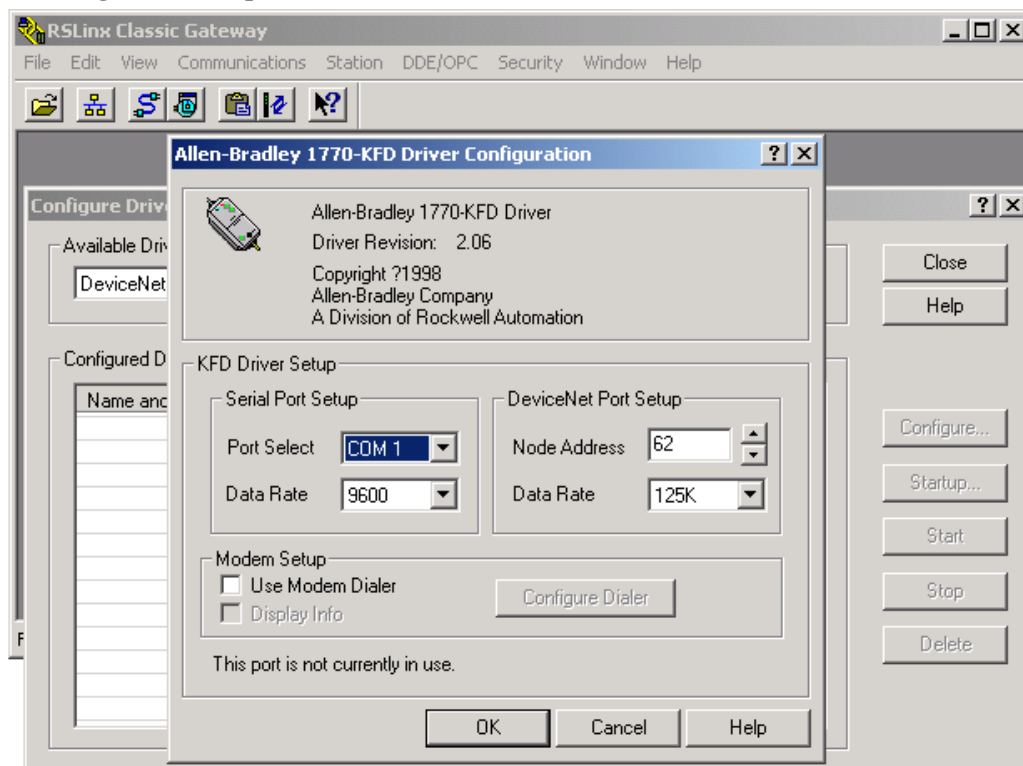
4. Entonces se abrirá la ventana de selección de Drivers DeviceNet como indica la figura 104. Seleccione el driver Allen Bradley 1770-KFD dando click en el botón select.

Fig. 104 driver DeviceNet



5. A continuación aparece la ventana de configuración del puerto COM del computador en el cual se va a conectar el modulo 1770KFD como indica la figura 105. La configuración del driver dependerá de la configuración particular de su sistema (puerto COM, velocidad en baudios, dirección de nodo en la red DeviceNet). Seleccione los parámetros apropiados para su sistema. En la configuración del puerto, se debe tener en cuenta el Baud Rate de la red y la dirección del nodo, es importante configurar una dirección diferente de 63 debido a que los dispositivos inician automáticamente en la posición 63 (configuración de fábrica).

Fig. 105. Configuración del puerto COM



6. Posterior a la configuración del puerto COM dar clic en ok entonces el sistema inicializa la interfaz DeviceNet, esta opción toma unos segundos tiempo en el cual se observa la siguiente ventana de la figura 106. Al terminar el tiempo de espera aparecerá la ventana de la figura 107 en la cual se puede renombrar el driver creado si es pertinente.

Fig. 106 inicializando la interfaz DeviceNet

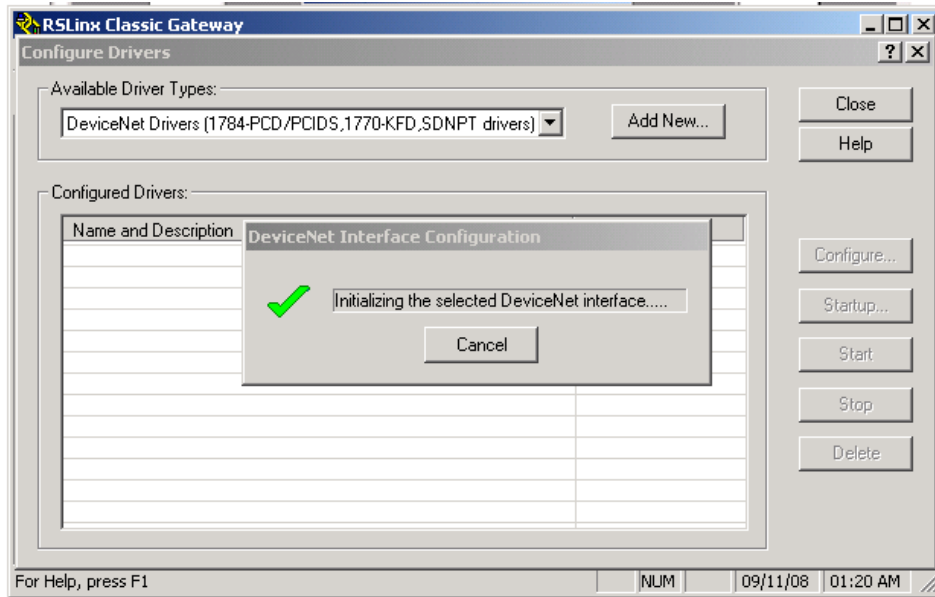
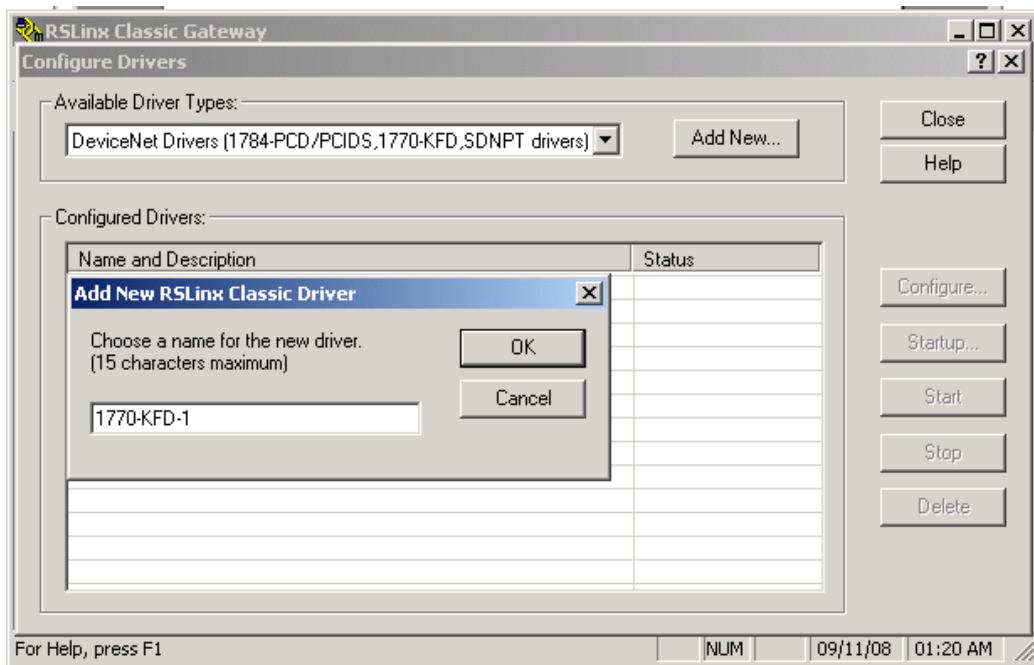
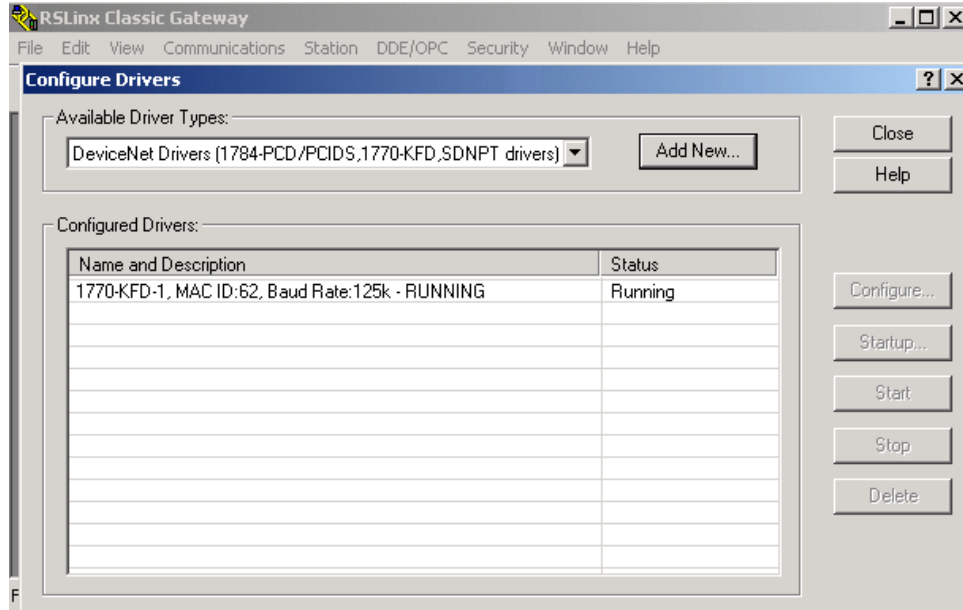


Fig. 107. Nombre del driver



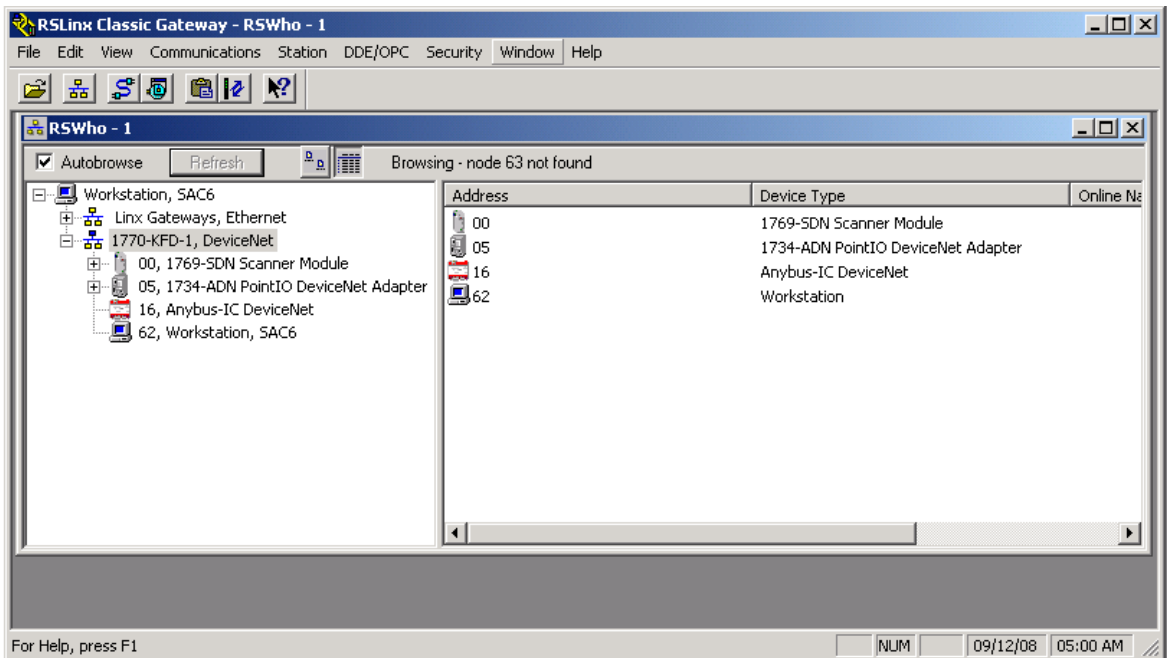
7. Posterior a la configuración, dar clic en ok. Si la configuración es correcta se observa el driver creado en modo *running* como indica la figura 108. Entonces se puede cerrar el RSLinx si es necesario.

Fig. 108. Driver en modo running



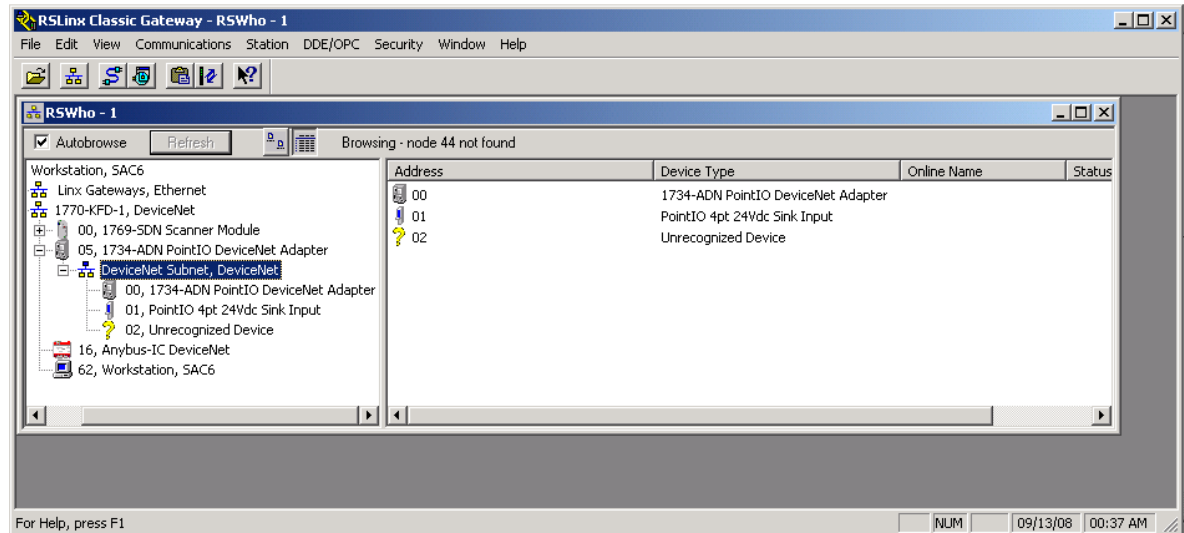
Si se desea se puede dar click en el boton RSWho para saber que dispositivos están en línea véase la figura 109.

Fig. 109. Nodos activos



Note que el nodo esclavo DeviceNet 1734ADN se comporta como un esclavo de la red principal, pero como un scanner en la sub-red de los módulos que conecta...véase la figura 110...

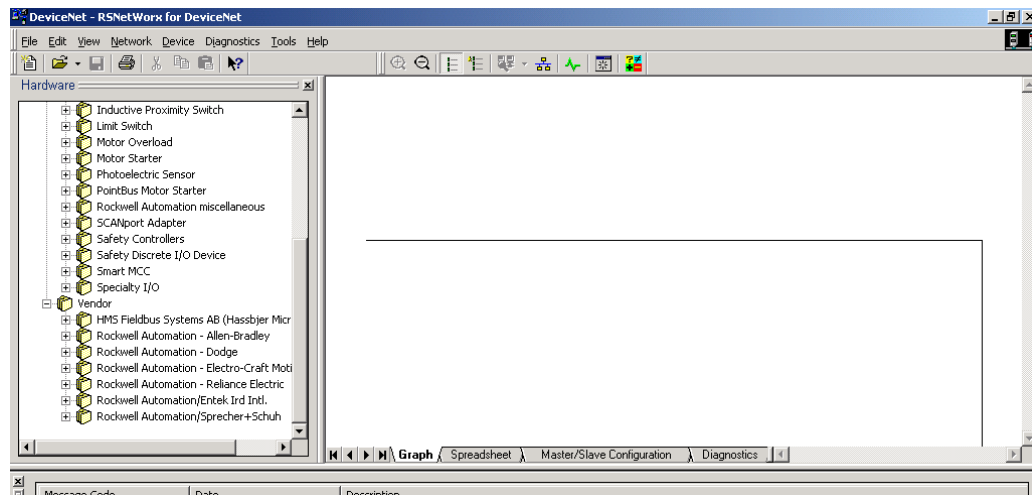
Fig. 110. Interfaces conectadas al nodo esclavo 1734ADN




E.2. CONFIGURACIÓN DE LA RED DEVICENET

1. Ejecute el programa RSNetworx que se encuentra en Inicio>Programas>Rockwell Software> RSNetworx > RSNetworx for DeviceNet. Una vez inicializado, cree un archivo nuevo. La figura 111 muestra la ventana inicial de RSNetworx para DeviceNet

Fig. 111. Ventana principal de RSNetworx para DeviceNet



2. Hacer doble click en el botón *online*  en la barra de herramientas, para que la red realice un Autobrowse (escaneo) de los dispositivos conectados a la red. Entonces aparece la ventana de selección del *path* (conducto) a la red deseada en este caso solo tenemos 1770KFD DeviceNet, donde se debe seleccionar 1770-KFD como indica la figura 112 y posteriormente dar doble click en ok, entonces aparecerá en pantalla la siguiente ventana mostrada en la figura 113.

1. Dar click en OK (figura 113) para subir (*Upload*) los dispositivos y pasar a modo online. Durante este proceso RSNetWorx busca los dispositivos disponibles en la red DeviceNet, ver la figura 114, cuando el software termina la búsqueda.

La figura 114 muestra el proceso de escaneo de dispositivos conectados físicamente a la red DeviceNet, este proceso tarda algunos segundos.

Fig. 112. Seleccione el Path

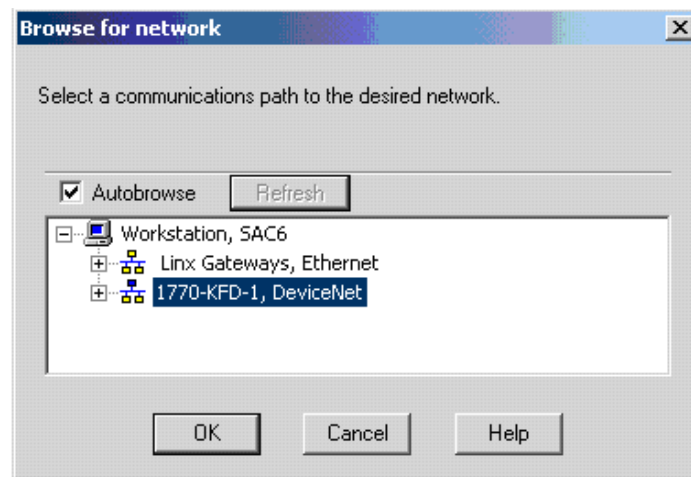
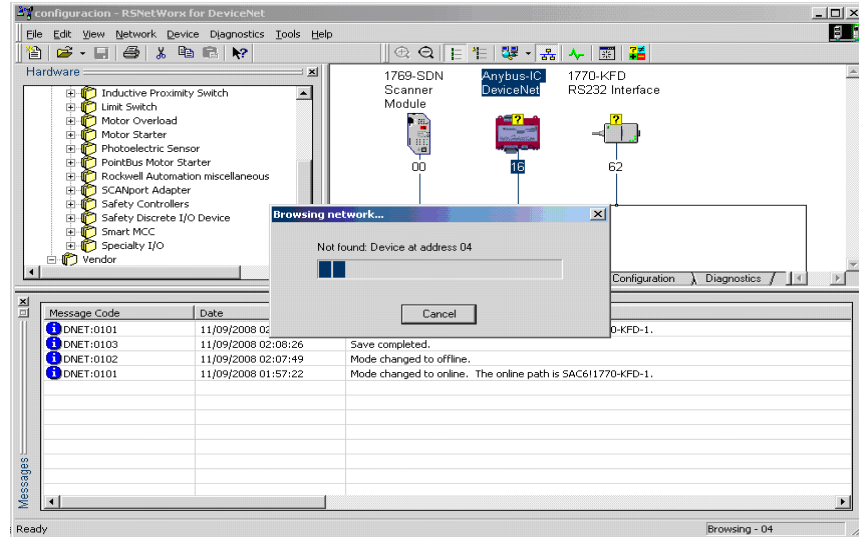


Fig. 113. Confirmación de búsqueda de dispositivos

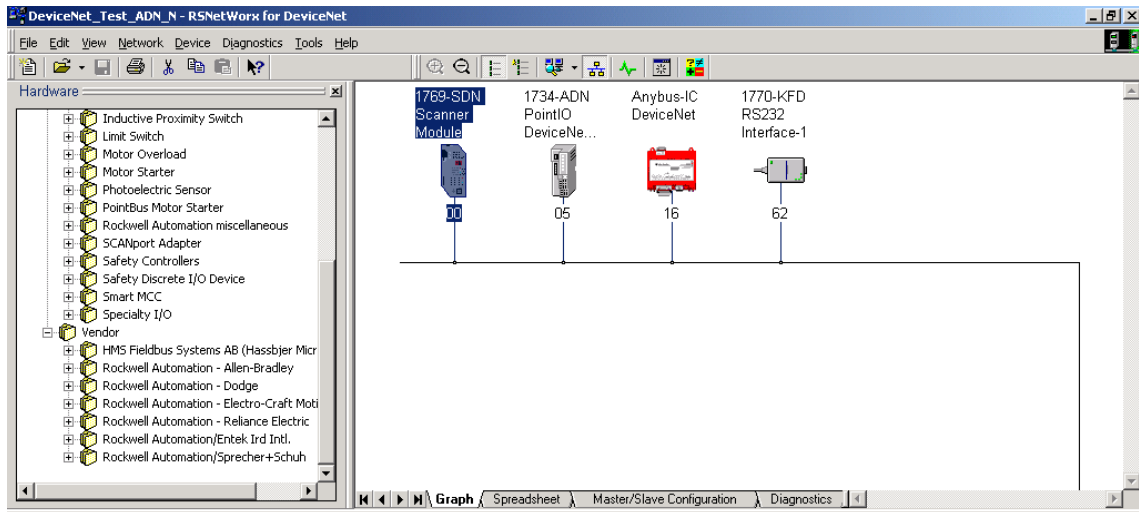


Fig. 114. Proceso de Autobrowse



Finalizado el Autobrowse el sistema entrega la lista de los dispositivos en línea indicando sus direcciones y estados en la red. En este caso solo hay dos nodos esclavos 1734ADN y nodo Unicauca, pero también se puede observar el nodo de configuración 1770KFD y el scanner 1769SDN...véase la figura 115...

Fig. 115. Nodos activos en la red



Si un nodo en línea pasa a estado fuera de línea no se mostrara en la red DeviceNet después del proceso de scan (busqueda), se puede buscar manualmente el nodo perdido usando la herramienta Browse Network de DeviceNet. Para ejecutar búsqueda manual, presionar el botón browse Network en la barra de herramientas, si la búsqueda falla

entonces se debe verificar la conexión física del dispositivo, también de debe verificar si el dispositivo tiene el mismo Baud Rate de la red.

El software de configuración de DeviceNet debe incluir los archivos EDS requeridos para cada dispositivo, para obtenerlos se puede remitir a la página <<http://www.ab.com/networks/eds>> para descargar los archivos. Posterior a la descarga se deben registrar los archivos, usando EDS wizard en RSNetWorx. Los archivos EDS del nodo DeviceNet Unicauca están ubicados en el anexo digital I...ver la sección 4.9.10...

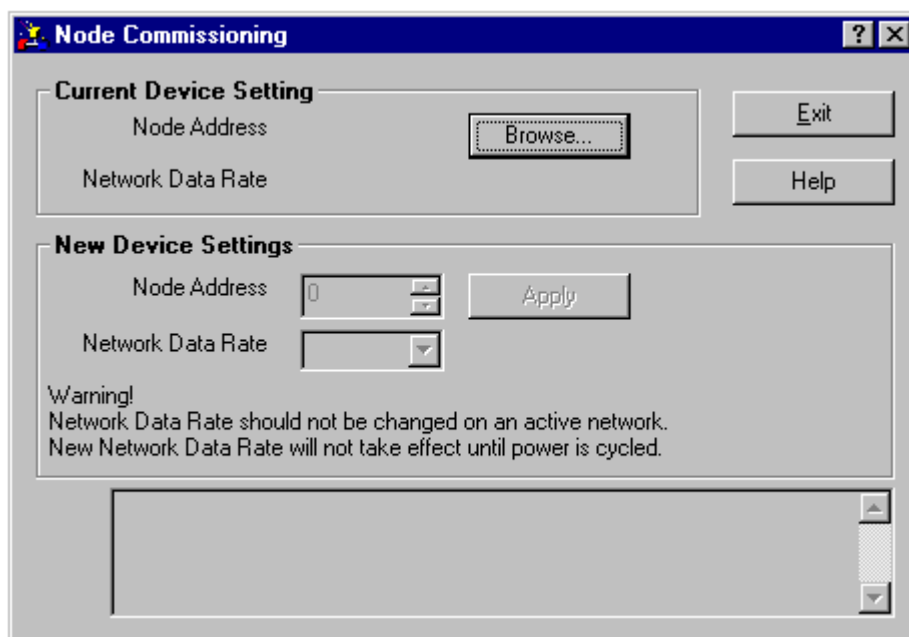
E.2.1. Direccionamiento de los nodos DeviceNet

Una vez que finaliza el escaneo de dispositivos, se puede cambiar la dirección de los nodos, si es pertinente, para evitar duplicaciones de MAC ID en la red, también es posible configurar el Baud Rate del dispositivo de acuerdo con la velocidad de la red.

El controlador debe estar en modo Program o el scanner en modo Idle para que el scanner acepte la nueva configuración. La figura 116 muestra la ventana de configuración.

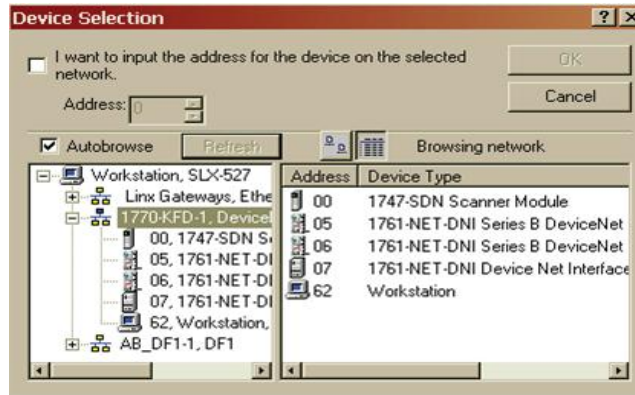
1. En el menú Tools seleccionar Node Commissioning.

Fig. 116. Comisionar dispositivos DeviceNet



2. Dar clic en el botón browse, como indica la figura 116, entonces aparecerá la ventana de selección de dispositivos. Ver la figura 117.

Fig. 117. Selección de dispositivos DeviceNet



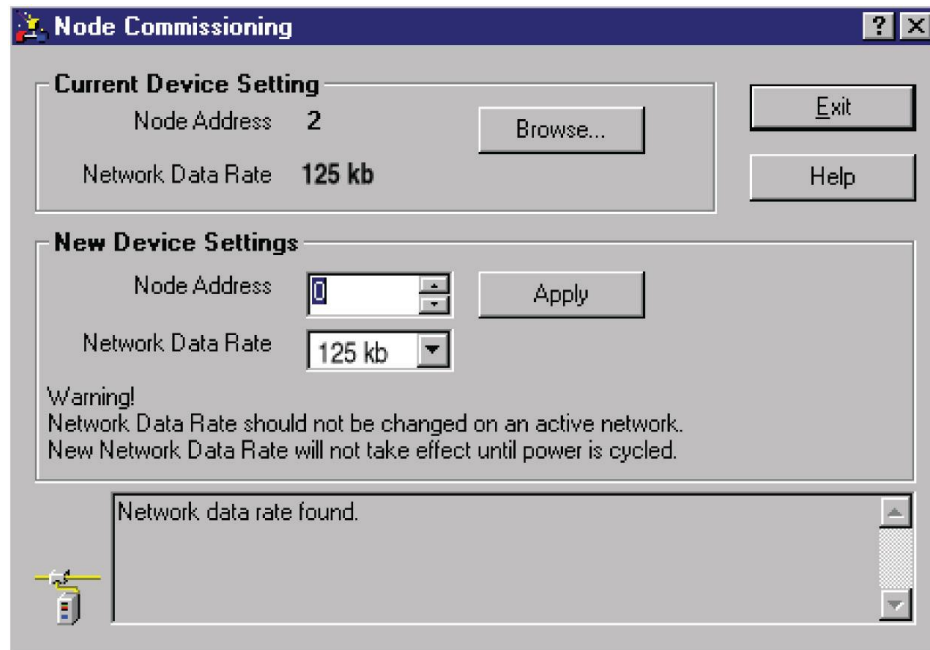
3. Seleccionar DeviceNet Network. Los dispositivos de la red DeviceNet aparecerán en el panel derecho de la ventana.

4. Seleccionar el dispositivo en el panel derecho y dar Click en ok.

Entonces aparece la siguiente ventana mostrada en la figura 118, donde se puede observar los valores por defecto del dispositivo. Por ejemplo los valores por defecto del scanner son: Baud Rate = 125K, Dirección del nodo = 63.

Se recomienda cambiar la dirección del nodo (diferente de 63) para evitar duplicaciones de MAC ID.

Fig. 118. Valores por defecto del scanner Devicenet



5. En la sección New Device Setting de la ventana, modificar *Node address* y *Network Data Rate* según la conveniencia teniendo en cuenta los parámetros de la red.

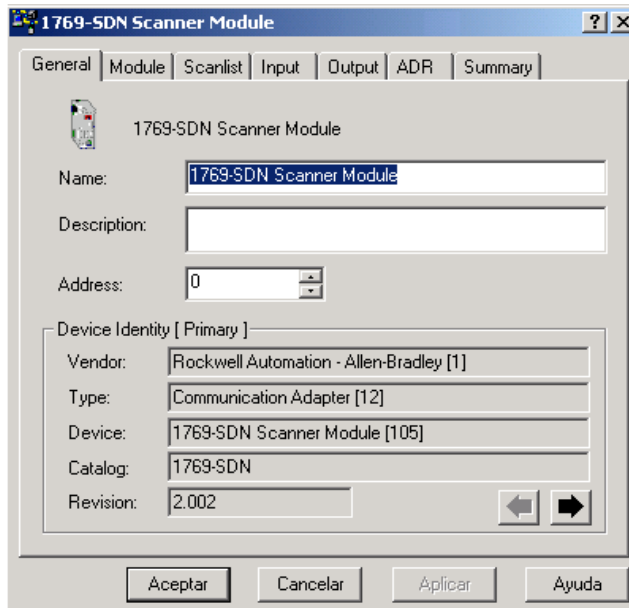
6. Dar clic en el botón apply y cerrar la ventana.

E.2.2. Configuración de dispositivos DeviceNet

Configure los parámetros de cada nodo esclavo en la red. En adelante cada vez que modifique un parámetro descárguelo sobre el dispositivo modificado y almacene su archivo de configuración. Mantenga el procesador (Micrologix 1500) en modo *Program* para poder realizar las descargas.

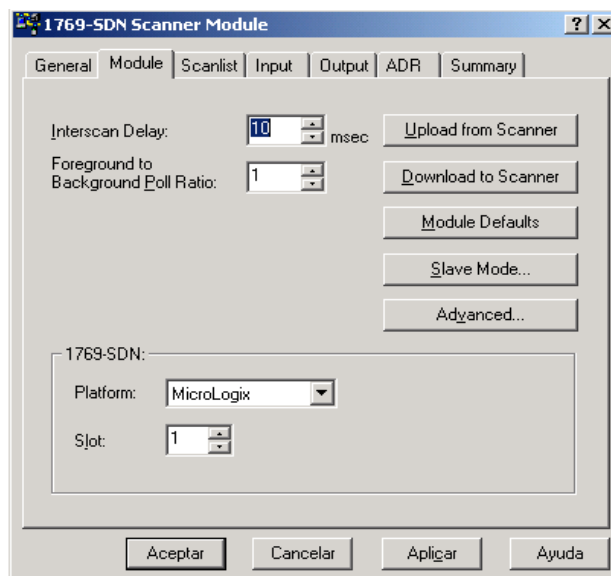
Dé doble clic sobre el 1769-SDN, de esta manera accederá a la ventana de propiedades (figura 119). En la pestaña 'General' puede modificar el nombre y la dirección en la que se ubica el scanner en la red. Para mas información de lo parámetros configurables del dispositivo. (*General, Module, Scanlist, Input, Output, ADR y Summary*) Véase el anexo C

Fig. 119. Pestaña General de la ventana de propiedades del scanner



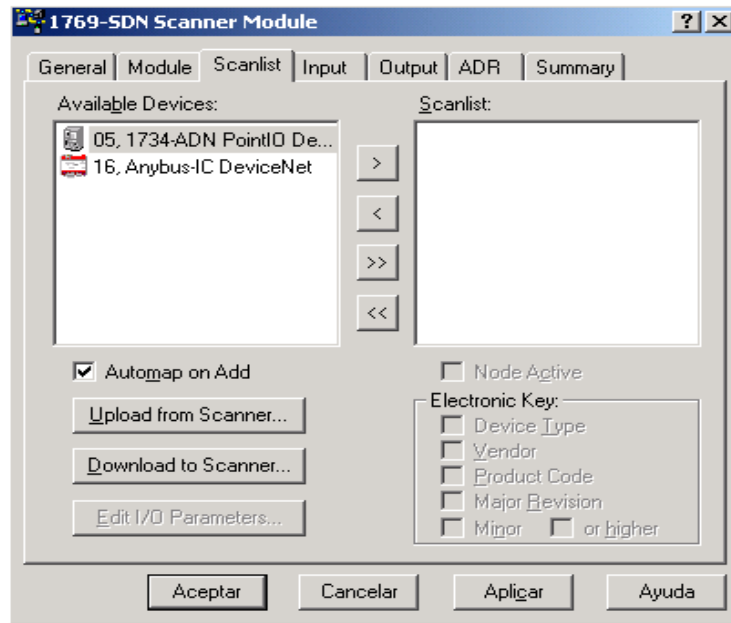
En la pestaña “Module” puede configurar el retardo de escaneo, la relación de escaneo entre nodos Foreground y Background, la plataforma sobre la cual esta conectado el scanner y el slot en el cual se ha situado entre otras opciones...véase el anexo C...si es pertinente cambie la plataforma a MicroLogix como aparece en la figura 120 y slot dependiendo de la posición del scanner en las ranuras del PLC, los demás parámetros se pueden dejar los dados por defecto.

Fig. 120. Pestaña Module



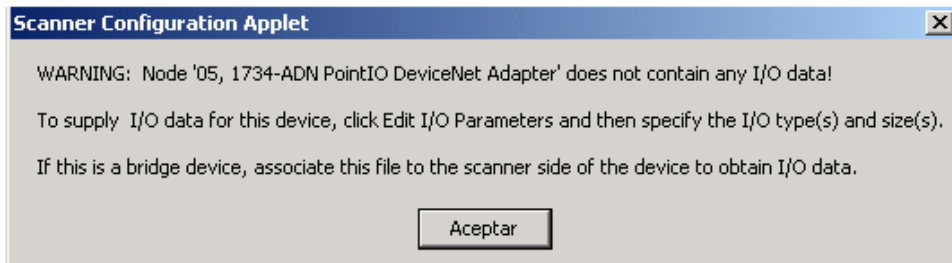
En la pestaña “ScanList” puede crear la lista de escaneo del 1769-SDN. Esta pestaña se divide en las ventanas: dispositivos disponibles y Scanlist. Mediante los controles entre estas ventanas puede adicionar o quitar dispositivos del Scanlist...véase la figura 121...

Fig. 121. Pestaña Scanlist



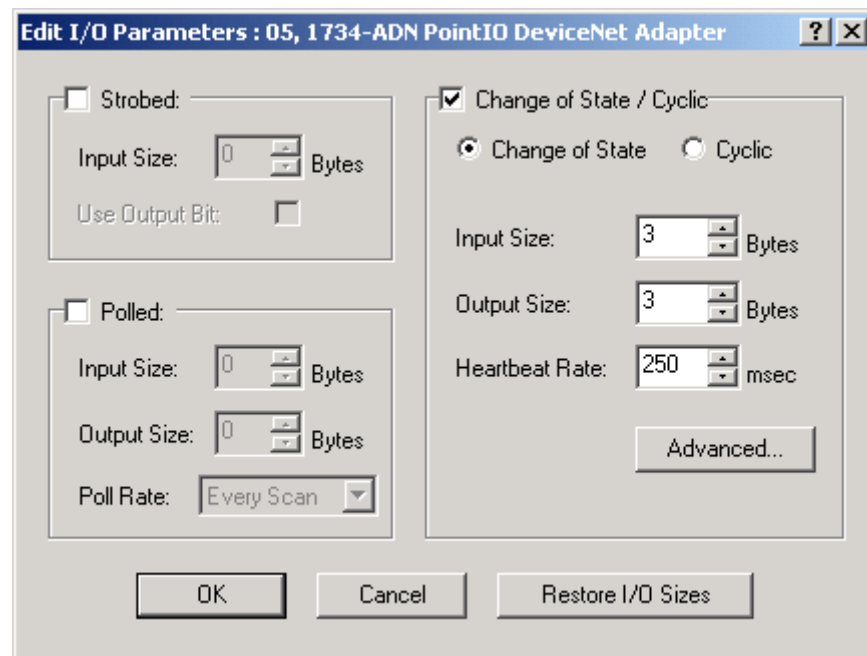
Adicione el nodo Unicauca y el nodo adaptador 1734-ADN en la lista, aparecerá un mensaje indicando que es necesario configurar la información de entradas y salidas del ADN (figura 122). Esta información aparece cuando esta el nodo ADN conectado a la red, con el nodo Unicauca no se debe tener en cuenta esta advertencia. Los parámetros del nodo Unicauca son automáticamente configurados por el sistema, no es necesario cambiar los parámetros, únicamente se debe adicionar al Scanlist y ver dónde se mapean I/O del dispositivo...véase anexo C...

Fig. 122. Advertencia ADN



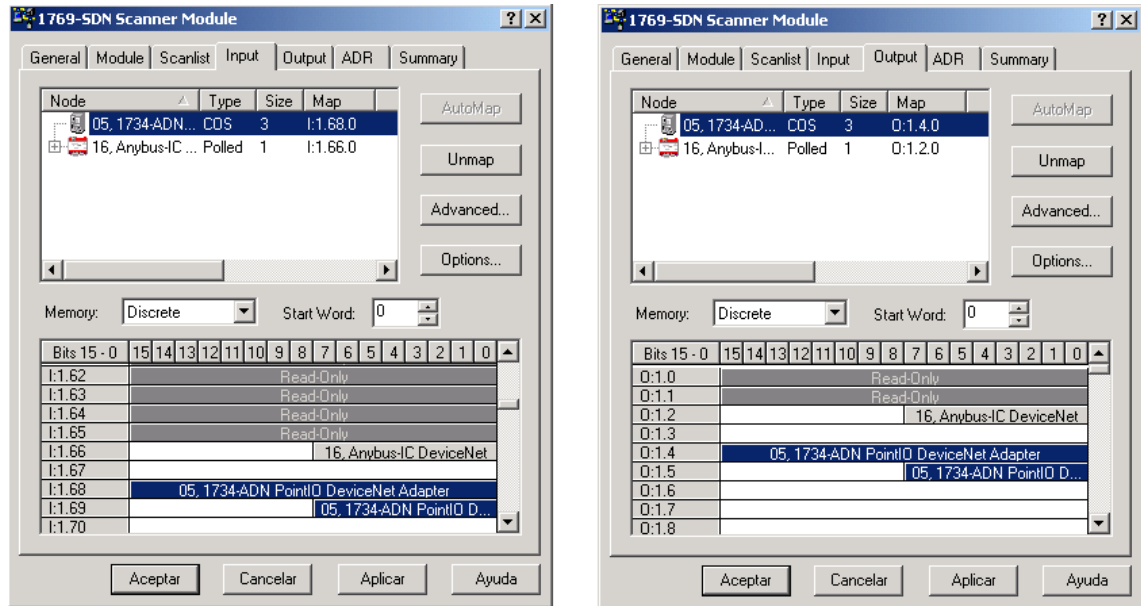
Para realizar la configuración de parámetros del ADN, seleccione el botón Edit I/O Parameters en la pestaña ScanList. En la configuración de este ejemplo, existen dos módulos de E/S conectadas al ADN: 1734 IB4 (4 entradas discretas), 1734 OA4 (4 salidas discretas). Cada dispositivo intercambia 1 byte con el scanner, adicionalmente el ADN intercambia 2 bytes de entrada y 2 bytes de salida con el scanner. En total, el adaptador y sus módulos intercambian 3 bytes de entrada y 3 bytes de salida con el scanner. Esta información es necesaria para configurar el mecanismo de activación del nodo como se muestra a continuación en la figura 123.

Fig. 123. Configuración de parámetros I/O del ADN



En la pestaña ScanList verifique que la casilla “Automap on add” este habilitada. Esta opción permite mapear automáticamente los datos intercambiados con los nodos en los registros de E/S del scanner. Para comprobar las direcciones de memoria del scanner en las cuales están siendo mapeados los nodos, revise las pestañas input, output. Note que existen espacios de memoria reservados que se identifican con la etiqueta **ReadOnly**, estos espacios corresponden a los Word [0 a 65] en el caso de las entradas y [0,1] en el caso de las salidas. Dichos espacios no pueden ser empleados para el mapeo de los nodos de la red. La figura 124 lista los dispositivos que son mapeados dentro de las imágenes de salida y entrada del scanner en este caso solo se puede observar el ADN.

Fig. 124. Posición de I/O del ADN mapeadas en el scanner



En este ejemplo se han mapeado los nodos según la tabla 19 pero se puede manualmente asignar la dirección deseada tanto a las salidas como a las entradas.

Tabla 19. Mapeo de nodos en la memoria del scanner

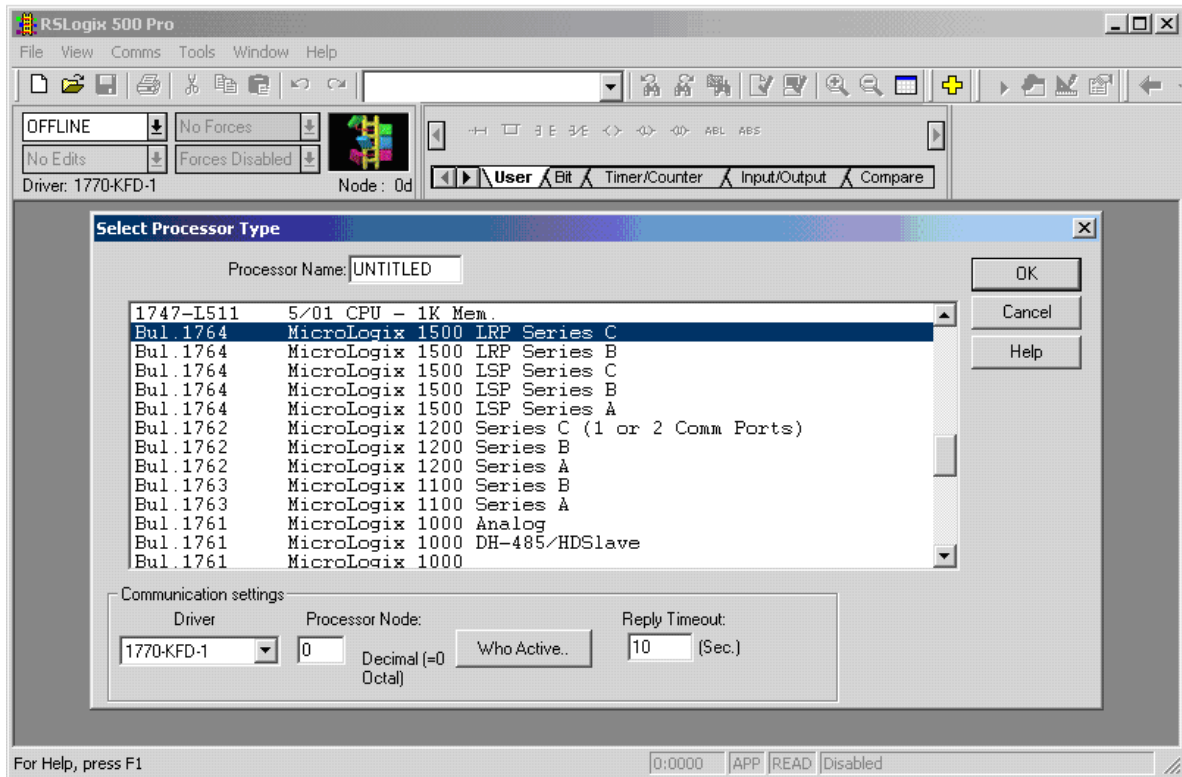
Dispositivo	Entradas	Salidas
Nodo Unicauca	Parte baja del Word 66	Parte baja del Word 2
ADN	Word 68	Word 4
1734 IB4	Parte baja del Word 69. (Específicamente los 4 primeros bits)	N/A
1734 OA4	N/A	Parte baja del Word 5. (Específicamente los 4 primeros bits)

Esta información es importante en el momento de realizar el ladder Logic o programa en escalera.

E.2.3. Creación de un programa en escalera para realizan control a través de la red DeviceNet

1. Ejecute el programa RSLogix 500 Pro que se encuentra en Inicio>Programas>Rockwell Software> RSLogix 500 English > RSLogix 500 English. Cree un proyecto nuevo tomando como procesador el Micrologix 1500 LRP Series C (figura 125) y almacénelo con un nombre apropiado.

Fig. 125. Selección del procesador MicroLogix 1500



2. Ingrese a la ventana de configuración de entradas/salidas y presione el boton Read I/O Config para identificar los módulos conectados al PLC (figura 126). A continuación aparece la ventana de selección de la ruta a través de la cual se realizará la lectura de módulos conectados al PLC. (Figura 127) Presione el boton Who Active para abrir la ventana Communications (figura 128) y seleccione el 1769-SDN, luego presione OK.

Fig. 126. Configuración de entradas y salidas dispuestas en el PLC

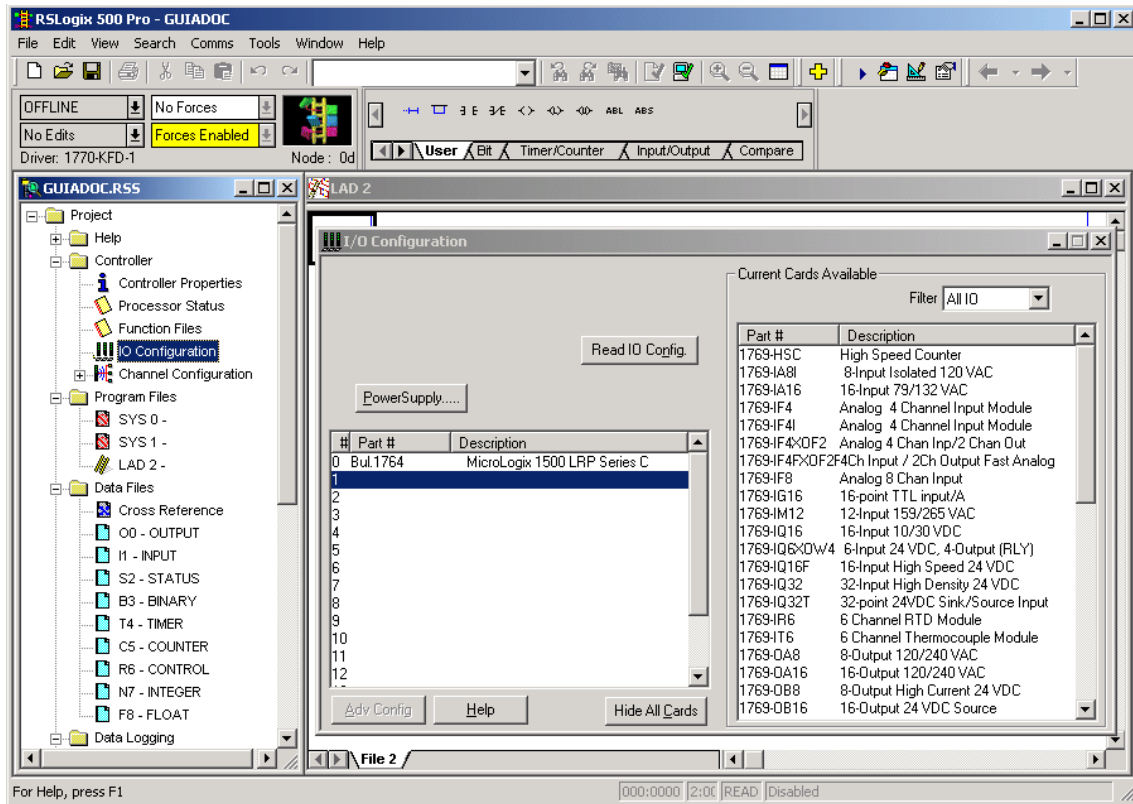


Fig. 127. Selección del Path para detectar los módulos de salidas y entradas en el PLC

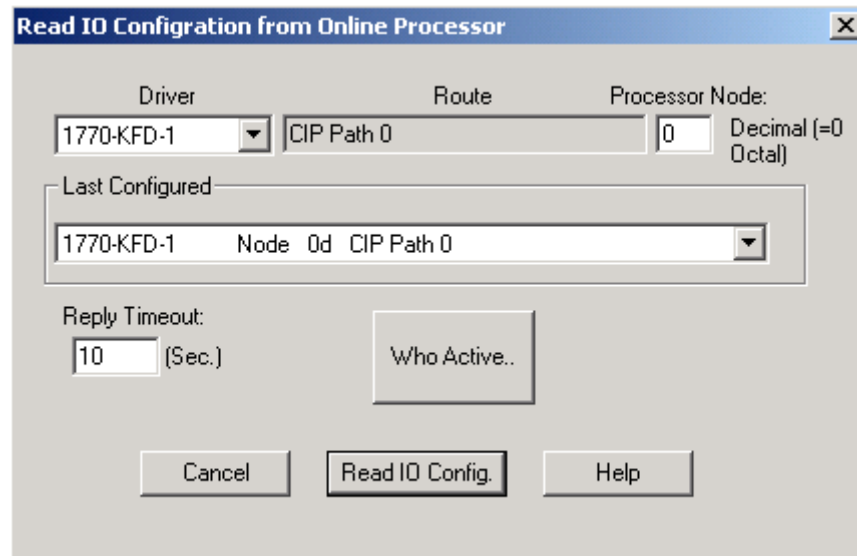
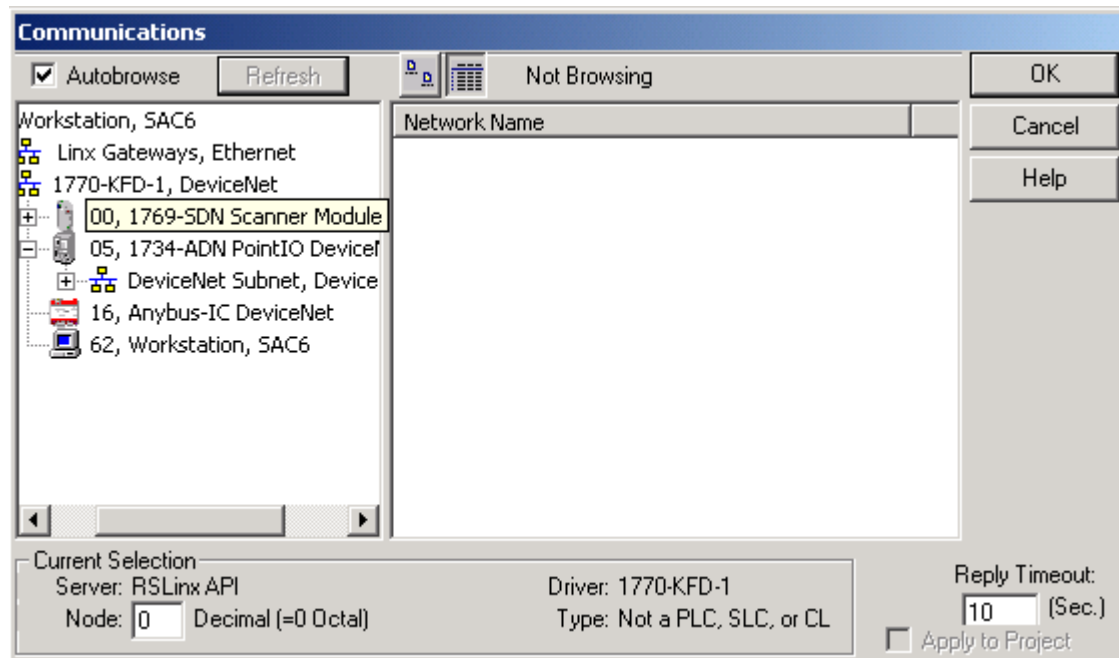
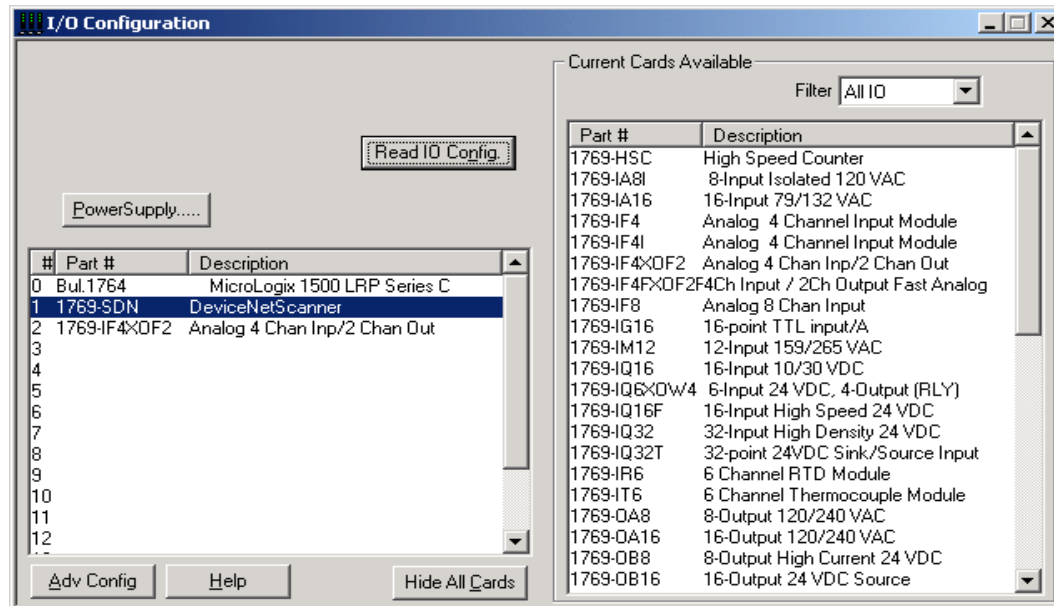


Fig. 128. Ventana de selección del dispositivo de comunicaciones



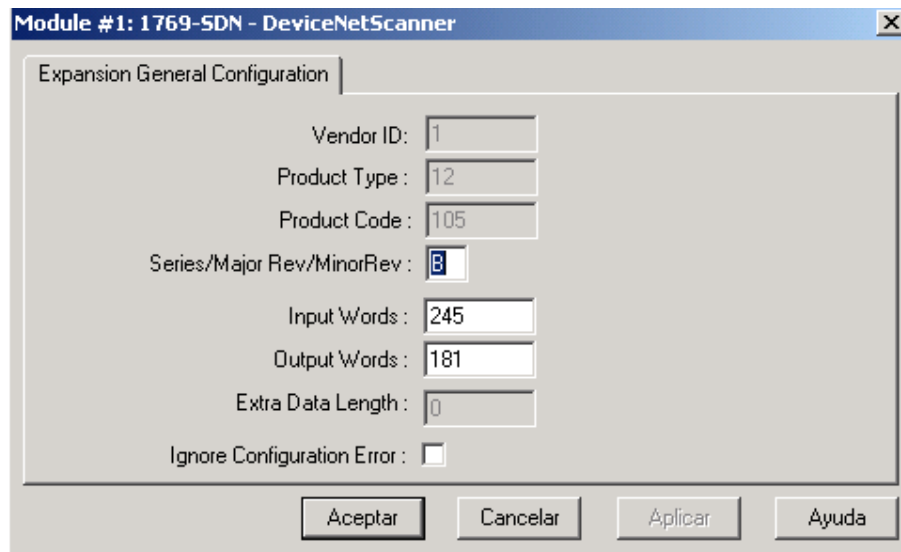
Nuevamente en la ventana de lectura de entradas y salidas presione el botón *Read I/O Config*. Si el procedimiento fue correcto, debe aparecer una ventana como la ilustrada en la figura 129. En este ejemplo se ha identificado el scanner DeviceNet y un módulo de entradas y salidas analógicas

Fig. 129. Módulos conectados en las ranuras del PLC



Ahora dé doble click sobre DeviceNet Scanner, configure el número de Words de entradas y salidas como se indica la figura 130. El scanner soporta 181 palabras de salida y 245 de entrada...véase anexo B...

Fig. 130. Configuración general del modulo scanner (Word I/O)

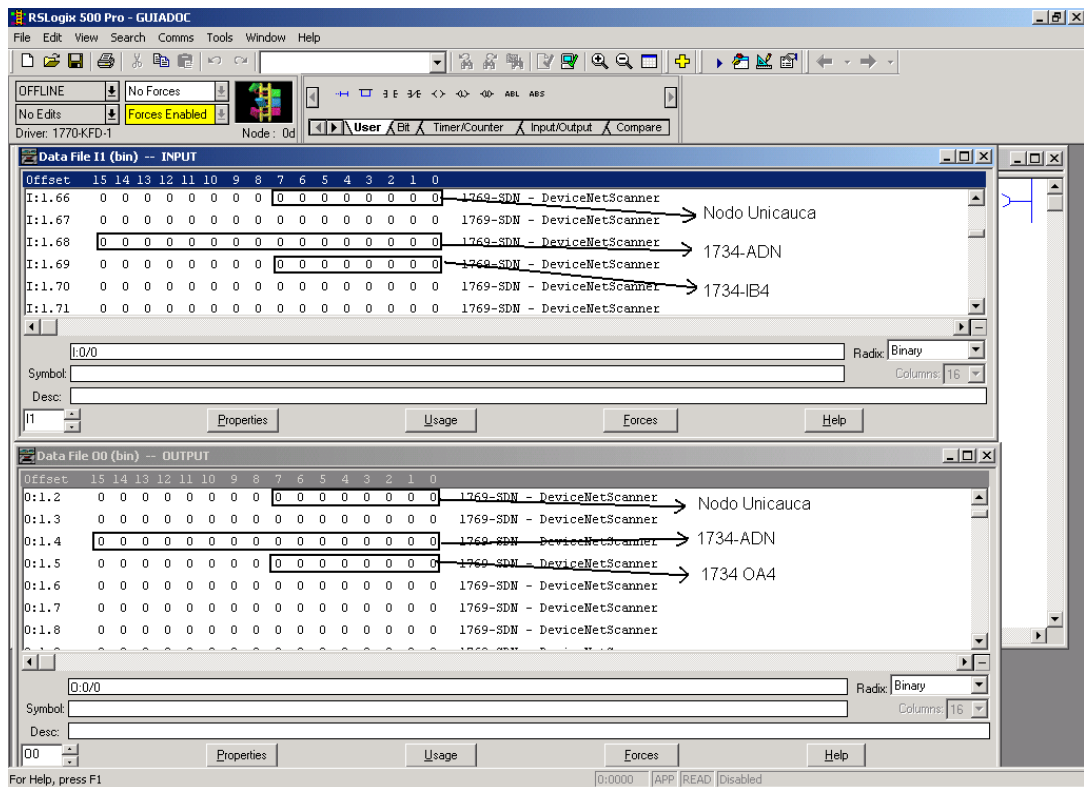


De clic sobre el botón Aceptar y cierre la ventana de configuración de I/O. Después de configurar los periféricos del PLC, es posible disponer de las entradas y salidas de de los dispositivos ubicados en la red directamente como si se tratara de un modulo adicional ubicado en alguno de los slots del PLC y están disponibles en RSLogix como O-Output y I-Input. Las imágenes de entrada y salida del scanner véase la figura 124 ahora hacen parte de las imágenes del procesador Micrologix. Se debe tener en cuenta que las salidas y entradas mapeadas en RSNetworkx deben estar en la misma posición en RSLogix...véase la figura 131...

Las salidas y entradas de los dispositivos se pueden mapear manualmente desde RSNetworkx.

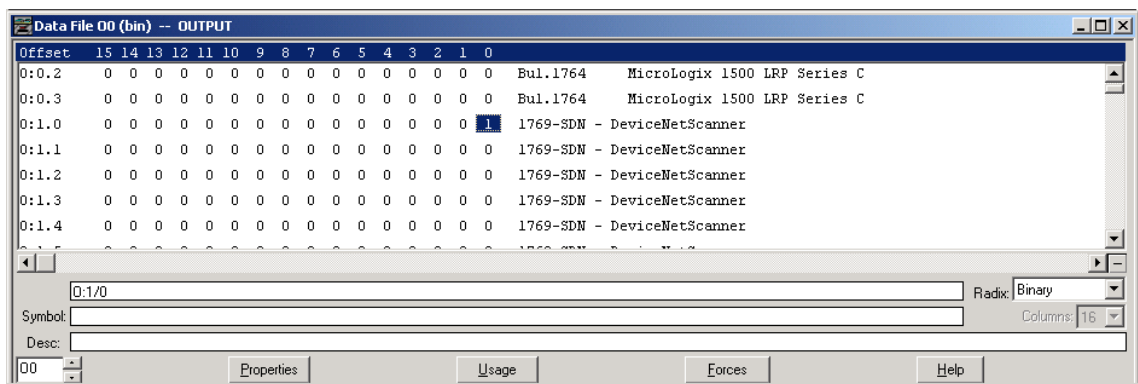
La figura 131 muestra la posición de las imágenes de entrada y salida del scanner DeviceNet, se puede evidenciar la posición de las entradas y salidas del nodo Unicauca y del ADN.

Fig. 131. Imágenes de entrada y salida mapeadas en el controlador



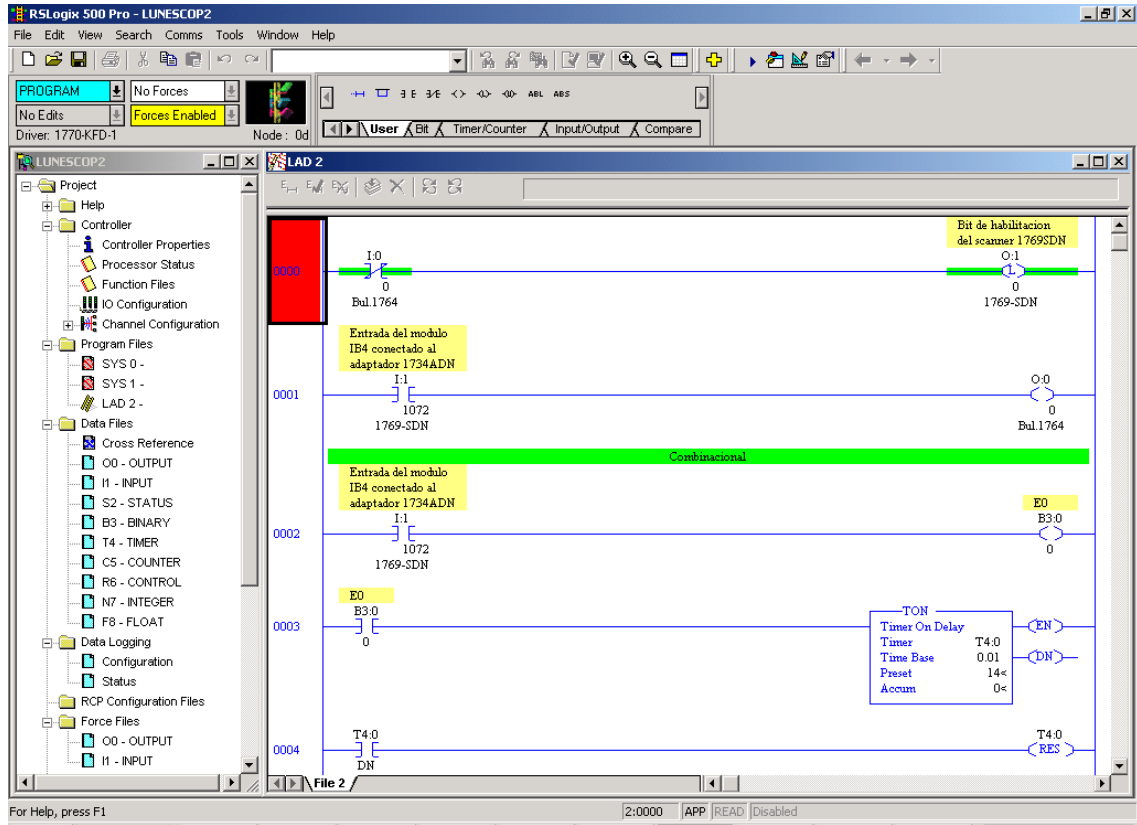
De aquí en adelante se puede iniciar el proceso de creación del ladder logic. El programa debe comenzar activando el bit 0 de la primer Word de salidas donde esta mapeado el scanner como indica la figura 132; esto es necesario para inicializar el scanner.

Fig. 132. Inicializando el Scanner



Las demás salidas y entradas del scanner se consideran como salidas y entradas convencionales (discretas) en el Ladder Logic...véase la figura 133...

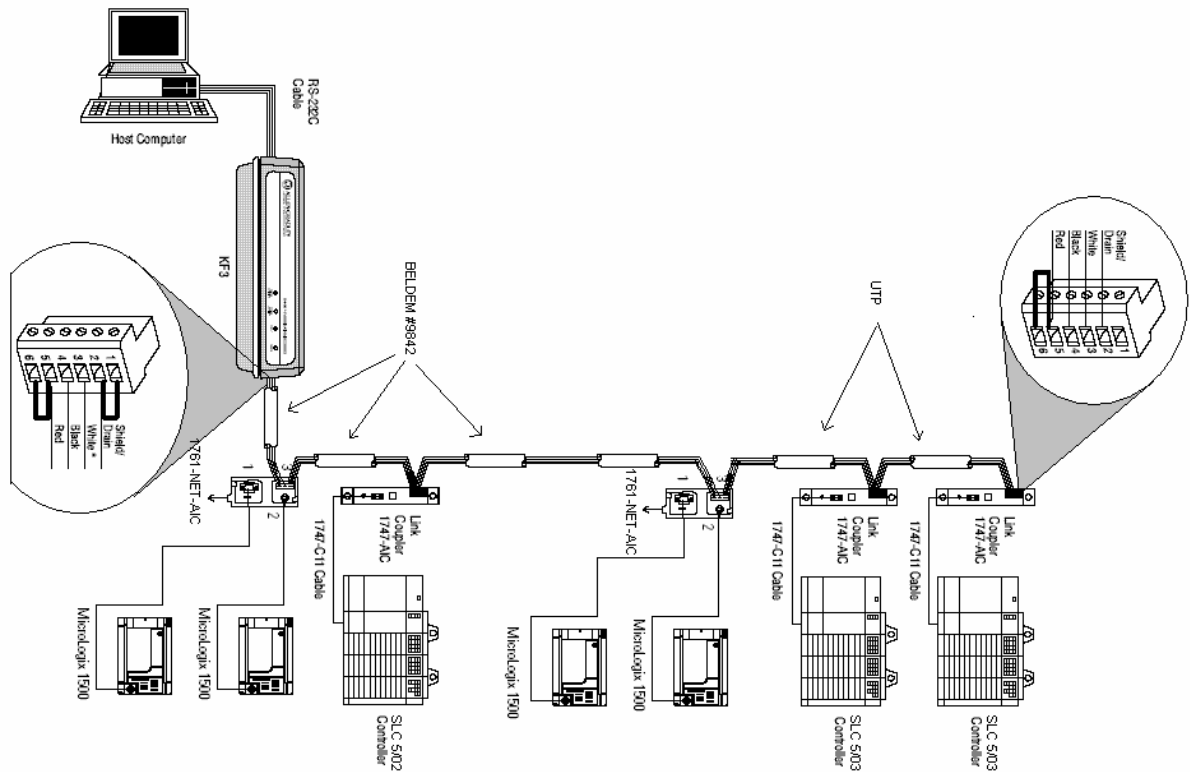
Fig. 133. Ladder Logic



ANEXO F MANUAL DE CONFIGURACIÓN DE LA RED DH485

En este documento se resume la configuración de la red DH-485 de los laboratorios del PIAI de la universidad del Cauca, la red esta conformada inicialmente por de 3 PLCs de la familia SLC500 y 4 PLCs de la familia Micrologix ambos de Allen Bradley. La figura 134 muestra la arquitectura física de la red de PLCs DH485 instalada en la Universidad del Cauca

Fig. 134. Arquitectura física de la red DH485



F.1. LISTADO DE ELEMENTOS NECESARIOS

Tabla 20. Elementos hardware de la red

Q	Nombre	Referencia
1	Interfaz de comunicación RS232/DH485	1770 KF3
2	Acopladores de enlace aislado + Conversor de interfaz avanzada	1761 NET AIC
3	Acopladores de enlace aislado	1747-AIC
2	Controladores Programables SLC500	SLC 5/03, 5/02
4	Controlador Programable Micrologix	Micrologix 1500
	Cable Beldem9842	
	Cable UTP	
5	Conectores RS-485	
1	Interfaz de comunicación DH485/RS232	1747-PIC

Tabla 21. Elementos Software

descripción	Referencia
Software de configuración de comunicaciones	RSLinx 2.52
Software de programación de PLCs de la familia Micrologix	RSLogix 500 pro
Software para ver el ladder de los PLCs Online	RSLader

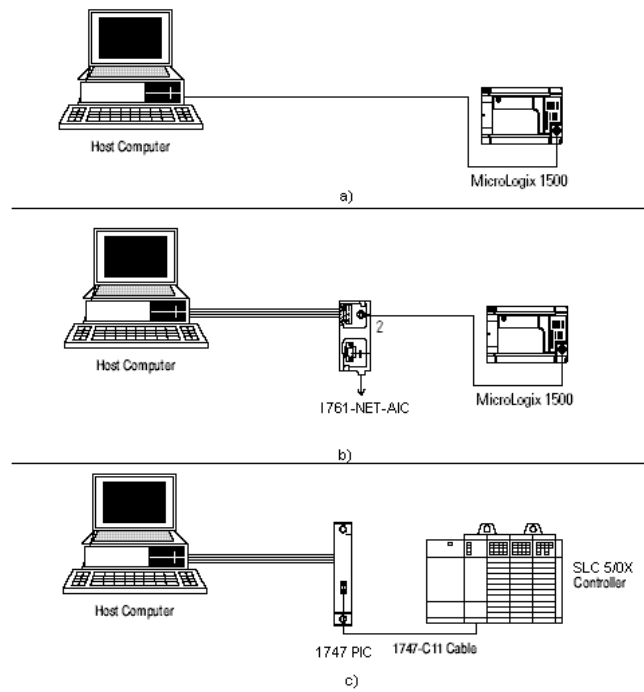
Las tablas 20 y 21 listan las herramientas hardware y software que conforman la red DH485

F.1.1. Configuración de la red DH485

Asignación de direcciones a cada nodo la red DH485 asigna automáticamente la dirección cero (0) a cada nodo que se adicione a la red, es por eso que para evitar duplicaciones de dirección se debe adicionar una dirección diferente de cero a cada nodo integrado a la red. La dirección se asigna desde RSlogix 500 desde el menú Chanel configuration. En el árbol de herramientas ubicado a la derecha de RSLogix 500.

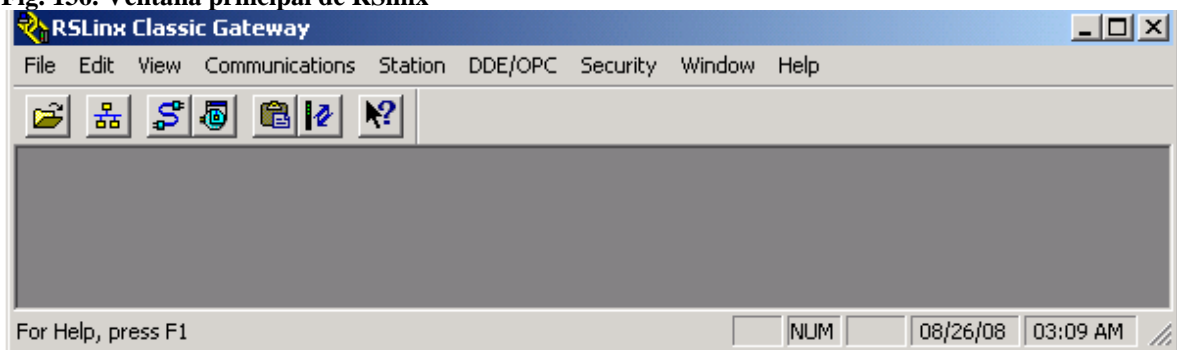
1. Conecte el primer PLC al computador mediante cualquiera de las configuraciones indicadas en la Figura 135 o a través de la red DH485 siempre y cuando no exista un PLC con la dirección cero.
2. Configure el driver de comunicaciones indicado para la comunicación punto a punto mediante el RSLinx.


Fig. 135. Opciones de conexión punto a punto entre los PLCs y el PC. a) Empleando un AIC b) Conexión directa c) Conexión mediante PIC 1747



3. Ejecute el programa Rslinx que se encuentra en Inicio>Programas>Rockwell Software>RsLinx>Rslinx, apareciendo la ventana indicada en la figura 136

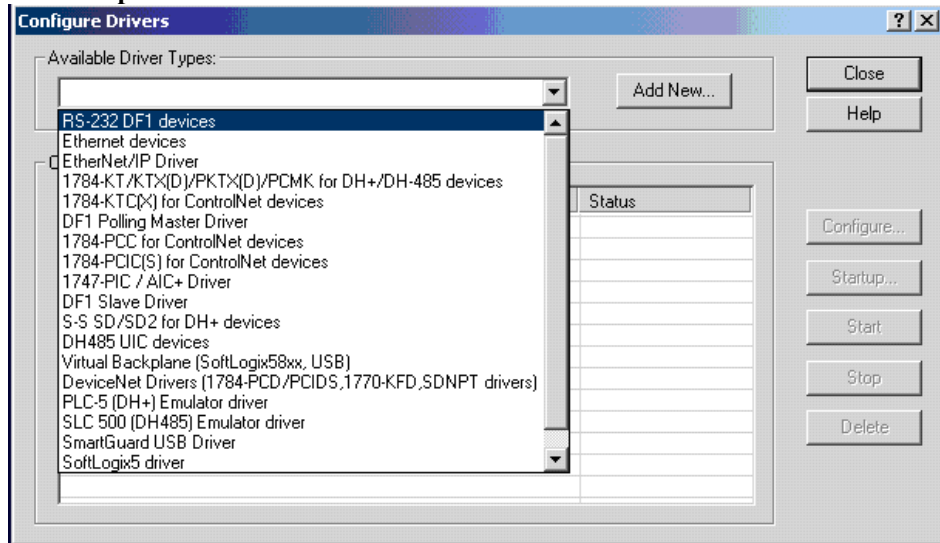
Fig. 136. Ventana principal de RSLinx



4. En la ventana RSLinx de click en el botón Configure Devices . De la figura 136. Entonces se abre la ventana de la figura 132 donde se lista los diferentes drivers disponibles en RSLinx .

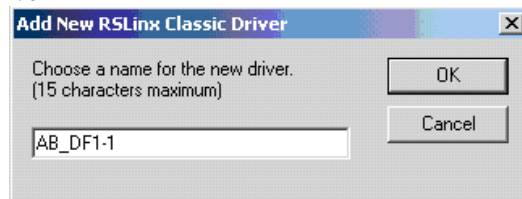
5. En la lista tipos de drives disponibles elija "RS 232 DF1 Devices", luego de click sobre el boton AddNew...véase la figura 137...

Fig. 137. Drivers disponibles



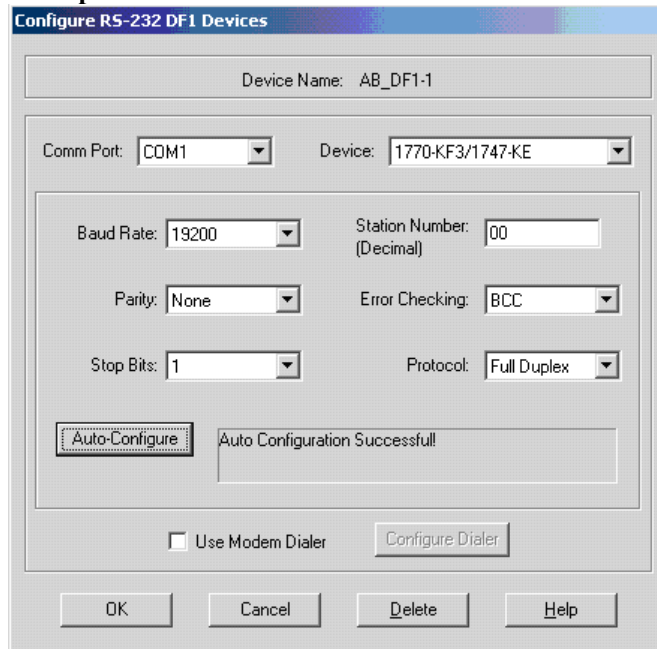
Seleccione el nombre del driver y de OK, como indica la figura 138

Fig. 138. Seleccionando el driver



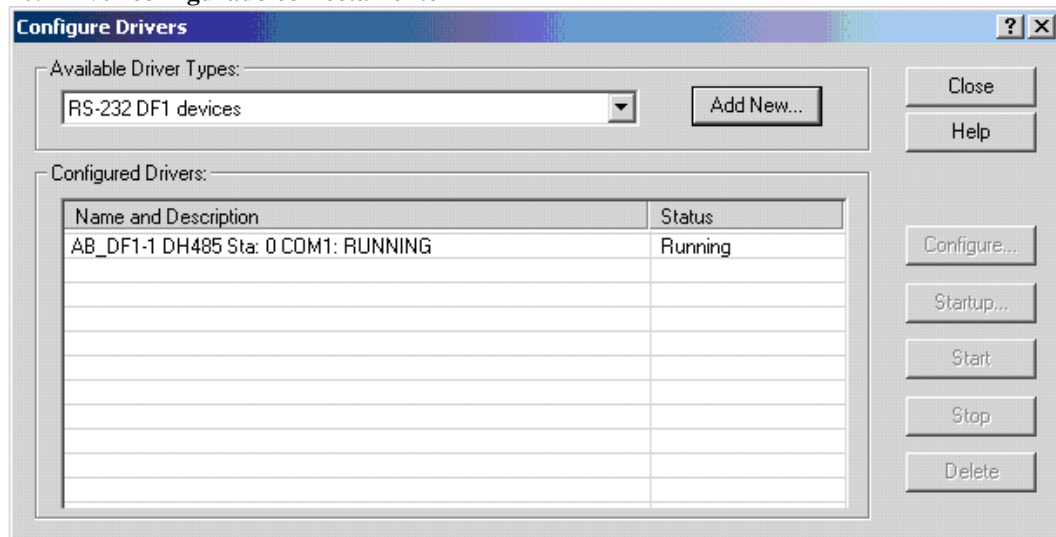
6. En la siguiente pantalla mostrada en la figura 139 configure las propiedades del driver creado: En el sub menú Device seleccione 1770 KF3/1747 KE. Posteriormente de clic sobre el botón Auto-Configure. Si las conexiones son correctas, la ventana de mensajes debe mostrar "Auto Configuration Succesfull". Finalmente de clic sobre el boton OK.

Fig. 139. Configurando el dispositivo RS232



El driver creado debe aparecer en estado Running como se indica en la siguiente figura 140.

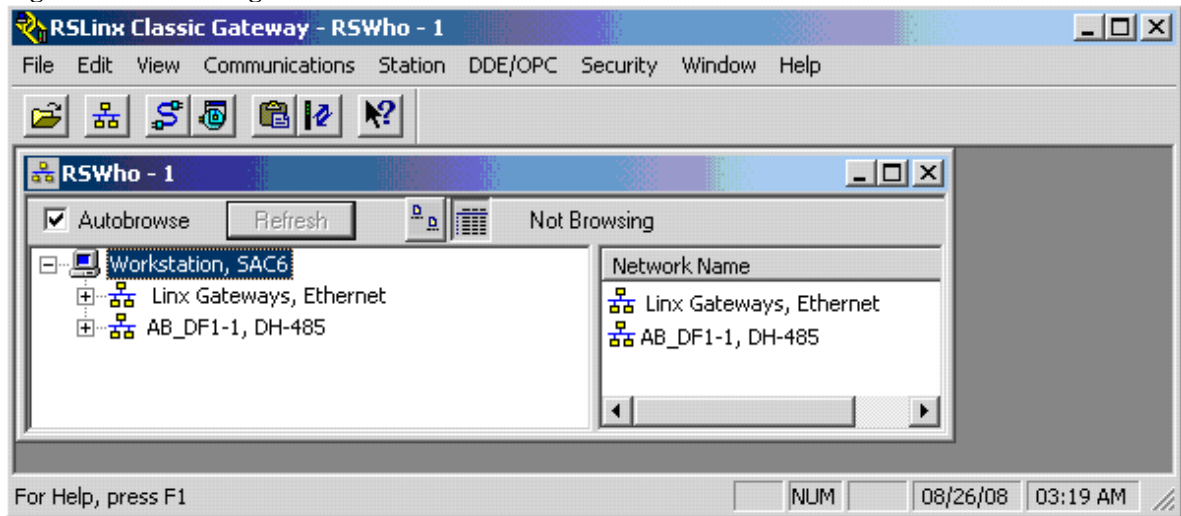
Fig. 140. Driver configurado correctamente



7. Cierre la ventana Configure Drives dando click sobre el botón Close...véase la figura 140...

8. De click sobre el boton RSWho  para comprobar si el controlador está en línea...véase la figura 141...

Fig. 141. Driver configurado correctamente



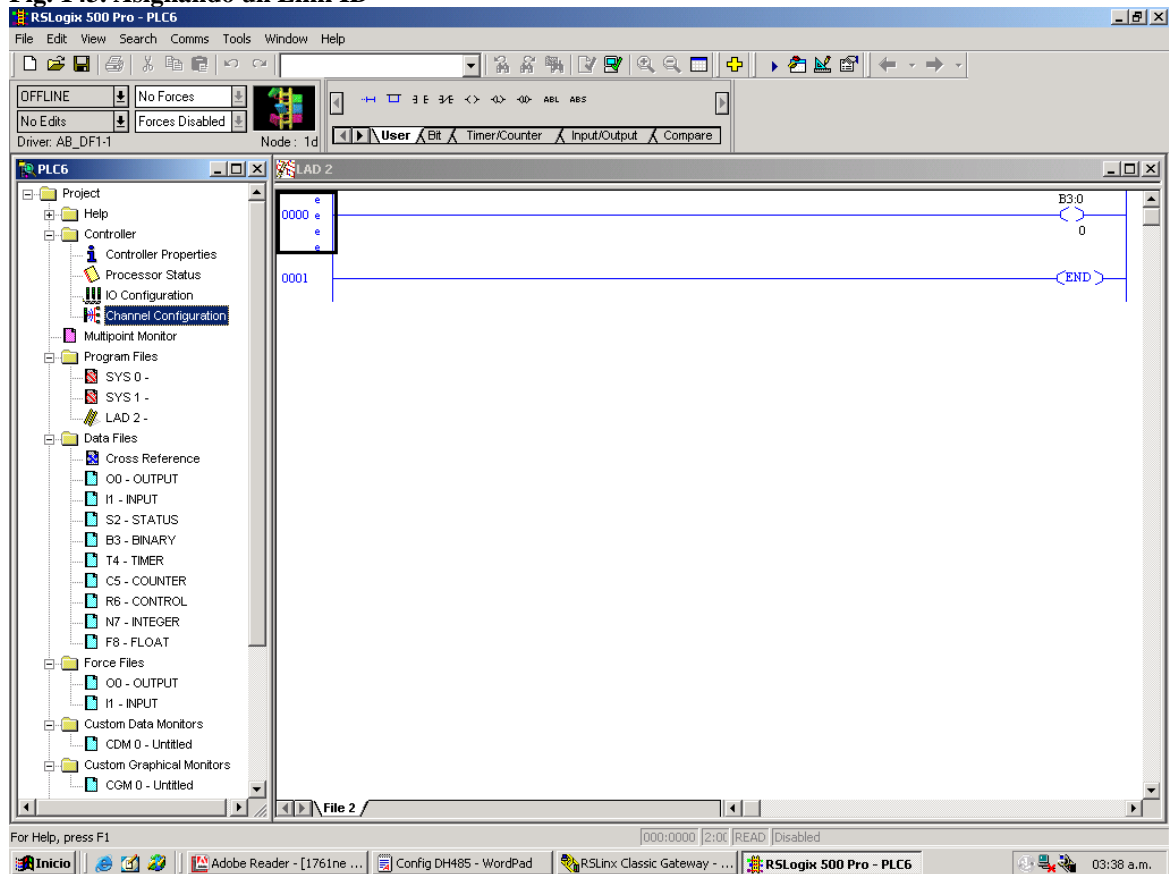
Ejecute el programa Rslgix que se encuentra en Inicio>Programas>Rockwell Software>Rslgix 500 English>Rslgix 500 English, entonces aparecerá la ventana de la figura 142.

Fig. 142. Ventana principal de RSLogix 500



La figura 143 indica como iniciar el procedimiento de cambio de link ID. Asigne la dirección del nodo al PLC mediante “Channel Configuration” mediante RSLogix. Nota: Direcciones entre 0 y 32 sin usar la dirección 0 ni la dirección 2 que esta reservada Panel Views.

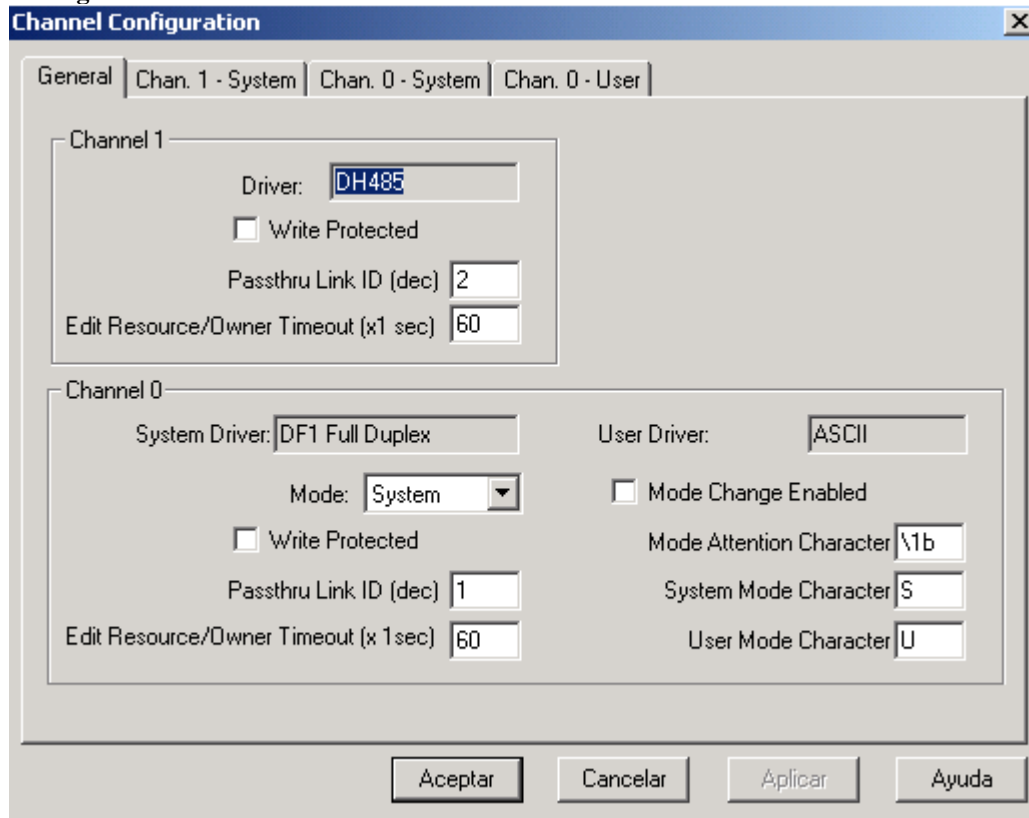
Fig. 143. Asignando un Link ID



La configuración del link ID de cada controlador se puede hacer en una conexión punto a punto o a través de la red DH485, si se configura usando la red se debe primero cambiar el Link ID a cada PLC, se recomienda inicialmente conectar un solo PLC a la red y posteriormente ir adicionando estos a medida que se va cambiando el Link ID a cada uno.

En la ventana del RSLogix 500 elija la opción "Channel Configure" y seleccione la velocidad en baudios (19200) y la dirección del nodo del PLC...vease la figura 144...

Fig. 144. Asignando un Link ID



Después de asignar una dirección diferente a cada nodo se debe configurar la red DH485, en este momento se pueden adicionar físicamente todos los PLCs a la red DH485.

Configuración de la red DH 485.


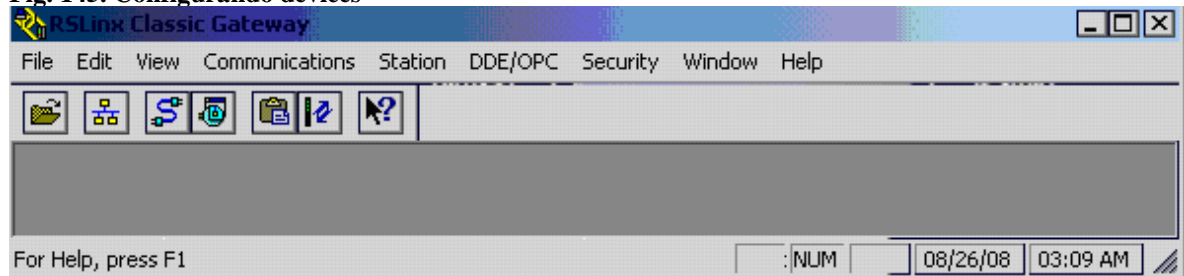
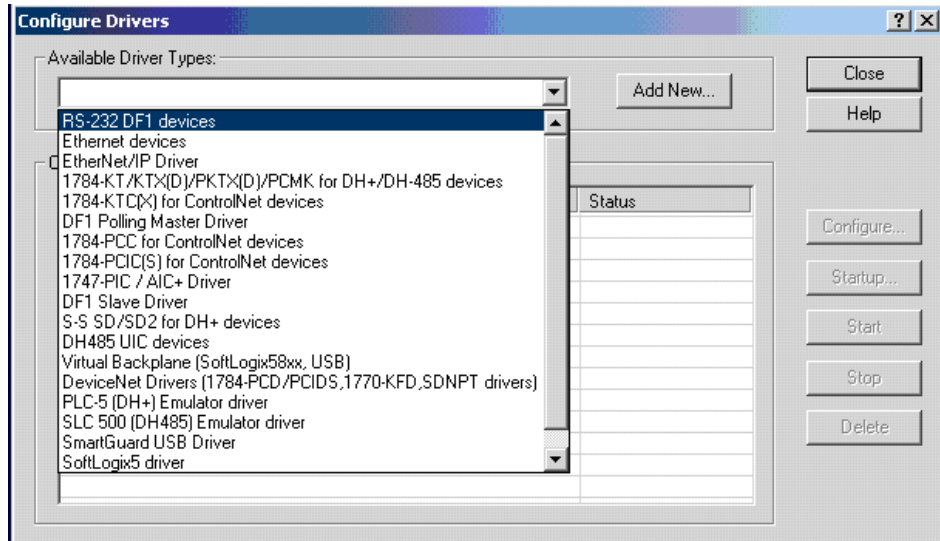
1. Configure el driver del 1770 KF3 a través de RSLinx, En la ventana RSLinx elija el botón Configure Devices . Como indica la figura 145.

Fig. 145. Configurando devices



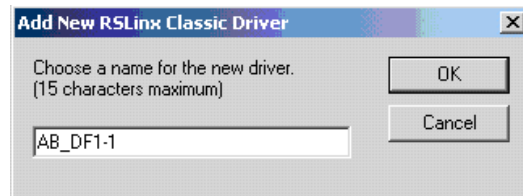
2. En la lista tipos de drives disponibles elija "RS 232 DF1 Devices", Luego de click sobre el boton AddNew...véase al figura 146...

Fig. 146. Seleccionando Devices



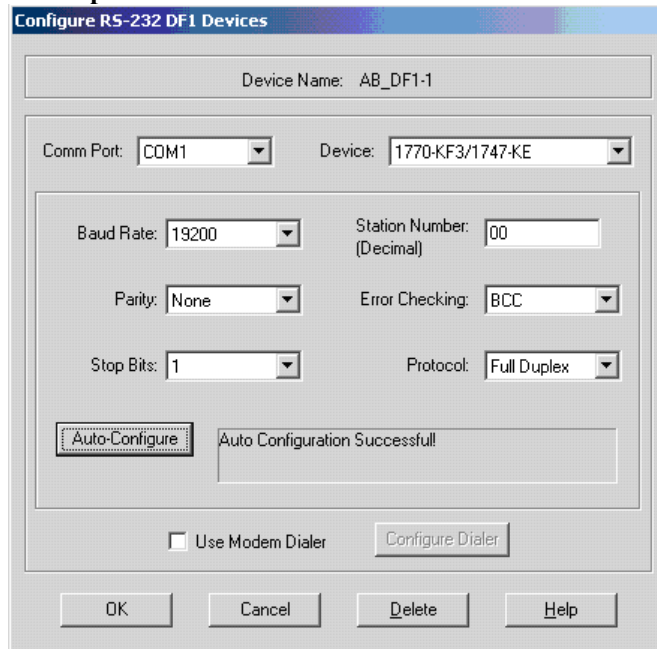
3. Seleccione el nombre del driver y de OK...ver la figura 147...

Fig. 147. Nombre del driver



4. En la pantalla siguiente mostrada en la figura 148 configure las propiedades del driver creado: En el sub menú Device seleccione 1770 KF3/1747 KE. Posteriormente de clic sobre el botón Auto-Configure.

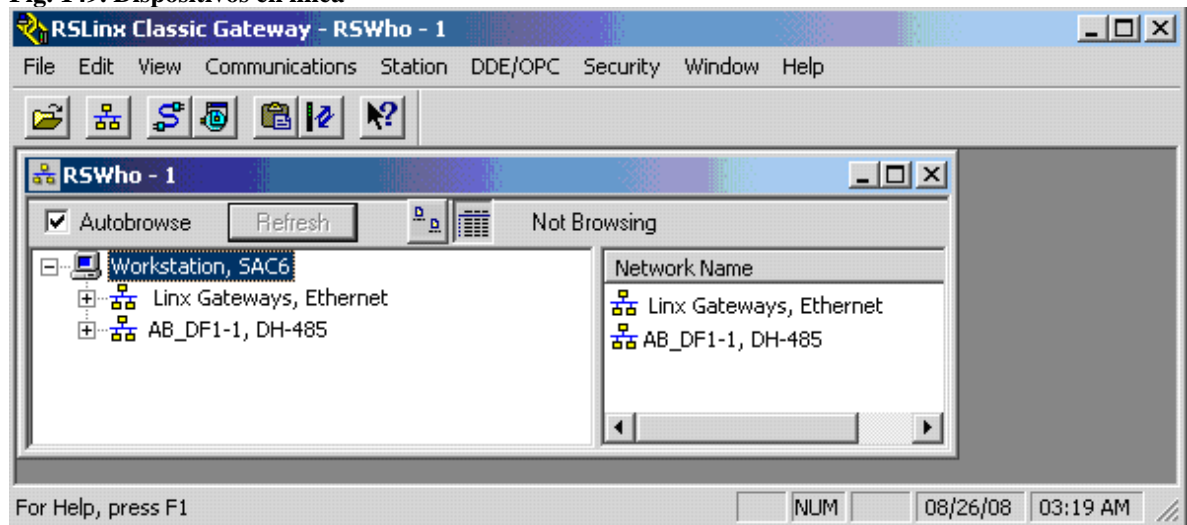
Fig. 148. Configuración de dispositivos



5. Cierre la ventana Configure Drives dando click sobre el boton Close.

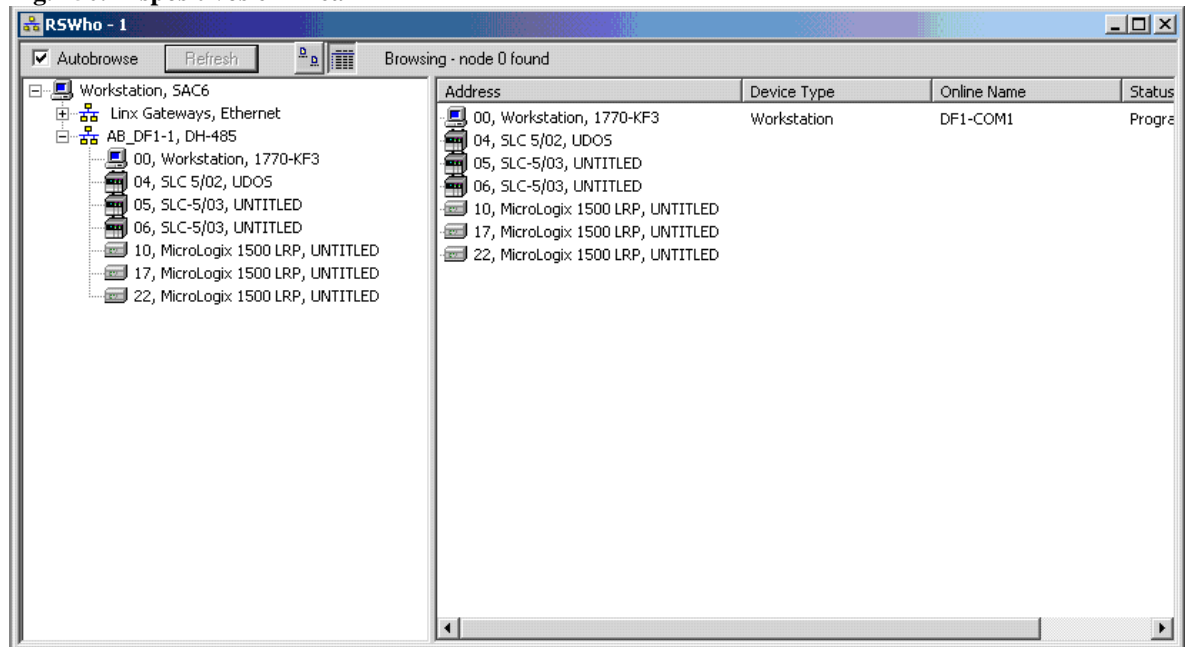
6. De click sobre el boton RSWho  como indica la figura 149 para ver los PLC conectados en red.

Fig. 149. Dispositivos en línea



Abra el árbol con el nombre del driver configurado (en este caso AB_DF-1, DH485). Donde se puede observar la siguiente pantalla mostrada en la figura 150

Fig. 150. Dispositivos en línea



F.2. NOTAS ADICIONALES

El acoplador de red aislado con número de catalogo 1747 AIC, se emplea para conectar los dispositivos de la familia SLC 500 a una red DH 485. El acoplador provee un terminal removible de 6 pines para conexión con el cable DH-485 (Cable Beldem).

Las conexiones de red para los procesadores SLC 500 son posibles mediante el cable 1747-C11 suministrado con el acoplador. Las conexiones de red con dispositivos periféricos, tales como conversores de interfaz personales (1747 PIC), Módulos de acceso a tablas de datos (1747-DTAM-E) son posibles mediante el cable 1747-C10 suministrado con cada uno de estos periféricos. Si requiere conectar uno de estos periféricos a distancias entre 1.8m y 6.1 m emplee cable 1747-C20.

Para proteger dispositivos conectados, el acoplador provee aislamiento de 1500V DC entre el cable de comunicación y el controlador SLC 500 y sus periféricos.

El acoplador también puede ser empleado para proveer conectividad entre dispositivos periféricos (PIC o DTAM) para distancias superiores a 1.8mt (máximo 1219mt). A continuación se ilustra un ejemplo de conexión remota entre un computador y el SLC500.

F.2.1. Consideraciones Hardware

Decida la longitud del cable de comunicación, en donde lo va a ubicar y como lo va a proteger del ambiente en donde será instalado. Cuando el cable este instalado, se necesita saber cuantos dispositivos se van a conectar durante la instalación y cuantos dispositivos se van a adicionar posteriormente.

Número de dispositivos y longitud del cable. Recuerde que en cada nodo debe instalar acopladores AIC. Si planea adherir nodos, reserve instalaciones durante el montaje para evita recableados cuando la red esté en operación.

La máxima longitud del cable es 1219mt (desde el primer nodo hasta el último nodo).

Planeación de rutas de cables. Como mínimo mantenga los cables de comunicación 1.52mt alejados de motores eléctricos, transformadores, rectificadores, generadores o fuentes de radiación de microondas.

Si debe extender el cable alrededor de líneas de alimentación de potencia, extienda el cable en ángulos rectos a dichas líneas.

F.2.1. Consideraciones software

- Número de nodos en la red.
- Dirección de los nodos.
- Ratas de comunicación.
- Dirección máxima de los nodos y direcciones restringidas.
- SLC 5 0x (x=3, 4, 5): factor de mantenimiento del token.
- Numero máximo de dispositivos de comunicación.

Número de nodos en la red. Afectan directamente el tiempo de transferencia de datos entre nodos. Los nodos innecesarios (como terminales de programación que no están siendo empleados) reducen la tasa de transferencia de datos. El número máximo de nodos es 32.

Configuración de las direcciones de nodos. El mejor desempeño de la red ocurre cuando las direcciones inician en 0 y son asignadas en orden secuencial. En los procesadores SLC 500 la dirección de nodo por defecto es 1. La dirección del nodo se almacena en el archivo de estado del procesador (S:15L). Los procesadores no pueden ser el nodo 0. Los iniciadores (p.ejm, el computador) deben tener las direcciones mas bajas para minimizar el tiempo requerido para inicializar la red.

Configuración de la tasa de comunicación del procesador. El mejor desempeño de la red ocurre a la máxima tasa de comunicación. Todos los dispositivos deben estar a la misma tasa. La tasa de comunicación se almacena en archivo de estado del procesador (S:15L).

Configuración de la máxima dirección de nodo. Debe ser lo mas baja posible, de esta manera se minimiza la cantidad de tiempo empleado en solicitar sucesores cuando se inicializa la red.

ANEXO G ESTÁNDAR DEVICENET

G.1. INTRODUCCIÓN

DeviceNet es un bus industrial que funciona sobre CAN, fue desarrollado originalmente por Allen-Bradley, A partir de 1995, la especificación se abrió y desde ese momento, fue mantenida por un organismo independiente llamado Open DeviceNet Vendor Association (ODVA). En su conjunto, DeviceNet puede ser visto como una red de bajo nivel que provee conexiones entre dispositivos industriales simples, como sensores y actuadores, y dispositivos de mayor nivel como por ejemplo, controladores, soportando modelos de comunicación peer-to peer y master / slave.

G.1.2. Modelo de objetos: clases y objetos

La capa de aplicación de DeviceNet es orientada a objetos. Se hace uso de un modelo abstracto de objetos para describir el conjunto de servicios de comunicación disponible, el comportamiento visible en forma externa de un dispositivo DeviceNet (por ejemplo, los servicios que ofrece un sensor o un actuador) y como medio común para acceder e intercambiar información. Este modelo se construye en base a dos entidades:

Clases una clase es un conjunto de objetos que representan el mismo tipo de componente con los mismos atributos, servicios y con un comportamiento similar. Una clase puede exponer por sí sola atributos y ofrecer servicios.

Objetos un objeto o instancia es la representación efectiva de un objeto dentro de una clase. Una instancia tiene atributos (cuyos valores determinan el estado del objeto), provee servicios (que son invocados para disparar alguna acción del objeto) e implementa un comportamiento (que indica como responder a eventos particulares). Un dispositivo se modela como una colección abstracta de objetos que luego, se mapea al dispositivo real

en una forma dependiente de la implementación usada. O sea, desde el punto de vista de DeviceNet, cada dispositivo está formado por una serie de objetos los cuales, al ser abstractos, no necesariamente se corresponden con objetos físicos de la implementación. Para acceder a estos objetos abstractos, DeviceNet define un esquema de direccionamiento que se basa en la asignación de identificadores a cada elemento (clases y objetos). Esto resulta en un esquema de direccionamiento no plano con los siguientes niveles de indirección:

MAC ID entero único asignado a cada nodo en cada red. Es similar a una dirección de red, asignada manualmente por los usuarios. La intervención humana hace factible que se presenten errores y que haya duplicados. DeviceNet define un protocolo para identificar direcciones repetidas y evitar un mal comportamiento del bus. Los valores posibles son de 0 a 63, utilizando 6 bits

Class ID entero único asignado a cada clase. Se pueden usar 1 o 2 bytes para guardar este valor. El tamaño es configurable. La especificación reserva una serie de identificadores de clase para su propia biblioteca de objetos.

Instance ID entero asignado a cada instancia, único dentro del espacio de direcciones MAC ID:Class ID. Se pueden usar 1 o 2 bytes para guardar este valor. El tamaño es configurable. Entonces, para poder acceder a un objeto en particular es necesario conocer, en primer lugar, la MAC ID del dispositivo donde corre. Luego la clase o sea, su Class ID y por último, el identificador de instancia Instance ID. Al conjunto de identificadores también puede llamárselo path del objeto. DeviceNet permite solicitar servicios o acceder a atributos de objetos y clases mediante un protocolo de mensajería explícita. Igualmente el acceso a un servicio o atributo se basa en el uso de un identificador único para el mismo. Así, se definen dos nuevos identificadores:

Attribute ID entero asignado a cada atributo de clase o instancia.

Service Code identificador de un método de instancia o clase. Los valores de códigos de servicio son de 0 a 127, consumiendo 7 bits. Existen una serie de servicios comunes, definidos en el estándar, que deben ser implementados por los objetos de todas las

clases. Para esto, se reservan los identificadores de 0x00 a 0x31 para los DeviceNet Common Services, asignados de la siguiente forma...véase la tabla 22...

Tabla 22. Service Codes de los DeviceNet Common Services

Código	Nombre del servicio	Descripción Funcional
0x00	Reservado	
0x01	Get_Attributes_All	Devuelve el contenido de todos los atributos de un objeto o de una clase.
0x02	Set_Attributes_All Request	Modifica el contenido de los atributos de un objeto o de una clase.
0x03-0x04	Reservado	
0x05	Reset	Depende del objeto o de la clase pero típicamente se va a pasar a un estado / modo inicial.
0x06	Start	Depende del objeto o de la clase pero típicamente se va a pasar a un estado / modo running.
0x07	Stop	Depende del objeto o de la clase pero típicamente se va a pasar a un estado / modo stopped/idle.
0x08	Create	Creación de una nueva instancia de la clase correspondiente.
0x09	Delete	Elimina una instancia de la clase especificada.
0x0A-0x0C	Reservado	
0x0D	Apply_Attributes	Hace que los valores de atributos cuyo uso está pendiente pasen a ser usados activamente.
0x0E	Get_Attribute_Single	Devuelve el contenido de un atributo específico.
0x0F	Reservado	
0x10	Set_Attribute_Single	Modifica el contenido de un atributo específico.

0x11	Find_Next_Object_Instance	Únicamente soportado a nivel de clase. Hace que la clase busque sus instancias y devuelva una lista de Instance IDs.
0x12-0x13 Reservado		
0x14	Error Response	Se usa para responder negativamente a un explicit request message recibido previamente.
0x15	Restore	Restaura los contenidos de los atributos de una clase / instancia de un almacenamiento accesible por el servicio Save (0x16).
0x16	Save	Copia los contenidos de los atributos de una clase / instancia a un almacenamiento accesible por el servicio Restore (0x15).
0x17	No Operation (NOP)	Hace que la clase / instancia destinataria simplemente conteste con un response.
0x18	Get Member	Devuelve información de un miembro de un atributo (que debe ser un array de un tipo básico).
0x19	Set Member	Guarda información de un miembro en un atributo (que debe ser un array de un tipo básico).
0x1A	Insert Member	Inserta un miembro en un atributo (que debe ser un array de un tipo básico).
0x1B	Remove Member	Elimina un miembro en un atributo (que debe ser un array de un tipo básico).
0x1C-0x31 Reservado		

Entonces, para poder solicitar un servicio a un objeto, operación similar a una invocación remota de método en otras tecnologías, es necesario conocer el Service Code del mismo una vez que fue ubicado el objeto.

Los posibles valores de cada uno de los identificadores antes definidos, principalmente Class ID y los Service Codes, se dividen en un rango abierto, llamado Open, otro específico del fabricante, llamado Vendor Specific, y por último, uno específico por clase, llamado Object Class Specific.

El primer rango, tiene significado definido por la ODVA y común a todos los participantes de dicho consorcio. O sea, todos están de acuerdo que el Service Code 0x0D corresponde al servicio Apply_Attributes. El segundo rango está reservado para el fabricante del dispositivo el cual, puede usar los identificadores dentro del mismo como considere más conveniente (normalmente para servicios específicos del dispositivo). Por último, el tercer rango es definido por cada clase y se aplica sólo para Service Codes.

G.1.3. Modelo de objetos: perfiles de dispositivo

Para mantener la interoperabilidad y promover la intercambiabilidad de dispositivos similares, tiene que existir cierta consistencia entre estos. O sea, dispositivos similares deberían tener el mismo comportamiento, producir y consumir el mismo conjunto básico de datos de I/O, tener los mismos atributos configurables, etc. En DeviceNet, la definición de estas características comunes se llama perfil de dispositivo (o por su nombre en inglés, device profile). Existen una serie de perfiles estándar que pueden ser adoptados o extendidos para nuevos dispositivos. A continuación se incluye una lista de los perfiles predefinidos listados en la tabla 23 [DevNet2 99]:

Tabla 23. Perfiles estándar de dispositivo definidos por DeviceNet

Tipo de Dispositivo	Profile ID
AC Drives	0x02
Barcode Scanner	0x0D
Communications Adapter	0x0C
Contactora	0x15
Control Station	0x01
DC Drives	0x13
Encoder	0x08
General Purpose Analog I/O	0x0A
General Purpose Discrete I/O	0x07
Generic Device	0x00
Human-Machine Interface	0x18
Inductive Proximity Switch	0x05
Limit Switch	0x04
Mass Flow Controller	0x1A
Message Display	0x12
Motor Overload	0x03
Motor Starter	0x16
Photoelectric Sensor	0x06
Pneumatic Valve	0x19
Position Controller	0x10
Resolver	0x09
Servo Drives	0x14
Soft Start	0x17
Weigh Scale	0x11

Existe un esquema de numeración para los perfiles de dispositivo. Puede haber perfiles públicos o propietarios no publicados. ODVA define los siguientes rangos...ver tabla24...

Tabla 24. Rangos de numeración de perfiles de dispositivo

Tipo	Rango	Cantidad de Perfiles
Público	0x0000-0x0063	100
Específico del fabricante	0x0064-0x00C7	100
Reservado	0x00C8-0x00FF	56
Público	0x0100-0x02FF	512
Específico del fabricante	0x0300-0x04FF	512
Reservado	0x0500-0xFFFF	64256

Un perfil de un dispositivo está formado por los siguientes elementos:

Modelo de Objetos cada producto DeviceNet contiene múltiples objetos que interactúan para proveer el comportamiento básico del dispositivo. Como el comportamiento de cada objeto individual es fijo, un mismo agrupamiento de objetos relacionado de igual forma, debería producir el mismo comportamiento. Ese agrupamiento es el modelo de objetos del dispositivo. Puede haber objetos requeridos y opcionales para los dispositivos de un perfil. Algunos objetos requeridos en cualquier perfil son los que modelan la interface DeviceNet (como Connection, DeviceNet, Identity y Message Router). Los objetos opcionales pueden formar parte de la DeviceNet Object Library o bien, ser definidos por el diseñador del dispositivo. El modelo debe incluir:

- Lista de todos los objetos presentes en el dispositivo.
- Cantidad máxima de instancias para cada una de las clases.
- Descripción de la forma en que los objetos afectan el comportamiento del dispositivo.
- Interfaces de todos los objetos.

Formato de Datos de I/O es una especificación de cómo el dispositivo se comunica en una red DeviceNet, incluyendo una especificación exacta del formato de datos a usar en las conexiones con formato libre (conexiones de I/O).

Se debe incluir la definición de todos los I/O assemblies (agrupamientos de datos relacionados) usados por el dispositivo, el formato de los mismos y un mapeo entre los miembros de cada assembly y los atributos de los objetos. Por ejemplo, una posible especificación de formato se puede observar en la tabla 25.

Tabla 25. Ejemplo de assembly de datos de I/O

Byte	7	6	5	4	3	2	1	0
0	Flow Value LSB							
1	Flow Value MSB							
2	Reservado					Diag #1	Diag #2	Diag #3

Un posible mapeo entre este assembly y objetos esta listada en la tabla26

Tabla 26. Mapeo entre assembly de datos de I/O y objeto

Nombre del Componente	Clase	Nro. Instancia	Atributo	Tipo Dato
	Nombre	Número	Nombre	Número
Flow Value	Low Pass Filter	0x63	Filter Output	0x03 UINT
Diag #1	Diagnostic	0x62	Wire Off Detect	0x07 BOOL
Diag #2	Diagnostic	0x62	Reverse Flow	0x08 BOOL
Diag #3	Diagnostic	0x62	Transducer Failure	0x09 BOOL

Configuración del Dispositivo definición de los parámetros configurables y la interfase pública de los mismos. Dispositivos similares deben comportarse igual por lo cual, deberían tener los mismos parámetros de configuración (al menos, los básicos ya que puede haber extensiones del fabricante).

Se debe incluir la siguiente información:

- Nombre del parámetro, path o dirección del atributo (Class ID, Instance ID, Attribute ID), tipo de dato, unidades y por último, valores mínimo, máximo y por defecto.
- Efecto de cada parámetro en el comportamiento del objeto.

- Grupos de parámetros si existen definidos algunos.
- Interface pública para configuración.

G.1.4. Modelo de comunicaciones

DeviceNet define sobre CAN un modelo de comunicaciones con conexión. Por el contrario, CAN ofrece un servicio sin conexión en la capa data-link, al ofrecer un servicio de LLC tipo 3 [Thomesse 99]. Las conexiones DeviceNet proveen, por lo tanto, un “camino de comunicación” entre múltiples end-points. Estos end-points son aplicaciones que necesitan compartir datos.

DeviceNet define dos tipos de conexiones en base a la funcionalidad que ofrecen y el uso que las aplicaciones van a hacer de las mismas:

Conexiones I/O (I/O Connections) Son conexiones dedicadas de propósito específico. Se establecen entre dispositivos productores y uno o más consumidores. No se imponen restricciones sobre el formato de los mensajes cuyo significado es transparente para DeviceNet. O sea, sólo lo conocen los dispositivos que intercambian estos mensajes. Este tipo de conexiones permite implementar modelos de comunicación master / slave.

Conexiones de Mensajería Explícita (Explicit Messaging Connections) Son conexiones genéricas de propósito múltiple. Se establecen entre dos dispositivos que intercambian mensajes cuyo formato está especificado por el estándar DeviceNet. Los mensajes se intercambian con una modalidad request / response que permite implementar modelos de comunicación peer-to-peer. Este tipo de conexiones son las usadas para solicitar servicios de objetos o bien, para acceder a los atributos de los mismos.

Las conexiones también pueden clasificarse dependiendo del mecanismo que se usó para su creación. Existen dos posibilidades:

Conexiones Dinámicas Se definen un conjunto de mensajes especiales que pueden ser intercambiados entre dispositivos sin necesidad de una conexión DeviceNet entre ellos.

Estos mensajes son los Unconnected Explicit Messages y permiten establecer conexiones de mensajería explícita.

Conexiones Predefinidas El estándar de DeviceNet define un conjunto de conexiones predefinidas (Predefined master / slave Connection Set) que facilitan las comunicaciones típicamente vistas en relaciones master / slave. Así, los dispositivos pueden evitar incluir el hardware necesario para manejo de conexiones dinámicas (registro maskand- match en el adaptador CAN).

Cada conexión entre end-points, se identifica mediante un Connection ID o CID asignado en el momento del establecimiento de la misma. Si la conexión involucra un intercambio bidireccional, entonces se asignan dos identificadores. Estos identificadores tienen que permitir mantener separados los mensajes de cada conexión. Además, para minimizar la carga de trabajo sobre el controlador de un dispositivo, su adaptador CAN debería interrumpirlo únicamente cuando llegan mensajes de conexiones existentes en el dispositivo en cuestión. Teniendo en cuenta el funcionamiento del bus CAN y los requerimientos anteriores, los CID deberían corresponderse con el valor del campo de arbitración de los frames CAN usados para intercambiar mensajes DeviceNet.

Así, el estándar DeviceNet particiona el espacio de posibles valores del campo de arbitración CAN, definiendo cuatro grupos de mensajes con características funcionales distintas y con distintos rangos de CIDs. Estos grupos son:

Tabla 27. Grupos de mensajes en DeviceNet

Bits de Identificación											Rango	Utilización	
10	9	8	7	6	5	4	3	2	1	0			
0		Group 1 MsgId				Source MAC ID					0x000-0x3FF	Grupo de mensajes 1	
1	0	MAC ID					Group 2 MsgId				0x400-0x5FF	Grupo de mensajes 2	
1	1	Group 3 MsgId			Source MAC ID					0x600-0x7BF	Grupo de mensajes 3		
1	1	1	1	1	Group 4 MsgId						0x7C0-0x7EF	Grupo de mensajes 4	
1	1	1	1	1	1	1	1	X	X	X	X	0x7F0-0x7FF	Identificadores CAN inválidos

Según la tabla 27, el campo de arbitración puede estar formado por los siguientes componentes:

Message ID identificador dentro del grupo de mensajes correspondiente. Este campo permite establecer más de una conexión dentro de un mismo end-point.

Source MAC ID MAC ID del nodo transmisor.

Destination MAC ID MAC ID del nodo receptor.

Tanto las conexiones de mensajería explícita como las de I/O pueden usar identificadores pertenecientes a los grupos 1, 2 o 3. Se debe tener en cuenta que estos grupos tienen distintas prioridades en el mecanismo de arbitración de CAN. Se debería considerar este aspecto al momento de diseñar las aplicaciones y de establecer conexiones en forma dinámica.

Este esquema de grupos de mensajes da la siguiente distribución de identificadores CAN para cada dispositivo [Etschberger 97]:

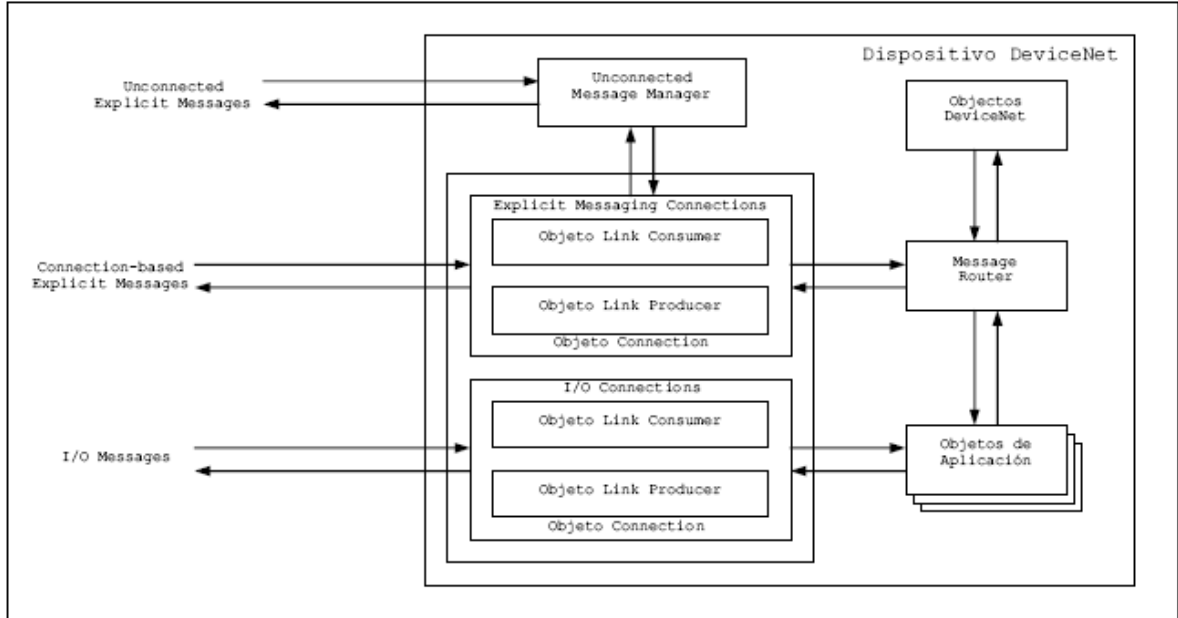
Tabla 28. Utilización de valores de arbitración en DeviceNet

CAN Id.	Utilización	Cantidad
0-63	Grupo 1 - Message ID 0 - Disp. 0 a 63	64
64-127	Grupo 1 - Message ID 1 - Disp. 0 a 63	64
...
960-1023	Grupo 1 - Message ID 15 - Disp. 0 a 63	64
Total Grupo 1: 1024 mensajes, 16 por dispositivo		
1024-1031	Grupo 2 - Disp. 0 - Message ID 0 a 7	8
1032-1039	Grupo 2 - Disp. 1 - Message ID 0 a 7	8
...
1528-1535	Grupo 2 - Disp. 63 - Message ID 0 a 7	8
Total Grupo 2: 512 mensajes, 8 por dispositivo		
1536-1599	Grupo 3 - Message ID 0 - Disp. 0 a 63	64
1600-1663	Grupo 3 - Message ID 1 - Disp. 0 a 63	64
...
1920-1983	Grupo 3 - Message ID 7 - Disp. 0 a 63	64
Total Grupo 3: 512 mensajes, 8 por dispositivo		
1984	Grupo 4 - Message ID 0	1
...
2031	Grupo 4 - Message ID 63	1
Total Grupo 4: 64 mensajes		

La especificación de CAN no define como se puede pasar un nodo en el estado BUS OFF al estado ERROR ACTIVE. DeviceNet define esta transición mediante el uso de un conjunto de conexiones específico llamado Offline Connection Set. Los mensajes del grupo 4 están reservados para estas conexiones.

El siguiente esquema describe el funcionamiento del modelo antes descrito en un dispositivo DeviceNet. Será usado para describir el establecimiento de conexiones (dinámicas o predefinidas) y el protocolo de mensajería explícita.

Fig. 151. Modelo de comunicaciones en un dispositivo DeviceNet



G.1.5. Capa de transporte.

Establecimiento dinámico de conexiones, DeviceNet implementa funciones típicas de la capa de transporte en el modelo de referencia ISO/OSI. En primer lugar, se describirá el mecanismo de establecimiento de conexiones dinámicas, el cual presenta diferencias dependiendo del tipo de conexión.

Conexiones de Mensajería Explícita Para solicitar el establecimiento de una de estas conexiones punto a punto se usan mensajes del grupo 3 con Message ID en 0x06,. Las respuestas a estas solicitudes se transmiten con mensajes del grupo 3 con Message ID en 0x05.

En ambos casos, se trata de mensajes sin una conexión asociada o Unconnected Explicit Messages. Estos son procesados por el Unconnected Message Manager (UCMM), mostrado en la figura 151. Para soportar este objeto DeviceNet, el adaptador del bus CAN del dispositivo debe permitir configurar su registro mask-andmatch para que tome mensajes del grupo 3 con Message ID en 0x05 o 0x06.

Los únicos servicios que se pueden solicitar con mensajes del grupo 3 con Message ID en 0x06 son:

- Open Explicit Messaging Connection Request
- Close Connection Request

De igual forma, los únicos mensajes que se pueden transmitir con un valor de arbitración correspondiente al grupo 3 con Message ID en 0x05 son:

- Open Explicit Messaging Connection Response
- Close Connection Response
- Error Response
- Device Heartbeat Message
- Device Shutdown Message

Conexiones I/O. Estas conexiones, que pueden ser punto a punto o multicast, se establecen en forma dinámica interactuando con el objeto Connection a través de una conexión de mensajería explícita previamente establecida.

Los pasos a seguir son, en primer lugar, abrir una conexión de mensajería explícita para luego enviar una solicitud de servicio Create Request a la clase Connection del dispositivo. Esto crea una nueva instancia de la clase I/O Connection la cual debe ser configurada. Finalmente, se envía una solicitud de aplicación de la configuración y se entrega la nueva instancia a la aplicación que va a consumir o producir datos (modelo master / slave o productor / consumidor).

Offline Connection Set. Este tipo de conexión es usado por herramientas cliente para recuperar dispositivos en el estado BUS OFF y pasarlos a ERROR ACTIVE. Esta es la única transición que CAN no define en su maquina de estados para confinamiento de fallas. Usando mensajes del grupo 4, una herramienta de este tipo puede:

- Identificar visualmente el dispositivo (flashing LED).

- Enviar mensajes de recuperación al dispositivo fallado.
- Recuperar el dispositivo fallado sin tener que desconectarlo.

Si hay más de una herramienta cliente que desea recuperar dispositivos fallados, para evitar problemas de consistencia, DeviceNet define un protocolo de ownership del Offline Connection Set. De esta forma, se tiene seguridad que en un momento dado, sólo existe un nodo usando alguna de las conexiones del conjunto de conexiones Offline.

El protocolo de obtención del ownership es el siguiente. La herramienta produce un Offline Ownership Request Message usando una conexión del grupo 4 con Message ID en 0x2F. Luego, espera una respuesta de tipo Offline Ownership Response Message por la conexión del grupo 4 con Message ID en 0x2E durante 1 segundo. Si no se recibe respuesta, se genera otro mensaje y se vuelve a esperar por la respuesta.

Transcurrido el segundo período de espera, si no llegó ninguna respuesta, la herramienta gana el ownership del Offline Connection Set. Una vez que la gana, la herramienta debe responder a todos los mensajes Offline Ownership Request Message para que el protocolo antes descrito funcione correctamente.

Cuando un dispositivo pasa al estado BUS OFF, sólo tiene que filtrar mensajes del grupo 4 con Message ID en 0x2D. De igual forma, sólo va a producir mensajes del grupo 4 con Message ID en 0x2C. Una herramienta cliente, que ya haya ganado el ownership del Offline Connection Set, transmite mensajes por la conexión del grupo 4 con Message ID en 0x2D y recibe respuestas por la conexión del grupo 4 con Message ID en 0x2C.

G.1.5. Capa de transporte: utilización de conexiones predefinidas

El modelo general de comunicaciones de DeviceNet define dos tipos de conexiones: las de mensajería explícita para comunicaciones peer-to-peer, y las de I/O para comunicaciones maestro / esclavo o bien, productor / consumidor.

El modelo utiliza conexiones de mensajería explícita para crear y configurar manualmente objetos Connection dentro de cada end-point y así, obtener conexiones de I/O. Este

enfoque impone mayores requerimientos de procesamiento sobre los dispositivos DeviceNet. Por ejemplo, supóngase un dispositivo DeviceNet que no trabaja con comunicaciones peer-to-peer y desea manejarse únicamente con un modelo maestro / esclavo. Bajo el modelo general, está obligado a soportar el UCMM y conexiones de mensajería explícita aunque estas sólo van a ser usadas para crear conexiones de I/O.

Para evitar estos requerimientos innecesarios y bajar los costos de los dispositivos DeviceNet, se define un conjunto de conexiones de I/O predefinidas que facilitan las comunicaciones típicas de relaciones master / slave. De esta forma, es posible construir dispositivos sin UCMM.

En primer lugar, es necesario definir qué criterio usa DeviceNet para decir que un dispositivo es master o slave. El master es el dispositivo que junta y distribuye datos de I/O al controlador del proceso. El slave es el dispositivo que envía datos de I/O a su master. Los masters inician las actividades mediante mensajes de solicitud y los slaves contestan con sus datos de I/O. Esto indica que los masters se comportan como clientes mientras que los slaves, como servidores.

Estos clientes y servidores pueden comunicarse mediante tres tipos de conexión, predefinidos dentro del conjunto de conexiones master / slave:

I/O Bit-Strobe Connection Esta conexión es usada para mover rápidamente pequeñas cantidades de datos de I/O entre un master y los bit-strobed slaves. El master, en base a su lista de slaves, construye una palabra de 64 bits (8 bytes) en la cual, pone un bit para cada slave (de ahí, bit strobed). Así, el bit 0 corresponde a la MAC ID 0 y el 63, a la MAC ID 63. Esta solicitud se envía usando el Grupo 2 con Message ID en 0x00.

En la respuesta, los slaves mandan directamente 8 bytes de datos de I/O dependiendo de cómo hayan interpretado su bit en la solicitud. Para esto, usan mensajes del grupo 1 con Message ID en 0x0E. Se trabaja con un único objeto Connection del lado del slave y múltiples del lado del master.

I/O Poll Connection La conexión I/O Poll, a diferencia de la anterior, sirve para mover cantidades mayores de datos (sólo punto a punto, en modo unicast). El master tiene un objeto Connection para cada slave. El master envía la solicitud de poll a un slave con un mensaje del grupo 2 con Message ID en 0x05. El slave correspondiente contesta con un mensaje del grupo 1 con Message ID en 0x0F.

I/O Change Of State/Cyclic Connection Es similar al tipo de conexión anterior siendo la única diferencia el uso de acknowledgement en ambos sentidos o sea, de master a slave y viceversa. La transmisión se dispara cuando se produce un cambio de estado en la aplicación o cuando expira un timeout. Este comportamiento es configurable al reservar la conexión. El master envía una solicitud usando una conexión polled o sea, envía mensajes del grupo 2 con Message ID en 0x05. El slave envía el acknowledgement correspondiente usando el grupo 1 con Message ID en 0x0F.

Luego, el slave envía al master el dato solicitado usando mensajes del grupo 1 con Message ID en 0x0D y el master contesta con el acknowledgement del grupo 2 y Message ID en 0x02. Como se puede ver, los servidores o slaves siempre reciben mensajes del grupo 2 por lo cual, es posible construirlos para que sólo soporten mensajes de dicho grupo. Esto permite usar adaptadores CAN más sencillos sin necesidad de registros mask-and-match para filtrar mensajes. Esta es una posibilidad contemplada por el estándar de DeviceNet. Un dispositivo con soporte para UCMM y mensajería explícita, puede usar conexiones de este conjunto predefinido solicitando el servicio Allocate_Master/Slave_Connection_Set del objeto DeviceNet. Si en cambio, se trabaja con dispositivos que sólo soportan el grupo 2 y no incluyen el UCMM, se define un mecanismo especial y se reserva el Message ID 0x06 del grupo 2 para solicitar el uso del conjunto predefinido.

G.1.6. Capa de sesión: protocolo de mensajes DeviceNet

El propósito de la capa de sesión es permitir la organización del diálogo, su sincronización y la administración del intercambio de datos [ISO7498]. En esta sección se describen los protocolos usados por los dispositivos DeviceNet para organizar su diálogo y así, comunicarse entre ellos. O sea, es una descripción del contenido que los dispositivos DeviceNet colocan en los DATA FIELDS de los frames CAN que intercambian.

En primer lugar, se describirán los protocolos usados por las conexiones de mensajería explícita y de I/O, que reciben el nombre de Control and Information Protocol (CIP) [VanGompel 01]:

Conexiones de Mensajería Explícita Un mensaje explícito o explicit message está formado por un encabezado de 1 byte y un cuerpo de 7 bytes. Si el cuerpo del mensaje no entra en estos 7 bytes, se aplica el protocolo de fragmentación, intercalando 1 byte entre el encabezado y el cuerpo (ahora de 6 bytes).

Los campos del encabezado del mensaje son:

- Frag (Fragment Bit, 1 bit): indica si el frame CAN contiene un fragmento (1, Fragmented) o el mensaje completo (0, Non-fragmented).
- XID (Transaction ID, 1 bit): usado por las aplicaciones para hacer corresponder una solicitud con su correspondiente respuesta. Teniendo en cuenta que el mecanismo de control de flujo sólo permite un mensaje pendiente de confirmación, un único bit es suficiente para esto. El server se limita a hacer un echo del valor recibido
- MAC ID (MAC ID, 6 bits): MAC ID o dirección del dispositivo de origen o de destino. Si en el campo de arbitración, al generar el Connection ID o CID, se utilizó la dirección de origen entonces, este campo contiene la dirección de destino y viceversa.

Los campos del cuerpo del mensaje son:

Service Field (1 byte) identificación del servicio que se está solicitando o respondiendo. Este campo, a su vez, se divide en:

- R/R (1 bit): determina si se trata de un request (bit en 0) o de un response (bit en 1).

- Service Code (7 bits): identificador del servicio que se está solicitando o respondiendo. Podrían ser, por ejemplo, los códigos de los DeviceNet Common Services.

Service Specific Arguments (1 o más bytes) parámetros específicos del servicio que se está solicitando. Por ejemplo, supóngase que se está llamando a un servicio particular de una instancia, este campo contendría los siguientes elementos:

- Class ID (1/2 bytes) define la clase a la cual se está enviando la solicitud. Se usan 1 o 2 bytes dependiendo de los parámetros usados al crear la conexión de mensajería explícita (Requested Message Body Format).
- Instance ID (1/2 bytes) define la instancia de la clase identificada por Class ID a la cual se dirige la solicitud. Se usan 1 o 2 bytes dependiendo de los parámetros usados al crear la conexión de mensajería explícita (Requested Message Body Format). El valor 0 está reservado para indicar que la solicitud va dirigida a una clase y no a una instancia.
- Service Data (0 o más bytes) datos específicos de la solicitud. En el caso de una respuesta exitosa a la solicitud, sólo se recibiría en este campo datos específicos de la solicitud. Podrían ser, por ejemplo, valores de atributos o valores de retorno del servicio solicitado.

Conexiones de I/O DeviceNet no define ninguna estructura ni protocolo dentro del DATA FIELD de un mensaje de I/O excepto el protocolo de fragmentación para mensajes de I/O mayores a 8 bytes.

Las condiciones de aplicación del protocolo de fragmentación son distintas para este tipo de conexión que para el anterior.

En secciones anteriores, se describió parcialmente el mecanismo de establecimiento de conexiones de mensajería explícita las cuales son usadas, posteriormente, para crear

conexiones de I/O. Se enumeraron los mensajes procesados por el UCMM los cuales, en cierta forma, son los servicios ofrecidos por este objeto DeviceNet.

Estos mensajes tienen el formato que se acaba de presentar, usado en las conexiones de mensajería explícita. A continuación se dan detalles, para cada tipo de unconnected explicit message, de los contenidos del campo Service Specific Arguments:

Open Explicit Messaging Connection Request: solicita el establecimiento de una conexión lógica entre dos módulos a través de la cual se transmitirán mensajes explícitos. Este mensaje se envía con un valor de arbitración del grupo 3 con Message ID en 0x06, usando un Service Code igual a 0x4B. Los argumentos específicos son:

Reserved Bits (4 bits): reservados para uso futuro. El receptor los ignora y el transmisor debe poner todos en 0.

Requested Message Body Format (4 bits): el formato en que se deberán transmitir los mensajes subsecuentes por esta conexión. Los posibles valores

Tabla 29. Request Message Body Format en Open Connection Request

Body Format	Descripción
0x0	DeviceNet(8/8), 8 bits de Class ID, 8 bits de Instance ID
0x1	DeviceNet(8/16), 8 bits de Class ID, 16 bits de Instance ID
0x2	DeviceNet(16/16), 16 bits de Class ID, 16 bits de Instance ID
0x3	DeviceNet(16/8), 16 bits de Class ID, 8 bits de Instance ID
0x4-0xF	Reservado por DeviceNet

El receptor del mensaje (a partir de ahora, el servidor) es el que toma la decisión final sobre el formato que se va a usar, dependiendo si puede cumplir o no con lo que pide el emisor (a partir de ahora, el cliente). En la respuesta se manda el valor efectivo para la conexión.

Group Select (4 bits) grupo de mensajes a usar. Los valores son 0 para el grupo 1, 1 para el 2 y 3 para el 3. Si el servidor no soporta o temporalmente no puede usar algún grupo, debe responder con un mensaje de error. En este caso, no hay negociación como en el anterior.

Source Message ID (4 bits) depende del valor de Group Select. Si este último es 0 o 3, especifica el Message ID asignado por el cliente de su pool de IDs del grupo 1 o grupo 3. Si por el contrario, Group Select es 1 este campo es ignorado y se llena con 0.

Open Explicit Messaging Connection Response este servicio se utiliza para responder una solicitud exitosa de establecimiento de conexión. Este mensaje se envía con un valor de arbitración del grupo 3 con Message ID en 0x05, usando un Service Code igual a 0x4B y con el campo R/R en 1. Los argumentos específicos son:

Reserved Bits (4 bits) reservados para uso futuro. El receptor los ignora y el transmisor debe poner todos en 0.

Actual Message Body Format (4 bits) el formato en que se deberán transmitir los mensajes subsecuentes por esta conexión. Puede ser un valor definido por el servidor o bien, el que solicitó el cliente si éste puede cumplirlo.

Destination Message ID (4 bits) depende del valor de Group Select. Si este es 1, especifica el Message ID asignado por el servidor de su pool de IDs del grupo 2. Si por el contrario, Group Select es 0 o 3 este campo es ignorado y se llena con 0.

Source Message ID (4 bits) el Message ID que el servidor asignó de su pool de IDs para el grupo 1, 2 o 3. El servidor usa este valor junto con su MAC ID como identificador de la conexión o CID.

Connection Instance ID (1/2 bytes) identificador de la instancia de la clase Connection creada por el UCMM. Este valor puede ser usado posteriormente para cerrar la conexión o bien, para abrir una conexión I/O invocando servicios del objeto Connection.

Close Connection Request servicio usado para cerrar una conexión I/O o mensajería implícita, es indistinto. Al recibir este mensaje, el UCMM llama al servicio Delete de la clase Connection. Este mensaje se envía con un valor de arbitración del grupo 3 con Message ID en 0x06, usando un Service Code igual a 0x4C. Los argumentos específicos son:

- **Connection Instance ID (2 bytes)** identificador de la instancia de la clase Connection a ser eliminada. Como estos mensajes se transmiten como unconnected messages, el receptor puede no conocer el message body format así que se usan siempre 2 bytes.

Close Connection Response este servicio es usado para responder en forma exitosa a un close connection request. Este mensaje se envía con un valor de arbitración del grupo 3 con Message ID en 0x05, usando un Service Code igual a 0x4C y con el campo R/R en 1. No tiene argumentos específicos.

Error Response este tipo de mensajes se envía cuando se produce un error tratando de dar servicio a una solicitud recibida en un explicit message. Este mensaje se envía con un valor de arbitración del grupo 3 con Message ID en 0x05, usando un Service Code igual a 0x14. Los argumentos específicos del mismo son:

- **General Error Code (1 byte)** identificador general del error.
- **Additional Error Code (1 byte)** valor específico del objeto o servicio, que aporta información adicional sobre el error. Si no hay información adicional, se coloca en este campo 0xFF.

Estos son los errores que se pueden presentar frente a una solicitud Open Explicit Messaging Connection Request o Close Connection Request:

Tabla 30. Posibles códigos de error en mensajes Error Response de DeviceNet

Condición de Error	Nombre del Error	Código de Error General	Código de Error Adicional
El Service Code no es Open ni Close.	Service not supported	0x08	0xFF
Error de recursos al seleccionar el grupo de mensajes.	Resource unavailable	0x02	0x01
El grupo de mensajes está fuera del rango válido.	Invalid Parameter	0x20	0x01
El servidor no dispone de más conexiones.	Resource unavailable	0x02	0x02
El servidor no dispone de más Message IDs para nuevas conexiones.	Resource unavailable	0x02	0x03
El Message ID de origen, enviado por el cliente, no es válido.	Invalid Parameter	0x20	0x02
El Message ID de origen, enviado por el cliente, está repetido.	Resource unavailable	0x02	0x04
El Instance ID de la conexión no es válido.	Object does not exists	0x16	0xFF

Por último, es necesario describir los servicios de fragmentación y de control de flujo end-to-end provistos por las conexiones DeviceNet. Si bien estos servicios normalmente se asocian con la capa de transporte, en el presente trabajo se incluyen junto con la capa de sesión para facilitar el orden de exposición de los temas.

La fragmentación en DeviceNet está planteada como un servicio opcional provisto por el objeto Connection. La lógica que decide si hay que aplicar fragmentación es distinta según el tipo de conexión:

Conexiones de Mensajería Explícita se examina la longitud de cada mensaje y si supera los 8 bytes, se aplica fragmentación acknowledged (con control de flujo).

Conexiones de I/O se examina el atributo produced_connection_size del objeto Connection y se decide en base a su valor, configurado al momento de crear el objeto. Si es mayor que 8, se aplica fragmentación unacknowledged (sin control de flujo). El

protocolo de fragmentación ocupa un byte entre el encabezado y el cuerpo del mensaje, si es necesario aplicarlo. Los campos definidos por este protocolo dentro de ese byte son:

Fragment Type (2 bits) indica si se trata del primer, último o un fragmento intermedio. Los posibles valores son 0 para el primer fragmento, 1 para fragmentos intermedios, 2 para el último fragmento y finalmente, 3 para el acknowledge de un fragmento.

Fragment Count (6 bits) índice o número de secuencia del fragmento que se está enviando. Se incrementa en 1 para cada nuevo fragmento de forma de poder efectuar un control de secuencia y determinar si se perdió algún fragmento.

En el caso de una conexión I/O, cuando se invoca el servicio `Send_Message` se examina el valor del atributo `produced_connection_size`. Si este valor es mayor a 8, indistintamente del tamaño del dato a transmitir, se coloca en el mensaje el byte del protocolo de fragmentación. Si el tamaño del dato a transmitir es mayor a `produced_connection_size` se produce un error.

Si el receptor detecta que falta un fragmento, al ver valores de `Fragment Count` no consecutivos, descarta todos los mensajes subsiguientes. La aplicación no se entera de que llegó un mensaje y la conexión se reinicia para esperar nuevamente por un fragmento inicial. Por otro lado, en el caso de una conexión de mensajería explícita, cada vez que se transmite un fragmento, se espera el `acknowledge` del nodo receptor. La disciplina de línea aplicada corresponde a un algoritmo `stop-and-wait`. Los campos de un mensaje de `acknowledgement` son:

Fragment Type (2 bits) indica que se trata de un `fragment acknowledge`. Tiene siempre valor 3.

Fragment Count (6 bits) es el valor del índice del último frame recibido.

Ack Status (1 byte) indica si se produjo (1) o no (0) un error en el receptor del mensaje. Los valores de 0x02 a 0xFF están reservados. El nodo transmisor envía un fragmento y espera el `acknowledgement` del receptor. Se inicia un timer para controlar la expiración del tiempo de espera. Si se da un `timeout`, el objeto `Connection` reintentará. Si nuevamente

se da un timeout, se le informa a la aplicación del error. Al recibir un acknowledge, se controlan los campos Fragment Count y Ack Status, aceptándose el fragmento si así corresponde. Para las capas superiores, la utilización o no de fragmentación es completamente transparente.

G.1.7. Capa de presentación: Compact Encoding

En DeviceNet, las funciones de la capa de presentación son realizadas por Compact Encoding. Se trata de una sintaxis de transferencia que permite pasar de una estructura compleja a un stream de bytes y viceversa.

En primer lugar, es necesario definir los tipos de dato de estas estructuras complejas a convertir. Para mantener la independencia entre capas, estos tipos se definen mediante una sintaxis abstracta. En el caso de Compact Encoding, se usa la Abstract Syntax Notation One (ASN.1) especificada en la normas ISO 8824.1-4 / ITU-T X.680-683 [ISO8824 02].

En segundo lugar, se debe aplicar un conjunto de reglas de codificación que permitan pasar de los datos abstractos a una secuencia de bytes que pueda ser transmitida. A diferencia de lo que sucede con la sintaxis abstracta, en el caso de las reglas de codificación, no se adopta la solución OSI dada por las Basic Encoding Rules (BER) definidas en los estándares ISO 8825.1-4 / ITU-T X.690-693 [ISO8825 02].

DeviceNet define su propio conjunto de reglas de codificación, respetando las siguientes restricciones:

- Las entidades involucradas en una conexión (los end-points de la misma) tienen conocimiento previo del tipo de las variables por lo cual, no es necesario transmitirlo. Así se evita enviar el tipo de cada variable en cada mensaje.
- El tipo de la variable es de longitud fija y no tiene campos condicionales u opcionales.

- La codificación de una variable es representada con un número constante de bytes, derivados del tipo de la misma. Así se evita transmitir la longitud de cada variable.
- Si la variable es entera y ocupa más de un byte, se usa siempre ordenamiento Least Significant Byte (LSB) o little endian. Esto es exactamente lo contrario del ordenamiento Most Significant Byte (MSB) o big endian, usado en ISO/OSI.

A partir de estas restricciones, se obtiene las siguientes reglas de codificación para tipos básicos:

Tabla 31. Reglas de codificación de Compact Encoding

Tipo de Dato	Bytes	Regla de Codificación
BOOL	1	TRUE → 0x01 FALSE → 0x00
Codificación de SignedInteger (enteros con signo)		
SINT	1	0LSB
INT	2	0LSB 1LSB
DINT	4	0LSB 1LSB 2LSB 3LSB
LINT	8	0LSB 1LSB 2LSB 3LSB ...
Codificación de UnsignedInteger (enteros sin signo)		
USINT	1	0LSB
UINT	2	0LSB 1LSB
UDINT	4	0LSB 1LSB 2LSB 3LSB
ULINT	8	0LSB 1LSB 2LSB 3LSB ...
ENGUNIT	2	0LSB 1LSB
Codificación de Strings		
STRING	2 de longitud + 1 por caracter	
STRING2	2 de longitud + 2 por caracter	Los caracteres multibyte (más de un byte) van siempre en LSB.

STRINGN	2 de tamaño de carácter + 2 de longitud + N por caracter	Los caracteres multibyte (más de un byte) van siempre en LSB.
SHORT_STRING	1 de longitud + 1 por caracter	
Codificación de FixedLengthReal (punto flotante)		
REAL	4	Van en notación de IEEE 754 y después en LSB.
LREAL	8	Van en notación de IEEE 754 y después en LSB.
Codificación de Date y Time		
TIME	4	Se codifica como FixedLengthReal.
DATE	2	
TIME_OF_DAY	4	Se codifica como FixedLengthReal.
DATE_AND_TIME	6	Se codifica como FixedLengthReal (4 para la hora y 2 para la fecha)
FTIME	4	Se codifica como FixedLengthReal
LTIME	8	Se codifica como FixedLengthReal
Codificación de FixedLengthBitStrings (cadenas de bits)		
BYTE	1	Orden de bits: 7-0
WORD	2	Orden de bits: 7-0, 15-8
DWORD	4	Orden de bits: 7-0, 15-8, 23-16, 31-24
LWORD	8	Orden de bits: 7-0, 15-8, 23-16, 31-24, ...

Los arreglos (arrays) se codifican como una simple concatenación repetitiva de los elementos que lo componen (usando las reglas anteriores para cada uno de estos). Un array de 1 dimensión, definido en ASN.1, sería:

Código 1 - Ejemplo de array DeviceNet en ASN.1

```

ARRAY ::= SEQUENCE OF {
    array_dimension_low_bound,
    array_dimension_high_bound,
    DeviceNetData
}

```

Para un array de 2 dimensiones:

```
ARRAY ::= SEQUENCE OF {  
    array_dimension_low_bound,  
    array_dimension_high_bound,
```

Código 2 - Ejemplo de array multidimensional DeviceNet en ASN.1

```
SEQUENCE OF {  
    array_dimension_low_bound,  
    array_dimension_high_bound,  
    DeviceNetData  
}  
}
```

Las estructuras (structs) se codifican de forma similar. Para cada uno de los campos, se aplican las reglas anteriores y luego se concatenan los bytes resultantes.

[ANEXO J: MANUAL DE CONFIGURACIÓN DEL NODO DEVICENET UNICAUCA](#)

BIBLIOGRAFÍA

ALLEN BRADLEY Sistema de cables DeviceNet, Manual de planificación e instalación (Núm. cat. DN-6.7.2ES) Publicación DN-6.7.2ES - Mayo de 1999. <<http://literature.rockwellautomation.com/idc/groups/literature/documents/um/dn-um072-es-p.pdf>>.

_____ Reference Manual 1770_6.5.22. DeviceNet RS_232 Interface Module Communication Protocol (Cat. No. 1770_KFD) Disponible en internet: <http://www.bitman.ca/DF1.pdf>.

_____ User manual 1769SDN DeviceNet scanner module. Publicacion 1769-UM 009C-EN-P mayo de 2005 Disponible en internet: <<http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1769-um009-en-p.pdf>>.

TOCCI, Ronald j. Sistemas digitales principios y aplicaciones. Prentice hall Hispanoamericana SA. Edición 6

ANYBUS. Datasheet Anybus-IC-5124-ABIC_DG_1_52_ROHS

