

SISTEMA PROGRAMABLE DE CONTROL Y SUPERVISIÓN REMOTA DE
EQUIPOS

SERGIO FRANCISCO RUEDA ALVARADO

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
ESCUELA DE CIENCIAS NATURALES E INGENIERÍA
FACULTAD DE INGENIERÍA MECATRÓNICA
BUCARAMANGA

2006 ♡

SISTEMA PROGRAMABLE DE CONTROL Y SUPERVISIÓN REMOTA DE
EQUIPOS

MEMORIAS PROYECTO DE GRADO

SERGIO FRANCISCO RUEDA ALVARADO

DIRECTOR

ING. CARLOS IVÁN PATIÑO GONZÁLEZ

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
ESCUELA DE CIENCIAS NATURALES E INGENIERÍA
FACULTAD DE INGENIERÍA MECATRÓNICA
BUCARAMANGA

2006

DEDICATORIAS

A Dios que nunca me desampara

A mi Mama, mi Papa, Tía Amparo, mis Hermanos, Antonie, Ruth,

Tía Teresa, Anita y a toda mi familia por que siempre han sido incondicionales conmigo y siempre serán lo más importante para mi vida.

A la Flaca mi novia, que me llena de paciencia y fortaleza cuando más

lo necesito.

AGRADECIMIENTOS

Este proyecto surge de la combinación de varias ideas que nacieron y se cultivaron en circunstancias totalmente diferentes. Primero agradezco a las personas que influyeron en lo que fue el proyecto irresuelto "*Diseño Y Desarrollo De Sistema Domótico Para Control De Seguridad Con Tecnología X-10*", a la empresa Oleaginosas las Brisas S.A. que me apoyo durante el desarrollo de mi práctica empresarial; allí gracias al apoyo que recibí de mis compañeros de trabajo, la idea prospero y se convirtió en el proyecto que es ahora.

Agradezco también a los profesores que de una u otra manera me apoyaron compartiendo sus conocimientos y experiencias durante el desarrollo de mi carrera. A mis amigos, por que sin su ayuda nada de esto hubiera sido posible.

Agradezco a mi familia que es lo más grande que Dios me ha dado.

Por ultimo pero lo mas importante, le doy gracias a Dios mi compañero durante el desarrollo de este proyecto.

TABLA DE CONTENIDO

1. MARCO TEORICO	13
1.1. ANÁLISIS DE COMPONENTES	14
1.1.1. CONTROLADOR LÓGICO PROGRAMABLE (PLC)	14
1.1.1.1. COMPONENTES BÁSICOS.....	15
Rack principal.....	15
Fuente De Alimentación	15
CPU.....	15
Periféricos Entradas/Salidas Digitales	15
Periféricos Entradas/Salidas Analógicos	15
Periféricos Especiales	16
1.1.1.2. FUNCIONES BÁSICAS DE UN PLC	16
1.1.1.3. NUEVAS FUNCIONES	16
Redes De Comunicación	16
Sistemas De Supervisión.....	16
Entradas- Salidas Distribuidas	17
Buses De Campo	17
1.1.1.4. VENTAJAS Y DESVENTAJAS	17
Ventajas.....	17
Desventajas	18
1.1.2. SISTEMA SCADA	18
1.1.2.1. ARQUITECTURA.....	18
Hardware.....	19
Software	19
Comunicación	19
Comunicación Interna.....	19
Acceso a Dispositivos	19
Interfaz	20
Escalabilidad.....	20
Redundancia	20
1.2. FUNDAMENTOS TEORICOS	21
1.2.1. MODELO OSI	21
1.2.1.1. CAPAS O NIVELES DEL MODELO OSI.....	22
Capa Física	22
Capa de Enlace.....	22
Capa de RED	23

Capa de Transporte.....	23
Capa de Sesión	24
Capa de Presentación.....	24
Capa de Aplicación.....	25
1.2.2. INTERFASES DE COMUNICACIÓN.....	26
1.2.2.1. ELECTRONIC INDUSTRIES ASSOCIATION RECOMMENDED STANDARD 232.....	26
1.2.2.2. ELECTRONIC INDUSTRIES ASSOCIATION RECOMMENDED STANDARD 485.....	28
1.2.3. BUSES DE DATOS Y PROTOCOLOS	28
1.2.3.1. INTER IC BUS	28
Características Principales Del Bus I ² C.....	29
Condiciones De Inicio (Start Y Stop).....	29
Transfiriendo Datos	30
Reconocimiento.....	30
Arbitraje Y Generación De Señales De Reloj.....	31
Sincronización	31
Arbitraje	31
1.2.3.2. PROTOCOLOS ASCII.....	32
Protocolo ASCII Para Transmisores Digitales.....	32
Características.....	32
1.2.4. MICROCONTROLADOR	33
2. DISEÑO.....	35
2.1. DISEÑO MECATRÓNICO	36
2.1.1. DISEÑO DE DPBR Y PESD22	38
2.1.1.1. SELECCIÓN DE COMPONENTES	38
CPU	39
Memoria.....	41
Comunicación	42
RTC	42
2.1.1.2. DISEÑO DE FIRMWARE.....	44
Firmware DPBR	44
Firmware PESD22	48
2.1.1.3. DISEÑO DE PROTOCOLOS DE COMUNICACIÓN:	50
DPBR ⇔ PC	50
DPBR ⇔ Periféricos.....	52
Acciones en PESD22	55
2.1.1.4. DISEÑO FÍSICO.....	57
Esquema y circuitos impresos	57
Carcazas	57
2.1.2. DISEÑO DE PAQUETE DE SOFTWARE	58
2.1.2.1. ESTRUCTURA DE PAQUETE	58

Aplicación De Servidor	59
Aplicación Programación	61
Aplicación Supervisión	66
3. DESARROLLO	69
3.1. DESARROLLO DE SOFTWARE Y FIRMWARE	70
3.1.1. DESARROLLO DE PAQUETE DE SOFTWARE.	70
3.1.2. FIRMWARE'S.....	72
3.1.3. INTEGRACIÓN	73
3.1.4. PRUEBAS DE FUNCIONAMIENTO.....	74
3.2. DESARROLLO FÍSICO.....	75
4. MANUAL DE OPERACIÓN	76
4.1. INSTALACION Y PUESTA EN MARCHA	77
4.1.1. APLICACIÓN SERVIDOR	82
4.1.2. APLICACIÓN PROGRAMADOR.....	88
4.1.2.1. PANEL DE PERIFÉRICOS Y PUERTOS	91
4.1.2.2. BARRA DE MENÚ	93
4.1.2.3. BARRA DE HERRAMIENTAS.....	95
Creación de Reglas DPBR	95
Programación	99
Descarga de Memorias	100
Herramientas de Emergencia.....	100
Periféricos	101
Panel de Conexión	102
4.1.3. APLICACIÓN SUPERVISIÓN	103
4.1.3.1. VISUALIZADOR	104
4.1.3.2. MENÚ.....	106
4.2. MONTAJE DE EQUIPOS	107
4.3. INDICADORES DE FUNCIONAMIENTO	109
5. CONCLUSIONES	112
6. RECOMENDACIONES.....	115
7. BIBLIOGRAFÍA.....	118
8. ANEXOS	120
8.1. ANEXO 1.....	121
8.1.1. DATASHEET PIC16F87X	121
8.1.2. DATASHEET MEMORIA EEPROM I ² C 24LC256	122
8.1.3. DATASHEET TRANSCEIVER MAX-232	123
8.1.4. DATASHEET TRANSCEIVER MAX-485	124
8.1.5. DATASHEET RTC DS1307	125
8.2. ANEXO 2.....	126
8.2.1. NORMA TIA/EIA-568-B	126

8.2.2. CONECTOR RJ-45	128
8.3. ANEXO 3.....	129
8.3.1. DISEÑO ELECTRÓNICO.....	129
8.3.2. LISTA DE MATERIALES.....	137
8.3.3. CIRCUITOS IMPRESOS	139
8.3.4. DISEÑO DE CARCAZAS	141
8.4. ANEXO 4.....	145
8.4.1. CÓDIGO FUENTE PAQUETE DE SOFTWARE	145
8.4.1.1. ADMINISTRACIÓN Y GRUPOS	145
8.4.1.2. AGREGAR PERIFÉRICO.....	152
8.4.1.3. CONFIGURACIÓN.....	156
8.4.1.4. CONFIGURACIÓN DPBR.....	177
8.4.1.5. CONFIGURACION USUARIO	192
8.4.1.6. DESCARGAR MEMORIAS.....	197
8.4.1.7. HERRAMIENTAS DE EMERGENCIA.....	207
8.4.1.8. INICIO DE SESION	212
8.4.1.9. PERIFERICOS.....	215
8.4.1.10. SIMULACION	221
8.4.1.11. SUBIR BINARIO	228
8.4.1.12. AGREGAR SECCION	239
8.4.1.13. SELECCIÓN DE IMAGEN	244
8.4.1.14. VISOR	254
8.4.1.15. CONSOLA	293
8.4.2. FIRMWARE.....	321
8.4.2.1. DPBR.....	321
8.4.2.2. PESD22.....	331

LISTA DE FIGURAS

FIGURA 1. DRIVER – RECEIVER RS-232	26
FIGURA 2. PINES RS-232	27
FIGURA 3. FUNCIONAMIENTO RS-485	28
FIGURA 4. PROTOCOLO I ² C.....	29
FIGURA 5. DISEÑO MECATRÓNICO I.....	36
FIGURA 6. DISEÑO MECATRÓNICO II	37
FIGURA 7. DIAGRAMA DE BLOQUES DPBR.....	38
FIGURA 8. DIAGRAMA DE BLOQUES PESD22	39
FIGURA 9. ALGORITMO FIRMWARE DPBR I.....	44
FIGURA 10. MÓDULO ANÁLISIS DE CONDICIONES DPBR.....	45
FIGURA 11. MENÚ PC ↔ DPBR.....	47
FIGURA 12. ALGORITMO PESD22	48
FIGURA 13. ACCIONES PESD22	49
FIGURA 14. PROTOCOLO PC ↔ DPBR	50
FIGURA 15. PROTOCOLO DPBR ↔ PERIFÉRICOS.....	54
FIGURA 16. ESTRUCTURA DE PAQUETE	58
FIGURA 17. JERARQUÍA SERVIDOR	59
FIGURA 18. COMUNICACIÓN SERVIDOR.....	59
FIGURA 19. ESTRUCTURA SERVIDOR.....	60
FIGURA 20. ESTRUCTURA PROGRAMACIÓN.....	61
FIGURA 21. DIAGRAMA REGLAS DPBR	62
FIGURA 22. PROGRAMACIÓN DPBR.....	63
FIGURA 23. ACCESO A HERRAMIENTAS DE EMERGENCIA	64
FIGURA 24. ACCESO A PERIFÉRICOS	65
FIGURA 25. ESTRUCTURA DE SUPERVISIÓN.....	66
FIGURA 26. EQUIPOS EN SECCIONES	67
FIGURA 27. VISUALIZADOR SUPERVISIÓN.....	68
FIGURA 28. CONEXIÓN VISUALIZADOR ↔ DPBR	68

LISTA DE IMÁGENES

IMAGEN 1. PLC's	14
IMAGEN 2. MODELO OSI.....	21
IMAGEN 3. SUPERVISIÓN REMOTA – PRIMERA VERSIÓN.....	70
IMAGEN 4. CARCAZAS	75
IMAGEN 5. CONFIGURACIÓN DE SEGURIDAD LOCAL.....	77
IMAGEN 6. VENTANA EJECUTAR	78
IMAGEN 7. VENTANA DE CONFIGURACIÓN – EQUIPO SERVIDOR.....	79
IMAGEN 8. VENTANA DE CONFIGURACIÓN – UBICACIÓN DE BASE DATOS.....	79
IMAGEN 9. NUEVA BASE DE DATOS.....	80
IMAGEN 10. VENTANA DE CONFIGURACIÓN – UBICACIÓN DE CARPETAS	80
IMAGEN 11. VENTANA DE CONFIGURACIÓN – INFORMACIÓN EMPRESARIAL	81
IMAGEN 12. VENTANA DE CONFIGURACIÓN – INFORMACIÓN DE PUERTOS	81
IMAGEN 13. VENTANA DE INICIO	82
IMAGEN 14. SERVIDOR - ICONO BARRA DE TAREAS	82
IMAGEN 15. ESTADOS DE ESPERA.....	83
IMAGEN 16. ESTADOS DE SUPERVISIÓN	83
IMAGEN 17. ESTADOS DE PROGRAMACIÓN	83
IMAGEN 18. VENTANA DE MONITORIZACIÓN Y CONTROL.....	84
IMAGEN 19. VENTANA DE MONITORIZACIÓN Y CONTROL (EXPANDIDA).....	85
IMAGEN 20. ESTADOS DE ACCIONES - SERVIDOR	85
IMAGEN 21. VENTANA DE MONITORIZACIÓN Y CONTROL (NORMALIZAR).....	86
IMAGEN 22. VENTANA DE MONITORIZACIÓN Y CONTROL (SEGUNDO PLANO).....	86
IMAGEN 23. MENÚ DE BARRA.....	86
IMAGEN 24. CONEXIÓN DPBR ↔ PC.....	87
IMAGEN 25. DPBR ↔ PC (SERVIDOR)	87
IMAGEN 26. DPBR ↔ PERIFÉRICOS	88
IMAGEN 27. CONEXIÓN DE FUENTE	88
IMAGEN 28. VENTANA DE INICIO DE SESIÓN.....	89
IMAGEN 29. MENSAJE INICIO DE SESIÓN	89
IMAGEN 30. CONFIGURACIÓN DE USUARIO	90
IMAGEN 31. VENTANA PRINCIPAL DE PROGRAMACIÓN	90
IMAGEN 32. PANEL DE PERIFÉRICOS Y PUERTOS	91
IMAGEN 33. NUEVO PERIFÉRICO.....	91
IMAGEN 34. AGREGAR/MODIFICAR PERIFÉRICO.....	92
IMAGEN 35. PUERTOS	92
IMAGEN 36. NUEVO PUERTO	93
IMAGEN 37. AGREGAR/MODIFICAR NUEVO PUERTO	93
IMAGEN 38. BARRA DE MENÚ.....	93
IMAGEN 39. ADMINISTRACIÓN Y GRUPOS	94
IMAGEN 40. PANEL DE HERRAMIENTAS.....	95
IMAGEN 41. CREAR REGLAS PARA DPBR.....	95
IMAGEN 42. SELECCIÓN DE PERIFÉRICO	96

IMAGEN 43. ESTADO DE ENTRADA	96
IMAGEN 44. CONDICIÓN HORARIA.....	97
IMAGEN 45. ESTADOS DE SALIDA.....	97
IMAGEN 46. PUERTO DE SALIDA.....	97
IMAGEN 47. ALMACENAR	98
IMAGEN 48. REGLAS DPBR - ABRIR GUARDAR	98
IMAGEN 49. REGLAS DPBR - SIMULACIÓN	98
IMAGEN 50. PROGRAMACIÓN DPBR.....	99
IMAGEN 51. DESCARGAR MEMORIAS.....	100
IMAGEN 52. HERRAMIENTAS DE EMERGENCIA.....	100
IMAGEN 53. VENTANA PERIFÉRICOS.....	101
IMAGEN 54. CONFIGURACIÓN PERIFÉRICO (ASIGNAR ID).....	102
IMAGEN 55. PANEL DE SERVIDORES.....	102
IMAGEN 56. APLICACIÓN SUPERVISIÓN	103
IMAGEN 57. AGREGAR NUEVO PUERTO - SUPERVISIÓN	104
IMAGEN 58. IMAGEN REPRESENTATIVA.....	105
IMAGEN 59. CREAR SECCIÓN.....	105
IMAGEN 60. PESD22	107
IMAGEN 61. BORNERAS PESD22	107
IMAGEN 62. ATORNILLAR BORNERAS PESD22.....	108
IMAGEN 63. INDICADORES EN DPBR	109
IMAGEN 64. RJ-45	128
IMAGEN 65. DISEÑO ELECTRÓNICO DPBR COMPLETO.....	129
IMAGEN 66. DISEÑO ELECTRÓNICO DPBR (CPU)	130
IMAGEN 67. DISEÑO ELECTRÓNICO DPBR (MEMORIAS).....	131
IMAGEN 68. DISEÑO ELECTRÓNICO DPBR (RTC)	131
IMAGEN 69. DISEÑO ELECTRÓNICO DPBR (COMUNICACIÓN RED).....	132
IMAGEN 70. DISEÑO ELECTRÓNICO DPBR (COMUNICACIÓN PC)	133
IMAGEN 71. DISEÑO ELECTRÓNICO DPBR (FUENTE)	133
IMAGEN 72. DISEÑO ELECTRÓNICO PESD22 COMPLETO.....	134
IMAGEN 73. DISEÑO ELECTRÓNICO PESD22(CPU)	135
IMAGEN 74. DISEÑO ELECTRÓNICO PESD22 (COMUNICACIÓN).....	136
IMAGEN 75. DISEÑO ELECTRÓNICO PESD22 (FUENTE)	136
IMAGEN 76. CIRCUITO IMPRESO DPBR.....	139
IMAGEN 77. CIRCUITO IMPRESO PESD22	140
IMAGEN 78. VISTA FRONTAL DPBR	141
IMAGEN 79. VISTA LATERAL DPBR.....	141
IMAGEN 80. VISTA SUPERIOR DPBR	142
IMAGEN 81. ISOMETRÍA DPBR.....	142
IMAGEN 82. VISTA FRONTAL PESD22	143
IMAGEN 83. VISTA LATERAL PESD22.....	143
IMAGEN 84. VISTA SUPERIOR PESD22	144
IMAGEN 85. ISOMETRÍA PESD22.....	144

LISTA DE TABLAS

TABLA 1. SELECCIÓN CPU	40
TABLA 2. SELECCIÓN MEMORIA	41
TABLA 3. REGLA DE FUNCIONAMIENTO	46
TABLA 4. REGISTRO ESPECIAL.....	46
TABLA 5. REGISTRO DE BITÁCORA	46
TABLA 6. REGISTRO DE CONFIGURACIÓN	49
TABLA 7. CONFIGURACIÓN DE CABLE DPBR⇄PERIFÉRICOS.....	53
TABLA 8. PRUEBAS DE FUNCIONAMIENTO.....	74
TABLA 9. INDICADORES DPBR.....	110
TABLA 10. INDICADORES PESD22.....	111
TABLA 11. PAR TRENZADO	127
TABLA 12. MATERIALES DPBR.....	137
TABLA 13. LISTA MATERIALES PESD22.....	138

INTRODUCCIÓN

A finales de los años 60 se crearon los PLC's por la necesidad de comunicación en las empresas que hasta el momento utilizaban sistemas de control basados en relays. El PLC tenía que ser fácilmente programable, su vida útil tenía que ser larga y ser resistente a ambientes difíciles. Esto se logró con técnicas de programación conocidas y reemplazando los relays por elementos de estado sólido.

En la actualidad algunas empresas cuentan con sistemas SCADA y redes de PLC's para la supervisión y control de sus equipos, pero debido al elevado costo que puede tener implementar estos equipos de automatización industrial, no todas las empresas cuentan con un sistema para monitorear sus procesos y sus equipos. Este proyecto surgió debido a la necesidad de desarrollar un sistema de monitoreo de manera económica para recolectar información precisa y en tiempo real sobre un proceso.

JUSTIFICACION

Con el fin de crear un sistema económico e intermedio para la supervisión y control de equipos, se diseño y desarrollo un sistema de monitoreo y control basado en reglas programadas mediante un software especialmente diseñado para este fin.

El sistema no tiene las mismas prestaciones que un sistema SCADA.

El sistema completo resulta de gran utilidad en la industria, ya que puede ser usado como soporte para el mantenimiento de equipos basado en las horas de trabajo, igualmente en el área de producción para estimar cuanta materia prima pueden haber utilizado y cuanto producto final deberían tener, y debido a la flexibilidad de su programación, también puede resultar una gran ayuda en el campo de la Domótica, para economizar recursos energéticos o como dispositivo de seguridad alertando sobre intrusiones no autorizadas y guardado una bitácora sobre lo que el usuario le programe o simplemente para dar mayor confort y comodidad al usuario manejando la iluminación, el acceso y la seguridad.

No se pretende desarrollar un controlador lógico programable (PLC) con sistema SCADA, se busca dar una solución económica e intermedia a problemas sencillos en la industria o el hogar.

PLANTEAMIENTO DEL PROBLEMA

En la actualidad, adquirir un sistema de supervisión, control y adquisición de datos (SCADA) puede resultar muy costoso debido a las grandes prestaciones que este ofrece a los usuarios de la industria.

Por lo tanto para una empresa que no cuente con los recursos suficientes resultaría muy difícil tomar la decisión de adquirir un sistema de supervisión de alto nivel como por ejemplo los equipos desarrollados por empresas como National Instrument, Siemens o Telemecanique entre otros.

Estos sistemas son de gran importancia ya que ayudan a simplificar el trabajo de análisis y generan bases sólidas para la toma de decisiones.

OBJETIVOS

OBJETIVO GENERAL

Diseñar y construir un sistema programable de supervisión remota para el control de equipos.

OBJETIVOS ESPECÍFICOS

- × Diseñar y construir un dispositivo programable basado en reglas (DPBR).
- × Diseñar y construir los periféricos para el DPBR.
- × Diseñar e implementar un protocolo de comunicación entre el PC - DPBR y el DPBR - Periféricos mediante el uso de buses de información.
- × Desarrollar un paquete de software para programación y supervisión remota del DPBR y la configuración de sus Periféricos.
- × Desarrollar un módulo dentro del paquete de software para usar la computadora como cliente/servidor aprovechando las ventajas que ofrece del protocolo TCP/IP.

1. MARCO TEORICO

1.1. ANÁLISIS DE COMPONENTES

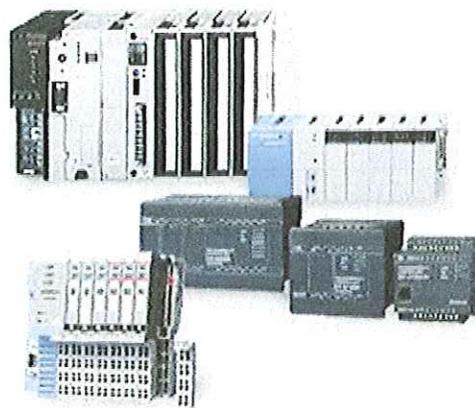
Para el desarrollo del proyecto se tomaron en cuenta ciertos fundamentos teóricos para el diseño del paquete de software, periféricos y dispositivo programable.

A continuación se enumeran los componentes en los que se basa el diseño y desarrollo del proyecto.

1.1.1. CONTROLADOR LÓGICO PROGRAMABLE (PLC)

El diseño del Dispositivo Programable Basado en Reglas y el Periférico de Entradas y Salidas Digitales de 22 puertos se basó en el funcionamiento de este componente electrónico usado comúnmente en el campo de la automatización industrial.

Imagen 1. PLC's



nyco-systems.com

Un Controlador Lógico Programable, es un dispositivo electrónico, programado en lenguaje no informático, diseñado para controlar en tiempo real procesos secuenciales.

Un PLC trabaja en base a la información recibida por sus captadores y opera basado en su programa lógico interno, actuando sobre sus periféricos de salida.

1.1.1.1. COMPONENTES BÁSICOS

RACK PRINCIPAL

Sobre el se conectan todos los elementos. Va atornillado a la placa de montaje. Puede alojar a un número limitado de elementos dependiendo del fabricante y conectarse a otros Racks llamados Racks de Expansión.

FUENTE DE ALIMENTACIÓN

La fuente es la encargada de proporcionar la tensión y corriente necesarias en la CPU y las tarjetas (Depende del Fabricante).

CPU

Consta de uno o varios microprocesadores que se programan mediante un software propio. La mayoría de ellos ofrecen distintos lenguajes para desarrollo y trabajan según lógica digital. El programa almacenado en la CPU del PLC esta escrito en un lenguaje propio, se ejecuta en una secuencia programable y tiene un encabezado y un final. El tiempo que transcurre entre los dos se llama ciclo de *scan* y hay un temporizador interno que vigila que este programa se ejecute de principio a fin.

PERIFÉRICOS ENTRADAS/SALIDAS DIGITALES

Se comunican con la CPU a través buses de comunicación. En el caso de las entradas digitales transmiten los estados del proceso a la CPU. En el caso de las salidas, la CPU determina el estado de las mismas tras la ejecución del programa.

PERIFÉRICOS ENTRADAS/SALIDAS ANALÓGICOS

Se conectan al Rack de igual manera que las anteriores. Estas tarjetas leen un valor analógico que internamente es convertido en un valor digital para su

procesamiento en la CPU. Los rangos de entrada están normalizados siendo lo más frecuente el rango de 4-20mA y 0-10 DCV.

PERIFÉRICOS ESPECIALES

Se utilizan normalmente para control o monitorización de variables o movimientos críticos en el tiempo, ya que usualmente realizan esta labor independientemente de la CPU. Son algunas muestras las siguientes:

- × Tarjetas de montaje rápido
- × Tarjetas de posicionamiento de motores
- × Tarjetas de regulación

1.1.1.2. FUNCIONES BÁSICAS DE UN PLC

- × Detección: Lectura de la señal de los periféricos de entrada.
- × Mando: Elabora y envía las acciones al sistema mediante los periféricos de salidas.
- × Diálogo hombre máquina: Mantener un diálogo el operario de producción, obedeciendo sus consignas e informando del estado del proceso.
- × Programación: Para introducir, elaborar y cambiar el programa de aplicación del PLC. La interfaz de programación debe permitir modificar el programa incluso con el PLC controlando los equipos.

1.1.1.3. NUEVAS FUNCIONES

REDES DE COMUNICACIÓN

Permiten establecer comunicación con otros sistemas de control. Las redes industriales permiten la comunicación y el intercambio de datos entre PLC's a tiempo real.

SISTEMAS DE SUPERVISIÓN

También los PLC's permiten comunicarse con ordenadores provistos de programas de supervisión industrial (SCADA). Esta comunicación se realiza por

una red industrial o por medio de una simple conexión por el puerto serie del ordenador.

ENTRADAS- SALIDAS DISTRIBUIDAS

Los módulos de entrada salida no tienen porqué estar en el gabinete del PLC. Pueden estar distribuidos por la instalación, se comunican con la unidad central del autómata mediante un cable de red y un protocolo de comunicación.

BUSES DE CAMPO

Mediante un solo cable de comunicación se pueden conectar al bus los periféricos de salida y los periféricos de entrada, reemplazando al cableado tradicional. El PLC consulta cíclicamente el estado de los periféricos de entrada y actualiza el estado de los periféricos de salida.

1.1.1.4. VENTAJAS Y DESVENTAJAS

No todos los PLC's tienen iguales prestaciones debido a las innovaciones técnicas que surgen constantemente. Pero en un PLC promedio se pueden observar las siguientes ventajas:

VENTAJAS

- × No es necesario dibujar el esquema de contactos.
- × No es necesario simplificar las ecuaciones lógicas, ya que, por lo general la capacidad de almacenamiento del módulo de memoria es lo suficientemente grande.
- × La lista de materiales queda sensiblemente reducida.
- × Da la posibilidad de introducir modificaciones sin cambiar el cableado ni añadir equipos adicionales.
- × Ocupa un espacio físico reducido.
- × Economía de mantenimiento.

- × Aumentar la fiabilidad del sistema, al eliminar contactos dinámicos, los mismos autómatas pueden indicar y/o detectar averías.
- × Posibilidad de gobernar varios equipos con un mismo PLC.
- × Si por alguna razón el equipo controlado queda fuera de servicio, el PLC sigue siendo útil para otro equipo similar.

DESVENTAJAS

La principal desventaja es que se hace necesario un programador, lo que obliga a capacitar a una persona para esta tarea, aunque hoy en día ese inconveniente no es tan influyente ya que las universidades ya se encargan de dicha capacitación a sus ingenieros y técnicos. Otra desventaja puede ser la inversión inicial, ya que puede ser muy elevada.

1.1.2. SISTEMA SCADA

El acrónimo SCADA significa en inglés *Supervisory Control And Data Acquisition* que traduce Sistema de Supervisión, Control y Adquisición de Datos. Su nombre nos indica que no es un sistema completo de control, mas bien se enfoca en el campo de la supervisión, es decir, un software sobre un hardware intercomunicado mediante PLC's u otros módulos de hardware comerciales, basado en este concepto se estructuro el software de supervisión.

Los sistemas SCADA han cambiado a través de los años, mejorando su confiabilidad, funcionalidad y recientemente se puede observar su uso para el desarrollo en sistemas de supervisión para la casa.

1.1.2.1. ARQUITECTURA

La arquitectura de un Sistema SCADA consiste en su hardware, software, comunicación, interfaz, escalabilidad y redundancia.

HARDWARE

En un sistema SACADA se distinguen dos capas diferentes que son la capa cliente que se refiere a la interacción del usuario con el sistema y la capa de servidor de datos que se encarga de los procesos relacionados con el control de la información.

Los servidores de datos se comunican en el campo a través de controladores de proceso como lo son PLC's los cuales se comunican mediante buses de campo de tipo propietario (Ej. Siemens H1) o no-propietario (Ej. Profibus). Estos servidores de datos son conectados entre si mediante Ethernet LAN.

SOFTWARE

Es un producto multi-tarea y esta establecido en bases de datos de tiempo real (RTDB) y localizadas en uno o mas servidores de datos. Los servidores son responsables de la adquisición de datos y manejan los parámetros y la configuración usada para conectarse.

Generalmente los servidores son de tipo dedicados, es decir son usados únicamente para las tareas mencionadas anteriormente.

COMUNICACIÓN

Comunicación Interna

La comunicación interna se refiere a la comunicación Cliente – Servidor y Servidor – Servidor usando el protocolo TCP/IP¹

Acceso a Dispositivos

Los servidores de datos se comunican con los controladores usando protocolos definidos. Los protocolos pueden requerir diferentes parámetros, con esto garantizan que la información pase de los controladores o PLC's a los servidores

¹ TCP/IP: Protocolo de control de transmisión/Protocolo de Internet, Conjunto de reglas que gobiernan el intercambio de información entre ordenadores conectados a Internet.

de datos. Existen protocolos ya definidos en la industria y los mas comúnmente usados en los PLC's son llamados field-buses o buses de campo como por ejemplo Modbus. Algunos servidores de datos pueden soportar varios protocolos de comunicación, estos generalmente portan tarjetas adicionales fabricadas para este fin.

INTERFAZ

La OPC² se ha encargado de definir interfases abiertas para el acceso a los dispositivos de manera estandarizada y económica. Entre los productos estandarizados por OPC están:

- × ODBC (*Open Data Base Connectivity*): Interfaz para gestión con bases de datos.
- × ASCII (*American Standard Code for Information Interchange*): Importación y exportación de datos.
- × API (*Application Programming Interface*): Soportado por VB, C, C++, acceso y gestión en Windows.

Otros productos que proporciona Microsoft como *Dynamic Data Exchange* (DDE) por ejemplo para la exportación a Excel, *Dynamic Link Library* (DLL) y *Object Linking and Embedding* (OLE) entre otros también hacen parte de los estándares de OPC.

ESCALABILIDAD

Escalabilidad se entiende como la posibilidad de expandir el sistema SCADA añadiendo más dispositivos y variables a controlar

REDUNDANCIA

² OPC: estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar.

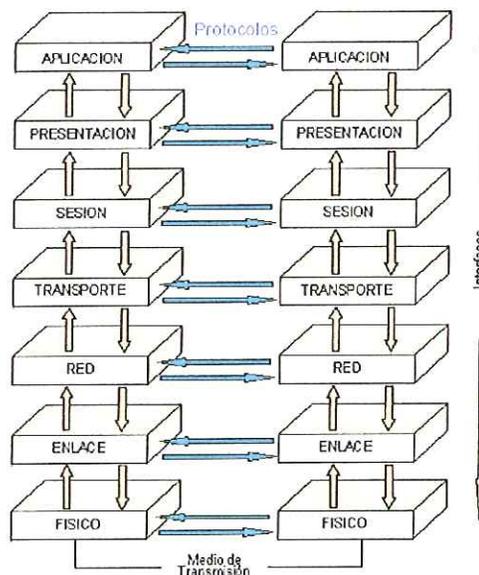
Generalmente el software es redundante a nivel del servidor y es transparente al usuario, facilitando la interpretación de la información para el mismo.

1.2. FUNDAMENTOS TEORICOS

1.2.1. MODELO OSI

Respondiendo a la teoría general imperante el mundo de la computación, de diseñar el hardware por módulos y el software por capas, en 1978 la organización ISO, propuso un modelo de comunicaciones para redes al que titularon "*The Reference Model of Open Systems Interconnection*", generalmente conocido como modelo OSI. Su filosofía se basa en descomponer la funcionalidad de la cadena de transmisión en diversos módulos, cuya interfaz con los adyacentes esté estandarizada.

Imagen 2. Modelo OSI



ietf.org

1.2.1.1. CAPAS O NIVELES DEL MODELO OSI

El Modelo OSI se basa en siete niveles definidos que realizan funciones concretas, estos interactúan entre sí por medio de distintas interfaces sin que un nivel afecte el sus adyacentes.

CAPA FÍSICA

Esta capa es la encargada de enviar la información por un medio de transición. Se ocupa de la parte física (características electrónicas, mecánicas, ópticas, etc.) de la comunicación. En resumen la finalidad de la capa física a nivel de transmisión es convertir la información binaria en impulsos adecuados para la entrega y a nivel de recepción, se encarga de convertir dichos impulsos en información para entregarla a la siguiente capa. Por ejemplo, en este nivel se define las medidas y el tipo de cable y de los conectores utilizados en una conexión.

CAPA DE ENLACE

Esta capa se encarga de mover los datos hacia/desde la capa física y la capa de red. Especifica como se organizan los datos cuando se transmiten. Por ejemplo esta capa define como son los *frames*, las direcciones y las sumas de control de los paquetes Ethernet.

Además del direccionamiento local, se ocupa de la detección y control de errores ocurridos en la capa física, del control del acceso a dicha capa y de la integridad de los datos y fiabilidad de la transmisión. Los datagramas recibidos son comprobados por el receptor. Si algún datagrama se ha corrompido se envía un mensaje de control al remitente solicitando su reenvío. El protocolo PPP³ es ejemplo de esta capa.

³ PPP: Protocolo Punto a Punto, protocolo utilizado en Internet para establecer enlace entre dos ordenadores remotos.

CAPA DE RED

Se ocupa de la transmisión de los datagramas o paquetes y de encaminar cada uno en la dirección adecuada, tarea esta que puede ser complicada en redes grandes como Internet, pero no se ocupa para nada de los errores o pérdidas de paquetes. Por ejemplo, define la estructura de direcciones y rutas de Internet. A este nivel se utilizan dos tipos de paquetes: paquetes de datos y paquetes de actualización de ruta. Como consecuencia esta capa puede considerarse subdividida en dos:

- × Transporte: encapsula los datos a transmitir. Utiliza los paquetes de datos. en esta categoría se encuentra el protocolo IP
- × Conmutación: es la encargada de intercambiar información de conectividad específica de la red. Los *routers* son dispositivos que trabajan en este nivel y se benefician de estos paquetes de actualización de ruta. En esta categoría se encuentra el protocolo ICMP⁴, responsable de generar mensajes cuando ocurren errores en la transmisión y de un modo especial de eco que puede comprobarse mediante PING⁵

CAPA DE TRANSPORTE

Garantiza la fiabilidad del servicio, describe la calidad y naturaleza del envío de datos. Por ejemplo, esta capa define cuando y como debe utilizarse la retransmisión para asegurar su llegada. Para ello divide el mensaje recibido de la capa de sesión en datagramas, los numera correlativamente y los entrega a la capa de red para su envío. Durante la recepción, si la capa de Red utiliza el protocolo IP, la capa de Transporte es responsable de reordenar los paquetes

⁴ ICMP: Protocolo Internet de Control de Mensajes, permite generar mensajes de error, paquetes de prueba y mensajes informativos relacionados con IP.

⁵ PING: Protocolo de Mensaje de Control de Internet, se utiliza para comprobar si un *host* está activo en la red.

recibidos fuera de secuencia. También puede funcionar en sentido inverso multiplexando una conexión de transporte entre diversas conexiones de datos. Este permite que los datos provenientes de diversas aplicaciones compartan el mismo flujo hacia la capa de red.

Un ejemplo típico de protocolo usado en esta capa es TCP que con su homólogo IP de la capa de Red, configuran la suite TCP/IP utilizada en Internet, aunque existen otros como UDP⁶ una capa de transporte utilizada también en Internet por algunos programas de aplicación.

CAPA DE SESIÓN

Es una extensión de la capa de transporte que ofrece control de diálogo y sincronización, aunque en realidad son pocas las aplicaciones que hacen uso de ella. Por ejemplo, las comunicaciones de Internet no la utilizan. La capa de sesión tiene la responsabilidad de asegurar la entrega correcta de la información. Esta capa tiene que revisar que la información que recibe este correcta; para esto, la capa de sesión debe realizar algunas funciones:

- × La detección y corrección de errores.
- × El controlar los diálogos entre dos entidades que se estén comunicando y definir los mecanismos para hacer las llamadas a procedimientos remotos (*Remote Procedure Control* - RPC).

CAPA DE PRESENTACIÓN

Esta capa se ocupa de los aspectos semánticos de la comunicación, estableciendo los arreglos necesarios para que puedan comunicar máquinas que utilicen diversa representación interna para los datos. Por ejemplo, describe como pueden transferirse números de coma flotante entre equipos que utilizan distintos

⁶ UDP: Protocolo de datagrama a nivel de usuario, protocolo donde se envían paquetes sueltos sin mantener una conexión continua, por ser abierto el usuario experto es quien define su tipo de paquete.

formatos matemáticos. Esta capa es buena candidata para implementar aplicaciones de criptografía.

En teoría esta capa "presenta" los datos a la capa de aplicación cogiendo los datos recibidos y transformándolos en formatos como texto imágenes y sonido. Como veremos a continuación, en realidad esta capa puede estar ausente, ya que son pocas las aplicaciones que hacen uso de ella.

En Internet, el único servicio que utiliza esta capa es TELNET⁷, que precisamente es un servicio de acceso a servidores desde terminales remotos. En este caso, la capa de presentación es la que se encarga de configurar el Terminal para conectar a un servidor de características particulares.

CAPA DE APLICACIÓN

Esta capa describe como hacen su trabajo los programas de aplicación (navegadores, clientes de correo, terminales remotos, transferencia de ficheros etc.). Por ejemplo, esta capa implementa la operación con ficheros del sistema. Por un lado interactúan con la capa de presentación; por otro representan la interfaz con el usuario, entregándole la información y recibiendo los comandos que dirigen la comunicación.

Ejemplos de protocolos utilizados por los programas de esta capa son HTTP⁸, SMTP⁹, POP¹⁰, IMAP¹¹, etc.

⁷ TELNET: Es el protocolo estándar de Internet que permite la conexión a un terminal remoto.

⁸ HTTP: Protocolo para transferir archivos o documentos hipertexto a través de la red.

⁹ SMTP: Protocolo simple de transferencia de correo electrónico, protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras y/o distintos dispositivos.

¹⁰ POP: Protocolo de Oficina de Correos, protocolo usado para la recuperación de correo electrónico. La versión actual es la 3 (POP3).

¹¹ IMAP: Protocolo Interactivo para el Acceso de Correos, mejora de POP3.

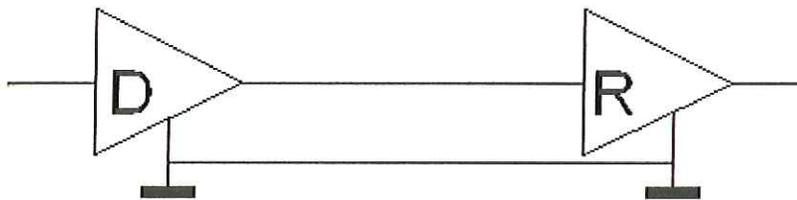
A continuación se definen algunos conceptos de componentes importantes durante el desarrollo del proyecto.

1.2.2. INTERFASES DE COMUNICACIÓN

Para facilitar la conexión entre DTE¹² y DCE¹³ se han desarrollado múltiples estándares que determinan todas las características físicas, eléctricas, mecánicas y funcionales de la conexión constituyendo lo que denominamos la definición de una interfaz. Estos estándares constituyen un ejemplo de los protocolos del nivel físico, y se encuadrarían en el nivel más bajo del modelo de referencia OSI.

1.2.2.1. ELECTRONIC INDUSTRIES ASSOCIATION RECOMMENDED STANDARD 232

Figura 1. Driver – Receiver RS-232



Del Autor

El RS-232C es una norma para la conexión entre un DTE y un DCE que define:

- × El tipo de conector a emplear.
- × Las características eléctricas.
- × Los niveles de tensión.
- × Las longitudes máximas a distintas velocidades.

¹² DTE: Equipo Terminal De datos, se refiere a cualquier dispositivo que esté preparado para recibir datos, aunque usualmente se refiere a un ordenador.

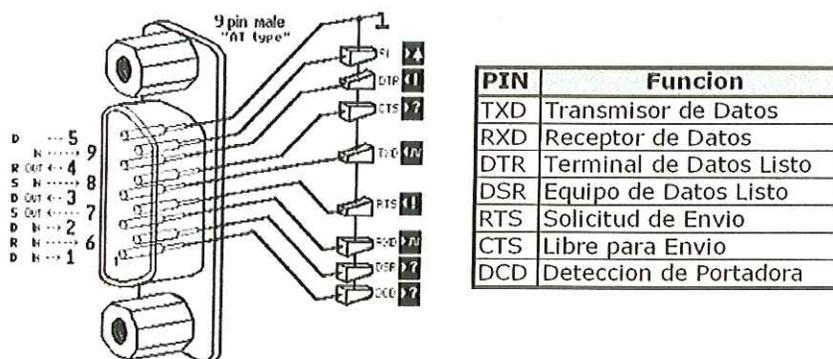
¹³ DCE: Equipo para Comunicaciones de Datos, se refiere a cualquier dispositivo que esté preparado para transmitir/recibir datos.

- × Los nombres de las señales que intervienen en el funcionamiento y la estructura del protocolo de comunicación.

Generalmente el RS-232C consiste en un conector tipo DB-25, aunque es normal encontrar la versión DB-9, mas barato e incluso mas extendido para cierto tipo de periféricos. Las señales con las que trabaja este puerto serie son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12V. Las velocidades de transmisión que puede soportar este estándar van desde los 0bps hasta los 20Kbps. Con respecto a las distancias máximas se propone que no sean superiores a 15 metros.

Cada pin tiene una función específica. Las más importantes se muestran en la Figura 2:

Figura 2. Pines RS-232

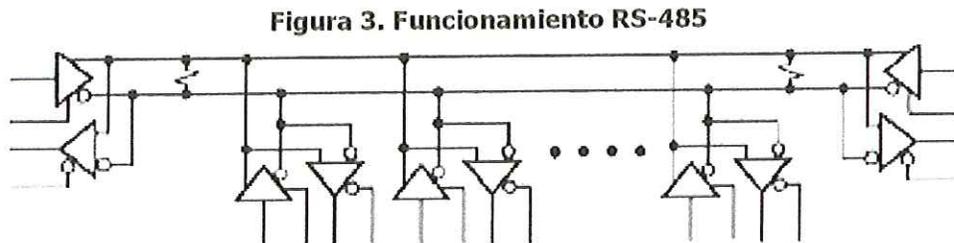


Del Autor

Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR, CTS y DCD son de entrada. La masa de referencia para todas las señales es SG (Tierra de Señal).

1.2.2.2. ELECTRONIC INDUSTRIES ASSOCIATION RECOMMENDED STANDARD 485

La *Electronic Industries Association* (EIA), en 1983 aprobó el nuevo estándar de transmisión diferencial al que llamo RS-485. El estándar RS485 especifica solamente las características eléctricas del *driver* y el *receiver* para ser utilizado en las líneas de comunicación, pero no especifica ni recomienda ningún protocolo (Figura 3).



Jordi Mayne Grau – Sistemas de Comunicaciones Industriales

La interfaz RS-485 es ideal para conexiones multi-terminal (donde hay múltiples drivers). Pueden tolerar una diferencia de potencial de hasta 7V. Son seguros y térmicamente protegidos. Estos soportan hasta 32 trancceptores de carga, maneja velocidades de hasta 10Mb/s.

1.2.3. BUSES DE DATOS Y PROTOCOLOS

1.2.3.1. INTER IC BUS

El bus I²C fue desarrollado hace casi 20 años atrás por Philips para facilitar la comunicación entre componentes ubicados sobre una misma placa. El nombre traduce "Inter IC" o "Circuitos Inter.- Integrados".

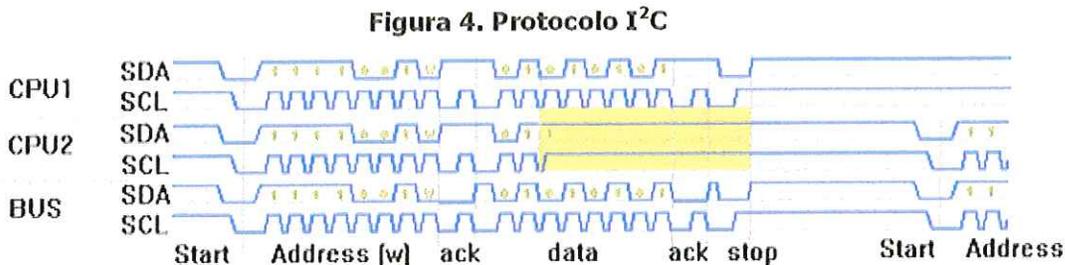
La velocidad original fue de 100Kbps ya que la mayoría de circuitos no necesitaban mayores velocidades de transmisión, pero en 1998 la velocidad aumenta de 400Kbps a 3.4Mbps

CARACTERÍSTICAS PRINCIPALES DEL BUS I²C

- × Solo dos líneas son requerías para la transmisión de datos
- × No requiere un rango de baudios como en RS-232 ya que la señal de reloj es generada por el maestro
- × Todos los componentes conectados al bus tienen su propia dirección de memoria
- × Permite conexiones multi - maestro para evitar colisiones

Para operar un esclavo sobre el Bus I²C solo son necesarios seis simples códigos, suficientes para enviar o recibir información (Figura 4).

- × Un bit de Inicio
- × 7-bit o 10-bit de direccionamiento
- × Un R/W bit que define si el esclavo es transmisor o receptor
- × Un bit de reconocimiento
- × Mensaje dividido en bytes
- × Un bit de Stop



ESAcademy - <http://www.esacademy.com/faq/i2c/general/i2carbit.htm>

CONDICIONES DE INICIO (START Y STOP)

Dentro del proceso de transferencia de datos en el Bus I²C hay dos situaciones básicas que son el Inicio y la Parada de toda transferencia de datos.

- × INICIO o Start: Una transición de "1" a "0" en la línea de datos mientras la línea del reloj está a "1".

- × PARADA o Stop: Una transición de "0" a "1" en la línea de datos mientras la línea de reloj esta a "1".

Las condiciones de Inicio y Stop son siempre generadas por el Maestro o Master. El Bus I²C se considera ocupado después de la condición de Inicio. El Bus se considera libre de nuevo después de un cierto tiempo tras la condición de Stop.

TRANSFIRIENDO DATOS

El número de bytes que se envíen a la línea SDA no tiene restricción. Cada byte debe ir seguido por un bit de reconocimiento, el byte de datos se transfiere empezando por el bit de mas peso (7) precedido por el bit de reconocimiento (ACK).

RECONOCIMIENTO

El bit de reconocimiento es obligatorio en la transferencia de datos. El pulso de reloj correspondiente al bit de reconocimiento (ACK) es generado por el Master. El Transmisor desbloquea la línea SDA ("1") durante el pulso de reconocimiento. El receptor debe poner a "0" la línea SDA durante el pulso ACK de modo que siga siendo "0" durante el tiempo que el master genera el pulso "1" de ACK.

Normalmente un receptor cuando ha sido direccionado esta obligado a generar un ACK después de que cada byte a sido recibido.

Cuando un dispositivo esclavo no genera el bit ACK (porque esta haciendo otra cosa y no puede atender el Bus) debe mantener el esclavo la línea SDA a nivel "1" durante el bit ACK. El Master entonces puede generar una condición de STOP abortando la transferencia de datos o repetir la condición de Inicio enviando una nueva transferencia de datos.

ARBITRAJE Y GENERACIÓN DE SEÑALES DE RELOJ

Sincronización

Todos los Master generan su propia señal de reloj sobre la línea SCL al transferir datos sobre el Bus I²C. Los bits de datos son solo validos durante los periodos "1" del reloj. Un control es necesario para mantener un orden en los diversos bits que se generan.

La sincronización del reloj se realiza mediante una conexión AND de todos los dispositivos del Bus a la línea SCL. Esto significa que una transición de un Master de "1" a "0" en la línea SCL hace que la línea pase a "0", esto mantiene la línea SCL en ese estado. Sin embargo la transición de "0" a "1" no cambia el estado de la línea SCL si otro reloj esta todavía en su periodo de "0". Por lo tanto la línea SCL permanecerá a "0" tanto como el periodo mas largo de cualquier dispositivo cuyo nivel sea "0". Los dispositivos que tienen un periodo mas corto de reloj "0" entran en un periodo de espera.

Cuando todos los dispositivos conectados al Bus han terminado con su periodo "0", la línea del reloj se desbloquea y pasa a nivel "1". Por lo que hay que diferenciar entre los estados de reloj de los dispositivos y los estados de la línea SCL, y todos los dispositivos empiezan a nivel "1". El primer dispositivo que completa su nivel "1" pone nuevamente la línea SCL a "0".

Arbitraje

Un master puede iniciar una transmisión solo si el bus esta libre. Dos o más master pueden generar una condición de Inicio en el bus lo que da como resultado una condición de Inicio general. Cada Master debe comprobar si el bit de datos que transmite junto a su pulso de reloj, coincide con el nivel lógico en la línea de datos SDA. El sistema de arbitraje actúa sobre la línea de datos SDA, mientras la línea SCL esta a nivel "1", de una manera tal que el master que transmite un nivel "1", pierde el arbitraje sobre otro master que envía un nivel

"0" a la línea de datos SDA. Esta situación continua hasta que se detecte la condición de Stop generada por el master que se hizo cargo del Bus

1.2.3.2. PROTOCOLOS ASCII¹⁴

Los protocolos ASCII son muy populares debido a su simplicidad, lo cual los hace apropiados para instalaciones sencillas, generalmente una Maestra y una Remota. Su principal desventaja es su lentitud y su incapacidad para manejar sistemas más complicados, por ejemplo, sistemas multipunto de más de 32 remotas.

En la práctica podemos encontrar dos tipos de protocolo ASCII: el protocolo ASCII para Transmisores Digitales y el protocolo ASCII ANSI¹⁵ X3.28-2.5-A4; este último es un poco más elaborado. La primera versión del protocolo ANSI X3.28 apareció en 1976.

PROTOCOLO ASCII PARA TRANSMISORES DIGITALES

En el comercio se encuentra una variedad de transmisores que aceptan la salida de sensores (de temperatura, flujo, densidad, etc.) los cuales procesan y transmiten asincrónicamente en un formato digital hacia un puerto serie de un procesador, algunas veces denominado "indicador". Generalmente en el procesador la información es almacenada y actualizada normalmente ocho veces por segundo y está disponible, si es solicitada, para ser enviada a un servidor de control. El transmisor digital puede también aceptar comandos desde el servidor.

Características

- × Control por Caracteres

¹⁴ ASCII: Código Estadounidense Estándar para el Intercambio de Información, código utilizado por las computadoras para representar los caracteres más habituales, como las letras, los números, los signos de puntuación o los caracteres de control.

¹⁵ ANSI: Instituto Nacional Estadounidense de Estándares

- × Transmisión HDX¹⁶ asincrónica.
- × Velocidades: entre 300 y 1200 bps.
- × Interfaces: RS-232C en operación punto a punto. Si acaso se llega a utilizar en operación multipunto, entonces hay que emplear la interfaz RS-485.

1.2.4. MICROCONTROLADOR

Circuito integrado o chip que incluye en su interior las tres unidades funcionales de un ordenador: CPU, Memoria y Unidades de E/S. Aunque sus prestaciones son limitadas, además de dicha integración, su característica principal es su alto nivel de especialización. Aunque los hay del tamaño de un sello de correos, lo normal es que sean incluso más pequeños, ya que, lógicamente, forman parte del dispositivo que controlan.

Un microcontrolador es un microprocesador optimizado para ser utilizado para controlar equipos electrónicos. Pueden encontrarse en casi cualquier dispositivo eléctrico como automóviles, lavadoras, hornos microondas, teléfonos, etc.

Un microcontrolador difiere de una CPU normal, debido a que es más fácil convertirla en un ordenador en funcionamiento, con un mínimo de chips externos de apoyo. La idea es que el chip se coloque en el dispositivo, enganchado a la fuente de energía y de información que necesite, y eso es todo. Un microprocesador tradicional no le permitirá hacer esto, ya que espera que todas estas tareas sean manejadas por otros chips.

Por ejemplo, un microcontrolador típico tendrá un generador de reloj integrado y una pequeña cantidad de memoria RAM¹⁷ y ROM¹⁸/EPROM¹⁹/EEPROM²⁰,

¹⁶ HDX: Half Duplex. No pueden recibir y enviar al mismo tiempo.

¹⁷ RAM: Memoria de Acceso Aleatorio.

¹⁸ ROM: Memoria de Solo Lectura.

¹⁹ EPROM: Memoria de solo lectura programable.

significando que para hacerlo funcionar, todo lo que se necesita son unos pocos programas de control y un cristal de sincronización. Los microcontroladores disponen generalmente también de una gran variedad de dispositivos de entrada/salida, como convertidores de analógico a digital, temporizadores, UART's²¹ y buses de interfaz serie especializados, como el I²C. Frecuentemente, estos dispositivos integrados pueden ser controlados por instrucciones de procesadores especializados.

Los microcontroladores negocian la velocidad y la flexibilidad para facilitar su uso. Debido a que se utiliza bastante sitio en el chip para incluir funcionalidad, como los dispositivos de entrada/salida o la memoria que incluye el microcontrolador, se ha de prescindir de cualquier otra circuitería.

²⁰ EEPROM: Memoria de sólo lectura programable y borrable eléctricamente.

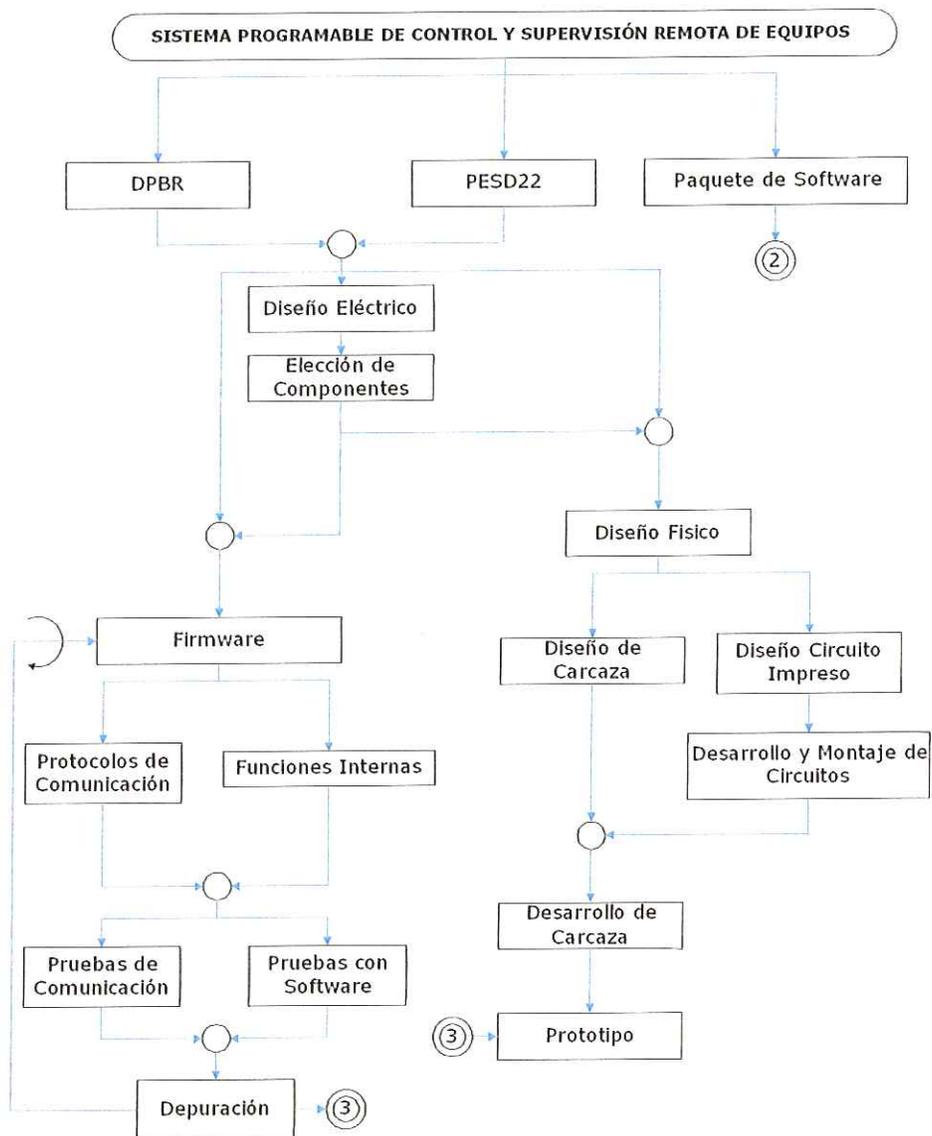
²¹ UART: Transmisor-Receptor Asíncrono Universal, es un bloque usado para convertir los datos en paralelo, que manda la CPU, en serie, con el fin de comunicarse con otro sistema externo. También realiza el proceso contrario, esto es, convierte los datos serie, recibidos de un sistema externo, en paralelo para ser procesados por la CPU.

2. DISEÑO

2.1. DISEÑO MECATRÓNICO

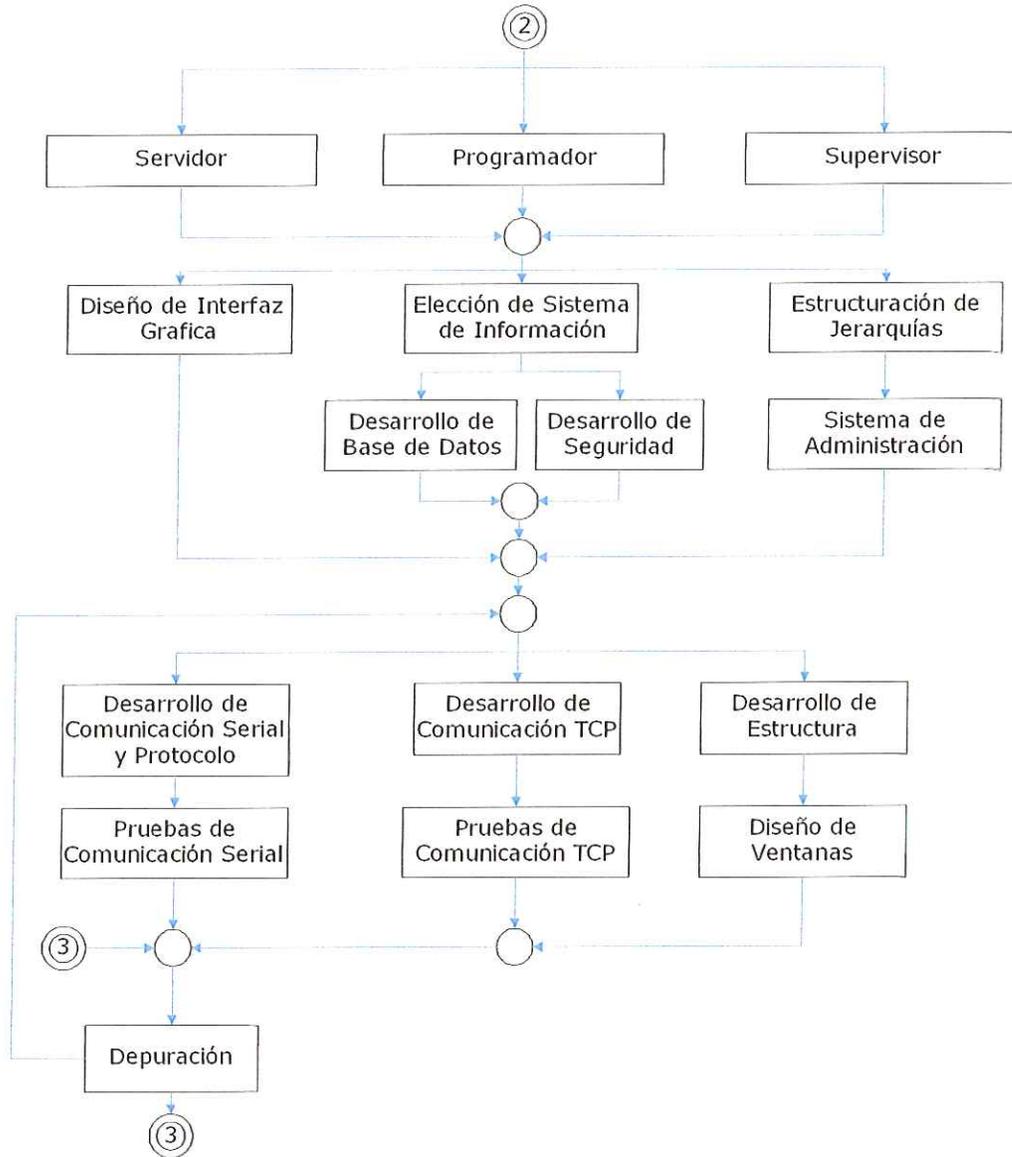
En el desarrollo de este proyecto se siguió el proceso de diseño de la Figura 5 y Figura 6.

Figura 5. Diseño Mecatrónico I



Del Autor

Figura 6. Diseño Mecatrónico II



Del Autor

Básicamente el Sistema Programable De Control Y Supervisión Remota De Equipos consta de tres elementos principales, el Dispositivo Programable Basado en Reglas (DPBR), el Periférico de Entradas y Salidas Digitales de 22 Puertos (PESD22) y el Paquete de Software (S&P). En este capítulo se expone el diseño de cada uno de estos componentes.

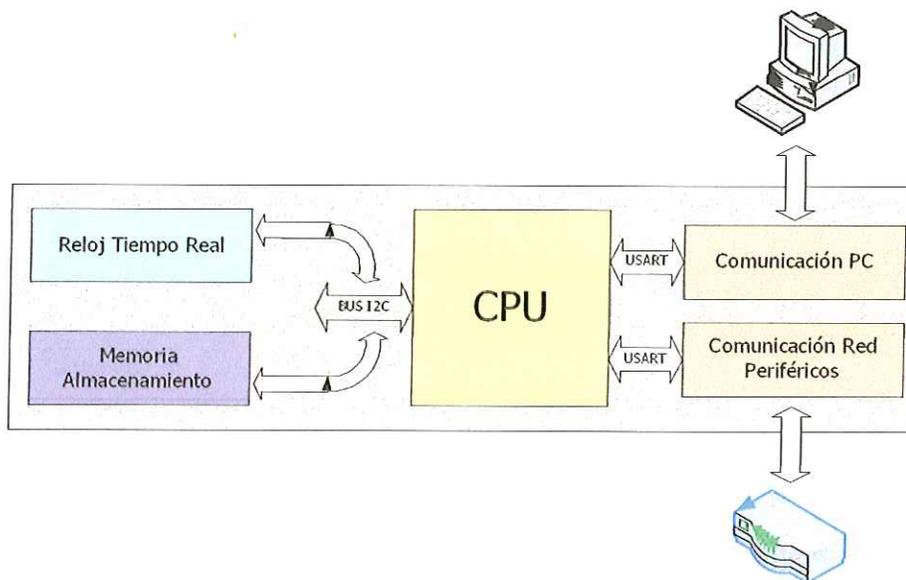
2.1.1.1. DISEÑO DE DPBR Y PESD22

El Dispositivo Programable Basado en Reglas es el administrador de la red de Periféricos obteniendo la información de sus entradas y actuando sobre sus salidas, dependiendo del paquete de reglas almacenadas en su memoria en donde también se almacenan la configuración de los periféricos y tiempos de encendido y apagado de equipos.

Los Periféricos, son dispositivos diseñados para entregar información del medio y actuar sobre el mismo atado a las órdenes recibidas por el DPBR, en este caso el Periférico de Entradas y Salidas Digitales de 22 Puertos, puede tomar señales digitales del medio para entregarlas al DPBR y este le envía las acciones correspondientes a sus reglas.

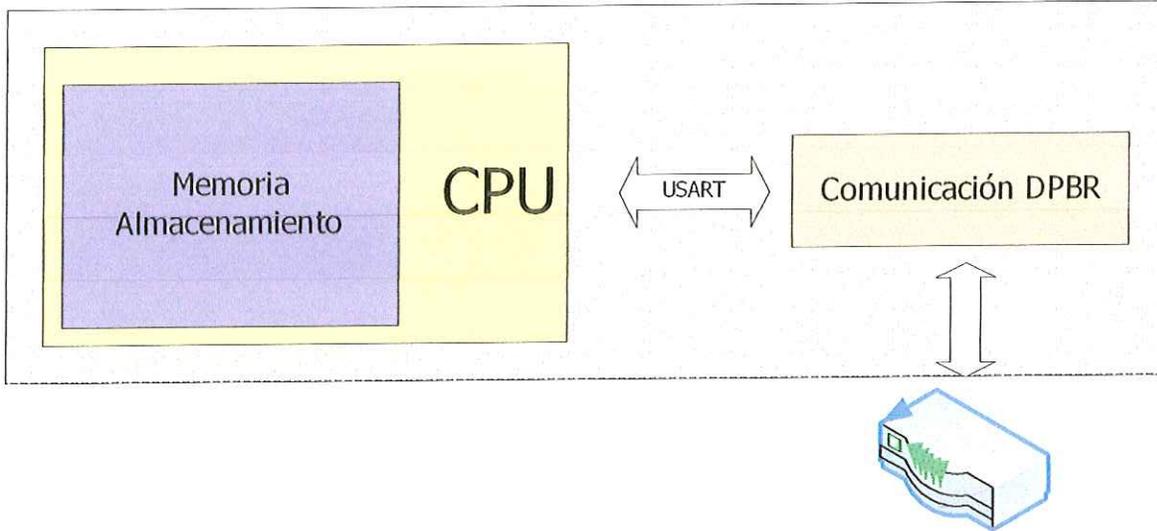
2.1.1.1.1. SELECCIÓN DE COMPONENTES

Figura 7. Diagrama de Bloques DPBR



Del Autor

Figura 8. Diagrama de Bloques PESD22



Del Autor

CPU

Para seleccionar la unidad central de procesamiento se tuvieron en cuenta criterios de costo, disponibilidad en el mercado local, funcionalidad (buses de comunicación, memoria) y desarrollador.

Siendo el costo la variable más influyente para la selección de los componentes, se descarta la opción de adquirir un sistema de desarrollo universal ya que el costo puede aumentar en casi \$1'500.000 adicionales a lo presupuestado, por lo tanto se adquirió el sistema de desarrollo "PPM PROGRAMADOR PARA PIC VS.ESTUDIANTIL" el cual alcanza un costo aproximado de \$100.000. Este desarrollador garantiza la programación de gran cantidad de microcontroladores de microchip, incluidas la serie 16FXXX, 12FXX.

Tabla 1. Selección CPU

		16F819	16F84A	16F873A	16F877A
Almacenamiento	EEPROM	256	64	128	256
	Flash (Kbytes)	3,5	1,75	7	14
Puertos	Entradas /Salidas	16	13	22	33
Comunicación	I2C	SI	NO	SI	SI
	USART	NO	NO	SI	SI
Velocidad	Externa Max (Mhz)	20	20	20	20
	Interna (Mhz)	8	-	-	-
Costo	Pesos Colombianos	\$ 8.600,00	\$ 8.000,00	\$ 17.400,00	\$ 16.000,00

Del Autor

Se estudiaron cuatro microcontroladores de la serie 16 de microchip debido a su disponibilidad en el mercado local y las especificaciones técnicas más puntuales para el diseño según se muestra en la tabla 1. Basados en este análisis se realizó la siguiente selección:

- × Para el DPBR fue seleccionado el integrado PIC16F877²² debido que maneja comunicación USART e I²C, 33 puertos de entrada y salida suficientes para la comunicación y los indicadores de funcionamiento mencionados en la pagina 109 y a la cantidad de memoria suficiente para programar todo el firmware sin correr el riesgo de llenar su memoria de programación.

²² Véase Anexo 1, pagina 121

- × Para el PESD22 fue seleccionado el integrado PIC16F877 al igual que el DPBR debido a la cantidad de puertos que este contiene, a la memoria EEPROM interna de 256 registros y a la comunicación USART utilizada para la red.

MEMORIA

Para seleccionar este componente importante para el funcionamiento del DPBR se realizo un proceso de selección basado en la disponibilidad en el mercado local, costo, interfaz de comunicación (I²C) y capacidad de almacenamiento.

Tabla 2. Selección Memoria

	24LC256	24LC64	24LC16B
Almacenamiento (Kbit)	256	64	16
Ciclos (Borrado/Escritura)	1.000.000	1.000.000	1.000.000
Ciclo Escritura (ms)	5	5	5
Cascada	8	8	NO
Protección Escritura	SI	SI	SI
Costo	\$ 6.500,00	\$ 10.000,00	\$ 4.200,00

Del Autor

Debido al diseño del Firmware (véase "Diseño de Firmware" pagina 44) el sistema necesita 11 bits por regla de funcionamiento, 6 bits para almacenar condiciones y 8 bits para almacenar la bitácora interna como se ve en las tablas 3,4 y 5 de la pagina 46 y en la tabla 6 de la pagina 49.

Para evitar limites en el sistema se escogió la memoria 24LC256²³ con la cual se pueden almacenar mas de 1200 reglas, mas de mil configuraciones y mas de 1500 registros de bitácora, además esta se puede usar en cascada por lo que se

²³ Véase Anexo 1, pagina 122

pueden almacenar hasta 8 veces estas cantidades sin alterar drásticamente el hardware.

COMUNICACIÓN

La comunicación entre los sistemas se divide en dos partes. Primero la comunicación entre el PC y el DPBR que se realiza mediante la interfaz RS232 que maneja niveles entre 11V (Bajos) y -11V (Altos) y debido a que estos niveles no son adecuados para el microcontrolador seleccionado, se requiere de un *transceiver* que realice esta conversión. Para esto se selecciono el integrado MAX232²⁴ el cual tiene varios equivalentes como el HIN232, ICL232, ADM232, etc. de los cuales solo difieren entre si por el fabricante que los desarrolla.

Para la comunicación del DPBR con la Red de Periféricos se uso la interfaz RS485 ya que teóricamente puede alcanzar distancias de más de 1000m y puede trabajar en lugares de gran interferencia electromagnética, para esto se hizo necesario el uso de un *transceiver* que sirviera como *driver* de comunicaciones. Para esto se selecciono el integrado MAX485²⁵ el cual al igual que el MAX232 tiene varios equivalentes de diferentes fabricantes.

RTC

El Reloj de Tiempo Real, otro importante componente del DPBR es el encargado de suministrar una entrada de datos para las diferentes reglas que se rigen por condiciones horarias y para almacenar la bitácora.

El proceso de selección del RTC no contó con varios componentes compatibles ya que en el mercado local solo se cuenta con el DS1307²⁶, este maneja una interfaz de comunicación mediante protocolo I²C y adicionalmente cuenta con un puerto

²⁴ Véase Anexo 1, Pagina 123

²⁵ Véase Anexo 1, Pagina 124

²⁶ Véase Anexo 1, pagina 125

de batería de respaldo para almacenar la hora aun cuando el sistema no ningún tipo de alimentación directa por fuente.

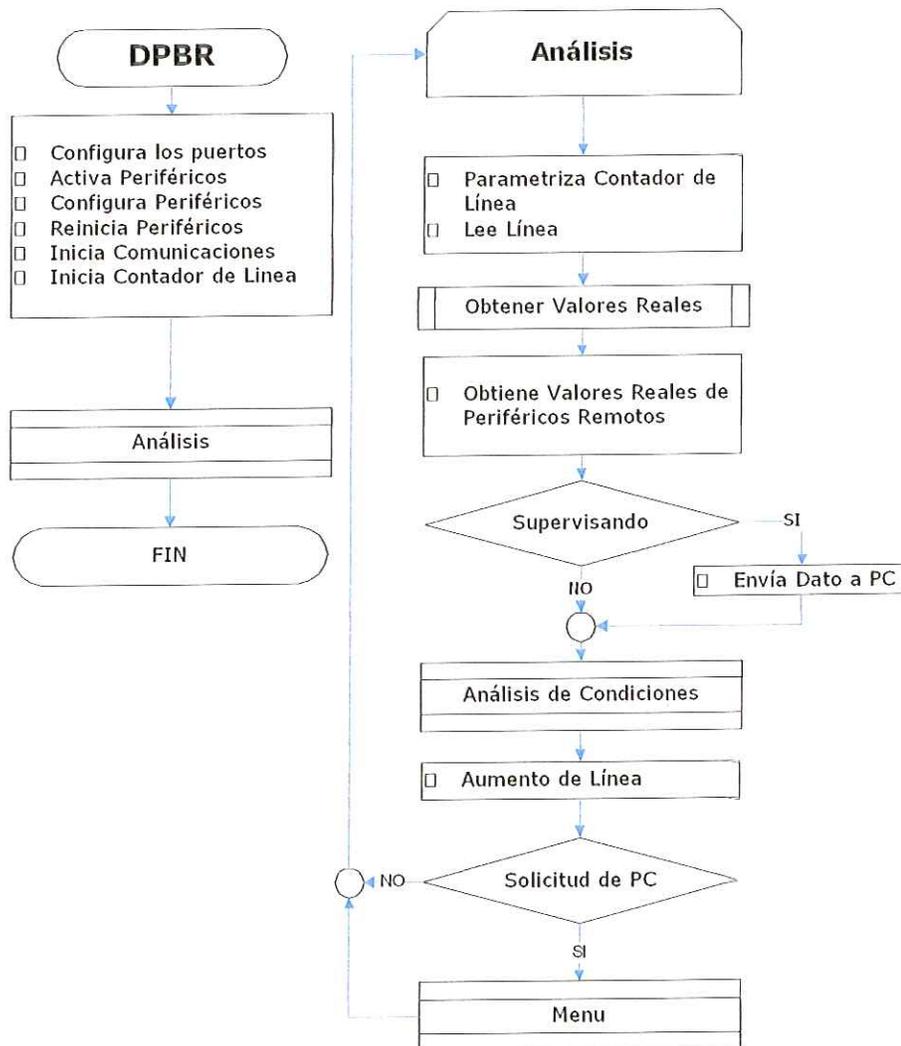
2.1.1.2. DISEÑO DE FIRMWARE

Los *Firmware's* fueron desarrollados con el software *MPLAB Integrated Development Environment* creado por Microchip®.

FIRMWARE DPBR

Para el desarrollo del Firmware del DPBR se siguió el algoritmo mostrado en de la Figura 9:

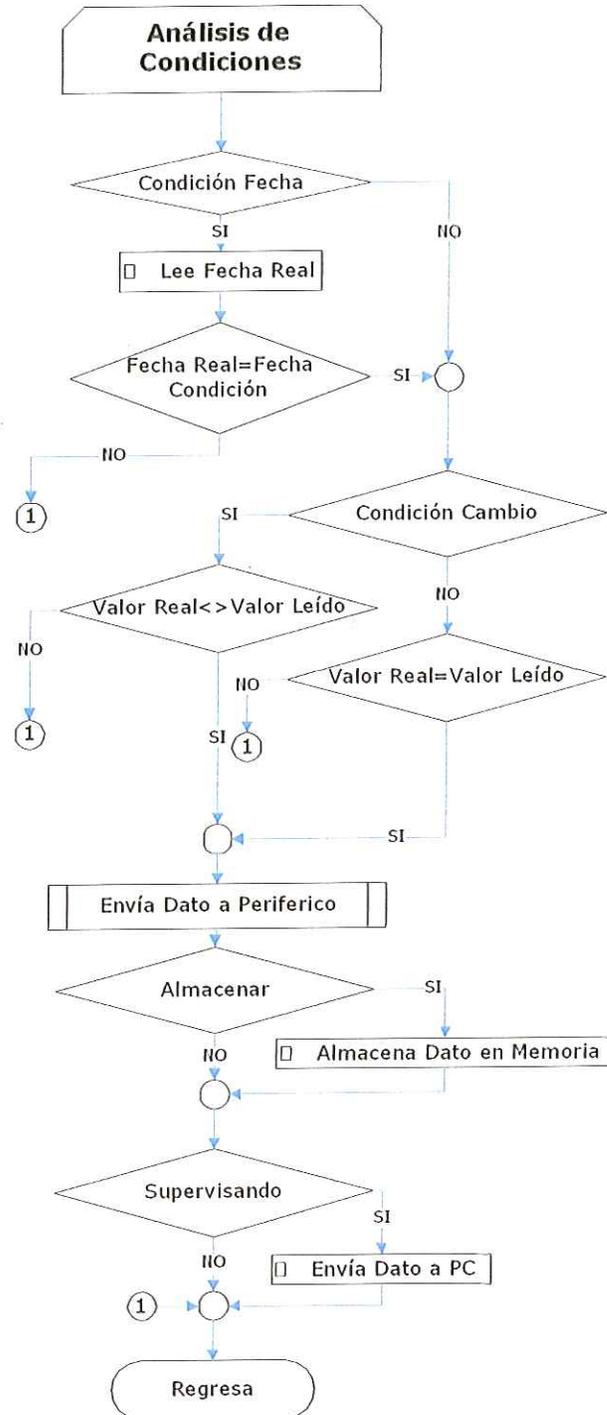
Figura 9. Algoritmo Firmware DPBR I



Del Autor

El programa comienza por configurar los puertos, la comunicación y los periféricos. Para el análisis de condiciones se usa un criterio basado en una tabla

Figura 10. Módulo Análisis de Condiciones DPBR



de 11 registros almacenados en la memoria EEPROM del DPBR.

Tabla 3. Regla de Funcionamiento

Periférico a Evaluar			Condición Horaria				Periférico a Actuar			
ID Periférico	Pto Periférico	Estado	Registro Especial	Hora I	Minuto I	Hora F	Minuto F	Id Periférico	Pto Periférico	Estado

Del Autor

Según el algoritmo del Análisis de condiciones, el sistema revisa el estado del puerto de que va a evaluar (Obtener Valores Reales) y según el registro especial revisa la condición horaria y/o si debe actuar cuando existe un cambio de estado (Alto – Bajo – Alto). Si al final del análisis la prueba fue satisfactoria entonces el DPBR actúa sobre el periférico especificado en "Periférico a Actuar" colocando el

Tabla 4. Registro Especial

SPE	SPS		C	R	B	>>>	B	Bandera de Bitacora	
7	6	5	4	3	2	1	0	R	Bandera de Reloj
								C	Bandera de Cambio
								SPS	Sub puerto de Salida (A-D)
								SPE	Sub Puerto de Entrada (A-D)

Del Autor

estado que el usuario determino previamente, adicionalmente envía el dato al PC si esta tiene activada la bandera de supervisión y lo almacena si el bit de bitácora esta en alto en un registro de 8 bits según se muestra en la tabla 5

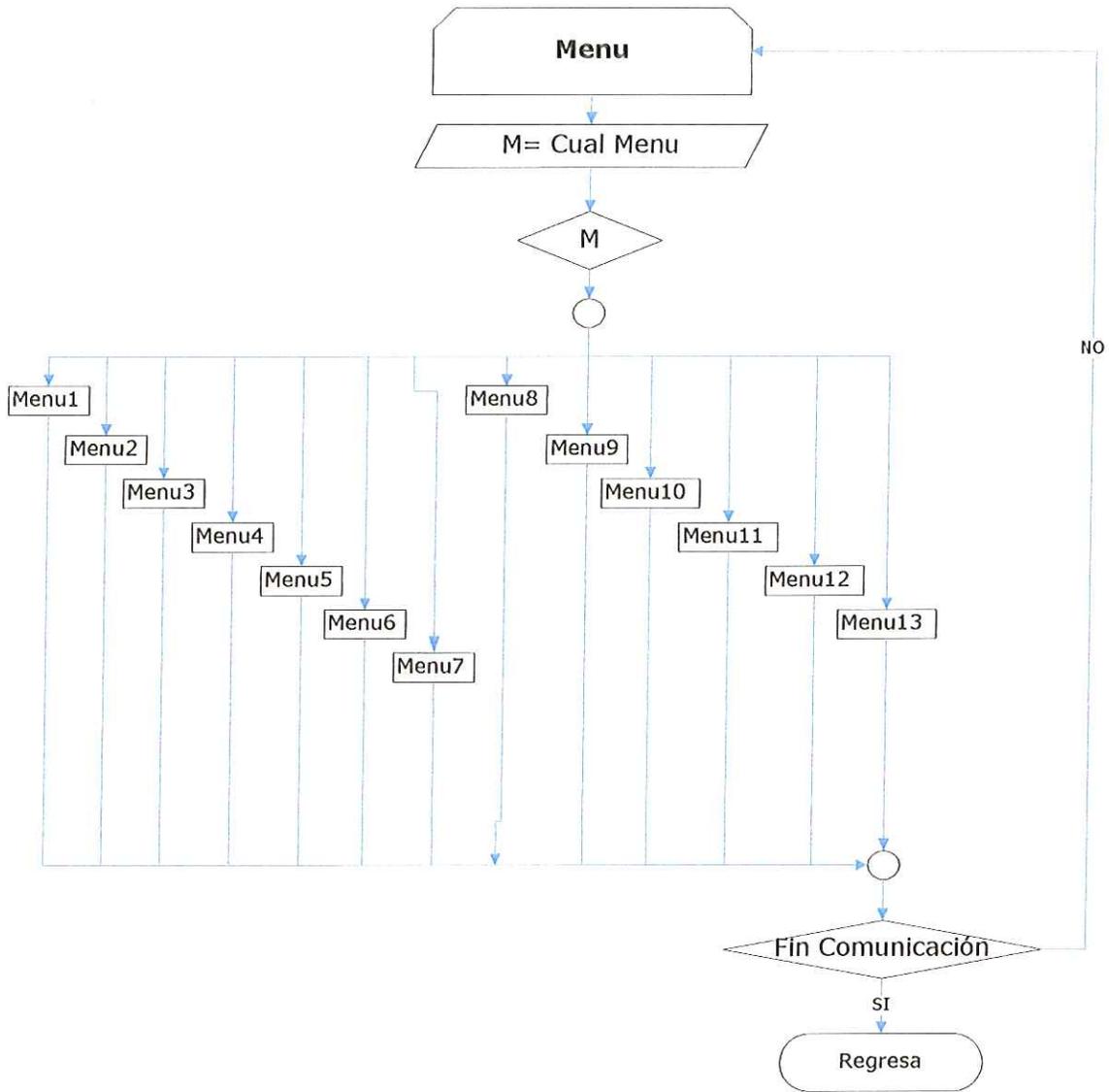
Tabla 5. Registro de Bitácora

Día	Mes	Año	Hora	SP	Minuto	Periferico	Puerto	Estado
-----	-----	-----	------	----	--------	------------	--------	--------

Del Autor

Si el PC solicita comunicación va al módulo de Menú (Figura 11).

Figura 11. Menú PC ⇔ DPBR

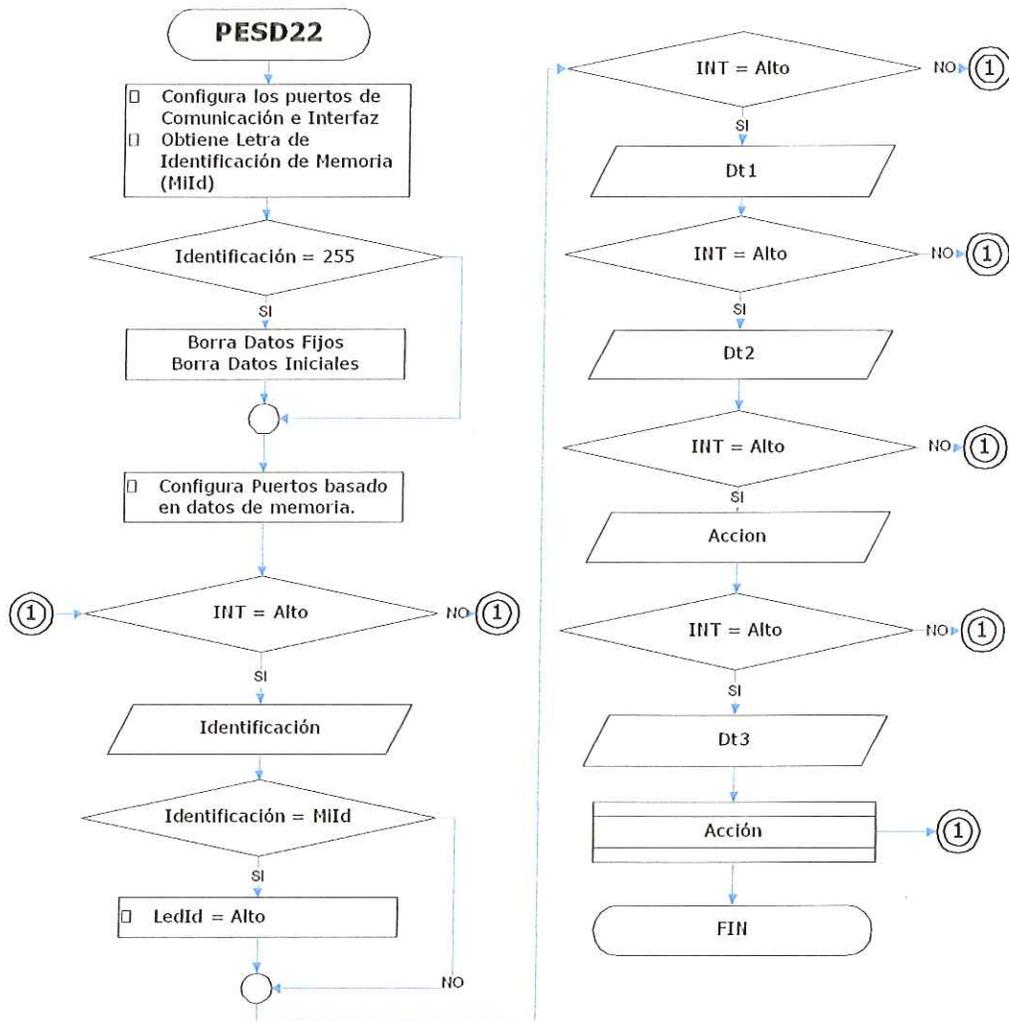


Del Autor

FIRMWARE PESD22

El Periférico de Entradas y Salidas Digitales de 22 Puertos, fue diseñado bajo el algoritmo de la Figura 12.

Figura 12. Algoritmo PESD22



Del Autor

El PESD22 inicialmente se configura obteniendo datos previamente almacenados por el DPBR según el registro de configuración (Tabla 6) y obtiene su letra de Identificación de la información obtenida en su memoria. Si detecta que no se ha asignado una Identificación elimina los datos fijos y los datos iniciales.

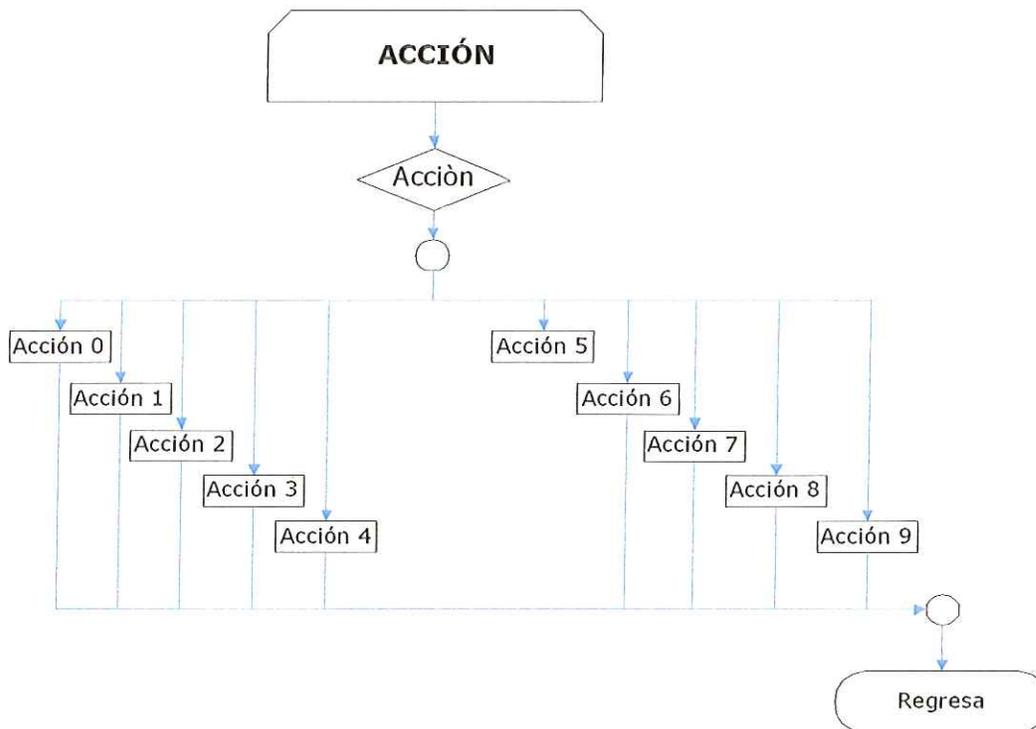
Tabla 6. Registro de Configuración

ID Periférico	Sub A	Sub B	Sub C	Sub D	Tipo
---------------	-------	-------	-------	-------	------

Del Autor

Luego espera a que el bit de inicio de transferencia (INT) este alto (Figura 12). Mientras este bit se encuentre en alto, el Periférico obtiene todos los datos del protocolo. Durante el proceso, este revisa si la letra de identificación obtenida mediante comunicación coincide con su letra de identificación, si esta revisión fue satisfactoria, el indicador luminoso de identificación se pone en Alto.

Figura 13. Acciones PESD22



Del Autor

Las acciones son tareas que se llevan a cabo después de recibir los 5 datos (véase pagina 52 "DPBR ⇔ Periféricos") provenientes del protocolo.

2.1.1.3. DISEÑO DE PROTOCOLOS DE COMUNICACIÓN:

Los protocolos que se utilizan en las comunicaciones son una serie de normas que deben aportar las siguientes funcionalidades:

- × Permitir localizar un dispositivo de forma inequívoca.
- × Permitir realizar una conexión entre dispositivos.
- × Permitir intercambiar información entre dispositivos de forma segura, independiente del tipo de equipos que estén conectados.
- × Permitir liberar la conexión de forma ordenada.

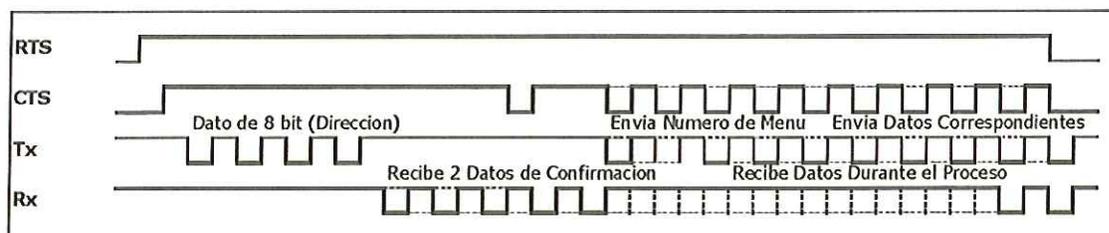
Durante el desarrollo de este proyecto se diseñaron dos tipos de protocolos con el fin de administrar la comunicación entre el PC ↔ DPBR y el DPBR ↔ Periféricos.

Para aislar la RED de DPBR's y la RED de Periféricos cada una de estas redes se administra por un canal de comunicación diferente.

DPBR ↔ PC

El protocolo que garantiza la correcta comunicación entre el DPBR y el PC es el siguiente según se muestra en la Figura 14:

Figura 14. Protocolo PC ↔ DPBR



Del Autor

- × El PC pone el bit de RTS en Alto ↑
- × El DPBR responde poniendo el bit CTS del PC en Alto ↑
- × El PC envía un dato para establecer comunicación.
- × El DPBR envía dos datos de confirmación.

- × Establecida la Comunicación el DPBR esta a la espera de los datos y va al módulo de MENÚ (Figura 11, pagina 47).
- × Durante el proceso el DPBR envía datos al PC y el PC envía al DPBR los datos correspondientes al Menú donde se acceso. El DPBR cada vez que envía un dato genera un pulso en CTS.
- × El PC pone RTS en Bajo ↓
- × El DPBR continúa con su proceso de Análisis.

Menú de Comunicación DPBR ⇔ PC

Si durante el proceso de análisis el PC requiere realizar un cambio u obtener algún tipo de información con el DPBR entonces este entra al módulo de MENÚ (Figura 11).

- × Menú 1 (Sincronizar Fecha): El propósito de este menú, es configurar la hora del DPBR (Hora en el RTC). El PC tiene que enviar en formato BCD²⁷ los segundos, minutos, horas, día de la semana, día, mes y año.
- × Menú 2 (Solicitar Fecha): Con esta orden el DPBR envía su fecha al PC.
- × Menú 3 (Guardar Dato en Memoria): El propósito de este menú es almacenar un dato en la memoria, banco y dirección seleccionada.
- × Menú 4 (Leer Dato en Memoria): El propósito de este menú es obtener el dato guardado en la memoria, banco y dirección seleccionada.
- × Menú 5 (Enviar Datos Condiciones): El propósito de este menú es obtener la información almacenada en los bancos seleccionados para las reglas, esto sirve para comprobar que los datos estén correctamente almacenados cuando se programa el paquete de reglas.
- × Menú 6 (Enviar Datos Configuración): El propósito de este menú es obtener la información almacenada en los bancos seleccionados para la

²⁷ BCD: Código binario decimal utilizado para almacenar números enteros positivos y facilitar las operaciones aritméticas.

configuración de las condiciones, sirve para comprobar que los datos estén correctamente almacenados cuando se programa el paquete de reglas.

- × Menú 7 (Enviar Fin de Datos): El propósito de este menú es obtener el número de la última memoria, banco y dirección de las condiciones, configuraciones y datos almacenados durante el proceso.
- × Menú 8 (Enviar Dato a Red): El propósito de este menú es el de enviar y recibir datos de la red de periféricos, con este menú se convierte la comunicación obtenida desde el PC a el protocolo de comunicación.
- × Menú 9 (Configuración de Periféricos): Este menú activa una macro que configura los periféricos que estén almacenados en los bancos de configuración.
- × Menú 10 (Descarga de Memorias): Este menú activa una macro que envía al PC todos los datos almacenados durante el proceso de Análisis.
- × Menú 11 (Reinicio de Comunicación): Este menú genera un reset en los periféricos.
- × Menú 12 (Carga de Velocidades): Este menú activa una macro que configura la velocidad la comunicación del sistema.
- × Menú 13 (Velocidad Predeterminada): Este menú establece una velocidad predeterminada de 9600 baudios para todo el sistema.

Luego de ejecutar las tareas el evalúa si el PC sigue en modo de programación (RTS en Alto), para luego volver al módulo de análisis y seguir evaluando las condiciones.

DPBR ⇔ PERIFÉRICOS

Usando la interfaz RS-485, se estableció el siguiente protocolo para la comunicación entre el dispositivo programable basado en reglas y sus Periféricos.

Físicamente se cuentan con un cable UTP²⁸ Categoría 5 (aunque se recomienda usar cable STP²⁹ ya que es mas reciente a ruidos generados por ondas electromagnéticas) con conectores tipo RJ45 Ensamblados bajo la Norma EIA/TIA 568B³⁰ en donde cada hilo tiene una función según la Tabla 7.

Tabla 7. Configuración de Cable DPBR ↔ Periféricos

Cable	Función
Blanco Naranja	-
Naranja	VCC
Blanco Verde	MCLR
Azul	A
Blanco Azul	B
Verde	INT
Blanco Marrón	GND
Marrón	GND

Del Autor

- × VCC³¹: Los periféricos son alimentador directamente por el DPBR mediante este hilo.
- × MCLR: reinicio de comunicaciones, este hilo esta en Alto por defecto. Cuando se desactiva este hilo, los periféricos sufren un reinicio general o reset. Este es activado cuando los periféricos sufren bloqueos generados por errores de comunicación o cuando se reconfiguran luego de una programación del DPBR.
- × A y B: Este par es usado para la Transmisión y Recepción de Información con el DPBR mediante la interfaz RS-485.

²⁸ UTP: Cable de par trenzado sin apantallar.

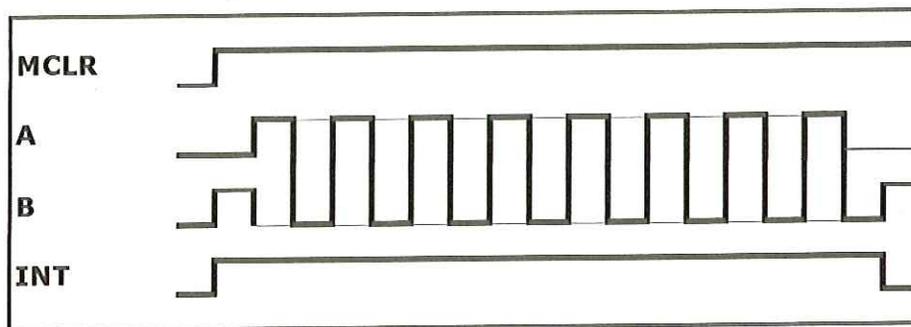
²⁹ STP: Par Trenzado Apantallado.

³⁰ Ver Anexo 2, pagina 126

³¹ VCC: Voltaje de colector común, sigla usada para la fuente de voltaje en un esquema eléctrico según ISO.

- × INT: inicio de transferencia, este hilo es usado para indicarle al periférico que un dato viene en camino, así se evitan bloqueos de comunicación si un periférico es conectado mientras el sistema este funcionando.
- × GND: Este par es usado para unir física y eléctricamente las tierras del DPBR y los Periféricos.
- × Los hilos que intervienen directamente en la comunicación son INT, A, B, MCLR como se ve en la Figura 15

Figura 15. Protocolo DPBR ↔ Periféricos



Del Autor

Para poder establecer una comunicación con los periféricos, el periférico pone constantemente en Alto el bit que corresponde al hilo de MCLR, como se menciona anteriormente el DPBR genera un pulso en MCLR cuando el PC se lo solicite o cuando configura los puertos de sus Periféricos.

Con los Periféricos funcionando el DPBR procede a poner el bit correspondiente al hilo INT en alto, esto genera que todos los Periféricos se pongan en nivel de escucha configurándose como *receiver*, luego el DPBR envía por las líneas de transmisión los siguientes datos:

- × Identificación: Dato que corresponde a la letra que identifica al periférico en la RED, esta letra puede ser asignada mediante el software de programación.
- × Dt1: Depende de la acción.

- × Dt2: Depende de la acción.
- × Acción: Dato correspondiente a la acción que se ejecutara en el Periférico (Depende del Periférico)
- × Dt3: Depende de la acción

Acciones en PESD22

Las acciones se ejecutan luego de recibir todos los datos según se muestra en la Figura 12.

- × Acción 0, escritura de Dato en Puerto: para esta acción Dt1 corresponde el sub.-puerto donde se va a escribir, Dt2 corresponde al puerto y Dt3 al dato.
- × Acción 1, lectura de Dato en Puerto: para esta acción al igual que la anterior Dt1 y Dt2 corresponden al sub.-puerto y puerto respectivamente, pero a Dt3 por defecto se envía un dato nulo, luego el Periférico envía el dato obtenido del puerto indicado al DPBR, para esto se configura como *transceiver* y el DPBR a su vez se configura como *receiver*
- × Acción 2, configuración de sub.-puerto: con esta acción se configuran los sub.-puertos para determinar si sus puertos son de entrada o de salida, Dt1 contiene el número del sub.-puerto a configurar, Dt2 el valor de los puertos que van a ser configurados y a Dt3 se envía un valor nulo.
- × Acción 3, asignación de identificación: por defecto la letra de identificación corresponde al carácter con valor ASCII de 255, esta es configurada mediante el software de programación. En la acción 3, Dt1 porta la letra de identificación y para los datos de Dt1 y Dt2 se envían valores nulos. Para que la asignación sea valida el usuario debe presionar el pulsador de "Asignar Identificador" ubicado en la parte trasera del Periférico.
- × Acción 4, confirmación: cuando el DPBR envía esta acción el Periférico se configura como *transceiver* y envía su letra de identificación. Para esta acción se envía a Dt1, Dt2 y Dt3 valores nulos.

- × Acción 5, almacenamiento en memoria: esta acción es usada para almacenar algún dato en la memoria interna del periférico, en esta acción Dt1 corresponde la dirección de memoria y Dt2 corresponde al dato a almacenar, a Dt3 se envía un valor nulo.
- × Acción 6, configuración inicial: con esta acción el Periférico activa una macro que configura todos sus puertos según lo que este almacenado en su memoria interna. Para esta acción a Dt1, Dt2 y Dt3 se envían valores nulos.
- × Acción 7, enviar configuración sub.-puerto: cuando el periférico recibe esta acción, se configura como *transceiver* y envía al DPBR el dato correspondiente a la configuración del sub.-puerto especificado. Para esta acción Dt1 corresponde al sub.-puerto, a Dt2 y Dt3 se envían valores nulos.
- × Acción 8, dato fijo: con esta acción se configuran valores fijos en el puerto del sub.-puerto seleccionado. Para esta acción Dt1 corresponde al sub.-puerto, Dt2 al puerto y Dt3 el dato fijo que se pretende almacenar.
- × Acción 9, sin datos fijos: con esta acción se activa una macro que elimina los datos fijos. Para esta acción a Dt1, Dt2 y Dt3 se envían datos nulos.

2.1.1.4. DISEÑO FÍSICO

ESQUEMA Y CIRCUITOS IMPRESOS

Para el diseño de estos componentes es necesario definir el tipo de empaquetadura a usar que en este caso es PDIP³². Debido a que este tipo de circuitos integrados son más económicos y comerciales a nivel local, además el producto final de la investigación es el desarrollo de un prototipo sobre el cual se ejecutan cierta cantidad de pruebas para detectar y depurar errores y la mayoría de desarrolladores de microcontroladores están diseñados para este tipo de empaquetadura.

El diseño se realiza mediante software CAD³³ para facilitar el enrutamiento y obtener el mejor resultado en cuanto a tamaño y costo para su posterior desarrollo. Para mas detalle remítase al Anexo 3, pagina 129.

CARCAZAS

Este componente también fue diseñado mediante software CAD para su posterior desarrollo. Para mas detalle remítase al Anexo 3, pagina 141.

³² PDIP: (*Plastic dual in-line package*) Encapsulado plástico dual en línea.

³³ CAD: (Computer Aided Design) Diseño Asistido por Computador.

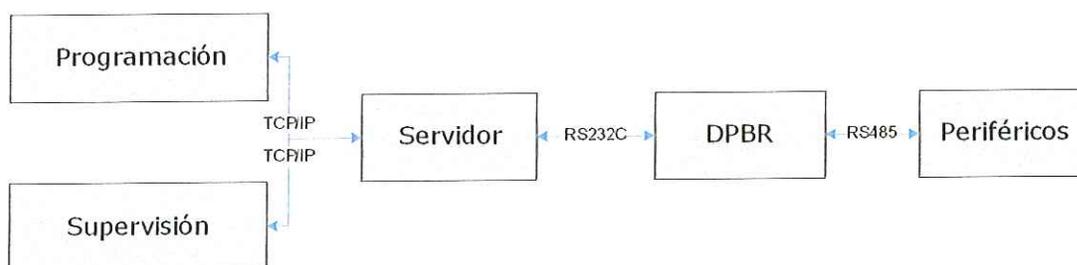
2.1.2. DISEÑO DE PAQUETE DE SOFTWARE

Inicialmente el software consistía en una aplicación la cual integraba las tres aplicaciones que actualmente se encargan del proceso de supervisión remota y programación. La aplicación tenía varias desventajas, en primera medida solo se podía efectuar la supervisión remota ya que la programación requiere de permisos especiales que no era posible otorgar con tan solo una aplicación.

2.1.2.1. ESTRUCTURA DE PAQUETE

El paquete consta de tres aplicaciones las cuales se comunican entre si mediante el protocolo TCP/IP. Estas tres aplicaciones utilizan una base de datos común la cual almacena la programación y configuración del paquete.

Figura 16. Estructura de Paquete



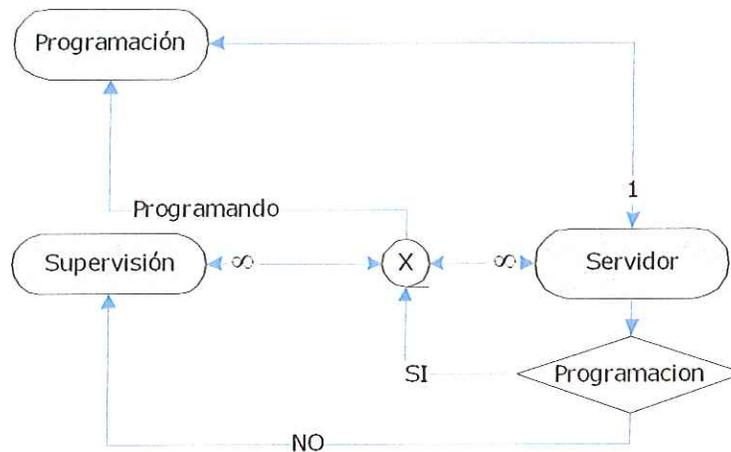
Del Autor

La aplicación encargada de comunicarse directamente con el DPBR es el Servidor el cual administra la comunicación y evita conflictos entre las aplicaciones remotas. La aplicación de Programación como su nombre lo indica es la encargada de crear las reglas para la Programación, Descargue de Datos, Configuración y Administración de Periféricos en el DPBR. La aplicación de Supervisión se encarga de recibir e interpretar la información que el Servidor recibe del DPBR para mostrarla de manera simbólica y grafica.

APLICACIÓN DE SERVIDOR

El servidor maneja una jerarquía en cuanto a la manera de entregar el control de acceso al DPBR según se muestra en la Figura 17 (Véase "APLICACIÓN SERVIDOR", Pagina 82)

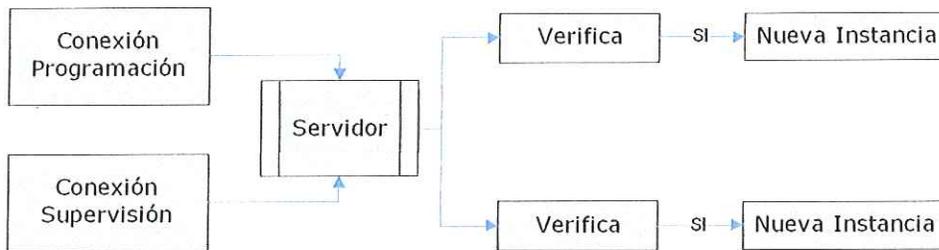
Figura 17. Jerarquía Servidor



Del Autor

Inicialmente el servidor se encuentra en estado de reposo donde escucha los puertos de entrada de datos de las aplicaciones de programación y supervisión como se muestra en la Figura 18. Cuando un usuario se conecta, el Servidor

Figura 18. Comunicación Servidor



Del Autor

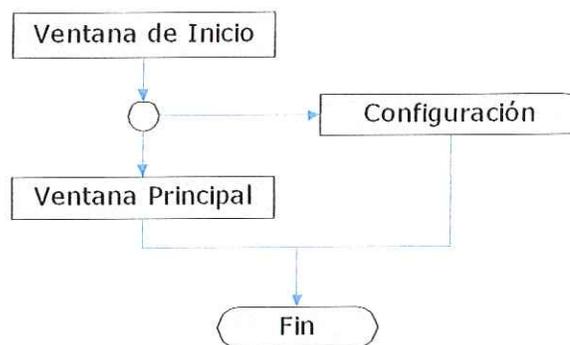
verifica que se encuentre en la base de datos, luego acepta la conexión y crea una nueva instancia para seguir escuchando el puerto. Si algún ordenador en la Intranet solicita Programar, el servidor revisa su estado, si ya esta en estado de

Programación retorna un error al equipo que envió la solicitud, pero si no es así, el servidor se pone en estado de Programación y si estaba al mismo tiempo en estado de Supervisión envía un mensaje de Pausa a todos los clientes que estén en modo Supervisión. Luego cuando el cliente salga de modo de Programación, el servidor revisa si hay clientes de Supervisión en Pausa y envía a estos un mensaje de Continuar.

Con esto se garantiza que solo un usuario a la vez pueda programar el DPBR, pero varios usuarios puedan Supervisarlos.

El software tiene la siguiente estructura en cuando a su organización de ventanas:

Figura 19. Estructura Servidor



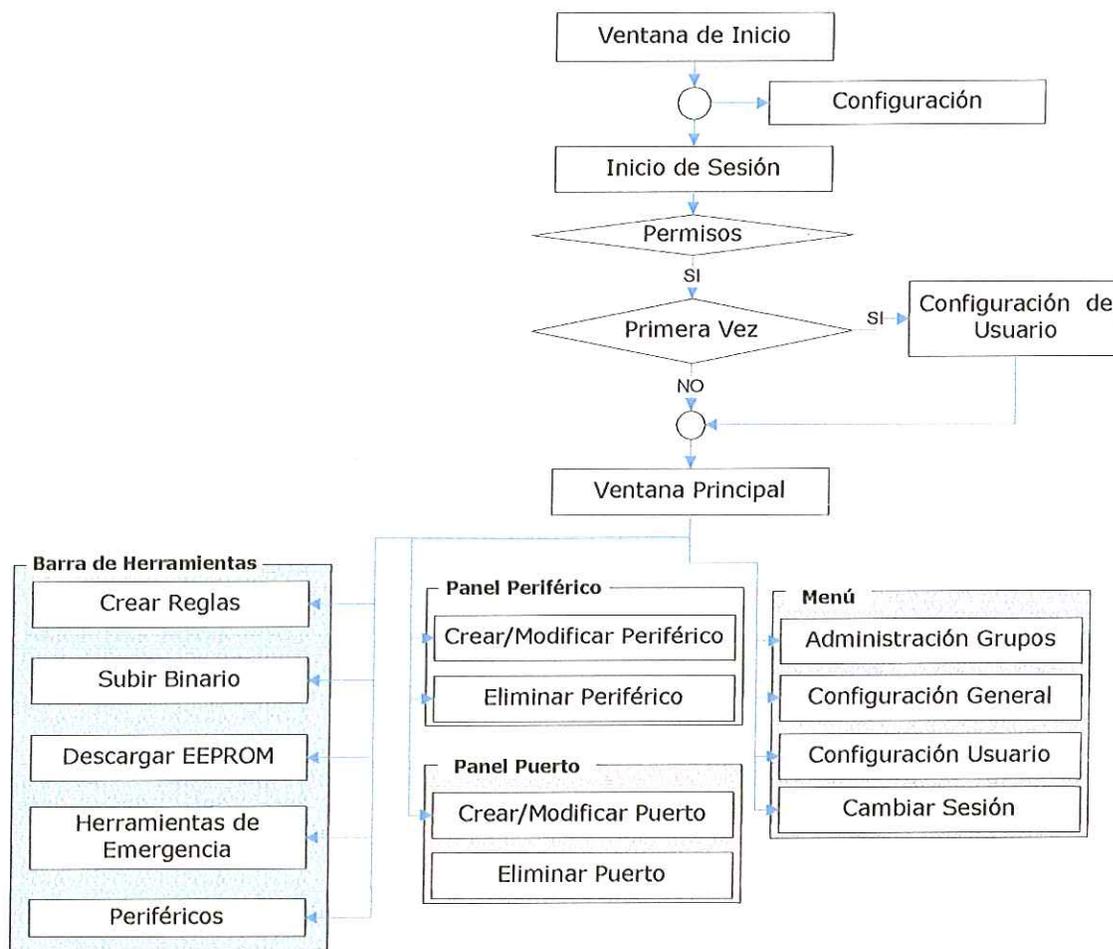
Del Autor

Como las otras dos aplicaciones del paquete, este muestra su ventana de inicio en donde carga todas sus variables y verifica que no existan errores de comunicación con los equipos remotos y base de datos, si se presenta algún error se va a la ventana de configuración, de lo contrario este ingresa a la ventana principal donde esta en constante escucha de los puertos configurados.

APLICACIÓN PROGRAMACIÓN

La aplicación de Programación, como su nombre lo indica funciona para que el usuario con los permisos respectivos pueda crear las reglas, programar el DPBR y configurar sus Periféricos. (Véase "APLICACIÓN PROGRAMADOR", Pagina 88)

Figura 20. Estructura Programación



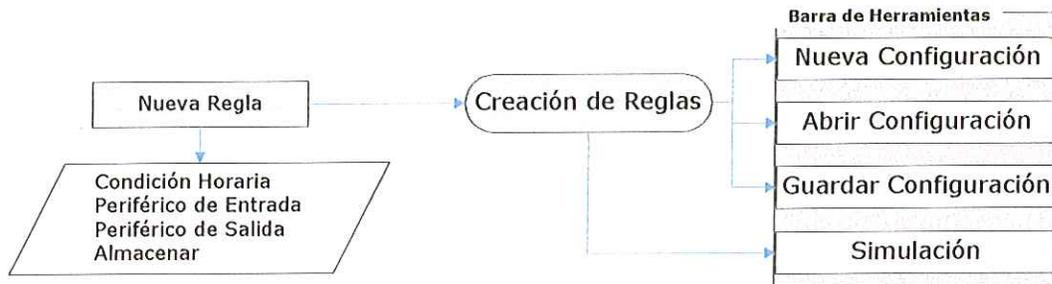
Del Autor

Luego de pasar por la ventana de inicio, la aplicación "Programación" continua a el módulo de inicio de sesión donde el usuario debe autenticarse para seguir a los siguientes módulos de la aplicación. Si el usuario es valido según la configuración guardada en la base de datos, puede crear reglas, subirlas en

modo binario al DPBR, descargar memoria de DPBR, configurar periféricos y acceder a herramientas de emergencia.

En el módulo de creación de reglas el usuario puede establecer configuraciones nuevas, abrir configuraciones almacenadas, guardar la configuración actual, exportar e importar configuraciones desde archivos y simular la configuración actual como se muestra en la Figura 21.

Figura 21. Diagrama Reglas DPBR

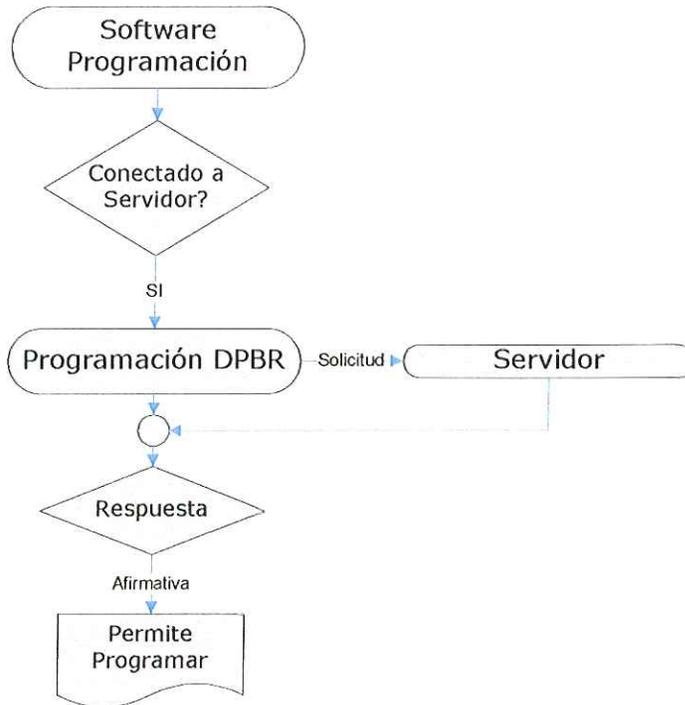


Del Autor

En el módulo de subir binario el usuario puede enviar las reglas creadas en el módulo anterior siguiendo el siguiente algoritmo (Figura 22):

- × Establecer Conexión con Servidor Remoto
- × Abrir ventana de Programación, donde esta envía un dato al Servidor solicitándole el control del DPBR para Programar.
- × Si el DPBR esta encendido y disponible, el Servidor envía un dato Afirmativo.
- × Si es Afirmativo el usuario Programa el DPBR.

Figura 22. Programación DPBR

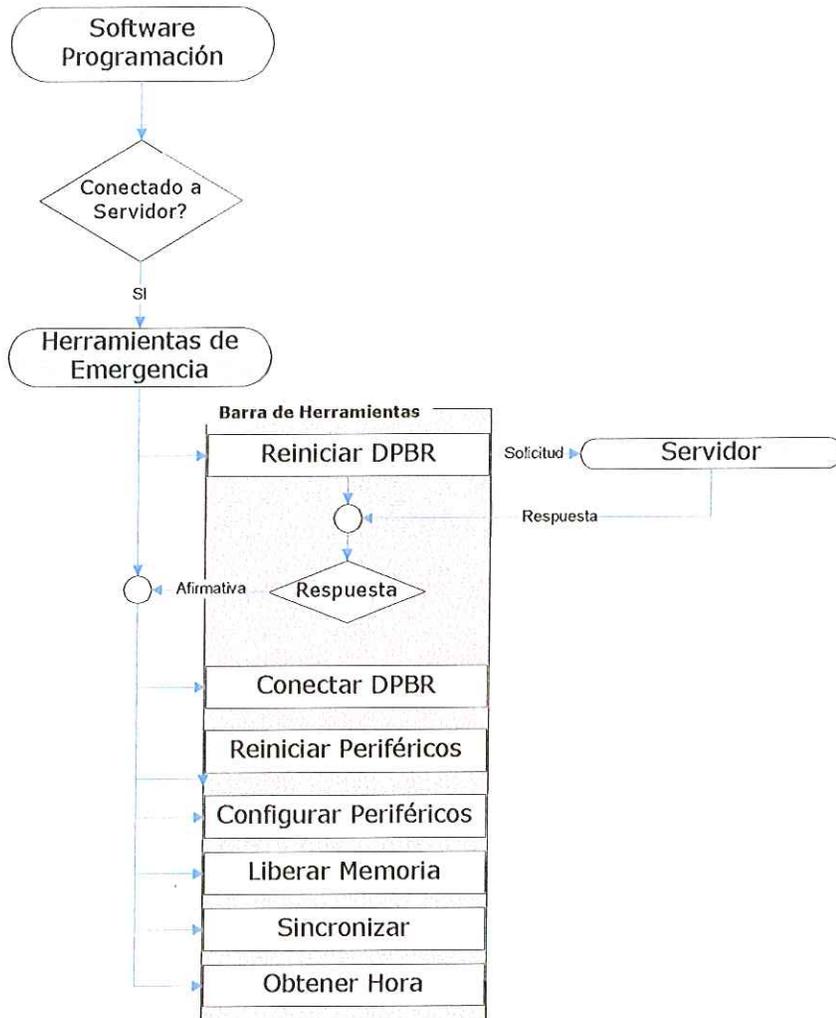


Del Autor

Mediante este mismo algoritmo se puede ingresar al módulo de descargue de memorias, a diferencia que el DPBR no se reprograma sino que se descarga para dar paso a información nueva y para que el usuario pueda realizar informes y de mas con la información obtenida.

El módulo de herramientas de emergencia se rige bajo el algoritmo mostrado a continuación:

Figura 23. Acceso a Herramientas de Emergencia



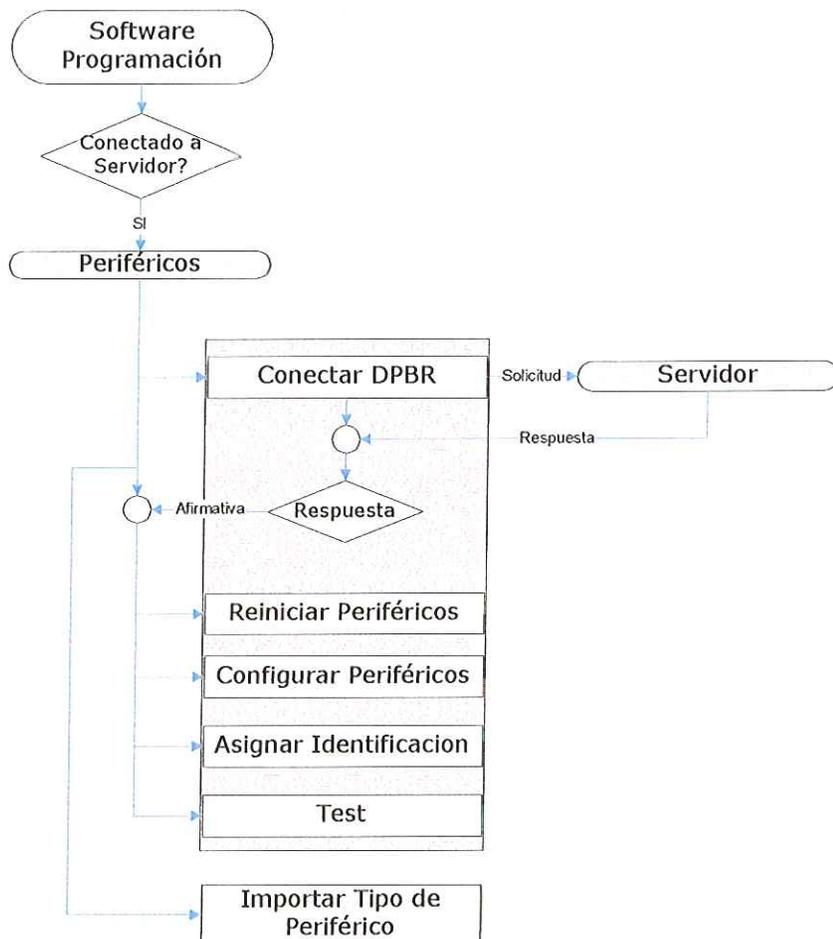
Del Autor

Según el algoritmo, para que el usuario pueda usar las herramientas en el panel debe conectarse manualmente servidor, desde este módulo el usuario puede reiniciar y configurar periféricos, liberar memoria (eliminar datos), sincronizar y obtener datos de fecha contenidos en el DPBR.

En el módulo de periféricos, luego de conectarse manualmente al servidor, el usuario puede reiniciar, configurar y hacer un test a los periféricos instalados y

configurados, también puede importar un nuevo tipo de periférico si este existiese.

Figura 24. Acceso a Periféricos

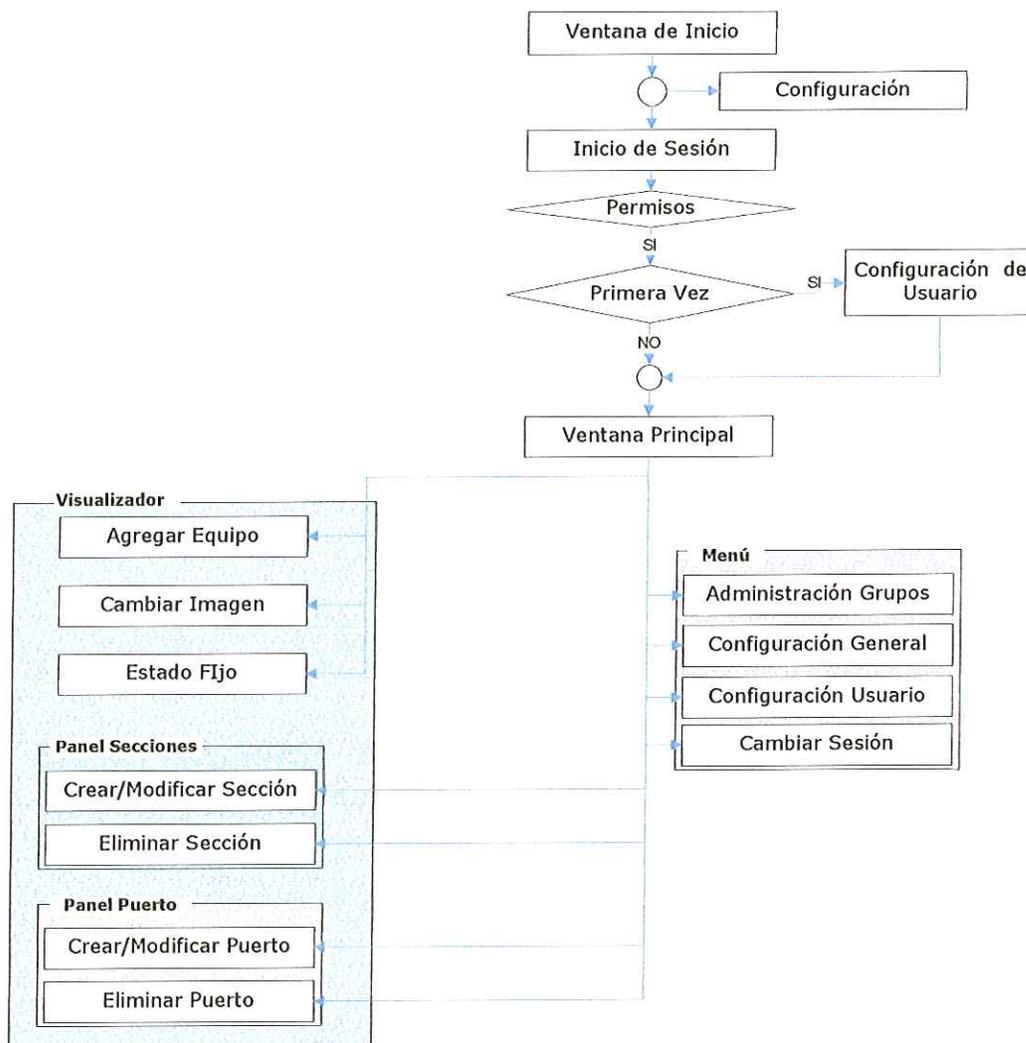


Del Autor

APLICACIÓN SUPERVISIÓN

Esta aplicación cumple la función de mostrar de forma grafica el estado en tiempo real de los puertos (equipos) configurados de los periféricos conectados en el DPBR (Vease "APLICACIÓN SUPERVISIÓN", Pagina 103). Igual que la

Figura 25. Estructura de Supervisión

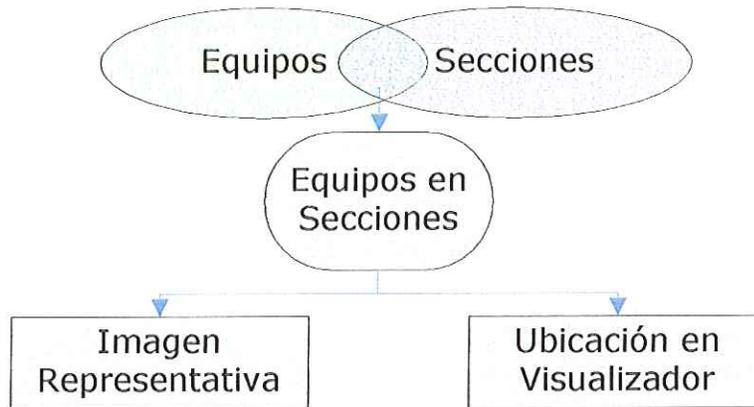


Del Autor

aplicación de Programación, para poder acceder a esta aplicación el usuario debe tener permisos previamente establecidos por el Administrador. En la ventana principal el usuario puede agregar equipos, asignarles imágenes representativas

y al supervisar puede asignar estados fijos a los equipos. Todo esto es administrado por secciones donde se agrupan los equipos que tienen cosas en común, una sección es un Sub - Grupo de equipos, es decir puede existir un equipo en varias secciones.

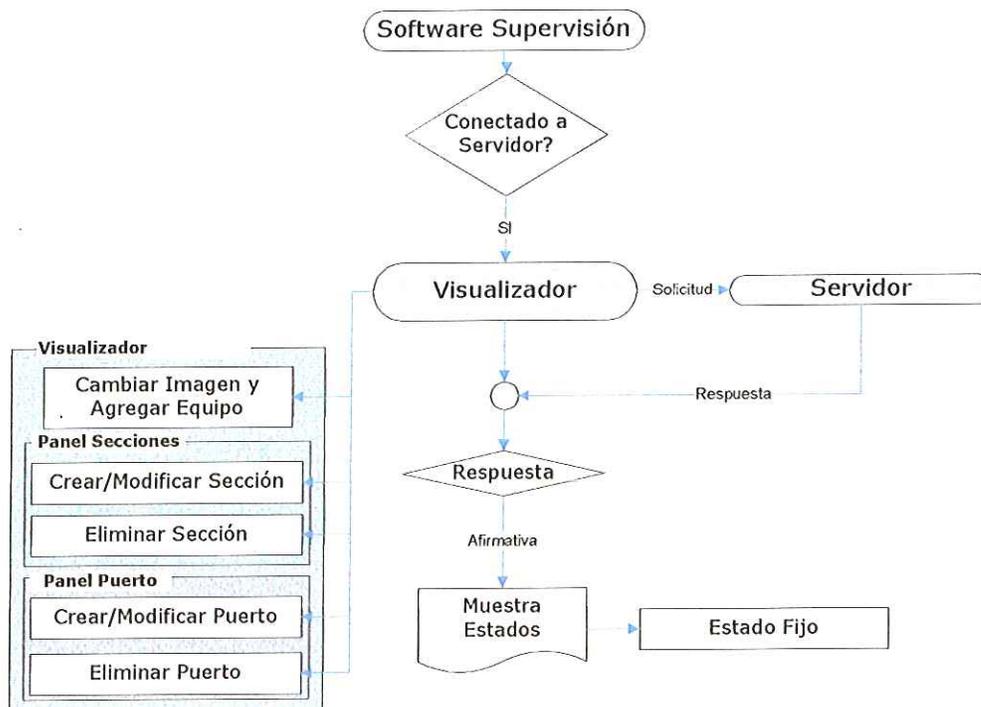
Figura 26. Equipos en Secciones



Del Autor

Desde esta aplicación se puede acceder a la configuración general, de usuario y de grupos para cambiar la manera como se muestra y administra el sistema. El diagrama de la Figura 27 muestra como es el funcionamiento del visualizador.

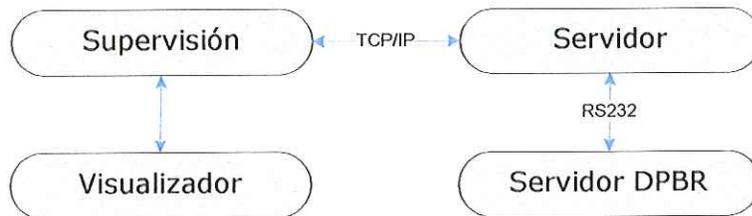
Figura 27. Visualizador Supervisión



Del Autor

Según la Figura 28 la aplicación de supervisión se conecta al servidor (Véase Figura 17), para que este le envíe los estados de los Periféricos.

Figura 28. Conexión Visualizador ↔ DPBR



Del Autor

3.DESARROLLO

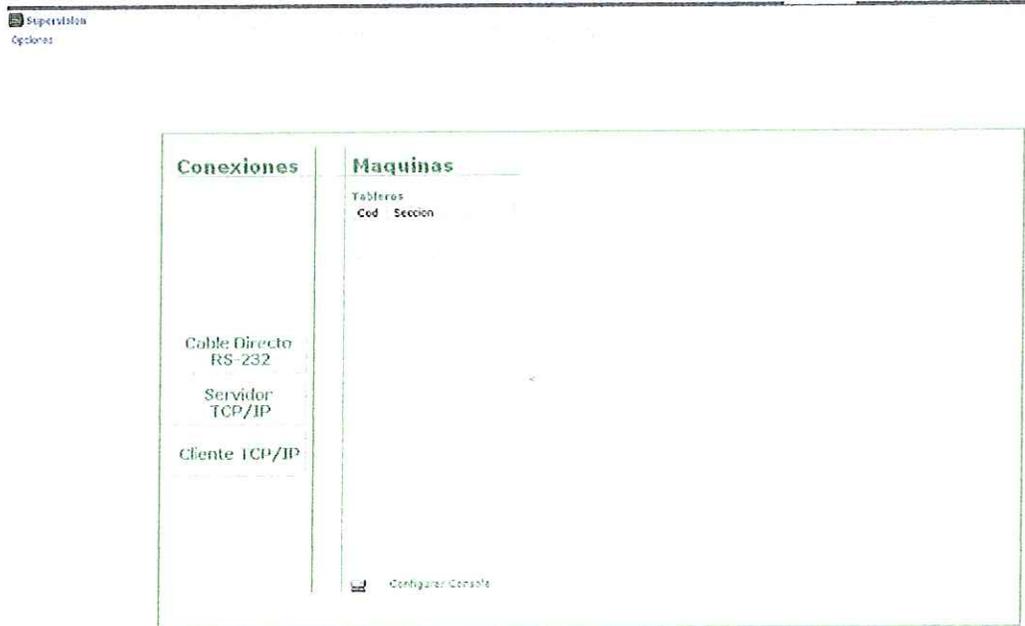
3.1. DESARROLLO DE SOFTWARE Y FIRMWARE

El desarrollo del Sistema Programable De Control Y Supervisión Remota De Equipos se llevo a cabo de forma paralela. Es decir el desarrollo del paquete de software se realizo simultáneamente con el desarrollo del *firmware*.

3.1.1. DESARROLLO DE PAQUETE DE SOFTWARE.

Como se menciona en el capítulo anterior, inicialmente el paquete de software consistía en una aplicación (Imagen 3).

Imagen 3. Supervisión Remota – Primera Versión



Del Autor – Software

Esta aplicación fue desarrollada en *Microsoft® Visual Basic 6.0.SP6* con motor de bases de datos *Microsoft® SQL Server 2000*.

Esta aplicación contenía un módulo de supervisión remota, módulo de reglas de DPBR, módulo de Programación DPBR y Periféricos, gestión de secciones y

equipos, y tenía un pequeño módulo que permitía que otro equipo se conectara remotamente al DPBR para realizar labores de Supervisión pero tenía varias desventajas. Primero su base de datos funcionaba bajo *Microsoft® SQL Server 2000* la cual es una aplicación para gestionar el manejo de bases de datos muy robustas. Solo la licencia de este software especializado puede alcanzar más de los dos mil dólares estadounidenses haciendo que la implementación de este paquete sea muy poco viable económicamente. Otra desventaja que tenía al integrar todo en una sola aplicación, era el hecho de que el usuario no podía acceder al módulo de programación desde un equipo remoto, ya que no era posible manejar permisos y una jerarquía para tomar el control del DPBR.

El primer paso a seguir para corregir estas falencias fue aislar las funciones de Supervisión y de Programación, y gestionar por medio de otra aplicación adicional la comunicación por medio del protocolo TCP/IP, luego se realizó un cambio en cuanto al motor de base de datos, usando *Microsoft® Jet 4.0* que viene dentro del paquete de aplicaciones de varios sistemas operativos como por ejemplo *Microsoft® Windows XP*. Este motor de bases de datos trabaja con bases de *Microsoft® Access 2000*. Un inconveniente que presenta este tipo de bases de datos, es que el usuario debe compartir una carpeta para poder acceder remotamente a ella, debido a esto se trabajó con autenticación de Windows por medio de WMI³⁴ para aumentar un poco la seguridad en la comunicación. Concluida la parametrización nueva de el paquete de aplicaciones, se procede a establecer un pequeño formato para la distribución de los datos entre las aplicaciones mediante el protocolo de comunicaciones TCP/IP, y se realizan las

³⁴ WMI: Instrumentación Administrativa de Windows. Es un API del sistema operativo Windows que permite controlar y administrar a los equipos de una red.

pruebas de comunicación entre ordenadores usando la herramienta *winsock*³⁵ ofrecida dentro del paquete de *Microsoft® Visual Basic 6.0*.

3.1.2. FIRMWARE'S

El desarrollo de los firmware's del Dispositivo Programable Basado en Reglas y el Periférico de Entradas y Salidas Digitales de 22 puertos también sufrió varias etapas de cambio y reestructuración para aumentar sus prestaciones y mejorar su capacidad de control.

La idea inicial del proyecto era el desarrollo de un dispositivo que tomara señales digitales y almacenara sus estados cuando estos cambiaran para llevar el control estadístico de un proceso para su depuración y la mejora del mantenimiento de sus equipos. La información almacenada por este dispositivo debía ser transportada mediante otro dispositivo externo de almacenamiento a un ordenador donde un software especialmente desarrollado descargara la información, la analizara y la almacenara en una base de datos para poder luego generar informes.

Con el fin de volver más funcional dicho dispositivo, la idea fue moldeada y se desarrollo el proyecto que es hoy.

Como primera medida se estudian los dispositivos ofrecidos actualmente en el mercado de la automatización industrial que tengan características similares al dispositivo que se desea desarrollar (PLC, Autómata Programable, Sistemas SCADA, etc.)

Con una idea clara de lo que se esperaba desarrollar, se establece un cronograma de actividades y se procede a comprar los materiales para el desarrollo del prototipo de lo que es el DPBR. Luego se establecieron las funciones para controlar el bus de datos I²C para comunicar los distintos

³⁵ Winsock: Basado en un conocido paquete de UNIX llamado Sockets, es una biblioteca que usan

componentes (memoria y el reloj de tiempo real), también se desarrollaron funciones para manejar el módulo USART del microcontrolador y comunicar el DPBR con el PC. Estas fueron las bases para comenzar a desarrollar el firmware del DPBR.

Luego de establecer una comunicación con el PC, se realizaron pruebas con el paquete de software y se estableció un protocolo preliminar que fue cambiando poco a poco según el desarrollo del proyecto.

Para la comunicación entre el DPBR y sus Periféricos se desarrollo una interfaz propia y un protocolo, pero debido a la inexactitud y a las distancias que se esperaban manejar se cambio totalmente este módulo interno en el PESD22 y el DPBR para usar comunicación serial mediante la interfaz RS-485, lo que forzó a realizar un módulo adicional USART en el DPBR para la comunicación con el PC mediante la interfaz RS-232C.

El firmware del PESD22 se desarrollo según la necesidad del momento (sensar, actuar) pero durante el desarrollo se implementaron las funciones que se mencionan en el capitulo anterior.

3.1.3. INTEGRACIÓN

Desarrollado los prototipos del DPBR y del PESD22 se comenzó a trabajar en la integración con la aplicación, depurando el protocolo e implementando las diferentes funciones (Acciones y Menús) mencionadas en el capitulo anterior. El proceso continuo de depuración dio como resultado la mejora del protocolo de comunicaciones.

los programadores para hacer más fácil la construcción de aplicaciones que trabajan con TCP/IP.

3.1.4. PRUEBAS DE FUNCIONAMIENTO

Las pruebas más importantes para el correcto funcionamiento del sistema se basaron en la siguiente tabla.

Tabla 8. Pruebas de Funcionamiento

		Elementos	Descripción	
Desarrollo	Comunicación	Microcontrolador-RTC	Comunicación el RCT con el Microcontrolador del DPBR	
		Microcontrolador-Memorias	Leer y Escribir un dato almacenado	
		DPBR-PC	Iniciar el modulo de menú de su firmware	
		DPBR-Periféricos	Enviar y Recibir datos en Periféricos	
		Servidor	Enviar y Recibir datos entre aplicaciones	
	Software	DB	Base de Datos	Se comprueba la comunicación con la base de datos del sistema.
		Manejo de Archivos	Manejo de Archivos	Generación y restauración de configuraciones de programación mediante archivos generados por la aplicación de programación
		Comunicación	PC-DPBR	Recibir datos de supervisión para la representación grafica en aplicación de supervisión
	Firmware	Análisis	DPBR	Almacenar programación en DPBR, configurar periféricos y RTC y comprobar mediante simulaciones
	Depuración	Estabilidad	Tiempo	Trabajo en el Tiempo

Del Autor

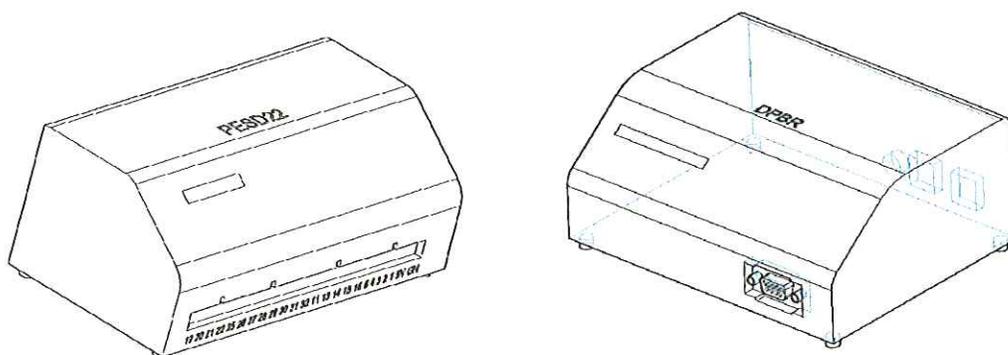
Estas pruebas pueden dar un diagnostico sobre el funcionamiento del sistema, ya que abarcan los aspectos mas importantes del mismo.

3.2. DESARROLLO FÍSICO

La última etapa del desarrollo fue la fabricación y el ensalme de los circuitos impresos al igual que el desarrollo de las carcazas.

Para el diseño electrónico se tuvo en cuenta diferentes aspectos tanto estéticos como funcionales. Por ejemplo el uso de conectores RJ45 y DB9 para la comunicación del DPBR con los periféricos y PC respectivamente.

Imagen 4. Carcazas



Del Autor

Finalmente la carcasa se desarrollo en acrílico por cuestiones de estética y economía puesto que se busca desarrollar un prototipo basándose en el tamaño y medidas de los circuitos impresos.

4.MANUAL DE OPERACIÓN

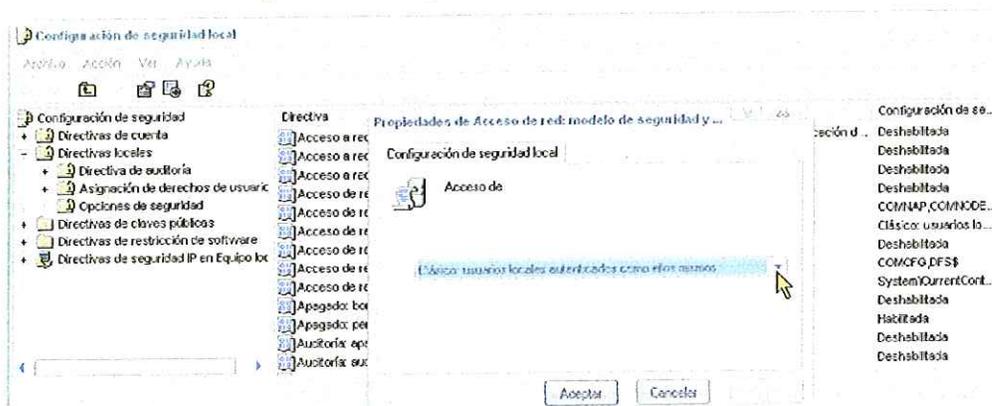
4.1. INSTALACION Y PUESTA EN MARCHA

Para poner en marcha el sistema el usuario debe seguir los siguientes pasos:

1 Primero el usuario debe instalar el paquete de software siguiendo los pasos del asistente de instalación automática, en el equipo servidor de DPBR y en los equipos remotos donde se desee realizar labores de supervisión y programación.

2 Luego de instalado el software ingrese a las "Herramientas Administrativas" en el panel de control y ejecute el módulo "Directiva de Seguridad Local", allí despliegue "Directivas Locales" \ "Opciones de Seguridad" y modifique la opción de "Acceso a red: modelo de seguridad y para compartir cuentas locales" colocando acceso "Clásico: usuarios locales autenticados como ellos mismos" como opción predeterminada.

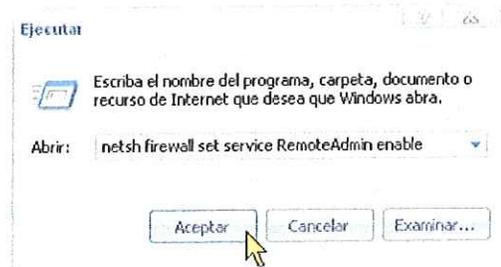
Imagen 5. Configuración de Seguridad Local



Microsoft Windows XP

Si el equipo usa *Microsoft Windows XP SP2* como sistema operativo y tiene el

Imagen 6. Ventana Ejecutar



Microsoft Windows XP

*firewall*³⁶ instalado realice este paso adicional: abra la ventana "Ejecutar..." desde el menú Inicio y digite las siguientes tres líneas de comando por separado:

- × *netsh firewall set service RemoteAdmin enable*
- × *netsh firewall add portopening protocol=tcp port=135 name=DCOM_TCP135*
- × *netsh firewall add allowedprogram program=%windir%\system32\wbem\unsecapp.exe name=UNSECAPP*

Si usa otro firewall, el usuario debe habilitar el puerto TCP 135, la administración remota y desbloquear el programa "unsecapp.exe" ubicado en la carpeta "wbem" del system32.

3 Luego de configurado el sistema operativo, ejecute la aplicación "Servidor" del paquete de software en donde debe seguir los pasos para la ubicación de la base de datos, y datos iniciales para el correcto funcionamiento del sistema, esto se realiza mediante la ventana de configuración que se ejecuta

³⁶ Firewall: Mecanismo de seguridad en Internet frente a accesos no autorizados. Básicamente consiste en un filtro que mira la identidad de los paquetes y rechaza todos aquellos que no estén autorizados o correctamente identificados

automáticamente cuando se detecta algún error o cuando este se ejecuta por primera vez la aplicación

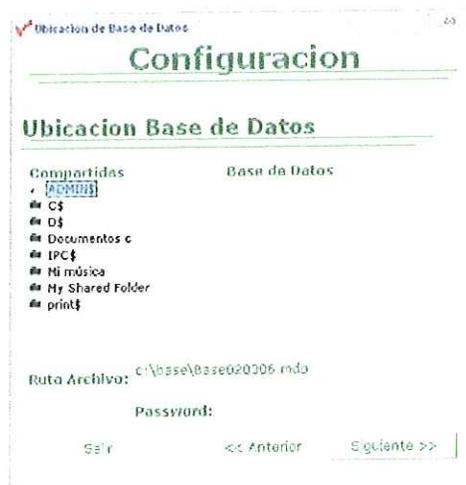
Imagen 7. Ventana de Configuración – Equipo Servidor



Del Autor

Esta ventana cumple la función de configurar los datos iniciales del sistema, aquí

Imagen 8. Ventana de Configuración – Ubicación de Base Datos



Del Autor

se apunta al equipo servidor remoto para luego validarlo mediante la autenticación de Windows (Nombre de Usuario y Contraseña). Si la autenticación

fue satisfactoria, se va al siguiente paso donde se ubica la base de datos en el equipo remoto (Imagen 8). Si la base esta protegida por contraseña, esta debe ser especificada en el campo "Password". Si no existe base de datos esta puede ser creada dando clic derecho sobre el panel de Base de Datos (Imagen 9).

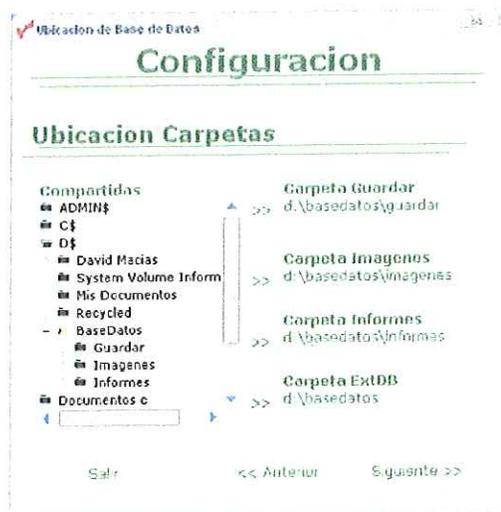
Imagen 9. Nueva Base de Datos



Del Autor

Aquí se especifica el nombre de la base y su contraseña.

Imagen 10. Ventana de Configuración – Ubicación de Carpetas



Del Autor

El tercer paso en esta ventana es la selección de las carpetas (Imagen 10) que van a ser usadas en el sistema para guardar configuraciones de DPBR, Imágenes, Informes y Otras Bases de Datos.

Luego debe establecerse la información empresarial (Imagen 11), datos como la razón social, el NIT³⁷, la dirección, teléfonos, e-mail y representante legal.

Imagen 11. Ventana de Configuración – Información Empresarial

The screenshot shows a software window titled 'Ubicación de Base de Datos' with a sub-header 'Configuración'. Below this is the section 'Información Empresarial'. The data is as follows:

Razon Social	DIRTEC		NIT	91529.026
Direccion	Cr 25a #145-20 Vista Campesino Casa 49			
Telefono	Telefono2	Fax		
6393227	3156763979	-		
e-Mail	ruedasera@gmail.com		Representante Legal	Sergio F. Rusda A.I

At the bottom, there are navigation buttons: 'Salir', '<< Anterior', and 'Siguiente >>'.

Del Autor

Por ultimo se debe especificar que puerto COM³⁸ (Imagen 12) que el PC usara

Imagen 12. Ventana de Configuración – Información de Puertos

The screenshot shows the same 'Configuración' window but with the 'Información de Puertos' section. The data is as follows:

Puerto COM	1		
TCP/IP Chat	0300	TCP/IP Comunicacion	0302

At the bottom, there are navigation buttons: 'Salir', '<< Anterior', and 'Finalizar'.

Del Autor

³⁷ NIT: Numero de Identificación Tributaria (Colombia)

para la comunicación con el DPBR (Servidor) y el rango de puertos para la configuración TCP/IP. Luego de dar clic en finalizar, la aplicación se cierra y tiene que ser abierta nuevamente manualmente para ejecutar la aplicación Servidor.

4.1.1. APLICACIÓN SERVIDOR

El Servidor igual que las otras dos aplicaciones comienza con la "Ventana de Inicio" en donde se cargan las variables principales usadas por el paquete para su funcionamiento.

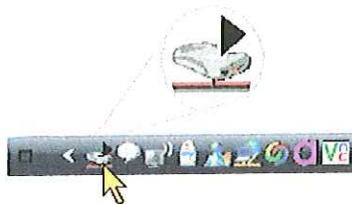
Imagen 13. Ventana de Inicio



Del Autor

En la ventana de inicio solo varía el nombre, la versión y la revisión con relación a las otras dos aplicaciones. Luego de ejecutar esta ventana, la aplicación se pone en segundo plano y sitúa un icono en la barra de tareas (Imagen 14).

Imagen 14. Servidor - Icono Barra de Tareas



Del Autor

³⁸ COM: Nombre que reciben bajo DOS los puertos serie

Este icono puede variar dependiendo del estado:

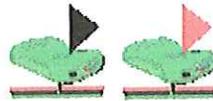
Imagen 15. Estados de Espera



Del Autor

Los estados de Espera son tres como se muestra en la Imagen 15, de izquierda a derecha, el primero se presenta cuando el software no recibe ningún dato del protocolo TCP/IP, el segundo se presenta cuando alguna aplicación envía un dato mediante el protocolo TCP/IP y el tercero cuando el servidor esta en modo de pausa.

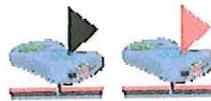
Imagen 16. Estados de Supervisión



Del Autor

Cuando el servidor esta a modo de supervisión los iconos varían entre los dos presentados en la Imagen 16, de izquierda a derecha, el primero representa que no se están enviando datos a los clientes de Supervisión TCP/IP y el segundo representa que se están enviando datos a los clientes mediante TCP/IP.

Imagen 17. Estados de Programación

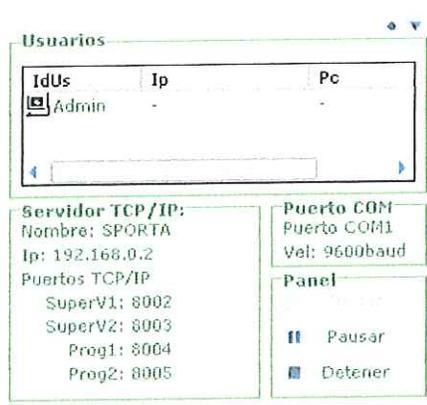


Del Autor

Cuando algún equipo solicita programar el DPBR, el servidor cambia su icono según la Imagen 17. De izquierda a derecha, el primero representa que no se

están enviando datos al cliente que esta Programando, y el segundo representa que se están enviando datos provenientes de las respuestas que genere el DPBR. Haciendo clic izquierdo en la barra de tareas la aplicación despliega una ventana en la esquina inferior derecha de la pantalla como se muestra en la Imagen 18.

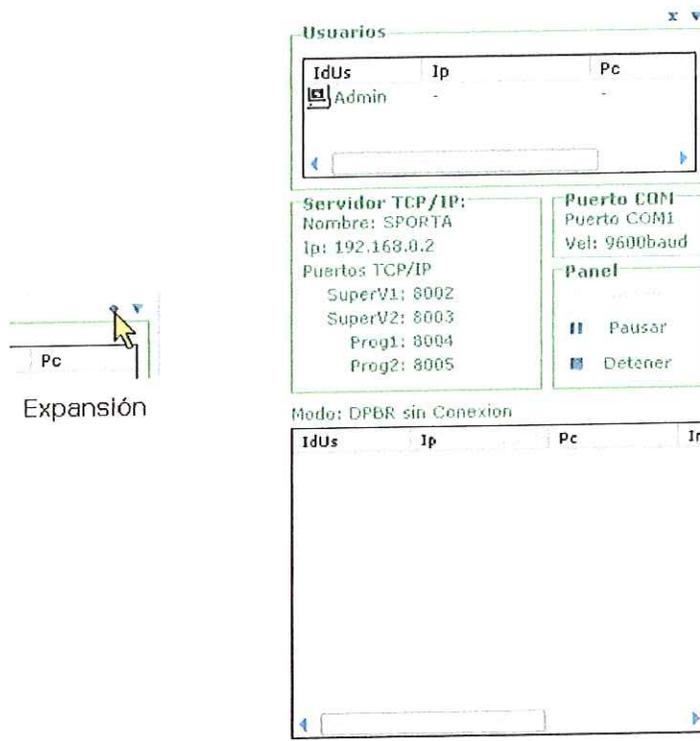
Imagen 18. Ventana de Monitorización y Control



Del Autor

Esta ventana monitoriza todos los usuarios de la base de datos, muestra que puerto COM se esta usando y a que velocidad. También muestra datos del equipo local, como el IP, el nombre del equipo, y los puertos usados para la Supervisión y la Programación. Mediante opciones del Panel se puede Iniciar, Pausar y Detener el servidor.

Imagen 19. Ventana de Monitorización y Control (Expandida)



Del Autor

Haciendo clic en el botón expansión la ventana muestra un listado de las últimas cincuenta acciones ejecutadas. Estas acciones poseen tres estados como se muestra en la Imagen 20:

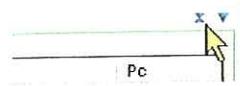
Imagen 20. Estados de Acciones - Servidor



Del Autor

De izquierda a derecha, el primer estado se muestra cuando la acción no se ha procesado y está en espera, la segunda se da cuando la acción se ejecutó con éxito y la tercera cuando durante la ejecución de la acción se produjo algún error en la comunicación.

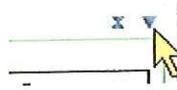
Imagen 21. Ventana de Monitorización y Control (Normalizar)



Del Autor

Para volver a la ventana normalizada se hace clic en el botón que se muestra en la Imagen 21 y para volver a la ventana a segundo plano se hace clic en el botón que se muestra en la Imagen 22.

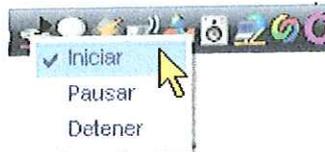
Imagen 22. Ventana de Monitorización y Control (Segundo Plano)



Del Autor

Si el usuario hace clic con el botón derecho del Mouse sobre el icono ubicado en la barra de tareas puede acceder a las opciones del Panel para Iniciar, Pausar o Detener el Servidor.

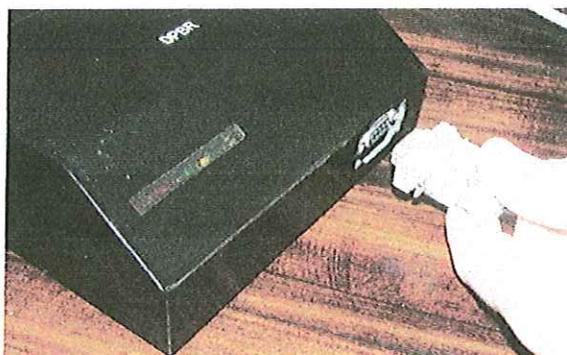
Imagen 23. Menú de Barra



Del Autor

4 En este paso debe preparar el Software, DPBR y sus Periféricos para su funcionamiento.

Imagen 24. Conexión DPBR ↔ PC



Del Autor

El DPBR debe estar ubicado cerca del PC servidor y conectado mediante cable al

Imagen 25. DPBR ↔ PC (Servidor)



Del Autor

puerto COM especificado en el paso anterior en el módulo de configuración. Los periféricos se conectan al DPBR mediante cable CAT5 y conectores RJ45.

Imagen 26. DPBR ↔ Periféricos



Del Autor

Por ultimo conecte el cable de la fuente de poder a la entrada del DPBR para que este comience a funcionar.

Imagen 27. Conexión de Fuente



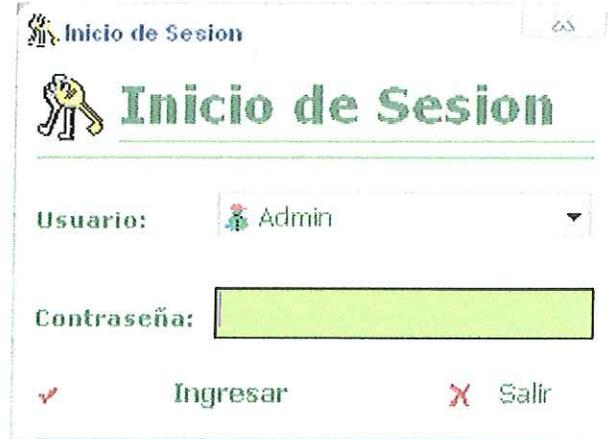
Del Autor

Para configurar el DPBR y sus periféricos ingrese a la aplicación "Programador" desde cualquier equipo en la Intranet

4.1.2. APLICACIÓN PROGRAMADOR

Igual que el software de Servidor, al iniciar se muestra la "Ventana de Inicio" que se muestra en la Imagen 13, luego va a la "Ventana de inicio de Sesión" mostrada en la Imagen 28

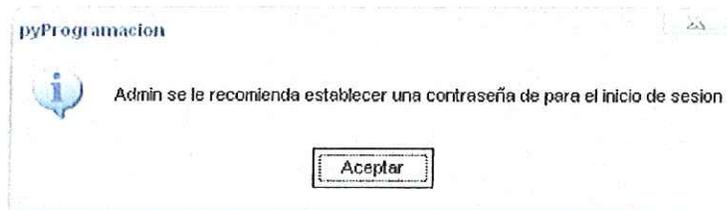
Imagen 28. Ventana de Inicio de Sesión



Del Autor

En la ventana de inicio de sesión el usuario debe autenticarse con su nombre de usuario y contraseña. Si este tiene los permisos correspondientes para acceder a este módulo revisa si el usuario accede por primera vez, si es así la aplicación muestra un mensaje (Imagen 29) y luego entra a la ventana de "Configuración de Usuario" mostrada en la Imagen 30.

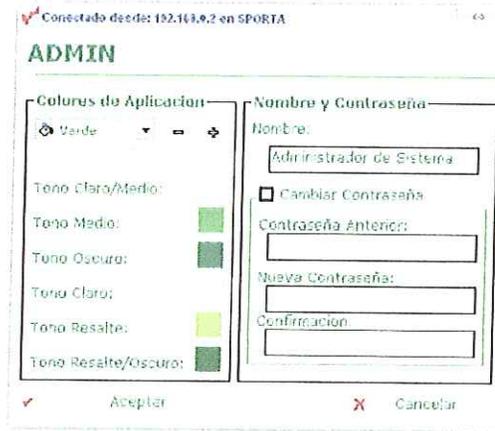
Imagen 29. Mensaje Inicio de Sesión



Del Autor

En esta ventana el usuario puede cambiar la apariencia de la aplicación durante su sesión, también puede cambiar su nombre y establecer una contraseña. Luego

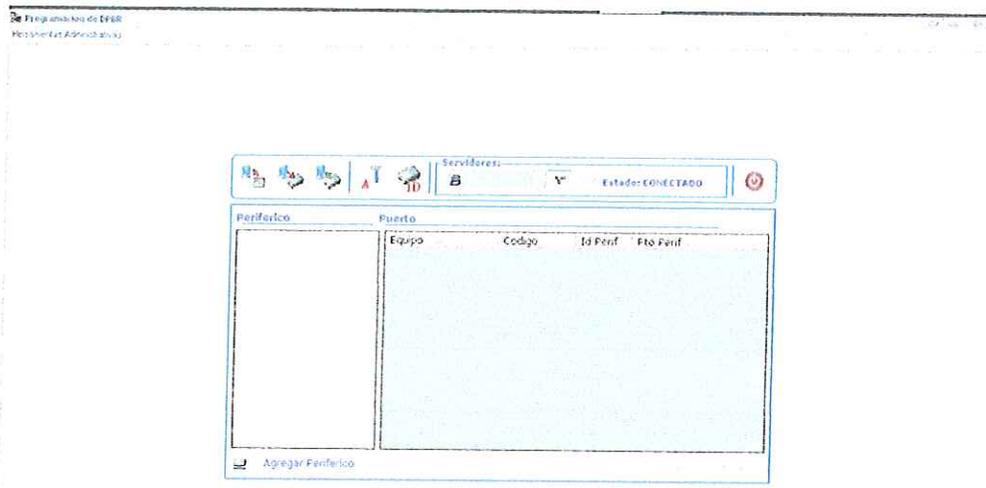
Imagen 30. Configuración de Usuario



Del Autor

de esto dando clic en aceptar, el software va a la ventana principal de Programación.

Imagen 31. Ventana Principal de Programación



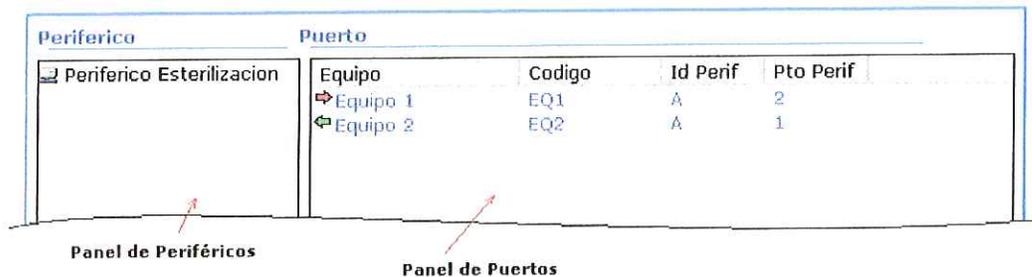
Del Autor

La ventana principal de programación consta de tres partes.

- × Panel de Periféricos y Puertos
- × Menú de Herramientas Administrativas
- × Barra de herramientas

4.1.2.1. PANEL DE PERIFÉRICOS Y PUERTOS

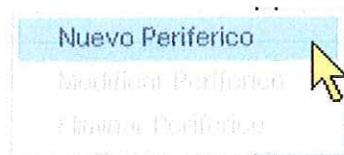
Imagen 32. Panel de Periféricos y Puertos



Del Autor

Para agregar un periférico el usuario debe hacer clic derecho sobre el panel "Periférico"

Imagen 33. Nuevo Periférico



Del Autor

Esto ocasiona que se abra el siguiente módulo de "Agregar/Modificar Periférico" como se muestra a continuación

Imagen 34. Agregar/Modificar Periférico



Del Autor

En esta ventana el usuario selecciona la letra de identificación, el tipo de periférico, su descripción y configuran sus puertos.

Imagen 35. Puertos



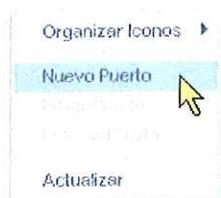
Del Autor

Los puertos son representados como se muestra en la Imagen 35, donde las verdes son entradas, las rojas son salidas y las amarillas son puertos bloqueados por el sistema ya que son usados por el para la comunicación.

Otra opción que ofrece el sistema es la de importar un tipo de Periférico diferente, de tal manera que el sistema no quede limitado al PESD22.

Para agregar un nuevo puerto representado por un equipo primero debe existir un Periférico creado luego el usuario debe hacer clic derecho sobre el panel de puertos donde aparece un menú emergente (Imagen 36).

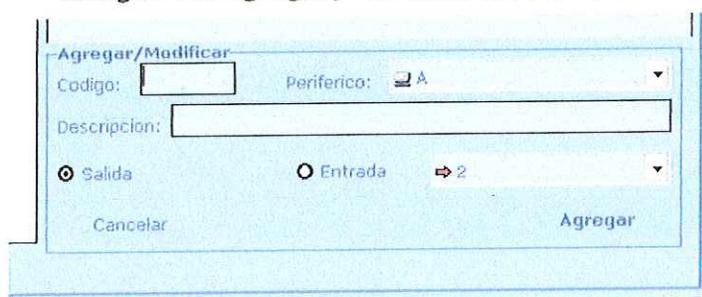
Imagen 36. Nuevo Puerto



Del Autor

Esto causa que se exponga el panel de Agregar/Modificar Puerto como se muestra a continuación.

Imagen 37. Agregar/Modificar Nuevo Puerto



Del Autor

En este panel el usuario debe asignar un código único al equipo, seleccionar el periférico, la descripción del equipo y el puerto al que va a representar. Esta información es almacenada en la base de datos donde a su vez es compartida con las demás aplicaciones y los otros equipos en la red.

4.1.2.2. BARRA DE MENÚ

Imagen 38. Barra de Menú

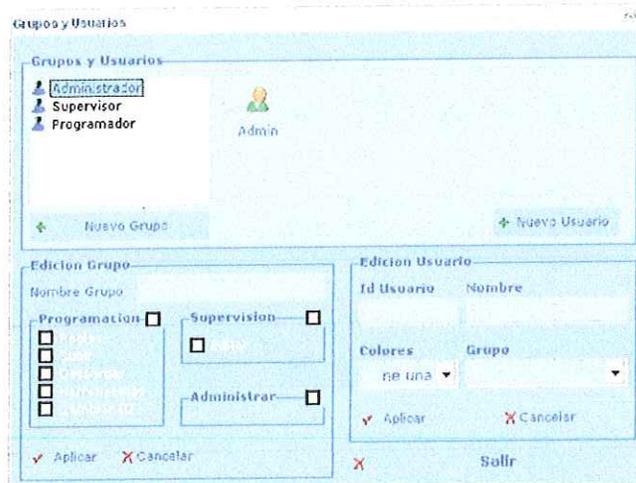


Del Autor

La barra de menú contiene los elementos mostrados en la Imagen 38. Desde este menú se puede acceder a la ventana "Configuración General" mostrada en el paso 3 para cambiar los parámetros iniciales y acceder a otras bases de datos. Si el usuario desea cambiar sus preferencias personales de colores, nombre y contraseña debe ingresar a la ventana de configuración de usuario mostrada en la Imagen 30 o para realizar cambio de usuario sin salir de la aplicación, el usuario desde el menú puede acceder a la ventana de "Inicio de Sesión" mostrada en la Imagen 28 y también puede acceder a la ventana de "Administración y Grupos" mostrada a continuación.

Esta ventana fue desarrollada para que el usuario "Administrador" pueda crear grupos con sus respectivos permisos, también puede mover usuarios entre grupos y crear nuevos usuarios.

Imagen 39. Administración y Grupos



Del Autor

4.1.2.3. BARRA DE HERRAMIENTAS

Imagen 40. Panel de Herramientas



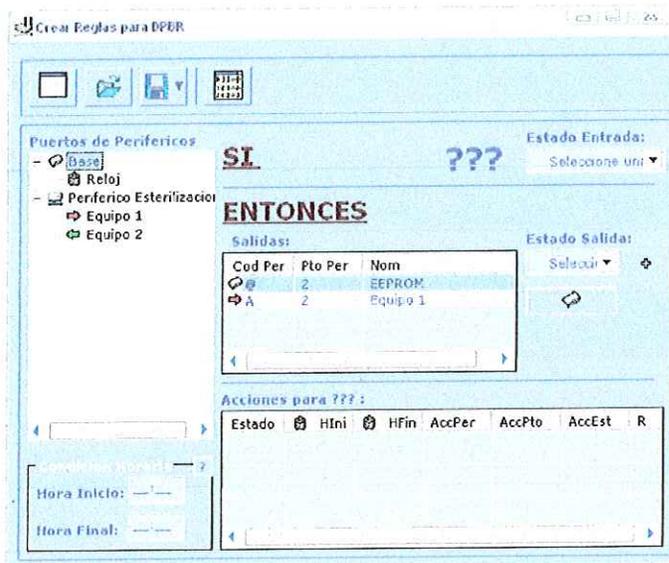
Del Autor

La barra de Herramientas consta de ocho botones los cuales cumplen diferentes funciones, de izquierda a derecha:

CREACIÓN DE REGLAS DPBR

Para crear y/o editar las reglas del DPBR, el usuario debe acceder al módulo de "Crear Reglas para DPBR" (Imagen 41) desde el módulo principal.

Imagen 41. Crear Reglas para DPBR



Del Autor

Para establecer una nueva regla el usuario debe seleccionar el periférico a evaluar según se muestra en la Imagen 42.

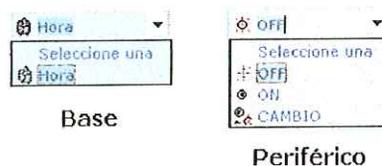
Imagen 42. Selección de Periférico



Del Autor

El periférico llamado Base representa a variantes dentro del mismo DPBR como lo son el reloj y la memoria representados con el código "@". Las reglas pueden ser

Imagen 43. Estado de Entrada



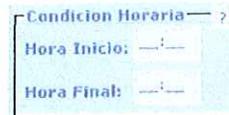
Del Autor

establecidas mediante reloj o mediante un puerto. Los estados dependen del periférico escogido (Imagen 43), en el caso de que el periférico sea Reloj, el estado único puede ser hora de lo contrario los estados pueden ser:

- × OFF: Ocurre cuando existe un 0 lógico en el puerto
- × ON: Ocurre cuando existe un 1 lógico en el puerto
- × Cambio: Ocurre cuando se detecta un cambio de 1 a 0 o viceversa.

También existe la posibilidad de complementar una regla con una condición horaria de tal manera que se deben cumplir ambas condiciones para que la regla sea válida y de respuesta afirmativa.

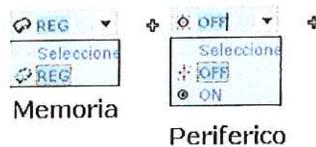
Imagen 44. Condición Horaria



Del Autor

Para esto se deben establecer la hora de inicio y la hora final y en este rango la regla puede ser satisfactoria (Imagen 44). Cuando el periférico seleccionado es

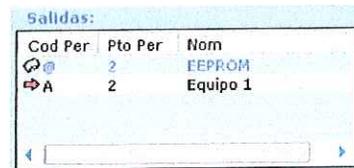
Imagen 45. Estados de Salida



Del Autor

Reloj, la condición horaria es obligatoria. Los estados de salida (Imagen 42) también dependen del periférico elegido en el panel de puerto de salida como se muestra en la Imagen 46.

Imagen 46. Puerto de Salida



Del Auto

Si el puerto elegido tiene como código "@", la única salida posible es REG (Registro), lo que ocasiona que el DPBR almacene un dato en la memoria EEPROM. Si el periférico tiene un código diferente a "@" la salida varia entre:

- × OFF: pone un 0 lógico en la salida seleccionada.
- × ON: pone un 1 lógico en la salida seleccionada.

Si el usuario desea que el DPBR almacene un registro en su memoria, debe seleccionar el botón de almacenar que se encuentra debajo de "Estado de Salida"

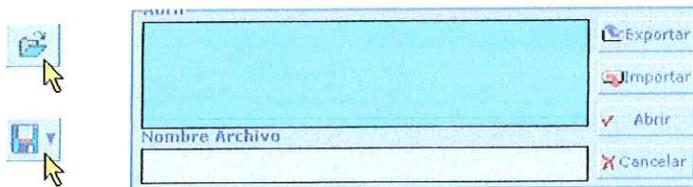
Imagen 47. Almacenar



Del Autor

Si el usuario desea abrir o guardar una configuración previamente establecida, o si desea importar o exportar un archivo externo, debe hacer clic en la barra de herramientas en los botones de abrir o guardar, esto ocasiona que sea expuesto el panel de "Abrir/Guardar" como se puede ver en la Imagen 48.

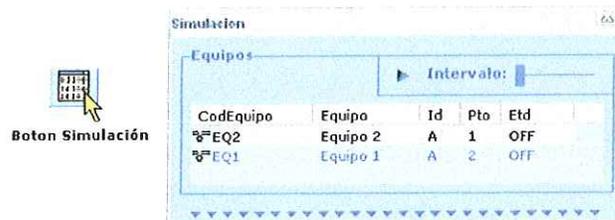
Imagen 48. Reglas DPBR - Abrir Guardar



Del Autor

Para simular la configuración actual, el usuario debe abrir el panel de "Simulación" dando clic en la barra de herramientas en el botón correspondiente a simulación.

Imagen 49. Reglas DPBR - Simulación



Del Autor

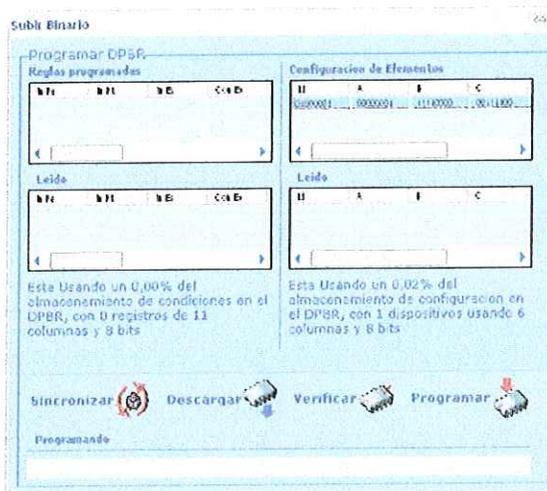
Esta pequeña ventana contiene el algoritmo de análisis que usa el DPBR para poder simular el comportamiento en el mismo con configuración elegida. Aquí se

puede modificar la velocidad entre reglas, ver los datos almacenados en la EEPROM virtual y ver que regla esta siendo satisfactoria.

PROGRAMACIÓN

Para almacenar las reglas establecidas previamente en la ventana de "Creación de Reglas", el usuario debe estar conectado al servidor remoto y en el servidor remoto el DPBR debe estar encendido. La ventana de Programación mostrada en Imagen 50 tiene cuatro botones para diferentes funciones.

Imagen 50. Programación DPBR



Del Autor

De izquierda a derecha:

- × Sincronizar: la función de esta herramienta es la sincronización de la fecha en el DPBR con la hora del Servidor remoto.
- × Descargar: al hacer clic en este botón, el servidor provoca que el DPBR envíe los datos que este tenga almacenado en los bancos de reglas y configuración.
- × Verificación: la función de la herramienta de verificación, es de comparar la configuración guardada y descargada con la que esta actualmente en la base de datos.

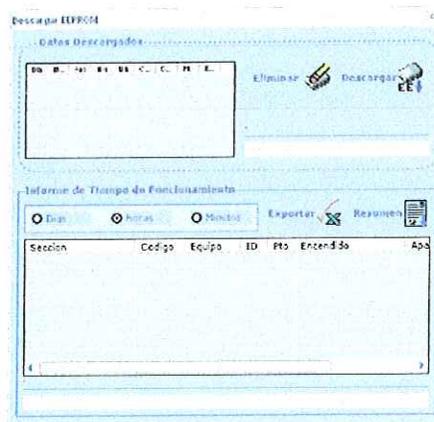
- × Programar: esta herramienta envía al servidor toda la programación para que este la envíe al DPBR.

Luego de Programar el DPBR, la herramienta reconfigura los Periféricos y reinicia el sistema de comunicaciones.

DESCARGA DE MEMORIAS

Para acceder al módulo de "Descargar Memorias" esta ventana el software sigue el mismo algoritmo que usa al ingresar a "Programar DPBR".

Imagen 51. Descargar Memorias



Del Autor

El módulo de "Descargar Memorias" cumple la función de descargar y liberar la memoria de los bancos de almacenamiento. Luego que el usuario descarga la memoria, puede generar un reporte detallado del tiempo de encendido de los equipos para luego exportarlo a Excel.

HERRAMIENTAS DE EMERGENCIA

El módulo de "Herramientas de Emergencia" consta de una barra de

Imagen 52. Herramientas de Emergencia



Del Autor

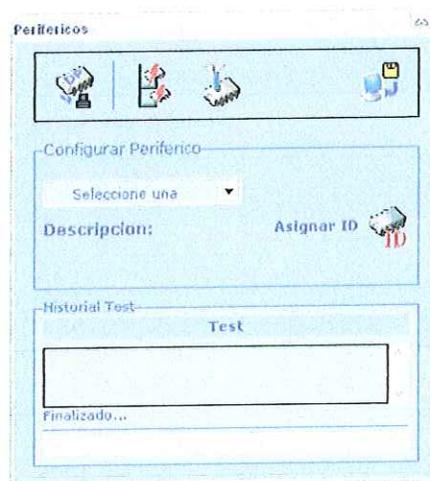
herramientas con siete botones que cumplen las siguientes funciones. (De izquierda a derecha).

- × Reinicio de DPBR: Genera un reset mediante software.
- × Conexión con Servidor↔DPBR: Conecta el DPBR al Servidor para poder usar las demás herramientas.
- × Reinicio de Periféricos: reinicia los periféricos
- × Configuración de Periféricos: configura y reinicia los periféricos.
- × Liberar Bancos de Almacenamiento: libera los bancos de almacenamiento.
- × Sincronizar Fecha: envía la fecha del servidor al DPBR
- × Obtener Fecha: obtiene la fecha del RTC del DPBR y la muestra en el indicador "DATE Consola"

PERIFÉRICOS

En el módulo de Periféricos el usuario puede importar un tipo diferente de periférico, para esto no es necesario estar conectado al DPBR remoto. Desde esta

Imagen 53. Ventana Periféricos



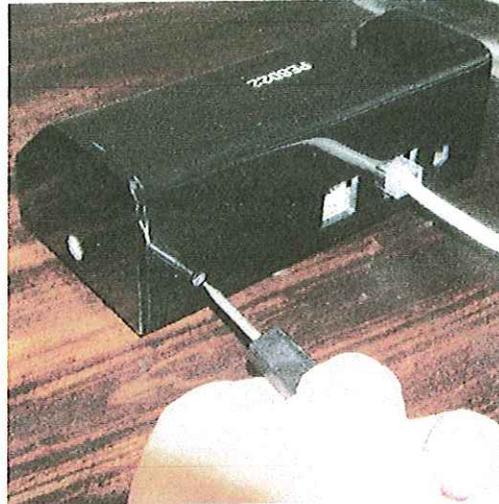
Del Autor

ventana el usuario también puede reiniciar los periféricos, configurar periféricos,

asignar la letra de identificación y hacer un "test" o prueba de comunicación con los periféricos que estén almacenados en la base de datos.

Para configurar los Periféricos el usuario debe presionar el botón rojo ubicado en

Imagen 54. Configuración Periférico (Asignar ID)



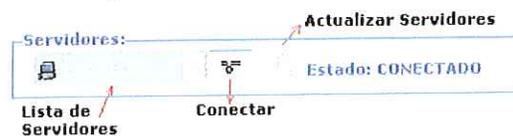
Del Autor

la parte trasera del mismo con un utensilio de punta fina (Imagen 54) y luego dar clic en el botón "Asignar ID"

PANEL DE CONEXIÓN

El panel mostrado en la Imagen 55 cumple la función de establecer la conexión con el servidor remoto.

Imagen 55. Panel de Servidores



Del Autor

En el combo "Lista de Servidores" se muestran los servidores que están registrados, el usuario debe escoger el servidor y luego dar clic en el botón "Conectar". Si la conexión fue satisfactoria, el "Estado" cambia a "CONECTADO"

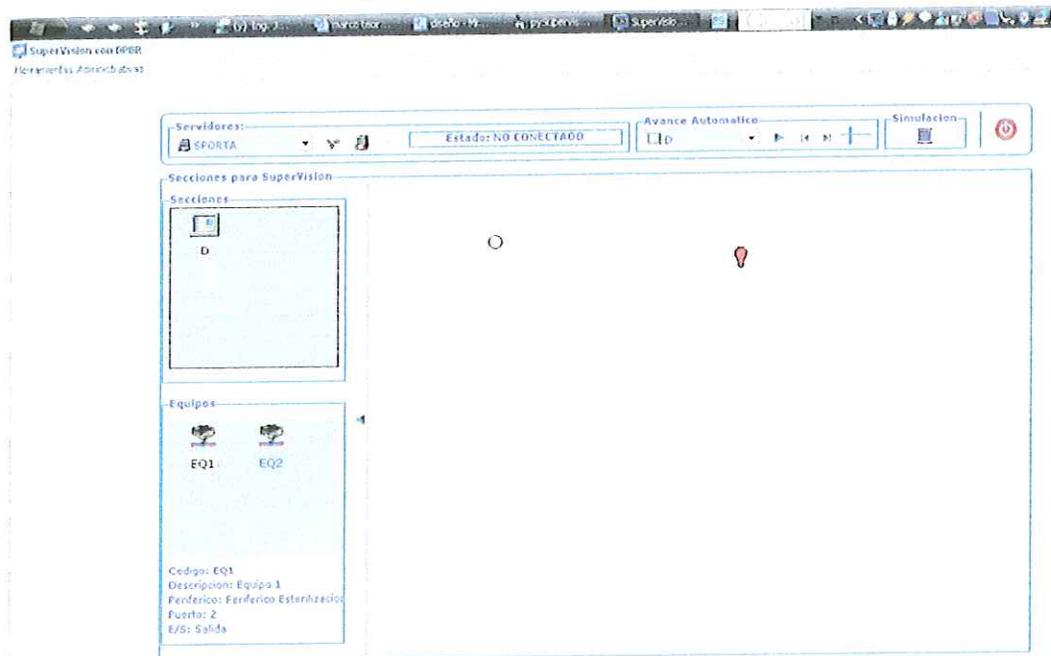
de lo contrario el "Estado" es de "NO CONECTADO". El usuario puede actualizar la lista de servidores para refrescar el combo "Lista de Servidores".

Por ultimo el botón salir cierra la aplicación almacenando la hora y fecha de salida del usuario.

4.1.3. APLICACIÓN SUPERVISIÓN

Al igual que las otras dos aplicaciones, al iniciar se muestra la "Ventana de Inicio" y luego va a la "Ventana de Inicio de Sesión" donde el usuario digita su nombre de usuario y contraseña, si el usuario posee los permisos necesarios va a la ventana principal mostrada a continuación.

Imagen 56. Aplicación Supervisión



Del Autor

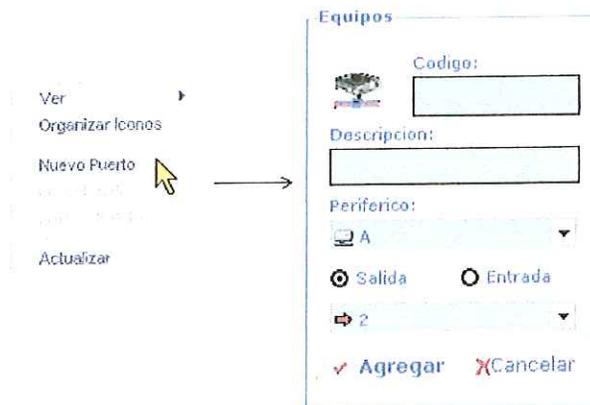
La aplicación de Supervisión consta de dos partes:

- × Visualizador
- × Menú

4.1.3.1. VISUALIZADOR

El visualizador es el panel en donde se exponen los equipos agregados al sistema en forma grafica. Aquí el usuario arrastra los equipos desde el panel de equipos al visualizador. Cada equipo tiene su representación grafica para diferenciarlo de los otros.

Imagen 57. Agregar Nuevo Puerto - Supervisión



The image shows a software interface for adding a new port. On the left, a menu contains options: 'Ver', 'Organizar Iconos', 'Nuevo Puerto', and 'Actualizar'. A mouse cursor is pointing at 'Nuevo Puerto'. An arrow points from this menu to a dialog box titled 'Equipos'. The dialog box contains the following fields and controls:

- Codigo:** A text input field.
- Descripcion:** A text input field.
- Periferico:** A dropdown menu with 'A' selected.
- Salida** (radio button selected) and **Entrada** (radio button unselected).
- Puerto:** A dropdown menu with '2' selected.
- Buttons: **✓ Agregar** and **✗ Cancelar**.

Del Autor

Al igual que en la aplicación de programación, desde la aplicación de supervisión el usuario también puede agregar un nuevo puerto para poder supervisarlos.

El procedimiento es el mismo, el usuario debe seleccionar el código único, la descripción, el periférico y el número de puerto.

Luego de creados los puertos, el usuario debe colocarlos directamente en el visualizador y seleccionar una imagen representativa para identificarlos en el panel de visualización.

Imagen 58. Imagen Representativa

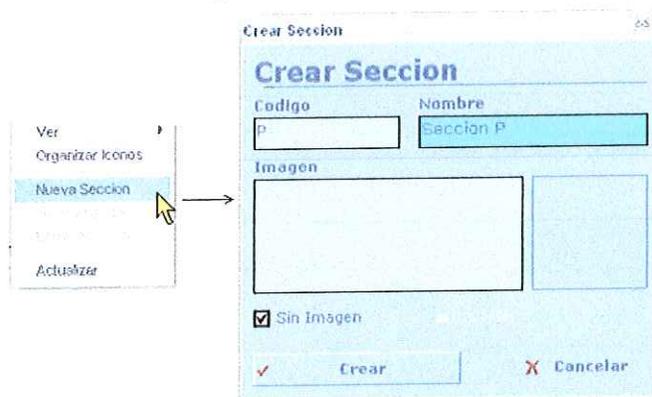


Del Autor

El usuario debe seleccionar el grupo de imágenes que se identifiquen mas con el equipo que se va a representar. Si el usuario no encuentra en los tres grupos predeterminados uno que identifique su equipo, este puede crear un nuevo grupo que sería guardado en la base de datos.

Para crear una sección el usuario debe hacer clic con el botón derecho del Mouse sobre el panel de Secciones, donde aparece un menú desplegable como se muestra en la Imagen

Imagen 59. Crear Sección



Del Autor

En esta ventana el usuario debe especificar el código único de sección y el nombre de la misma. Adicional a esto, el usuario puede escoger una imagen que represente la sección.

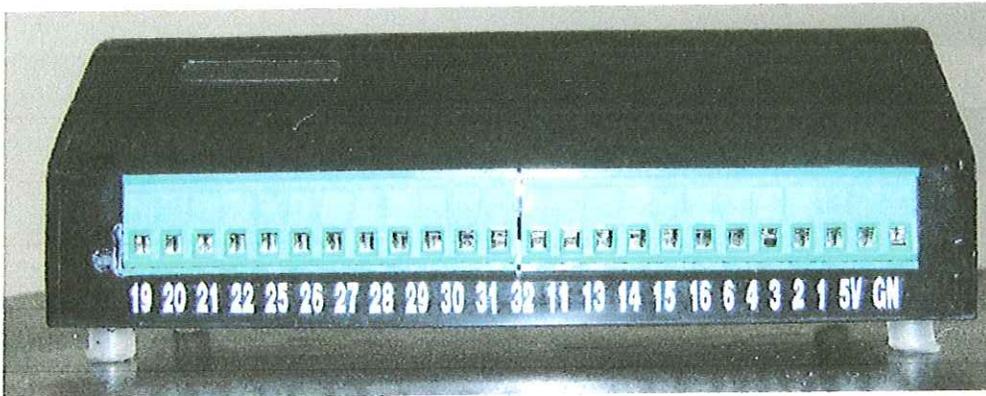
4.1.3.2. MENÚ

La barra de menú de esta aplicación contiene los elementos mostrados en la Imagen 38, igual a la aplicación de programación.

4.2. MONTAJE DE EQUIPOS

Usando el PESD22 el usuario dispone de 22 puertos que pueden ser usados como entrada o salida de manera digital. Estos puertos están distribuidos en el siguiente orden:

Imagen 60. PESD22



Del Autor

Adicionalmente se dispone de una salida constante de 5 voltios y una tierra lógica llamada GN. Para asegurar los equipos al PESD22, este tiene borneras que pueden ser divididas en dos partes de la siguiente manera:

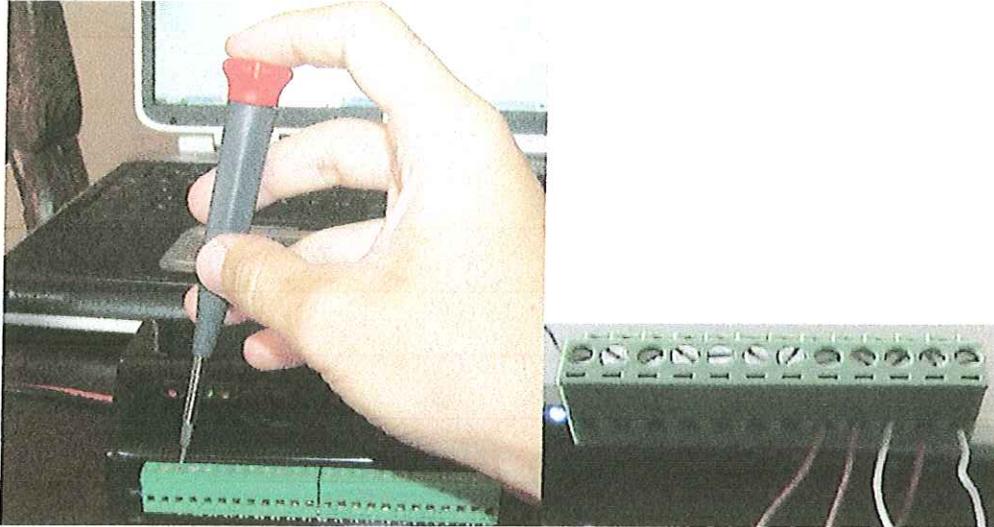
Imagen 61. Borneras PESD22



Del Autor

Las borneras son atornilladas directamente a los equipos como se muestra en la siguiente imagen:

Imagen 62. Atornillar Borneras PESD22



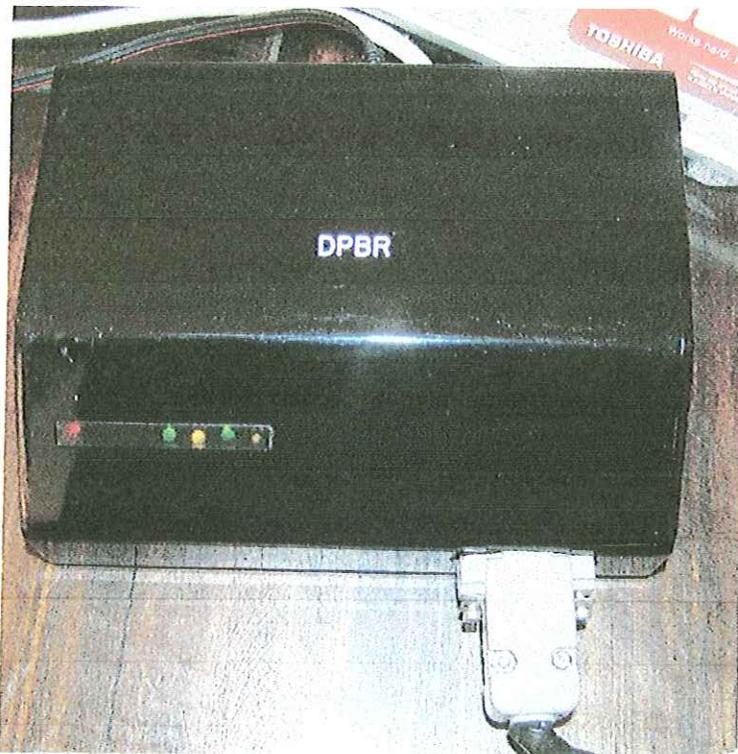
Del Autor

Finalmente los equipos son creados virtualmente en el software para su programación y supervisión (Véase "Panel de Periférico y Puertos, Pág. 86)

4.3. INDICADORES DE FUNCIONAMIENTO

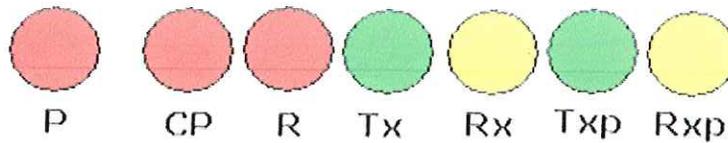
Tanto como el DPBR como el PESD22 contienen indicadores luminosos los cuales nos dicen que tarea esta realizando el controlador interno y también si existen errores en el funcionamiento.

Imagen 63. Indicadores en DPBR



Del Autor

Tabla 9. Indicadores DPBR



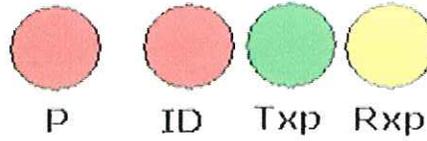
Indicador	Funcion
P	Recibiendo energía de la red
CP	El PC estableció comunicación
R	Frecuencia del Reloj interno
Rx	Recibe Dato de PC
Tx	Envía Dato a PC
Txp	Envía Dato a la red
Rxp	Recibe Dato a la red

Del Autor

El DPBR como lo muestra la Tabla 9 contiene indicadores que realizan las diferentes funciones. Tx y Rx funcionan de manera inversa, están encendidos mientras no tenga actividad con el PC. CP indica que hay supervisión o programación por parte del PC. Txp y Rxp indican que hay actividad en la RED. Si CP, Txp y Rxp iluminan de forma intermitente al mismo tiempo el DPBR nos esta indicando que el espacio en la memoria interna se agoto y si el indicador R no esta intermitente, puede ser por que el reloj interno esta des-configurado y requiere acción del usuario para funcionar.

El PESD22 contiene también indicadores luminosos como se muestra a continuación:

Tabla 10. Indicadores PESD22



Indicador	Funcion
P	Recibiendo energia de la red
ID	El DPBR establecio comunicaci3n
Txp	Envia Dato a la red
Rxp	Recibe Dato a la red

Del Autor

Txp y Rxp al igual que en el DPBR nos indican si existe actividad en la red. El indicador ID solo enciende cuando el DPBR establece comunicaci3n directa con el perif3rico en cuesti3n.

5. CONCLUSIONES

- × Usando la microelectrónica se llegan a soluciones confiables a problemas industriales siempre y cuando sea correctamente manipulada. Es decir, si se diseña correctamente un sistema este da los resultados esperados y puede ser fácilmente modificado de acuerdo a la necesidad real del usuario.
- × Empleando los buses de comunicación combinados con un protocolo de comunicaciones el sistema se puede expandir fácilmente sin sufrir cambios en su hardware si todo se basa en el modelo OSI.
- × Las redes de comunicación son un campo muy importante en una empresa ya que con estas se pueden unificar las diversas bases de datos para obtener información precisa que ayude a la toma de decisiones y automatice procesos no solo industriales sino de ofimática para crear un sistema de información eficiente y confiable.
- × Desarrollando el proyecto se investigo diversos componentes relacionados con el área de automatización (industrial, ofimática, sistemas de información), este es un campo extenso en donde se pueden aplicar conceptos multidisciplinarios, además que se adquieren conocimientos variados sobre campos diversos en la industria, e influye áreas como producción, mantenimiento, distribución, etc.
- × Aplicando el modelo OSI se pueden hacer muchas variaciones y mejoras en cuanto a la estructura del proyecto sin realizar cambios en su estructura básica de Hardware.
- × Usar cable blindado incrementa la distancia entre los puntos en un 25% (según el fabricante) es decir se podría distanciar 1200 metros los puntos

a conectar pero también se incrementa el costo el mismo en casi un 200% y se reduce la cantidad de periféricos comunicados por el bus.

- × Disponer de la fuente de alimentación correcta también influye en la comunicación y en el correcto funcionamiento. Una fuente diseñada correctamente y con respaldo a fallas eléctrica podría ser de gran ayuda para liberar información e impedir que el sistema deje de funcionar cuando exista un corte de luz.

6.RECOMENDACIONES

- × Trabajar con protocolos e interfaces existentes en la actualidad facilita la comunicación de dispositivos propios con dispositivos existentes en el mercado, y debido a que ya han sido desarrollados y probados son muy confiables y eficientes a la hora de desarrollar un sistema que use uno para la interacción. Con este proyecto se desarrollo una interfaz propia que además de ser muy ineficiente también era inestable ya que no siempre se obtenía la información requerida por eso se implemento la interfaz RS485 mencionada en capítulos anteriores. En cuando al protocolo, se desarrollo uno propio lo cual limita un poco la comunicación con componentes comerciales, esto se puede mejorar reemplazando el módulo del protocolo en el DPBR y el PESD22 por algún protocolo comercial pero incrementaría los costos debido al hardware y también a que algunos están protegidos por derechos de autor.
- × El diseño de las carcazas puede ser modificado al igual que el circuito impreso para hacerlo mas modular y así crear un rack para su ubicación final. El circuito impreso puede ser desarrollado con montura superficial lo que reduciría su tamaño en un 60% pero incrementaría casi en un 70% su fabricación, en cuanto a la carcaza, puede ser desarrollada en materiales mas resistentes al medio y ser sellada herméticamente para evitar alteraciones internas en su microelectrónica.
- × Debido a los costos en el desarrollo, los puertos del PESD22 están desprotegidos a fallas eléctricas, esta mejora en una futura versión haría más robusto el sistema. Por el momento es recomendable proteger los puertos para evitar daños irreversibles en el sistema.

- × En la actualidad el puerto serial de un PC tiende a desaparecer en los equipos nuevos, esto obliga en una futura mejora a desarrollar la comunicación por medio del puerto USB incrementando así el alcance del sistema.

7. BIBLIOGRAFÍA

- [1]** Automatas.org. "Scadas".
<http://www.automatas.org/redes/scadas.htm>
- [2]** Webopedia. "What is SCADA - A Word Definition From the Webopedia Computer Dictionary"
<http://www.webopedia.com/TERM/S/SCADA.html>
- [3]** Wales, Jimmy. "SCADA".
<http://es.wikipedia.org/wiki/SCADA>
- [4]** W.Salter, CERN, Geneva, Switzerland, A. Daneels, CERN, Geneva, Switzerland. "What is SCADA"
<http://www.elettra.trieste.it/ICALEPCS99/proceedings/papers/mc1i01.pdf>
- [5]** Loiola Zubia, Iban. "PLC".
http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/PLC/plc.htm
- [6]** Wales, Jimmy. "Controlador Lógico Programable".
http://es.wikipedia.org/wiki/Controlador_l%C3%B3gico_programable
- [7]** Ubiria Manzarraga, Marcos. "ARTICULOS"
<http://www.euskalnet.net/m.ubiria/ARTICULOS.htm>
- [8]** Zator Systems. "El Modelo OSI".
http://www.zator.com/Hardware/H12_2.htm
- [9]** Softtelecom. "El Modelo OSI"
http://www.geocities.com/txmetsb/el_modelo_de_referencia_osi.htm
- [10]** Sánchez Badillo, Eduardo. "El Modelo de referencia OSI de telecomunicaciones".
http://www.geocities.com/txmetsb/el_modelo_de_referencia_osi.htm
- [11]** Wales, Jimmy. "Modelo de interconexión de sistemas abiertos".
<http://es.wikipedia.org/wiki/OSI>
- [12]** Wales, Jimmy. "Microcontrolador".
<http://es.wikipedia.org/wiki/Microcontrolador>
- [13]** I2C Bus. "I2C-Bus: What's that?"
<http://www.i2c-bus.org/>
- [14]** Wales, Jimmy. "RS-232".
<http://es.wikipedia.org/wiki/RS232>
- [15]** Mayné Grau, Jordi. "Sistemas de Comunicaciones Industriales"
<http://tec.upc.es/ie/practi/Sistemas.pdf>

8. ANEXOS

8.1. ANEXO 1

8.1.1. DATASHEET PIC16F87X



PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

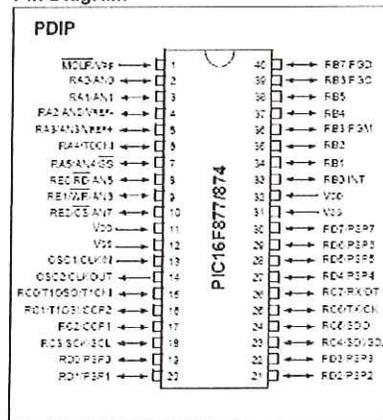
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input; DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory; Up to 368 x 8 bytes of Data Memory (RAM); Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 18-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

8.1.2. DATASHEET MEMORIA EEPROM I²C 24LC256



24AA256/24LC256

256K I²C™ CMOS Serial EEPROM

DEVICE SELECTION TABLE

Part Number	Vcc Range	Max Clock Frequency	Temp Ranges
24AA256	1.8-5.5V	400 kHz†	I
24LC256	2.5-5.5V	400 kHz‡	I, E

† 100 kHz for Vcc < 2.5V.
‡ 100 kHz for E temperature range.

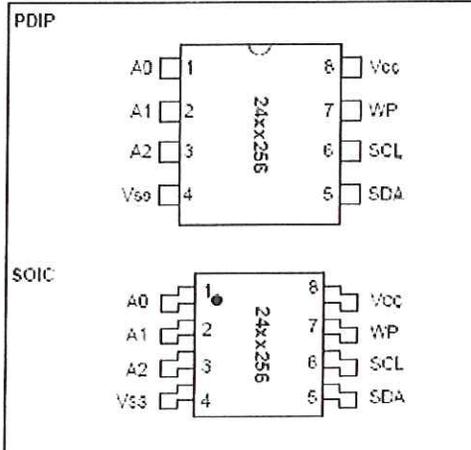
FEATURES

- Low power CMOS technology
 - Maximum write current 3 mA at 5.5V
 - Maximum read current 400 µA at 5.5V
 - Standby current 100 nA typical at 5.5V
- 2-wire serial interface bus, I²C compatible
- Cascadable for up to eight devices
- Self-timed ERASE/WRITE cycle
- 64-byte page-write mode available
- 5 ms max write-cycle time
- Hardware write protect for entire array
- Schmitt trigger inputs for noise suppression
- 100,000 erase/write cycles guaranteed
- Electrostatic discharge protection > 4000V
- Data retention > 200 years
- 8-pin PDIP and SOIC (208 mil) packages
- Temperature ranges:
 - Industrial (I): -40°C to +85°C
 - Automotive (E): -40°C to +125°C

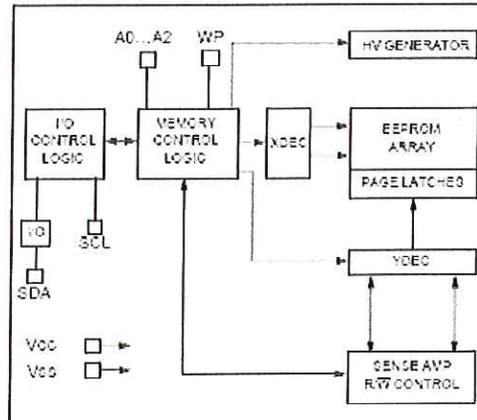
DESCRIPTION

The Microchip Technology Inc. 24AA256/24LC256 (24xx256*) is a 32K x 8 (256K bit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.8V to 5.5V). It has been developed for advanced, low power applications such as personal communications or data acquisition. This device also has a page-write capability of up to 64 bytes of data. This device is capable of both random and sequential reads up to the 256K boundary. Functional address lines allow up to eight devices on the same bus, for up to 2 Mbit address space. This device is available in the standard 8-pin plastic DIP, and 8-pin SOIC (208 mil) packages.

PACKAGE TYPE



BLOCK DIAGRAM



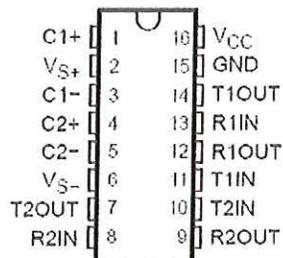
8.1.3. DATASHEET TRANSCEIVER MAX-232

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- \pm 30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22 - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202
- Applications
 - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept \pm 30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
		Tube of 40	MAX232D	MAX232
	SOIC (D)	Reel of 2500	MAX232DR	
		Tube of 40	MAX232DW	MAX232
	SOIC (DW)	Reel of 2000	MAX232DWR	
-40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
		Tube of 40	MAX232ID	MAX232I
	SOIC (D)	Reel of 2500	MAX232IDR	
		Tube of 40	MAX232IDW	MAX232I
	SOIC (DW)	Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

8.1.4. DATASHEET TRANSCEIVER MAX-485

120122; Rev 5; 2006



Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

General Description

The MAX481, MAX483, MAX485, MAX487-MAX491, and MAX1487 are low-power transceivers for RS-485 and RS-422 communication. Each part contains one driver and one receiver. The MAX483, MAX487, MAX488, and MAX489 feature reduced slew-rate drivers that minimize EMI and reduce reflections caused by improperly terminated cables, thus allowing error-free data transmission up to 250kbps. The driver slew rates of the MAX481, MAX485, MAX490, MAX491, and MAX1487 are not limited, allowing them to transmit up to 2.5Mbps.

These transceivers draw between 120µA and 500µA of supply current when unloaded or fully loaded with disabled drivers. Additionally, the MAX481, MAX483, and MAX487 have a low-current shutdown mode in which they consume only 0.1µA. All parts operate from a single 5V supply.

Drivers are short-circuit current limited and are protected against excessive power dissipation by thermal shutdown circuitry that places the driver outputs into a high-impedance state. The receiver input has a fail-safe feature that guarantees a logic-high output if the input is open circuit.

The MAX487 and MAX1487 feature quarter-unit-load receiver input impedance, allowing up to 128 MAX487/MAX1487 transceivers on the bus. Full-duplex communications are obtained using the MAX488-MAX491, while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are designed for half-duplex applications.

Applications

Low-Power RS-485 Transceivers
 Low-Power RS-422 Transceivers
 Level Translators
 Transceivers for EMI-Sensitive Applications
 Industrial-Control Local Area Networks

Features

- ◆ In µMAX Package: Smallest 8-Pin SO
- ◆ Slew-Rate Limited for Error-Free Data Transmission (MAX483/487/488/489)
- ◆ 0.1µA Low-Current Shutdown Mode (MAX481/483/487)
- ◆ Low Quiescent Current:
 120µA (MAX483/487/488/489)
 230µA (MAX1487)
 300µA (MAX481/485/490/491)
- ◆ -7V to +12V Common-Mode Input Voltage Range
- ◆ Three-State Outputs
- ◆ 30ns Propagation Delays, 5ns Skew (MAX481/485/490/491/1487)
- ◆ Full-Duplex and Half-Duplex Versions Available
- ◆ Operate from a Single 5V Supply
- ◆ Allows up to 128 Transceivers on the Bus (MAX487/MAX1487)
- ◆ Current-Limiting and Thermal Shutdown for Driver Overload Protection

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX481CFA	0°C to +70°C	8 Plastic DIP
MAX481CSA	0°C to +70°C	8 SO
MAX481CUA	0°C to +70°C	8 µMAX
MAX481C/D	0°C to +70°C	Dice*

Ordering Information continued at end of data sheet.

* Contact factory for dice specifications.

Selection Table

PART NUMBER	HALF/FULL DUPLEX	DATA RATE (Mbps)	SLEW-RATE LIMITED	LOW-POWER SHUTDOWN	RECEIVER/DRIVER ENABLE	QUIESCENT CURRENT (µA)	NUMBER OF TRANSMITTERS ON BUS	PIN COUNT
MAX481	Half	2.5	No	Yes	Yes	300	32	8
MAX483	Half	0.25	Yes	Yes	Yes	120	32	8
MAX485	Half	2.5	No	No	Yes	300	32	8

8.1.5. DATASHEET RTC DS1307



DS1307 64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C*, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the battery supply.

** I²C is a trademark of Philips Corp. Purchase of I²C components of Maxim Integrated Products, Inc., or one of its sublicensed Associated Companies, conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips Corp.*

FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, Nonvolatile (NV) RAM for Data Storage
- I²C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratory (UL) Recognized

Typical Operating Circuit and Pin Configurations appear at end of data sheet.

ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307

8.2. ANEXO 2

8.2.1. NORMA TIA/EIA-568-B

TIA/EIA-568-B es un conjunto de tres estándares que trata el cableado de edificios comerciales para productos y servicios de telecomunicaciones. Los tres estándares tienen los siguientes títulos oficiales: ANSI/TIA/EIA-568-B.1-2001, -B.2-2001 y -B.3-2001. Los estándares TIA/EIA-568-B se publicaron por primera vez en 2001. Sustituyen al conjunto de estándares TIA/EIA-568-A que han quedado obsoletos.

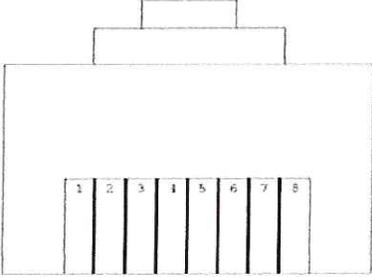
Tal vez la característica más conocida del TIA/EIA-568-B.1-2001 sea la asignación de pares/pines en los cables de 8 hilos y 100 ohmios. Esta asignación se conoce como T568A y T568B, y a menudo es nombrada (erróneamente) como TIA/EIA-568A and TIA/EIA-568B.

Respecto al estándar de conexión, los pines en un conector RJ-45 modular están numerados del 1 al 8, siendo el pin 1 el del extremo izquierdo del conector, y el pin 8 el del extremo derecho. Los pines del conector hembra se numeran de la misma manera para que coincidan con esta numeración, siendo el pin 1 el del extremo derecho y el pin 8 el del extremo izquierdo³⁹.

La asignación de pares de cables es como se muestra en la siguiente tabla:

³⁹ Según Wales, Jimmy - <http://es.wikipedia.org/wiki/TIA-568B>

Tabla 11. Par Trenzado

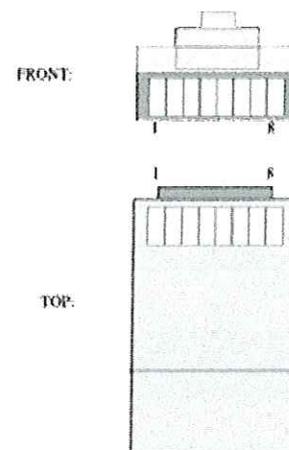
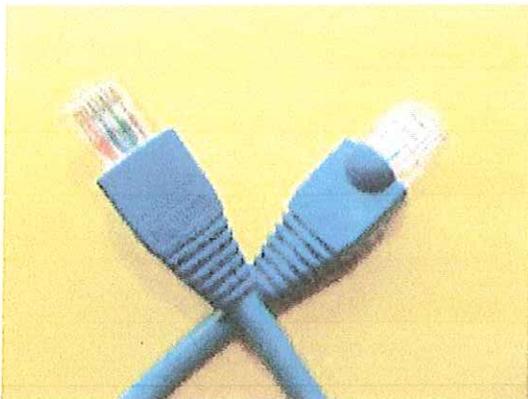
Pin	Color T568A	Color T568B	Pines en conector macho (hembra invertidos)
1	 Blanco/Verde (W-G)	 Blanco/Naranja (W-O)	
2	 Verde (G)	 Naranja (O)	
3	 Blanco/Naranja (W-O)	 Blanco/Verde (W-G)	
4	 Azul (BL)	 Azul (BL)	
5	 Blanco/Azul (W-BL)	 Blanco/Azul (W-BL)	
6	 Naranja (O)	 Verde (G)	
7	 Blanco/Marrón (W-BR)	 Blanco/Marrón (W-BR)	
8	 Marrón (BR)	 Marrón (BR)	

Wales, Jimmy - <http://es.wikipedia.org/wiki/TIA-568B>

8.2.2. CONECTOR RJ-45

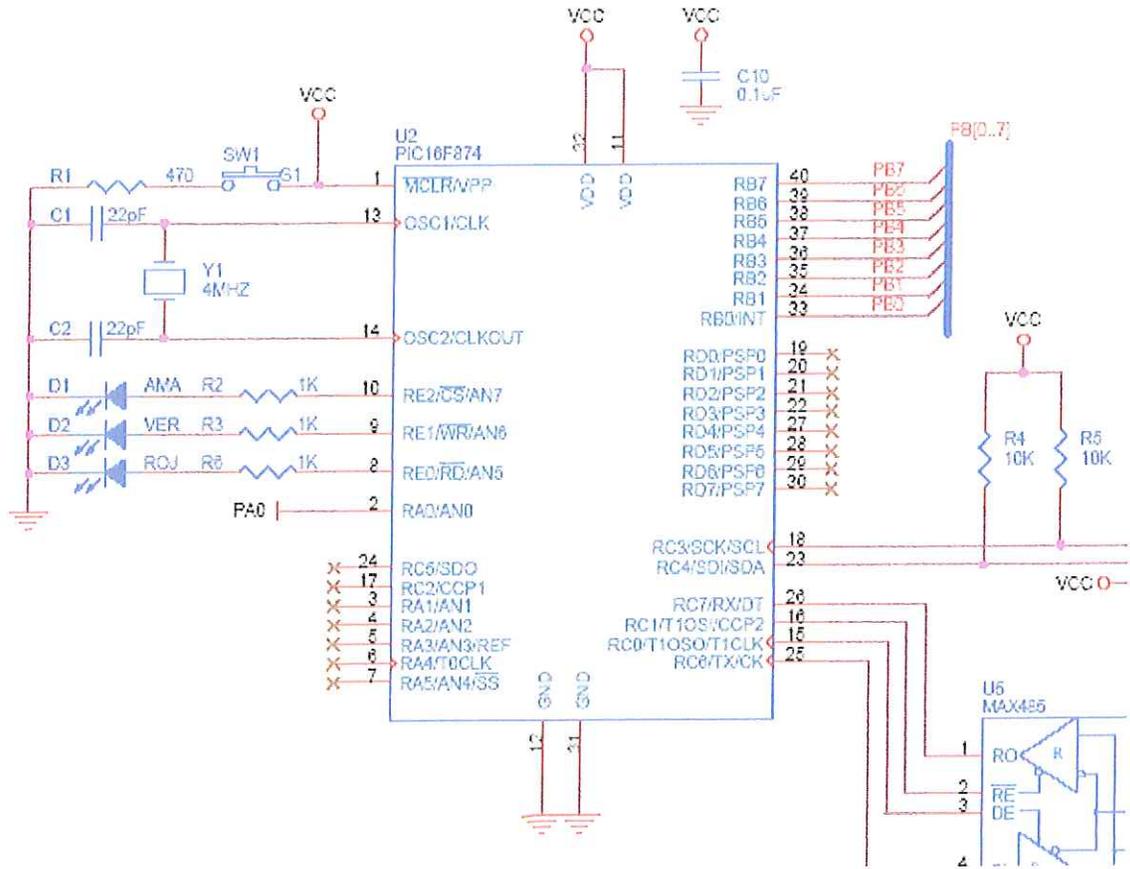
El RJ45 es una interfaz física comúnmente usada para conectar redes de cableado estructurado, (categorías 4, 5, 5e y 6). RJ es un acrónimo inglés de Registered Jack que a su vez es parte del Código Federal de Regulaciones de Estados Unidos. Posee ocho 'pines' o conexiones eléctricas, que normalmente se usan como extremos de cables de par trenzado. Es utilizada comúnmente con estándares como EIA/TIA-568B, que define la disposición de los pines o wiring pinout. Para que todos los cables funcionen en cualquier red, se sigue un estándar a la hora de hacer las conexiones.

Imagen 64. RJ-45



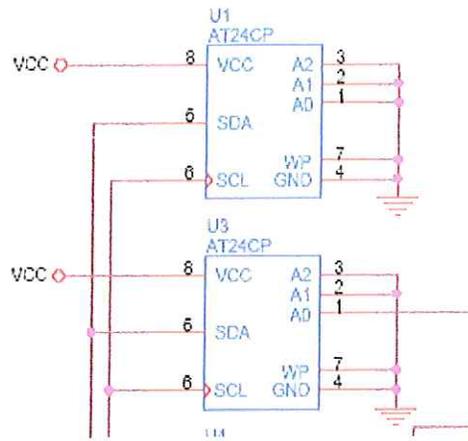
Wales, Jimmy - <http://es.wikipedia.org/wiki/RJ45>

Imagen 66. Diseño Electrónico DPBR (CPU)



Del Autor

Imagen 67. Diseño Electrónico DPBR (Memorias)



Del Autor

Imagen 68. Diseño Electrónico DPBR (RTC)

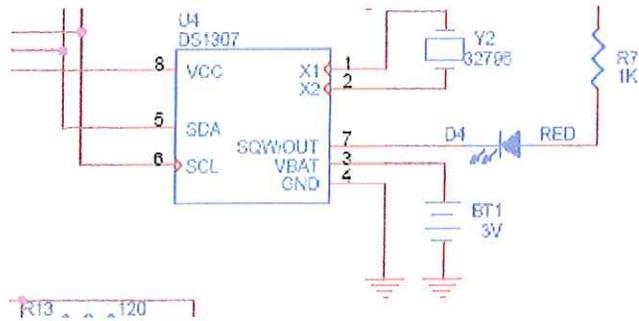


Imagen 69. Diseño Electrónico DPBR (Comunicación RED)

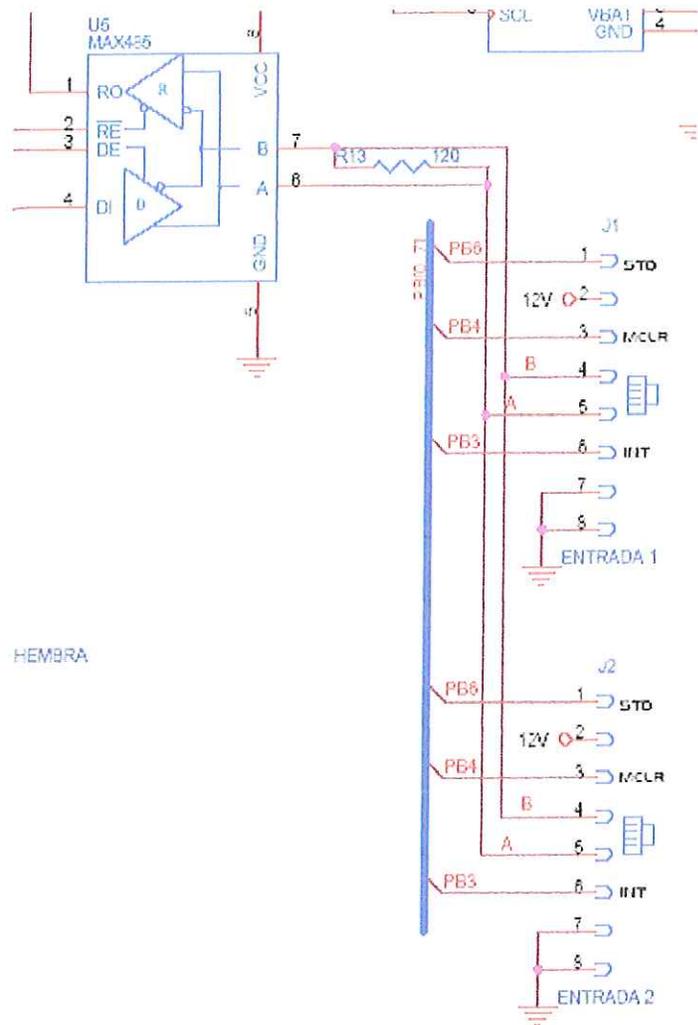


Imagen 70. Diseño Electrónico DPBR (Comunicación PC)

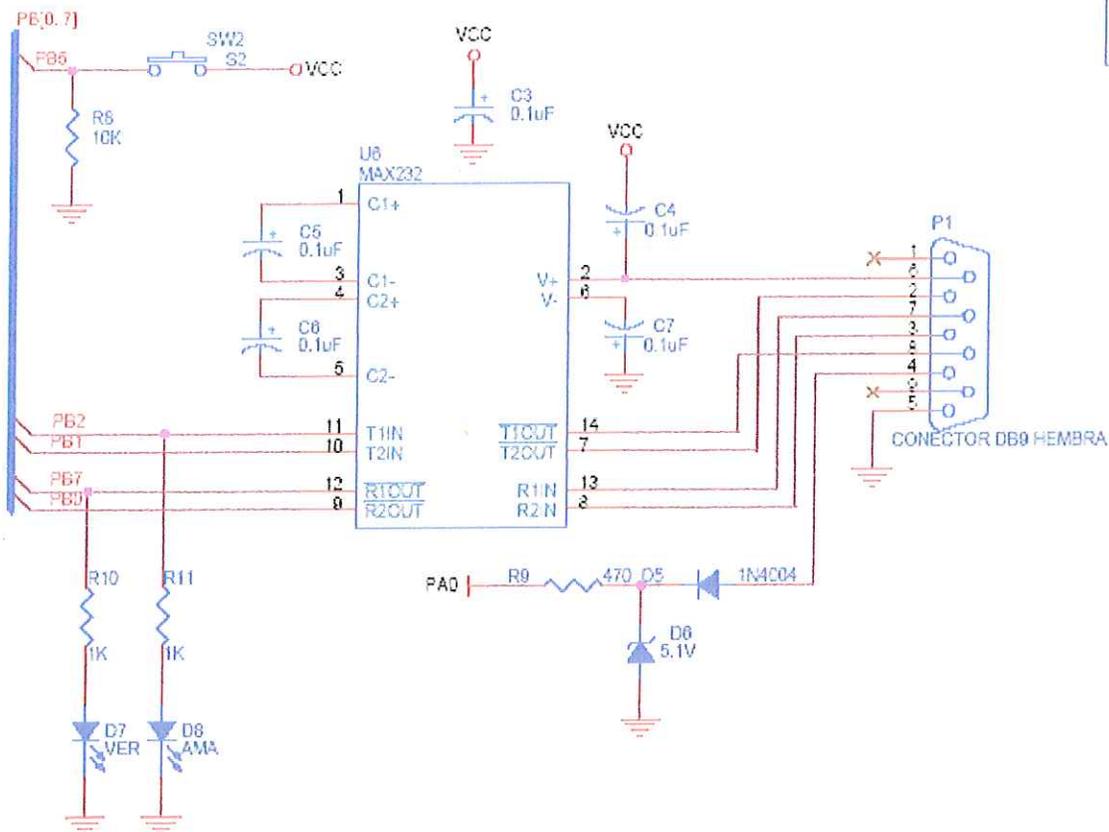


Imagen 71. Diseño Electrónico DPBR (Fuente)

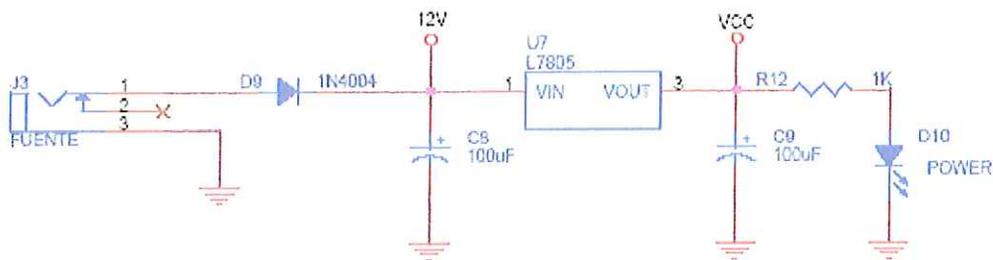
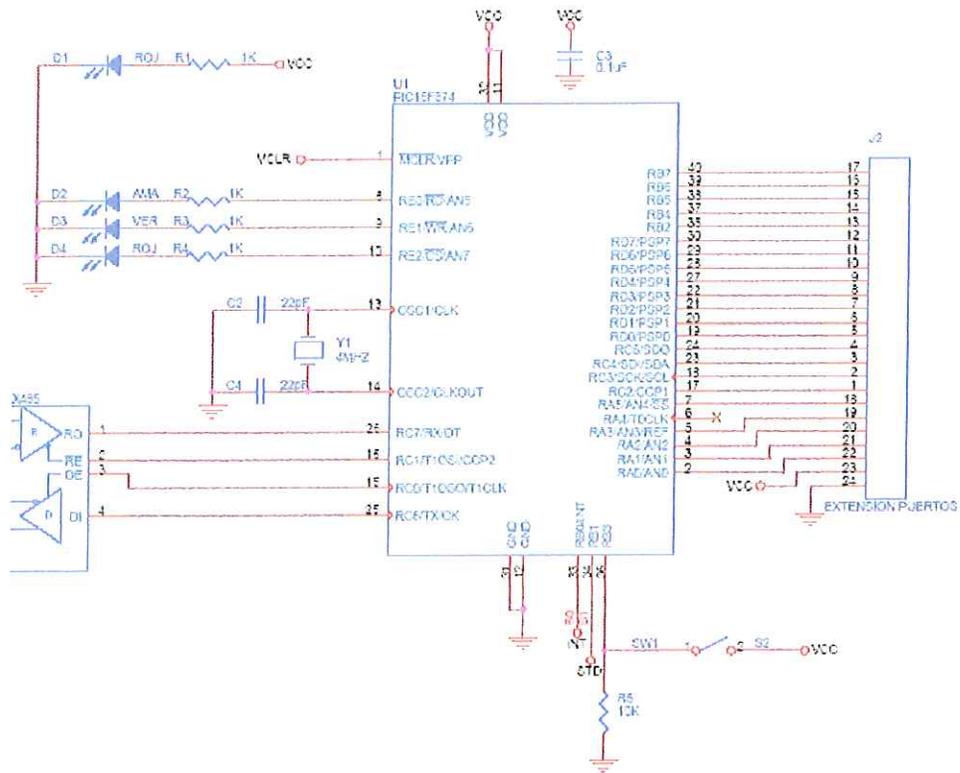
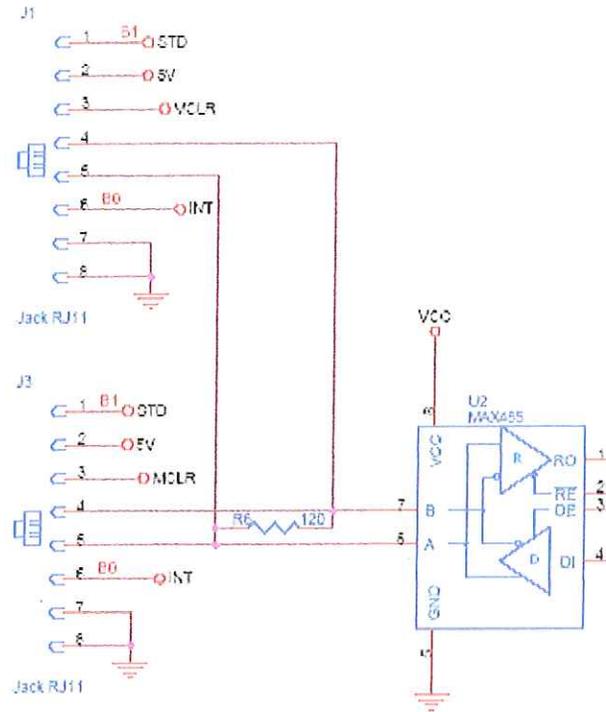


Imagen 73. Diseño Electrónico PESD22(CPU)



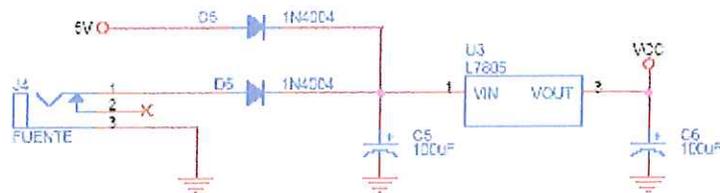
Del Autor

Imagen 74. Diseño Electrónico PESD22 (Comunicación)



Del Autor

Imagen 75. Diseño Electrónico PESD22 (Fuente)



Del Autor

8.3.2. LISTA DE MATERIALES

Tabla 12. Materiales DPBR

Fecha: 19/02/2006		Lista de Materiales					
Página: 1 de 1							
Versión: 1							
Equipo: DPBR							
Descripción: Dispositivo Programable Basado en Reglas							
Cantidad	Unidad	Tipo	Componente	Referencia	Valor Unitari	Valor Total	Valor IVA
2	UNO	CI	Memoria EEPROM	24LC256	\$ 6.465,52	\$ 12.931,03	\$ 15.000,00
1	UNO	CI	Microcontrolador	PTC16F877A	\$ 13.793,10	\$ 13.793,10	\$ 16.000,00
1	UNO	CI	Reloj Tiempo Real	DS1307	\$ 7.327,59	\$ 7.327,59	\$ 8.500,00
1	UNO	CI	Transceiver	MAX232	\$ 2.758,62	\$ 2.758,62	\$ 3.200,00
1	UNO	CI	Transceiver	MAX485	\$ 5.172,41	\$ 5.172,41	\$ 6.000,00
1	UNO	CI	Regulador	LM7805	\$ 950,00	\$ 950,00	\$ 1.102,00
4	UNO	Condensador	Condensador	1uF (Polarizado)	\$ 215,52	\$ 862,07	\$ 1.000,00
2	UNO	Condensador	Condensador	22pF	\$ 150,00	\$ 300,00	\$ 348,00
3	UNO	Diodo	Diodo	LED Rojo	\$ 129,31	\$ 387,93	\$ 450,00
2	UNO	Diodo	Diodo	LED Verde	\$ 129,31	\$ 258,62	\$ 300,00
2	UNO	Diodo	Diodo	LED Amarillo	\$ 129,31	\$ 258,62	\$ 300,00
1	UNO	Diodo	Diodo Rectificador	1N4004	\$ 43,10	\$ 43,10	\$ 50,00
1	UNO	Diodo	Diodo Zener	5,1V	\$ 258,62	\$ 258,62	\$ 300,00
2	UNO	Otros	Jack	RJ45	\$ 1.293,10	\$ 2.586,21	\$ 3.000,00
1	UNO	Otros	Base	Base 40 Pines	\$ 258,62	\$ 258,62	\$ 300,00
4	UNO	Otros	Base	Base 8 Pines	\$ 172,41	\$ 689,66	\$ 800,00
1	UNO	Otros	Base	Base 16 Pines	\$ 215,52	\$ 215,52	\$ 250,00
1	UNO	Otros	Cristal	32,768 Mhz	\$ 1.120,69	\$ 1.120,69	\$ 1.300,00
1	UNO	Otros	Cristal	4 Mhz	\$ 1.120,69	\$ 1.120,69	\$ 1.300,00
1	UNO	Otros	Jack	12V	\$ 344,83	\$ 344,83	\$ 400,00
2	UNO	Otros	Pulsadores	Pulsador Abierto	\$ 603,45	\$ 1.206,90	\$ 1.400,00
1	UNO	Otros	Zoquets	DB9	\$ 1.724,14	\$ 1.724,14	\$ 2.000,00
6	UNO	Resistencia	Resistencia	1K Ω	\$ 17,24	\$ 103,45	\$ 120,00
3	UNO	Resistencia	Resistencia	470 Ω	\$ 17,24	\$ 51,72	\$ 60,00
2	UNO	Resistencia	Resistencia	10K Ω	\$ 17,24	\$ 34,48	\$ 40,00
1	UNO	Resistencia	Resistencia	120 Ω	\$ 17,24	\$ 17,24	\$ 20,00
1	UNO	Otros	Bateria	3V	\$ 2.155,17	\$ 2.155,17	\$ 2.500,00
1	UNO	Otros	Porta Bateria	Porta Bateria	\$ 1.293,10	\$ 1.293,10	\$ 1.500,00
Total Componentes:						\$	\$ 67.540,00

Del Autor

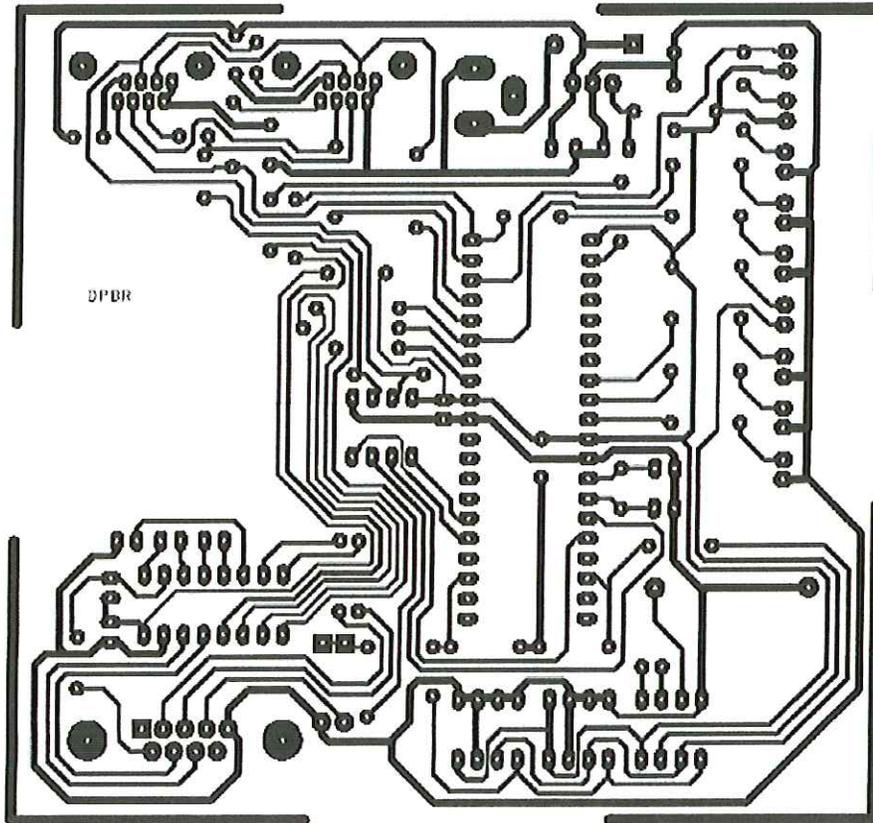
Tabla 13. Lista Materiales PESD22

Lista de Materiales							
Fecha:	19/02/2006						
Página:	1 de 1						
Versión:	1						
Equipo:	Periférico E/S Digitales						
Descripción:	Periférico de entradas y salidas digitales de 22 puertos						
Cantidad	Unidad	Tipo	Componente	Referencia	Valor Unitari	Valor Total	Valor IVA
1	UNO	CI	Microcontrolador	PTC16F877A	\$ 13,793.10	\$ 13,793.10	\$ 16,000.00
1	UNO	CI	Transceiver	MAX485	\$ 5,172.41	\$ 5,172.41	\$ 6,000.00
1	UNO	CI	Regulador	LM7805	\$ 818.97	\$ 818.97	\$ 950.01
2	UNO	Condensador	Condensador	22pF	\$ 129.31	\$ 258.62	\$ 300.00
2	UNO	Diodo	Diodo	LED Rojo	\$ 129.31	\$ 258.62	\$ 300.00
1	UNO	Diodo	Diodo	LED Verde	\$ 129.31	\$ 129.31	\$ 150.00
1	UNO	Diodo	Diodo	LED Amarillo	\$ 129.31	\$ 129.31	\$ 150.00
1	UNO	Otros	Base	Base de 40 Pines	\$ 258.62	\$ 258.62	\$ 300.00
1	UNO	Otros	Base	Base de 8 Pines	\$ 172.41	\$ 172.41	\$ 200.00
1	UNO	Otros	Cristal	4 Mhz	\$ 1,120.69	\$ 1,120.69	\$ 1,300.00
2	UNO	Otros	Pulsadores	Pulsador Abierto	\$ 603.45	\$ 1,206.90	\$ 1,400.00
2	UNO	Otros	Jack	RJ45	\$ 1,293.10	\$ 2,586.21	\$ 3,000.00
2	UNO	Otros	Jack	12 puertos	\$ 8,189.66	\$ 16,379.31	\$ 19,000.00
1	UNO	Otros	Jack	12V	\$ 344.83	\$ 344.83	\$ 400.00
3	UNO	Resistencia	Resistencia	470 Ω	\$ 17.24	\$ 51.72	\$ 60.00
5	UNO	Resistencia	Resistencia	1K Ω	\$ 17.24	\$ 103.44	\$ 119.99
1	UNO	Resistencia	Resistencia	120 Ω	\$ 17.24	\$ 17.24	\$ 20.00
Total Componentes:					\$	\$ 49,629.99	\$

Del Autor

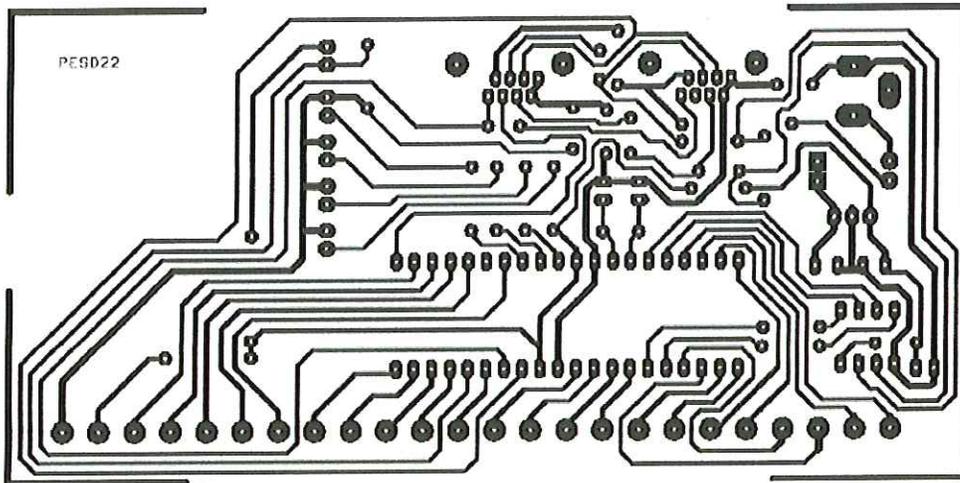
8.3.3. CIRCUITOS IMPRESOS

Imagen 76. Circuito Impreso DPBR



Del Autor

Imagen 77. Circuito Impreso PESD22

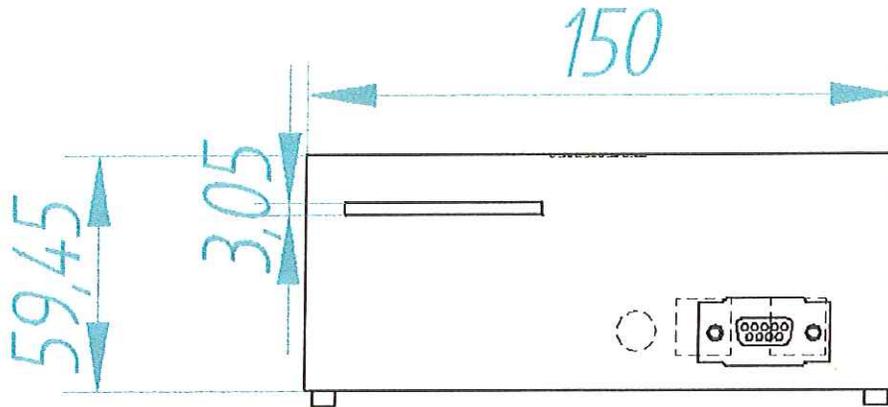


Del Autor

8.3.4. DISEÑO DE CARCAZAS

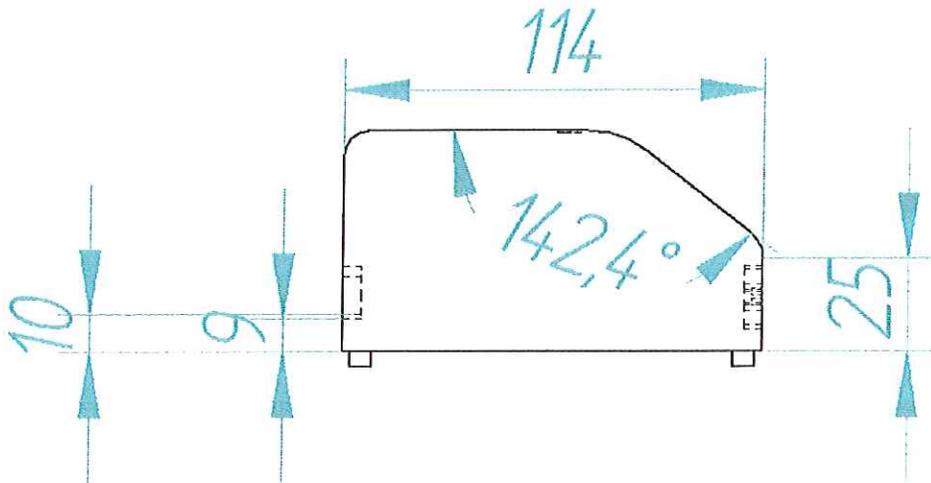
Las dimensiones están en milímetros.

Imagen 78. Vista Frontal DPBR



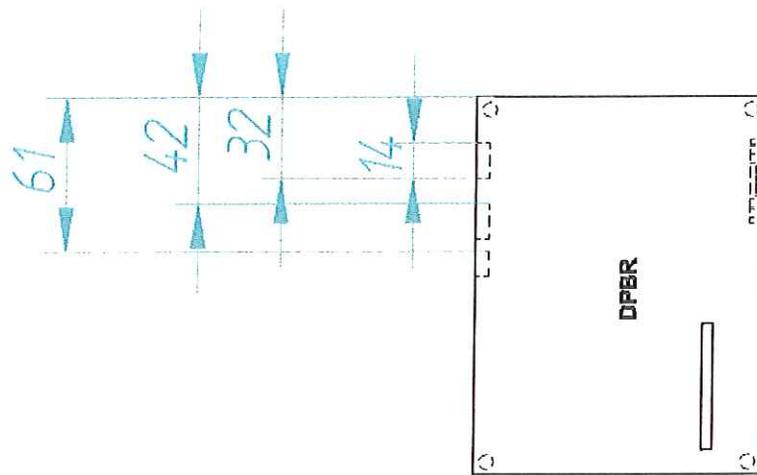
Del Autor

Imagen 79. Vista Lateral DPBR



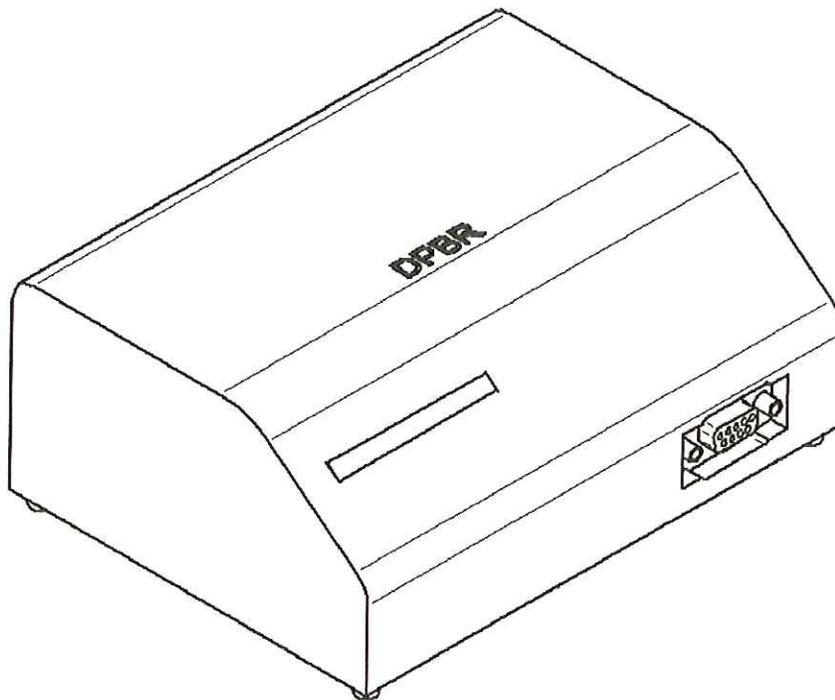
Del Autor

Imagen 80. Vista Superior DPBR



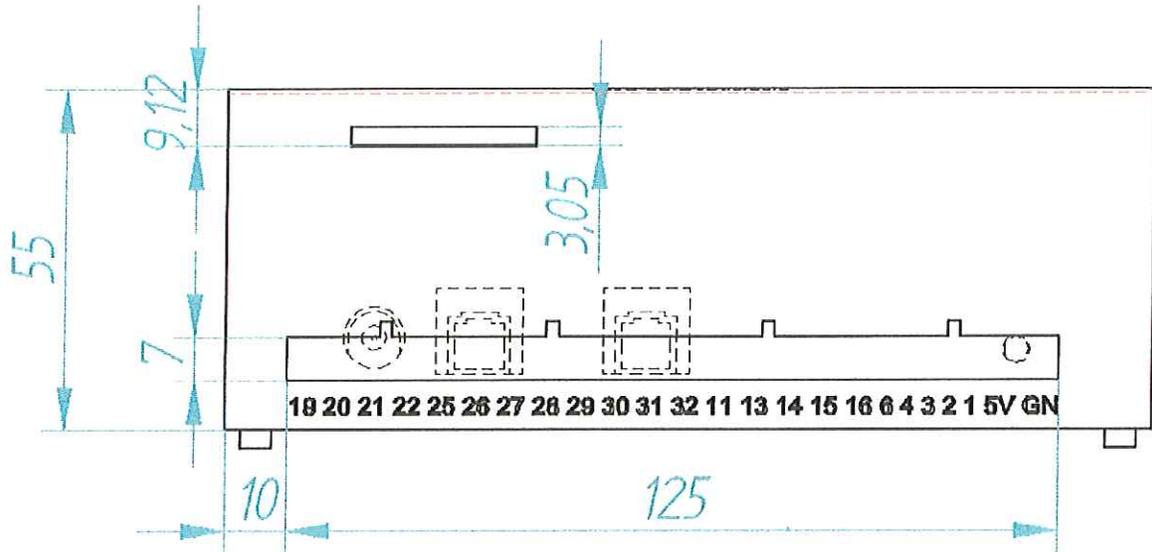
Del Autor

Imagen 81. Isometría DPBR



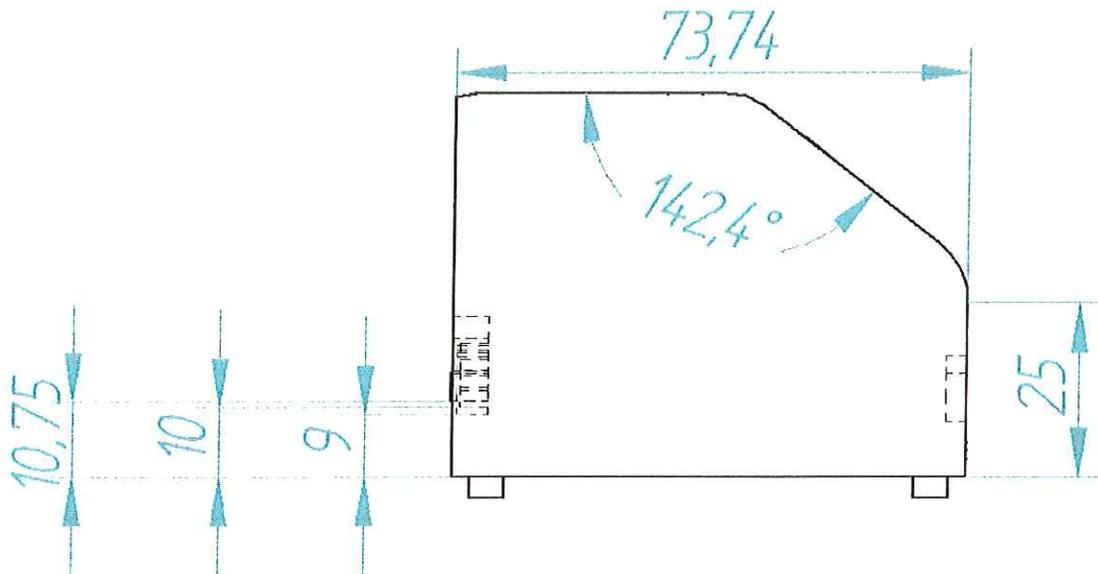
Del Autor

Imagen 82. Vista Frontal PESD22



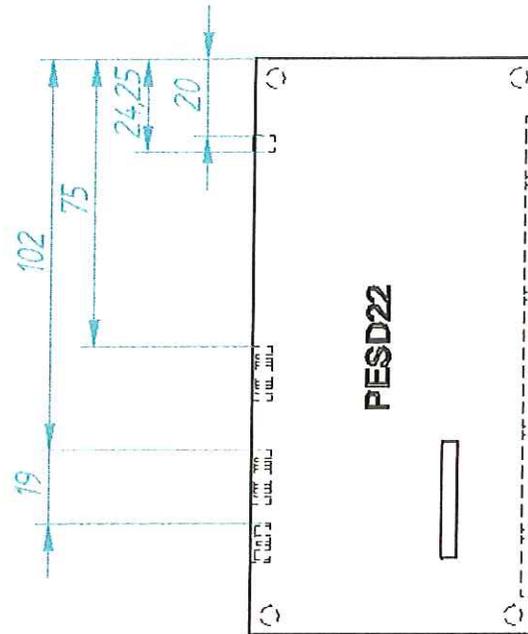
Del Autor

Imagen 83. Vista Lateral PESD22



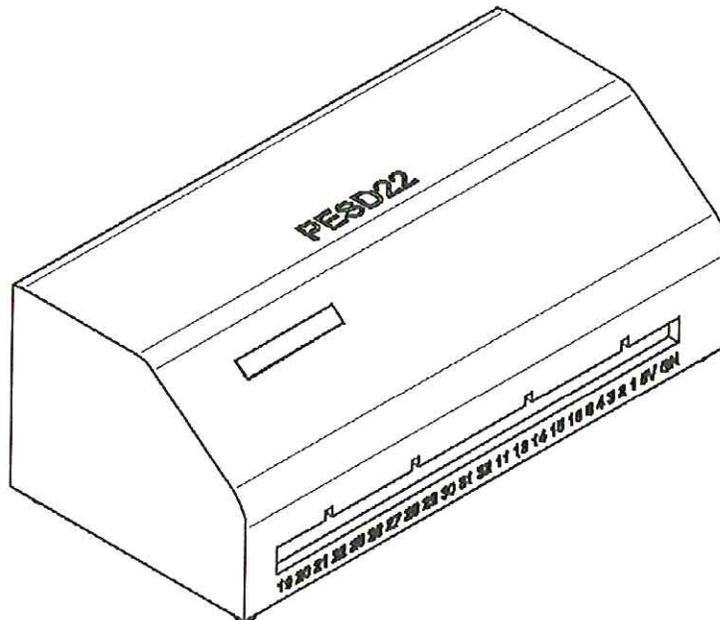
Del Autor

Imagen 84. Vista Superior PESD22



Del Autor

Imagen 85. Isometría PESD22



Del Autor

8.4. ANEXO 4

Véase contenido completo en CD anexo