

TRABAJO ESPECIAL DE GRADO

**DISEÑO DE UN SISTEMA DE CONTROL SUPERVISORIO Y DE
ADQUISICIÓN DE DATOS PARA EL CAUDAL DEL BLOQUE DE
ACONDICIONAMIENTO DE LA PLANTA DE ENSAYOS Y OPERACIÓN
DE REFORMADO DE METANO**

Presentado ante la ilustre
Universidad Central de Venezuela
por el Br. Pagliarone A. Carlos J.
para optar al Título de Ingeniero
Electricista

Caracas, 2016

TRABAJO ESPECIAL DE GRADO

DISEÑO DE UN SISTEMA DE CONTROL SUPERVISORIO Y DE ADQUISICIÓN DE DATOS PARA EL CAUDAL DEL BLOQUE DE ACONDICIONAMIENTO DE LA PLANTA DE ENSAYOS Y OPERACIÓN DE REFORMADO DE METANO

Prof. Guía: Ebert Brea
Tutor Industrial: Ing. Ricardo A., Olejnik A.

Presentado ante la ilustre
Universidad Central de Venezuela
por el Br. Pagliarone A. Carlos J.
para optar al Título de Ingeniero
Electricista

Caracas, 2016

CONSTANCIA DE APROBACIÓN

Caracas, 07 de noviembre de 2016

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Carlos Pagliarone:

**“DISEÑO DE UN SISTEMA DE CONTROL SUPERVISORIO Y DE
ADQUISICIÓN DE DATOS PARA EL CAUDAL DEL BLOQUE DE
ACONDICIONAMIENTO DE LA PLANTA DE ENSAYOS Y OPERACIÓN
DE REFORMADO DE METANO”**

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Electrónica y Control, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.

Prof. Alejandro González
Jurado

Prof. Panayotis Tremante
Jurado

Prof. Ebert Brea
Prof. Guía

DEDICATORIA

A Dios.

A mis padres, mi hermana y mi familia.

A mis amigos y compañeros.

A la Universidad Central de Venezuela y mis profesores.

RECONOCIMIENTOS Y AGRADECIMIENTOS

Ante todo le agradezco a Dios, por ser guía de mí transitar por la vida, dándome la oportunidad de alcanzar nuevas metas y desarrollarme integralmente como persona.

Igualmente le agradezco inmensamente a mis padres, hermana y familiares, por los valores impartidos, la confianza y ayuda incondicional.

A mi amigo Herman y su madre Nancy, por apoyarme en cada momento e impulsarme a culminar esta etapa de mi vida.

A mis compañeros de estudio, en especial a Luis, Jhonny y Rubén, por su motivación para alcanzar esta meta y estar cada vez que los necesité.

A mis amigas y compañeras de clases, Andrea, Claudia, Francismar, Marta, Naty, Loredana e Yvelc, por su apoyo durante la carrera.

A mis amigos de la Escuela de Ingeniería Mecánica, Alberto, David, Gabriel, Kristoffer y Luis, por los momentos compartidos desde el inicio de la carrera.

A mis amigos de la Facultad de Ingeniería, Bracho, David, Génesis, Karla, María, Luigi, Luis, Paola, Paolo, Pedro y Yuli, quienes comenzaron conmigo esta meta y hoy en día continúan apoyándome.

A mis amigos de toda la vida por su amistad incondicional durante todo este tiempo.

A mis compañeros de trabajo, Davilo Chiu y Otto Lambertus, por sus consejos y el apoyo en cada una de las fases de este trabajo.

Al Ing. Carlos Di Yorio e Ing. Luis Sarco, por el apoyo y ofrecerme la oportunidad de trabajar en EDT Ingeniería, lugar en el que complementé los conocimientos de mi formación profesional.

A mi profesor guía Ebert Brea y mi tutor Ricardo Olejnik, por el esfuerzo, disposición y conocimientos compartidos.

A los profesores Mercedes Arocha, Nerio Ojeda, Rafael Rivero, Servando Álvarez y Víctor Urbina, por los consejos brindados para la realización de este trabajo.

Al profesor Simón Morales y la Escuela de Ingeniería Eléctrica, por facilitarme el sistema embebido Intel Galileo.

A la profesora Omaira Camacaro, por prestarme las instalaciones y hacer posible la ejecución de esta investigación.

A los profesores de la Escuela de Ingeniería Eléctrica y de la Facultad de Ingeniería, por contribuir en mi formación personal y profesional.

A la Universidad Central de Venezuela por albergarme durante todo este tiempo y brindarme la posibilidad de llegar hasta aquí.

Sin Dios y éstas personas, este trabajo no hubiera sido posible.

Pagliarone C.

Pagliarone A., Carlos J.

**DISEÑO DE UN SISTEMA DE CONTROL SUPERVISORIO Y DE
ADQUISICIÓN DE DATOS PARA EL CAUDAL DEL BLOQUE DE
ACONDICIONAMIENTO DE LA PLANTA DE ENSAYOS Y
OPERACIÓN DE REFORMADO DE METANO**

Prof. Guía: Ebert Brea. Tutor Industrial: Ing. Ricardo A., Olejnik A. Tesis. Caracas U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Mención: Electrónica, Computación y Control. Institución: Escuela de Ingeniería Química Trabajo de Grado. 2016. páginas 114 + Anexos.

Palabras claves: SCADA; Reformado de metano; Controladores de flujo; Sistemas embebidos; Intel Galileo; Python; LabVIEW.

Resumen. Se presenta el diseño de un sistema de control supervisorio y de adquisición de datos del bloque de caudal de una planta de reformado de metano, para la producción de gas síntesis. Se utilizan cuatro controladores de flujo que miden el caudal, la temperatura y la presión de los gases que intervienen en la reacción. Esta información es adquirida por un sistema embebido, a través de un protocolo de comunicación, que procesa y retransmite los datos a un programa desarrollado en una computadora. Con el sistema se pueden ajustar los puntos de consigna de los controladores de flujo, se muestran las variables en tiempo real y se generan alarmas de seguridad. El resultado es una interfaz gráfica sencilla, mediante la cual se ejecutan acciones de control supervisorio, mejorando la eficiencia, la adquisición de datos y el control del bloque de caudal de la planta de reformado de metano.

ÍNDICE GENERAL

CONSTANCIA DE APROBACIÓN.....	iii
DEDICATORIA.....	iv
RECONOCIMIENTOS Y AGRADECIMIENTOS.....	v
RESUMEN.....	vii
ÍNDICE GENERAL.....	viii
LISTA DE TABLAS.....	xiii
LISTA DE FIGURAS, ILUSTRACIONES O GRÁFICOS.....	xiv
SIGLAS Y ACRÓNIMOS.....	xvii
INTRODUCCIÓN.....	1
CAPÍTULO I.....	3
1. MARCO REFERENCIAL.....	3
1.1 ANTECEDENTES.....	3
1.2 PLANTEAMIENTO DEL PROBLEMA.....	3
1.3 OBJETIVOS.....	4
1.3.1 OBJETIVO GENERAL.....	4
1.3.2 OBJETIVOS ESPECÍFICOS.....	4
CAPÍTULO II.....	6
2. MARCO TEÓRICO.....	6
2.1. REFORMADO DE METANO.....	6
2.2. SCADA.....	9
2.3. MICROPROCESADORES Y MICROCONTROLADORES.....	12
2.4. SISTEMAS EMBEBIDOS.....	14
2.5. COMUNICACIÓN SERIAL.....	14
2.6. ESTÁNDAR DE COMUNICACIÓN RS-232.....	17

2.7. INTEL GALILEO.....	21
2.7.1. Arduino IDE.....	22
2.7.2. Intel XDK.....	24
2.7.3. Intel System Studio IoT Edition.....	25
2.7.4. Python.....	26
2.8. MODBUS.....	26
2.8.1. Modbus TCP.....	29
2.8.2. Modbus RTU.....	30
2.8.3. Modbus ASCII.....	30
2.9. LABVIEW.....	31
CAPÍTULO III.....	33
3. OPERACIÓN Y DIAGNOSTICO DE LA PLANTA.....	33
3.1. DESCRIPCIÓN DE LA PLANTA.....	33
3.2. PROCESO DE OPERACIÓN DE LA PLANTA.....	33
3.2.1. Bloque de alimentación.....	34
3.2.2. Bloque de acondicionamiento.....	34
3.2.3. Bloque de reacción.....	36
3.2.4. Bloque de separación.....	37
3.2.5. Bloque de análisis.....	37
3.3. DIAGNÓSTICO DEL BLOQUE DE ACONDICIONAMIENTO DE LA PLANTA.....	38
3.3.1. Controlador de flujo.....	38
3.3.1.1 Principio de funcionamiento.....	39
3.3.1.2. Alimentación y conexiones de señales.....	41
3.3.1.3. Funciones del controlador de flujo.....	45
3.3.1.3.1. Pantalla de inicio.....	45
3.3.1.3.2. Pantalla de selección de menú.....	47
3.3.1.3.3. Pantalla de configuración de control.....	47
3.3.1.3.4. Pantalla de selección de gas.....	49

3.3.1.3.5. Pantalla de selección de comunicación.....	50
3.3.1.3.6. Pantalla de información de fabricante.....	51
3.3.1.3.7. Pantalla de misceláneos.....	52
3.3.1.4. Comunicación RS-232.....	53
3.3.2. Circuito impreso multiplexor RS-232.....	56
3.3.3. Módulos FP-1001 y DP-DO-401.....	58
CAPÍTULO IV.....	60
4. DISEÑO DEL SISTEMA DE SCADA.....	60
4.1. DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS.....	60
4.1.1. Selección del sistema embebido.....	61
4.1.1.1. Raspberry Pi 1 modelo B+.....	61
4.1.1.2. Intel Galileo Gen 1.....	62
4.1.1.3. BeagleBone Black.....	63
4.1.1.4. Requisitos del sistema y comparación.....	64
4.1.2. Selección del método de comunicación entre el sistema embebido y los controladores de flujo.....	65
4.1.2.1. RS-232.....	66
4.1.2.2. RS-422.....	66
4.1.2.3. RS-485.....	66
4.1.2.4. Requisitos del sistema y comparación.....	67
4.1.3. Diseño del circuito impreso.....	69
4.1.3.1. MAX232.....	69
4.1.3.2. CD4052B.....	71
4.1.3.3. Descripción del circuito impreso.....	72
4.2. DISEÑO DEL SISTEMA DE VISUALIZACIÓN Y CONTROL.....	77
4.2.1. Selección de software de visualización.....	77
4.2.1.1. SIMATIC WinCC.....	77
4.2.1.2. Factory Talk View SE.....	78
4.2.1.3. LabVIEW.....	78

4.2.1.4. Requisitos del sistema y comparación.....	78
4.2.2. Selección del criterio de control.....	79
4.2.3. Selección del método de comunicación entre el sistema embebido y el software de visualización.....	82
4.2.3.1. Modbus TCP.....	83
4.2.3.2. Ethernet IP.....	83
4.2.3.3. Secure Shell.....	83
4.2.3.4. Requisitos del sistema y comparación.....	83
CAPÍTULO V.....	85
5. PROGRAMACIÓN DE LOS ELEMENTOS QUE CONFORMAN EL SISTEMA SCADA.....	85
5.1. PROGRAMACIÓN DEL SISTEMA EMBEBIDO.....	85
5.1.1. Imagen del sistema operativo Linux.....	86
5.1.2. Python y librerías utilizadas.....	86
5.1.2.1 Wiring-x86.....	87
5.1.2.2. Pyserial.....	87
5.1.2.3. Pymodbus.....	87
5.1.2.4. Expresiones regulares.....	88
5.1.3. Descripción del programa.....	88
5.2. PROGRAMACIÓN DEL SOFTWARE DE VISUALIZACIÓN.....	91
5.2.1. LabVIEW y módulos utilizados.....	91
5.2.2. Descripción del programa.....	92
CAPÍTULO VI.....	95
6. VALIDACIÓN Y RESULTADOS.....	95
6.1. PRUEBAS DE FUNCIONAMIENTO INTEL GALILEO.....	96
6.1.1. Multiplexor.....	96
6.1.2. Comunicación serial.....	96
6.1.3. Flotantes a enteros sin signo.....	97
6.1.4. Comprobación de trama.....	98

6.1.5. Modbus.....	99
6.1.6. Apagar el sistema embebido.....	100
6.2. PRUEBAS DE FUNCIONAMIENTO LABVIEW.....	100
6.2.1. Inicio de sesión.....	101
6.2.2. Interfaz humano máquina.....	101
6.2.3. Monitoreo en tiempo real.....	102
6.2.4. Alarma de emergencia a la salida del controlador.....	103
6.2.5. Alarma de emergencia a la entrada del controlador.....	103
6.2.6. Registro de datos.....	104
CONCLUSIONES.....	105
RECOMENDACIONES.....	107
REFERENCIAS BIBLIOGRAFICAS.....	108
ANEXOS.....	114

LISTA DE TABLAS

Tabla N° 1. Comparación entre SCADA basado en objetos y “tags”.....	11
Tabla N° 2. Niveles de tensión en la comunicación RS-232.....	18
Tabla N° 3. Terminales comunes entre el conector DB-9 y DB-25.....	19
Tabla N° 4. Niveles lógicos aceptados por diferentes familias de circuitos integrados a distintos niveles de tensión.....	20
Tabla N° 5. Bloques de memoria del protocolo Modbus.....	26
Tabla N° 6. Relación entre la dirección de memoria y el número de referencia.....	26
Tabla N° 7. Códigos de funciones estándares del protocolo Modbus.....	28
Tabla N° 8. Conector mini-DIN de 8 terminales.....	42
Tabla N° 9. Selección de gas.....	54
Tabla N° 10. Comparación de los sistemas embebidos.....	64
Tabla N° 11. Comparación entre los estándares RS-232, RS-422 y RS485.....	67
Tabla N° 12. Funciones de los terminales del integrado MAX232.....	70
Tabla N° 13. Funciones de los terminales del integrado CD4052B.....	72
Tabla N° 14. Requisitos mínimos del sistema.....	78

LISTA DE FIGURAS, ILUSTRACIONES O GRÁFICOS

Figura N° 1. Arquitectura de un sistema SCADA.....	11
Figura N° 2. Microprocesador.....	13
Figura N° 3. Microcontrolador.....	13
Figura N° 4. Transmisión síncrona.....	15
Figura N° 5. Transmisión asíncrona.....	15
Figura N° 6. Conectores DB-9 y DB-25 macho y hembra.....	19
Figura N° 7. Conexiones entre terminales para convertir de DB-25 a DB-29.....	20
Figura N° 8. Configuración básica del integrado MAX232.....	21
Figura N° 9. Placa de desarrollo Intel Galileo.....	22
Figura N° 10. Arduino software IDE.....	23
Figura N° 11. Intel XDK.....	25
Figura N° 12. Intel System Studio IoT Edition.....	25
Figura N° 13. Transmisión de datos Modbus TCP.....	30
Figura N° 14. Transmisión de datos Modbus RTU.....	30
Figura N° 15. Transmisión de datos Modbus ASCII.....	31
Figura N° 16. Bloque de alimentación.....	34
Figura N° 17. Bloque de acondicionamiento.....	36
Figura N° 18. Bloque de reacción.....	36
Figura N° 19. Bloque de separación.....	37
Figura N° 20. Bloque de análisis.....	38
Figura N° 21. Conectores jack y mini-DIN de 8 terminales.....	42
Figura N° 22. Método analógico para transmitir el punto de consigna a los controladores.....	43
Figura N° 23. Conexión mini-DIN de 8 terminales a conector serial DB-9.....	44
Figura N° 24. Cable adaptador conector mini-DIN de 8 terminales a conector serial DB-9.....	45
Figura N° 25. Pantalla de inicio.....	46
Figura N° 26. Pantalla de selección de menú.....	47

Figura N° 27. Pantalla de configuración de control.....	48
Figura N° 28. Pantalla de selección de gas.....	50
Figura N° 29. Pantalla de selección de comunicación.....	50
Figura N° 30. Pantalla de información del fabricante “Mfg 1”.....	51
Figura N° 31. Pantalla de información del fabricante “Mfg 2”.....	52
Figura N° 32. Pantalla de misceláneos.....	52
Figura N° 33. Diseño esquemático del multiplexor.....	57
Figura N° 34. Diseño físico de la placa multiplexor.....	57
Figura N° 35. Módulo National Instruments FP-1001.....	58
Figura N° 36 Módulo de salidas digitales FP-DO-401.....	59
Figura N° 37. Módulo National Instruments instalados en el laboratorio.....	59
Figura N° 38. Raspberry Pi 1 modelo B+.....	62
Figura N° 39. Características del Intel Galileo.....	63
Figura N° 40. BeagleBone Black.....	64
Figura N° 41. Arquitectura multipunto para RS-422 y RS-485.....	67
Figura N° 42. Configuración de terminales del integrado MAX232.....	70
Figura N° 43. Configuración de terminales del integrado CD4052B.....	71
Figura N° 44. Diseño esquemático de la placa de multiplexado.....	74
Figura N° 45. Diseño físico de la placa de multiplexado.....	74
Figura N° 46. Diseño esquemático de la placa de acople al sistema embebido.....	75
Figura N° 47. Diseño físico de la placa de acople al sistema embebido.....	75
Figura N° 48. Circuito impreso con los integrados MAX232 y CD4052B.....	76
Figura N° 49. Circuito impreso con los bornes y “pin headers”.....	76
Figura N° 50. Circuitos elaborados acoplados al sistema embebido.....	76
Figura N° 51. Ajuste en el Dominio de la frecuencia Aström y Hägglung.....	80
Figura N° 52. Método de la curva de reacción Cohen y Coon.....	81
Figura N° 53. Diagrama de flujo del script principal de sistema embebido.....	89
Figura N° 54. Diagramas de flujo de las funciones principales.....	90

Figura N° 55. Diagramas de flujo de los scripts secundarios.....	91
Figura N° 56. Diagrama de flujo del programa en LabVIEW.....	94
Figura N° 57. Comparación entre el controlador de flujo y el flujómetro.....	95
Figura N° 58. Prueba de funcionamiento de control del multiplexor.....	96
Figura N° 59. Prueba de funcionamiento de la comunicación serial.....	96
Figura N° 60. Prueba de funcionamiento de la conversión de flotante a entero sin signo.....	98
Figura N° 61. Prueba de funcionamiento de la comprobación de trama.....	99
Figura N° 62. Prueba de funcionamiento del servidor y cliente Modbus TCP.....	99
Figura N° 63. Prueba de funcionamiento apagar el sistema embebido.....	100
Figura N° 64. Prueba de funcionamiento de inicio de sesión.....	101
Figura N° 65. Prueba de funcionamiento de la interfaz humano máquina.....	102
Figura N° 66. Prueba de funcionamiento de la monitorización en tiempo real.....	102
Figura N° 67. Prueba de funcionamiento de la alarma a la salida.....	103
Figura N° 68. Prueba de funcionamiento de la alarma a la entrada.....	103
Figura N° 69. Prueba de funcionamiento del registro de datos.....	104

SIGLAS Y ACRÓNIMOS

UCV: Universidad Central de Venezuela.

SCADA: *Supervisory Control and Data Acquisition* (Control supervisorio y adquisición de datos)

HMI: *Human Machine Interface* (Interfaz humano máquina)

CPU: *Central Processing Unit* (Unidad central de procesamiento)

ROM: *Read Only Memory* (Memoria de solo lectura)

RAM: *Random Access Memory* (Memoria de acceso aleatorio)

I/O: *In/Out* (Entrada/Salida)

I²C: *Inter-Integrated Circuit* (Circuito interintegrado)

SPI: *Serial Peripheral Interface* (Interfaz serial periférica)

UART: *Universal Asynchronous Receiver-Transmitter* (Receptor-Transmisor universal asíncrono)

TIA: *Telecommunications Industry Association* (Asociación de la industria de telecomunicaciones)

RS-232: *Recommended Standard 232* (Estándar recomendado 232)

TTL: *Transistor-Transistor Logic* (Lógica transistor-transistor)

CMOS: *Complementary Metal Oxide Semiconductor* (Semiconductor complementario de óxido metálico)

SoC: *System on a Chip* (Sistema en chip)

IDE: *Integrated Development Environment* (Entorno de desarrollo integrado)

ICSP: *In Circuit Serial Programming* (Programación serial en circuito)

IoT: *Internet of Thing* (Internet de las cosas)

USB: *Universal Serial Bus* (Bus universal en serie)

OSI: *Open System Interconnection* (Interconexión de sistemas abiertos)

PLC: *Programmable Logic Controller* (Controlador lógico programable)

TCP: *Transmission Control Protocol* (Protocolo de control de transmisión)

IP: *Internet Protocol* (Protocolo Internet)

RTU: *Remote Terminal Unit* (Unidad terminal remota)

ASCII: *American Standard Code for Information Interchange* (Código estándar estadounidense para el intercambio de información)

CRC: *Cyclic Redundancy Check* (Verificación por redundancia cíclica)

LRC: *Longitudinal Redundancy Check* (Verificación de redundancia longitudinal)

PID: *Proportional Integral Derivative* (Proporcional integral derivativo)

TCV: *Thermal Conductivity Detector* (Detector de conductividad térmica)

AC: *Alternating Current* (Corriente alterna)

DC: *Direct Current* (Corriente directa)

DIN: *Deutsches Institut für Normung* (Instituto alemán de estandarización)

LCD: *Liquid Crystal Display* (Pantalla de Cristal Líquido)

PSIA: *Pounds per Square Inch Absolute* (Libras por pulgada cuadrada absoluta)

GPIO: *General Purpose Input/Output* (Entrada/Salida de propósito general)

HDMI: *High-Definition Multimedia Interface* (Interfaz multimedio de alta definición)

SD: *Secure Digital* (Seguro digital)

CSI: *Camera Serial Interface* (Interfaz serial de cámara)

DSI: *Display Serial Interface* (Interfaz serial de pantalla)

PWM: *Pulse-Width Modulation* (Modulación por ancho de pulsos)

eMMC: *embedded Multimedia Card* (Tarjeta multimedia embebida)

STP: *Shielded Twisted Pair* (Par trenzado apantallado)

SSH: *Secure Shell* (Interprete de ordenes seguro)

FAT: *File Allocation Table* (Tabla de asignación de archivos)

API: *Application Programming Interface* (Interfaz de programación de aplicaciones)

UDP: *User Datagram Protocol* (Protocolo de datagrama de usuario)

DSC: *Datalogging and Supervisory Control* (*Registro de datos y control supervisorio*)

SLPM: *Standard Liters Per Minute* (Litros estándar por minuto)

INTRODUCCIÓN

El alza de la demanda de energía a nivel mundial y la necesidad de disminuir la emisión de gases de efecto invernadero, impulsa la búsqueda de formas alternas de generar energía. El gas síntesis sería para Venezuela una de las opciones a tomar en cuenta, ya que la emisión de dióxido de carbono a la atmósfera es poca; posee un gran reservorio natural y su producción es atractiva económicamente. Por ello en la Escuela de Ingeniería Química de la Universidad Central de Venezuela (UCV), se realizan investigaciones para profundizar sobre su obtención y aplicación práctica, utilizando una planta a escala para estudiar la reacción del reformado autotérmico de metano empleando catalizadores de níquel soportados sobre alúmina, para la producción de este gas.

En esta época la automatización de procesos se ha convertido en una necesidad, y sobre todo si es un proceso industrial, incluso a pequeña escala, del cual se necesitan analizar las variables que intervienen en él; por esta razón se requiere implementar tecnologías para el monitoreo, supervisión y control de datos.

Apoyando el trabajo que se viene realizando en la Escuela de Ingeniería Química de la UCV, y a sabiendas de que el equipo que se utilizaba para controlar y monitorear el proceso no funciona de forma adecuada, la investigación se enfocó en diseñar un sistema de control supervisorio y de adquisición de datos para el bloque de caudal de la planta, empleando cuatro controladores de flujo que midan el caudal, la temperatura y la presión de los gases utilizados en la reacción, los cuales transmiten esta información a través de un protocolo de comunicación hacia un dispositivo de control, que se encarga de procesar y enviar los datos a una interfaz humano máquina.

La realización de este Trabajo Especial de Grado, se diseñó por capítulos, los cuales permiten explicar los fines, desarrollo, propuesta y conclusiones de la investigación, quedando estructurados de la siguiente manera.

En el capítulo I, se muestran el antecedente, el planteamiento del problema presentado y la formulación de los objetivos que rigieron la ejecución del trabajo.

El capítulo II es el marco teórico, donde se reseñan las investigaciones y las bases teóricas que dan soporte a la investigación.

En el capítulo III, se describe la operación de la planta y diagnóstico de la situación presentada como eje de este trabajo.

En el capítulo IV, se presenta el diseño del sistema del control supervisorio y adquisición de datos.

En el capítulo V, se esboza la programación realizada para los elementos que conforman dicho sistema.

En el capítulo VI, se explican las pruebas realizadas para evaluar el desempeño del sistema diseñado.

Finalmente se exponen las conclusiones, recomendaciones y anexos.

CAPÍTULO I

MARCO REFERENCIAL

1.1. ANTECEDENTES

Jorge E., Jaimes L. en [1], implementó los lazos de control de flujo y temperatura del bloque de acondicionamiento de la Planta de Reformado de Metano ubicada en el Laboratorio de Petroquímica de la Escuela de Ingeniería Química. A través de su trabajo se logró ejecutar el control supervisorio y la adquisición de las variables del proceso, esto se realizó mediante la instalación de equipos de instrumentación, que permiten adquirir los datos necesarios para poder procesar la información y monitorizar el comportamiento, tanto del flujo como de la temperatura, en tiempo real, utilizando una interfaz humano máquina amigable que proporciona una excelente visualización y fácil ajuste de las variables. Gracias a este desarrollo la planta mejoró en aspectos como la eficiencia, el control, la modernización, entre otros.

1.2. PLANTEAMIENTO DEL PROBLEMA

En los procesos industriales es de vital importancia manejar de forma precisa el comportamiento de las variables que interfieren en cada etapa del sistema, para lograr esto se utilizan dispositivos e instrumentos para medir las variables físicas involucradas, que envían los datos hacia un equipo central en el que se visualiza la información y se procesa, para realizar acciones de control supervisorio. Estos sistemas son denominados “Supervisory Control and Data Acquisition” (SCADA), y brindan una mayor eficiencia, confiabilidad y precisión al proceso, mostrando en tiempo real el comportamiento y guardando registros de las variables que se requieran, pudiendo realizar acciones de control sobre ellas.

La Planta de Reformado de Metano presenta un problema en el manejo del flujo de los gases que se utilizan en las reacciones. A pesar que se tienen equipos especiales para controlar el flujo, que poseen control integrado y comunicación serial, entre otras características, el ajuste se hace de forma directa en el dispositivo, lo cual puede traer problemas ocasionados por el operador, además no se aprovecha todo el potencial del equipo y de esta forma no se puede visualizar el comportamiento del flujo en tiempo real o no se logra registrar la información en una computadora de forma eficiente.

A través del estudio del proceso y de los instrumentos que se utilizan para medir las variables, se plantea una solución al problema mediante un sistema SCADA, que se encargue de recolectar la información de los controladores y procesar dichos datos para visualizarlos a través de una interfaz humano máquina, en la cual se pueda ajustar de forma sencilla la tasa de flujo de cada gas y sacarle provecho a las demás características del controlador de flujo.

1.3. OBJETIVOS

1.3.1. Objetivo general:

Diseñar un sistema de control supervisorio y adquisición de datos para el caudal del bloque de acondicionamiento de la Planta de Ensayos y Operación de Reformado de Metano.

1.3.2. Objetivos específicos:

Analizar el proceso y operación de la planta.

Realizar el diagnóstico de los bloques de acondicionamiento de la planta.

Diseñar el sistema eléctrico físico para el control del flujo.

Diseñar e implementar los diferentes sistemas de transmisión de datos.

Realizar la programación para controlar los caudales mediante un programa de computación en tiempo real.

Seleccionar un método para el ajuste de los controladores.

Diseñar e implementar una interfaz gráfica que permita al usuario la manipulación vía software de las variables del proceso.

Diseñar e implementar el sistema de alarmas para fallas en los equipos de control y equipos del sistema.

Evaluar el desempeño de los sistemas elaborados.

Realizar los manuales de usuario y los instructivos de código fuente.

CAPÍTULO II

MARCO TEÓRICO

2.1. REFORMADO DE METANO

El contenido de esta unidad fue extraído de trabajos anteriores, realizados en la Escuela de Ingeniería Química, para mayor información consultar la referencia [6].

El metano es un hidrocarburo, que se presenta en forma de un gas incoloro, inodoro e insoluble en agua, se produce a través de la descomposición de materia orgánica y se encuentra en grandes cantidades en los depósitos de gas natural. En la actualidad posee una gran importancia debido al potencial que tiene para producir energía.

El metano es utilizado para la obtención de productos de mayor valor comercial, como son los hidrocarburos livianos, gas síntesis, alcoholes, entre otros. Es tratado de diferentes formas para la obtención de dichos productos, entre las que se destacan, conversión térmica para la obtención de hidrocarburos; conversión directa para la obtención de aldehídos y alcanos; y conversión indirecta para la producción de gas síntesis, y es sobre este último proceso en el que se basa la planta.

El gas síntesis es un combustible gaseoso que se obtiene a partir de sustancias ricas en carbono, está compuesto principalmente por hidrógeno y monóxido de carbono. Este combustible surge como una forma alternativa de generar energía, con la principal característica de producir una combustión menos contaminante que los demás combustibles fósiles y al mismo tiempo es una opción económicamente más factible que otros métodos alternos, como la energía eólica, hidráulica, solar e inclusive nuclear. Específicamente para Venezuela resulta una opción muy atractiva debido a las grandes reservas de gas natural que este país posee.

Debido a que el enlace covalente hidrógeno carbono es de los más fuertes, es necesario establecer condiciones severas de reacción para romper el enlace; a pesar de que el reformado de metano se pueda realizar a temperaturas menores a 430 °C, se suelen utilizar temperaturas alrededor de los 700 °C para producir mejores conversiones y altos rendimientos. Existen diferentes reacciones de reformado de metano, entre las cuales se pueden mencionar reformado de metano con vapor de agua; reformado de metano con dióxido de carbono y reformado de metano con oxígeno.

En el reformado de metano con vapor de agua se lleva a cabo la siguiente reacción:



Es un proceso utilizado a nivel industrial, que utiliza un catalizador de níquel soportado sobre alúmina. Su principal ventaja es que genera una alta relación de hidrógeno-monóxido de carbono, pero por otro lado esta reacción es altamente endotérmica por lo que genera grandes costos energéticos y se pierde la estabilidad térmica del catalizador debido a las altas temperaturas. Este método consume una gran cantidad de vapor de agua.

En el reformado seco o reformado con dióxido de carbono ocurre la siguiente reacción:



Al comparar este método con el reformado de vapor de agua, se observa que disminuye la relación de hidrógeno monóxido de carbono; pero el dióxido de carbono es más económico que el vapor de agua como materia prima, además al utilizar el dióxido de carbono se reducen las emisiones de este gas, el cual contamina el medio ambiente. Para esta reacción se utilizan catalizadores de metales nobles y a base de

níquel, pero en este proceso ocurre la desactivación y destrucción de dichos catalizadores por deposición de carbón vía desproporción del monóxido de carbono y descomposición del metano.

El reformado de metano por oxidación, es decir, utilizando oxígeno se ve representado por las siguientes reacciones:



Para este proceso se utilizan catalizadores de metales nobles. En comparación con los métodos mencionados anteriormente, esta reacción es más conveniente debido a que es altamente exotérmica y esto conlleva bajos requerimientos energéticos, además se puede obtener una relación de hidrógeno monóxido de carbono admisible. Tiene como desventajas una disminución de la selectividad hacia el gas síntesis, debido a la formación de dióxido de carbono y agua; además el oxígeno que se utiliza en la reacción se obtiene del aire por lo que se tiene que separar del nitrógeno en el ambiente, lo cual significa un aumento de los costos y es una reacción altamente riesgosa.

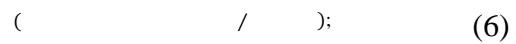
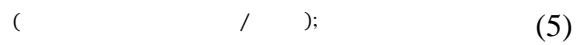
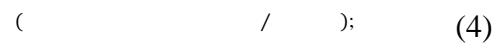
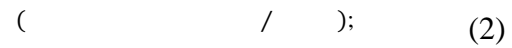
También existen otros métodos de reformado de metano que surgen a partir de la combinación de los procesos explicados. En el reformado autotérmico de metano se tiene como función principal que la mayoría de las reacciones que se producen en los métodos mencionados anteriormente se lleven a cabo; en este proceso se aprovecha la energía producida en las reacciones exotérmicas para las reacciones endotérmicas, haciendo prácticamente nulas las necesidades de calor adicionales. Las reacciones que participan se pueden dividir en principales y colaterales. Las principales son aquellas que a partir del metano, oxígeno y vapor de agua producen los componentes del gas síntesis, el hidrógeno y el monóxido de

carbono. Por otro lado las colaterales proporcionan la energía necesaria para llevar a cabo las demás reacciones.

Reacciones principales:



Reacciones colaterales:



2.2. SCADA

Un SCADA es un sistema completo que incluye “Human Machine Interface” (HMI), y además es capaz de registrar datos; generar alarmas y administrar un sistema de control distribuido a través de una red de hardware [3]. HMI y SCADA están relacionados entre sí a medida que uno o varios HMI son subconjuntos o componentes de un sistema SCADA. Un HMI es simplemente la manera en que los humanos interactúan con las máquinas. Los SCADA involucran los siguientes aspectos [3]:

redes de automatización industrial y máquinas;

telemetría y control remoto utilizando comunicaciones continuas o ráfaga;

sistemas de control de procesos y control de procesos estadísticos;
sistemas de adquisición de datos;
históricos y servidores de almacenamiento de datos;
sistemas de control industrial;
sistemas del entorno empresarial;
entorno de computación de nube industrial y
sistemas de seguridad y procesos.

Los SCADA se conciben principalmente como una herramienta de supervisión y mando. Poseen entre sus características principales: economía, accesibilidad, mantenimiento, ergonomía, gestión, flexibilidad y conectividad. Estos sistemas comprenden toda una serie de funciones con la finalidad de establecer una comunicación entre el operador y el proceso; entre sus prestaciones se tienen: la monitorización, la supervisión, la adquisición de datos de los procesos en observación; el mando, grabación de acciones, garantizan la seguridad de los datos y la seguridad en los accesos. Su arquitectura queda dividida en tres bloques principales tal y como lo muestra la figura 1[4]:

sistema de visualización y control;
sistema de adquisición de datos y
sensores e instrumentos de control.

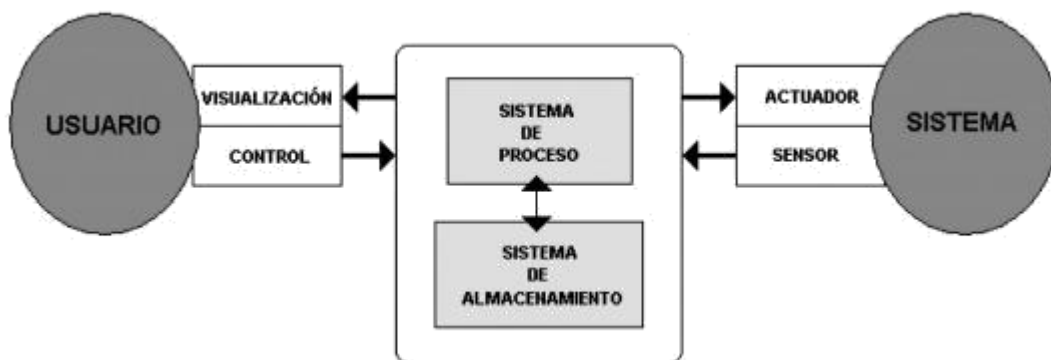


Figura 1: Arquitectura de un sistema SCADA.

En la tabla 1 se muestra la comparación entre los SCADA basados en objetos y puntos hacia una base de datos conocidos como “tags” [5]:

Tabla 1: Comparación entre SCADA basado en objetos y “tags”.

	Arquitectura basada en objetos		Arquitectura basada en “tags”	
	Desarrollo	Tiempo de ejecución	Desarrollo	Tiempo de ejecución
Estructura de la aplicación	Jerárquica – Objetos creados usando una metodología de flujo de trabajo orientado hacia los objetos.	Jerárquica – Los componentes representan dispositivos físicos y pueden coordinarse con componentes en diferentes computadoras.	Jerárquica – Contenido gráfico creado en ocasiones usando orientación hacia objetos.	Plana – Ejemplos monolíticos de software ejecutado en una o múltiples máquinas como “aplicaciones” separadas.
Desarrollo de gráficos	Realizado último.	N/A	Realizado primero.	N/A
“Scripting”	Desarrollado en plantillas de objetos que luego se despliega en una aplicación de tiempo de ejecución específico.	N/A	Desarrollado de manera separada, vinculando a una interfaz gráfica.	N/A
Estándares	Estrictamente cumplidos.	N/A	No son estrictamente cumplidos.	N/A

	Arquitectura basada en objetos		Arquitectura basada en “tags”	
	Desarrollo	Tiempo de ejecución	Desarrollo	Tiempo de ejecución
Cambios de aplicación	Propagados desde plantillas de objetos.	Los objetos pueden distribuirse, intercambiarse o mejorarse.	Basados en gráficos o cambiados usando herramientas como Excel.	Exige la recopilación de la aplicación.
Como se procesan los datos	Los constructos lógicos tales como los dispositivos físicos o dispositivos lógicos se representan como objetos.	N/A	Los dispositivos gráficos se representan como objetos o “tags”.	N/A

2.3. MICROPROCESADORES Y MICROCONTROLADORES

Un microprocesador es un dispositivo que ejecuta funciones equivalentes a los de la unidad de procesamiento de una pequeña computadora. Es capaz de realizar funciones de cómputos básicas, y puede ser incorporado en sistemas donde dichas funciones son requeridas. Un microprocesador por definición significa solo la unidad central de procesamiento, sin incluir memoria o componentes de entrada salida, para conectarse a ambos elementos utiliza buses de información, tal y como se muestra en la figura 2 [7].

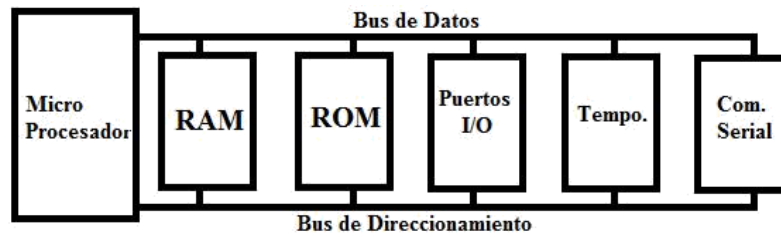


Figura 2: Microprocesador.

Un microcontrolador es un chip de alta gama, el cual incluye tanto el microprocesador como algunos periféricos de entrada salida y la memoria, tal y como se muestra en la figura 3. Estos dispositivos son diseñados para controlar algunos procesos o aspectos de cierto entorno [7].

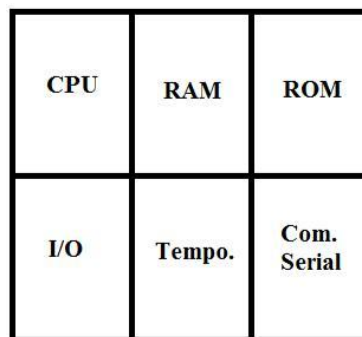


Figura 3. Microcontrolador

La principal diferencia entre los dispositivos mencionados anteriormente, es que los microprocesadores no contienen memoria o puertos de entrada salida, para que un microprocesador funcione es necesario agregarles todos estos componentes de forma externa, a pesar de que esta característica los hace mucho más caros que su contraparte, tienen la ventaja de poder escoger la cantidad de memoria o de puertos que se usarán acorde a la tarea a realizar. Por otra parte los microcontroladores tienen una cantidad fija de memoria y puertos, lo cual los hacen ideales para aplicaciones en los que el espacio y el costo son críticos [8].

2.4. SISTEMAS EMBEBIDOS

Un sistema embebido es un sistema basado en un microprocesador, el cual se programa para realizar una o varias tareas que forman parte de un sistema mayor, y no está diseñado para ser reprogramado por el usuario final. El usuario puede escoger entre las opciones que brinda el equipo, pero no puede alterar la funcionalidad de éste modificando el software [9]. Los sistemas embebidos son similares a los microcontroladores, ya que ambos tienen como núcleo un microprocesador. La diferencia entre estos radica, en que los sistemas embebidos poseen los elementos necesarios para el funcionamiento del microprocesador implementados en una placa base, en vez de integradas en un mismo circuito integrado. En el mercado actual de sistemas embebidos los más conocidos son los modelos producidos por Fundación Raspberry Pi, Texas Instrument y Arduino.

2.5. COMUNICACIÓN SERIAL

El contenido de esta unidad es una síntesis de lo expuesto en el documento [10]

La comunicación serial es un medio de transmisión de datos, por el cual se envía la información un bit a la vez. La mayoría de los puertos seriales son bidireccionales, es decir, pueden enviar y recibir datos al mismo tiempo, el cable entre los dispositivos posee típicamente un canal de comunicación dedicado para cada dirección. Sin embargo en algunos casos solo se tiene un camino compartido para ambas direcciones, en el cual los transmisores toman turnos para enviar la información.

Un protocolo de comunicación es un conjunto de reglas que definen como los dispositivos gestionan la comunicación. Pueden ser asíncronos o síncronos dependiendo del origen del reloj que se utiliza para sincronizar la transmisión. En el asíncrono, el conector no incluye un terminal de reloj, en lugar de eso cada

computadora proporciona un reloj interno como referencia de tiempo. Los equipos deben coincidir en una misma frecuencia de reloj, ver figura 4, un ejemplo de esto es el RS-232. En contraste, en una comunicación síncrona, el conector incluye un terminal para el reloj, que comúnmente es controlado por uno de los equipos y todos los bits transmitidos son sincronizados por dicho reloj, ver figura 5. Un ejemplo de comunicación serial síncrona serían “Inter-Integrated Circuit” (I²C), “Serial Peripheral Interface” (SPI), y Microwire.

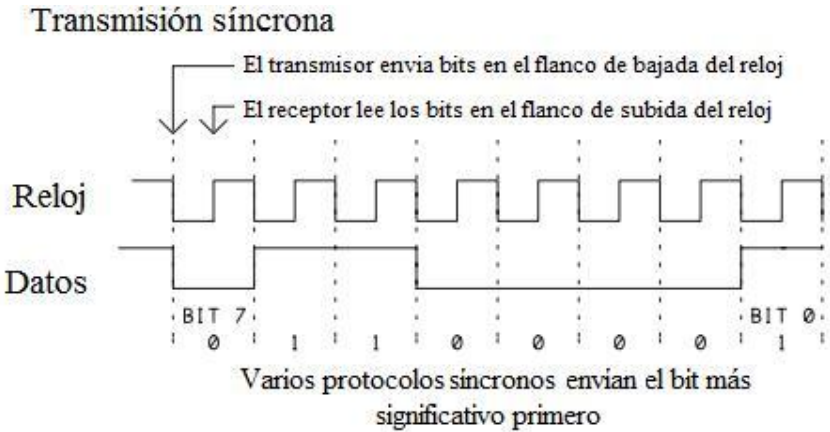


Figura 4. Transmisión síncrona.

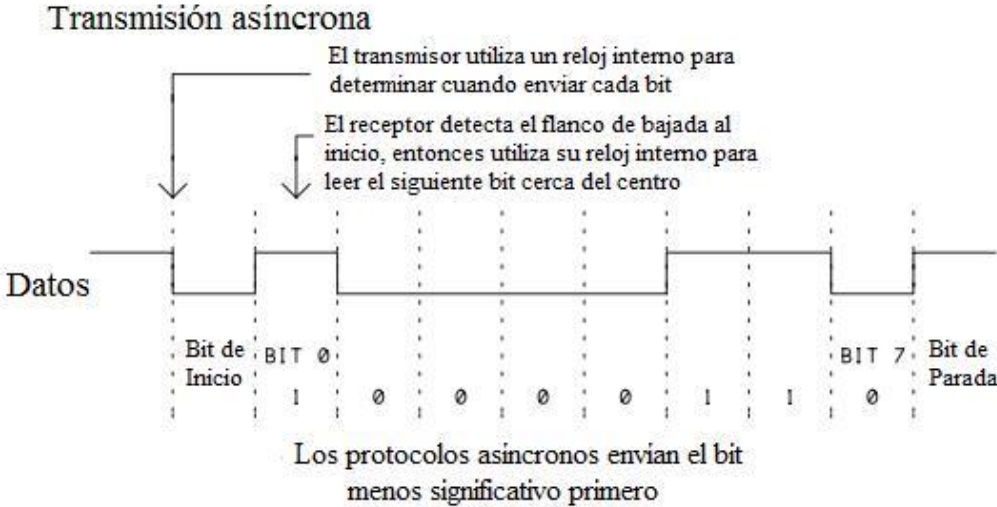


Figura 5. Transmisión asíncrona.

Los equipos con comunicación serial asíncrona normalmente contienen un componente en el hardware llamado “Universal Asynchronous Receiver-Transmitter” (UART). El UART usualmente gestiona los detalles de enviar los bits individuales y almacenar los bits recibidos en el puerto serial. La comunicación serial debe implementar un protocolo de comunicación de bajo nivel y un protocolo de mensaje de alto nivel.

El protocolo de comunicación define como los bits viajan, incluyendo cuando el equipo puede transmitir; la tasa de bits y el orden en que son transmitidos. Además puede incluir el uso de estados y líneas de control, estos canales indican cuando un transmisor tiene datos para enviar o cuando un receptor está disponible para aceptar nuevos datos. El proceso de intercambiar esta información es llamado control de flujo, se puede implementar mediante hardware o software.

En la comunicación serial usualmente se intercambian mensajes que consisten en bloques de datos con formatos definidos. El protocolo de mensaje puede especificar el tipo de datos en un mensaje y como está estructurada la información dentro de éste. Para que una computadora que actúa como receptor detecte el inicio y el final de un mensaje, éste puede incluir códigos que indiquen dichos eventos; una cabecera que almacene la longitud del mensaje o incluso puede tener un bloque de bytes que la computadora receptora utilizará para confirmar errores.

El UART transmite datos en paquetes normalmente llamados palabras, cada palabra contiene un bit de inicio, los bits de datos, un bit opcional de paridad, y uno o más bits de parada. La mayoría de los UART soportan múltiples formatos, un formato común es 8-N-1, que quiere decir 1 bit de inicio, 8 bits de datos, sin paridad y 1 bit de parada.

La tasa de bit es el número de bits por segundo transmitido o recibido por unidad de tiempo, usualmente expresado como bits por segundo. La tasa de baudios

es el número de posibles eventos, o transiciones de datos, por segundo. Popularmente el término tasa de baudios se refiere a la tasa de bit, ya que por lo general son iguales. El número de caracteres transmitido por segundo es igual al número de bits en una palabra. Para el formato 8-N-1, un byte es transmitido a una tasa de bit de 1/10 debido a que cada palabra contiene 10 bits, por lo tanto, a una tasa de baudios de 9600 utilizando dicho formato se pueden transmitir 960 bytes por segundo.

Los bits en la comunicación serial pueden contener cualquier tipo de información, incluidos comandos, lectura de sensores, estados de información, entre otros, pero al final toda la información transmitida son bytes, el software que gestiona la comunicación del puerto serial interpreta los datos como binarios o texto.

2.6. ESTÁNDAR DE COMUNICACIÓN RS-232

Al igual que la unidad anterior, la información presentada en esta unidad es extraída del documento [10].

El término RS-232 se refiere a un estándar de comunicación serial que cumple con la norma *TIA-232-F: Interface between Data Terminal Equipment*, con una edición más actualizada. El ente que se encarga de publicar el estándar es la Asociación de la Industria de las Telecomunicaciones (TIA).

El RS-232 es capaz de realizar varias tareas de comunicación básica entre dos equipos. Usualmente se utilizan dos canales de comunicación, cada señal posee un canal dedicado de voltaje y están referenciados a una tierra común, pero también se puede realizar la conexión con un solo canal, estableciendo una comunicación simplex, donde un solo equipo transmite y otro recibe; o una comunicación semi dúplex, donde ambos equipos pueden transmitir y recibir, pero la comunicación se realiza por turnos, ya que utilizan el mismo canal. En la tabla 2 se muestran los niveles de tensión que se usan comúnmente.

Tabla 2. Niveles de tensión en la comunicación RS-232.

Parámetro	Voltaje [V]
Nivel lógico 0 a la salida	+5 hasta +15
Nivel lógico 1 a la salida	-5 hasta -15
Nivel lógico 0 a la entrada	+3 hasta +15
Nivel lógico 1 a la entrada	-3 hasta -15

Los niveles de tensión son definidos positivos y negativos, como se observa en la tabla 2. El RS-232 usa lógica negativa, es decir, el voltaje más positivo representa un nivel lógico de 0 y mientras más negativo es el voltaje representa un nivel lógico de 1. Las señales de estados y control utilizan los mismos niveles de tensión pero, al contrario que las señales de datos, utilizan lógica positiva.

El estándar define el nombre y la función de cada señal, las características eléctricas, y las especificaciones mecánicas, incluida las disposiciones de los terminales. También indica que el conector utiliza 25 terminales, pero en la actualidad raramente se encuentran computadoras o equipos con el conector de 25 terminales, es más común encontrar que el conector serial utilizado posea nueve terminales. El resto de las señales que el conector de 25 terminales posee se utilizan con módems síncronos, para segundos canales de transmisión y/o seleccionar la velocidad de transmisión en módems duales. Para varias aplicaciones solo se necesitan tres señales (Transmisión, recepción y referencia). En la figura 6 se muestran los conectores macho y hembra de 9 y 25 terminales, conocidos popularmente como DB-9 y DB-25 respectivamente, y en la tabla 3 se muestran los terminales que tienen en común ambos conectores. Para poder comunicarse entre los conectores DB-9 y DB-25 se deben realizar las conexiones que se muestran en la figura 7.

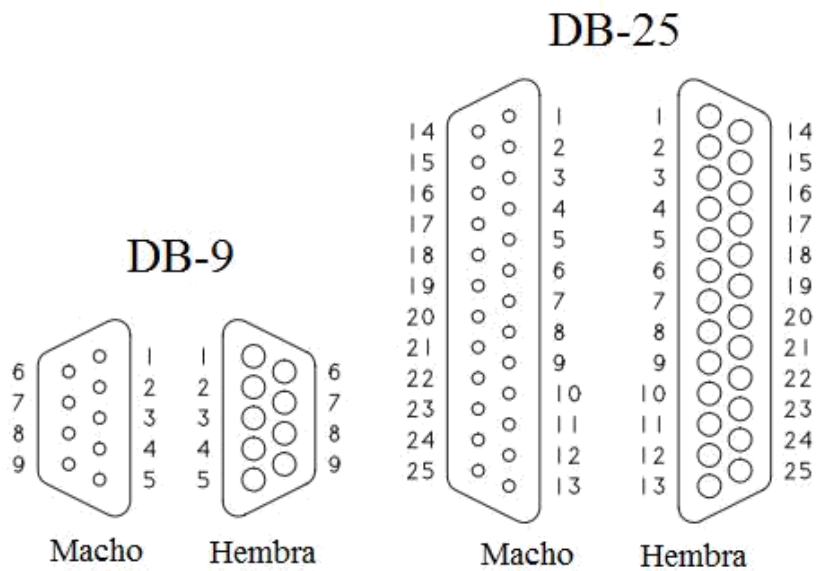


Figura 6. Conectores DB-9 y DB-25 macho y hembra.

Tabla 3. Terminales comunes entre el conector DB-9 y DB-25.

Número de terminal conector DB-9	Número de terminal conector DB-25	Señal	Tipo	Descripción
1	8	CD	Control	Detección de portadora
2	3	RX	Datos	Recepción de datos
3	2	TX	Datos	Transmisión de datos
4	20	DTR	Control	Terminal de datos lista
5	7	SG	-	Señal de tierra
6	6	DSR	Control	Fijación de datos lista
7	4	RTS	Control	Requerimiento de envío
8	5	CTS	Control	Borrar para envío
9	22	RI	Control	Indicador de llamada

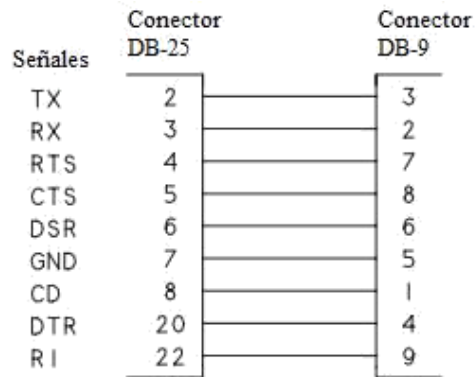


Figura 7. Conexiones entre terminales para convertir de DB-25 a DB-9.

Muchos microcontroladores tienen puertos seriales asíncronos, que utilizan niveles lógicos “Transistor Transistor Logic” (TTL), basados en fuentes de alimentación de 5 V, o niveles lógicos “Complementary Metal Oxide Semiconductor” (CMOS), que utilizan alimentación de 3, 3.3 o 5 V. Para poder establecer una conexión con un puerto serial RS-232 se requiere convertir dichos voltajes a los niveles de tensión con los que trabaja el estándar. En la tabla 4 se muestran los niveles lógicos para diferentes familias de circuitos integrados y niveles de tensión.

Tabla 4. Niveles lógicos aceptados por diferentes familias de circuitos integrados a distintos niveles de tensión

Parámetro	TTL (5 V)	74HC (5 V)	74HC (3 V)	74HC (3.3 V)	74HCT (5 V)
Voltaje máximo para nivel lógico bajo a la salida	0.4	0.1	0.1	0.1	0.1
Voltaje mínimo para nivel lógico alto a la salida	2.4	4.9	2.9	3.2	3.5
Voltaje máximo para nivel lógico bajo a la entrada	0.8	1.0	0.6	1.0	0.8
Voltaje mínimo para nivel lógico alto a la entrada	2.0	3.5	2.1	2.3	2.0

La mejor forma para convertir la lógica de 3 o 5 V a niveles de tensión de RS-232 es utilizar uno de los diferentes circuitos integrados diseñados con este propósito. El MAX232 incluye dos transmisores que convierten entradas de TTL o CMOS a salidas RS-232 y dos receptores que convierten las entradas RS232 a salidas compatibles con TTL o CMOS. En la figura 8 se muestra una configuración básica para utilizar el integrado MAX232.

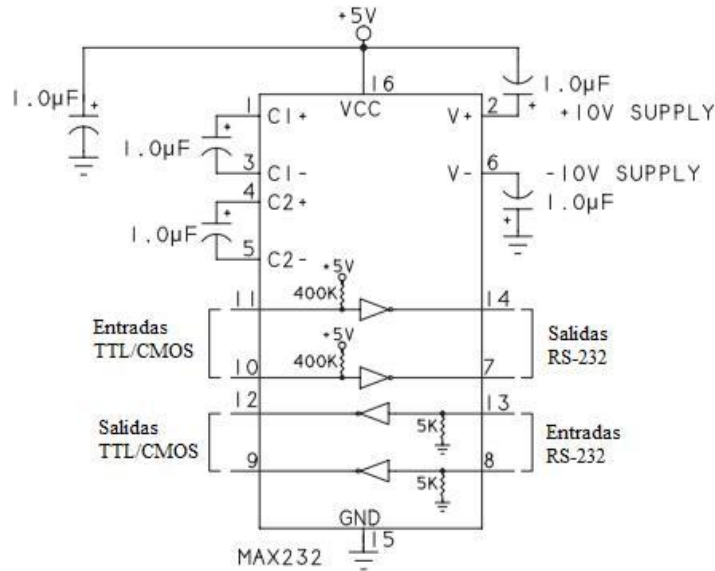


Figura 8. Configuración básica del integrado MAX232.

2.7. INTEL GALILEO

Intel galileo es una placa de desarrollo, basada en el procesador Intel Quark SoC X1000, el cual es un “System on a Chip” (SoC), de la clase Pentium de 32 bits. Es el primer sistema embebido basado en la arquitectura Intel, diseñado para ser compatible en hardware y software con los productos elaborados por Arduino, posee 14 terminales de salidas y/o entradas digitales; seis entradas analógicas; conectores de alimentación; “In Circuit Serial Programming” (ICSP), y un puerto UART, todo dispuesto en la misma posición que en la placa Arduino Uno R3, como se muestra en la figura 9 [11].



Figura 9: Placa de desarrollo Intel Galileo.

La placa Intel Galileo puede ser programada en distintos “Integrated Development Enviroment” (IDE), como lo son [12]:

Arduino IDE;

Intel XDK y

Intel System Studio IoT Edition.

2.7.1. Arduino IDE

Arduino IDE es un entorno de desarrollo integrado libre basado en Processing, contiene un editor de texto para escribir códigos; un área para mensajes; una consola de texto y una barra de herramientas con botones para funciones comunes y varios menús. Se conecta a los productos Arduino y Genuino para subir programas y comunicarse con ellos. Puede ser instalado en cualquier sistema operativo como Windows, Linux o Mac OS X. Posee una extensa cantidad de librerías, que proporcionan funcionalidades extras para facilitar el uso de distintos tipos de hardware o manipulación y procesamiento de datos, además se pueden desarrollar librerías propias para funciones más específicas [13]. En la figura 10 se

muestra la interfaz del software, donde se observan el editor de texto, el área para mensajes, la barra de herramientas, entre otras cosas.

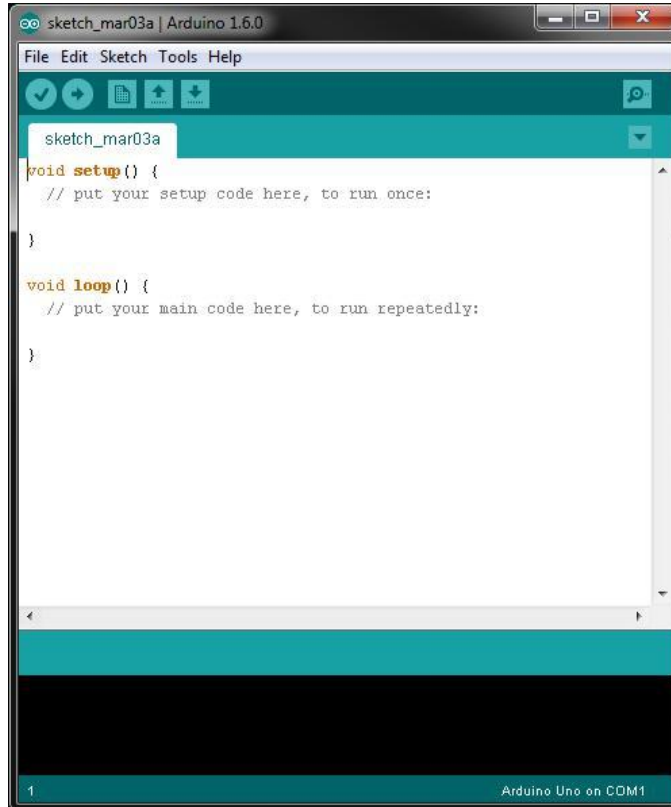


Figura 10. Arduino software IDE.

Para comenzar a cargar programas a la placa Intel Galileo usando el Arduino IDE, primero es necesario remover la tarjeta “Secure Digital” (SD) (si se tiene alguna insertada en el equipo); luego se debe conectar la alimentación y finalmente se conecta el cable micro “Universal Serial Bus” (USB) del Galileo a la computadora. La conexión de la alimentación y el cable micro USB se debe realizar tal y como se especificó anteriormente, en caso contrario puede ocasionar daños al equipo, para desconectar la placa se realiza el procedimiento contrario, primero se desconecta el cable micro USB y por último el cable de alimentación. Siempre y cuando se utilice el conector micro USB se tiene que realizar el proceso descrito para conectar y

desconectar la placa. Se debe de actualizar el firmware del equipo para tener acceso a las últimas características agregadas al IDE [14].

A diferencia del Arduino IDE, para programar la placa Intel Galileo en los IDE Intel XDK o Intel System Studio IoT Edition, es necesario arrancar el equipo utilizando una tarjeta micro SD, que contenga la última versión del Intel IoT Developer Kit, el cual es una imagen de Linux que utiliza la versión Yocto. El equipo trae de fábrica un sistema operativo Linux, pero a pesar de que ya tiene una versión de Linux programada, la versión de Yocto incluye más librerías y recursos, para ayudar a los desarrolladores a crear aplicaciones en diferentes lenguajes de programación, entre los que se pueden mencionar Python, JavaScript, OpenCV, entre otros [15].

2.7.2. Intel XDK

Intel XDK es un IDE que permite crear y correr aplicaciones basadas en JavaScript directamente en plataformas Intel basadas en “Internet of Things” (IoT). Las ediciones más novedosas de este IDE proporcionan plantillas para crear nuevas aplicaciones, ofrece la posibilidad de depurar los proyectos de forma remota y ejecutarlos desde una computadora de escritorio. Utilizar Intel XDK permite crear proyectos más ambiciosos y complejos, aprovechando el poderoso procesador que tienen los equipos para manejar un entorno completo de desarrollo [16]. En la figura 11 se muestra el entorno de programación de Intel XDK.

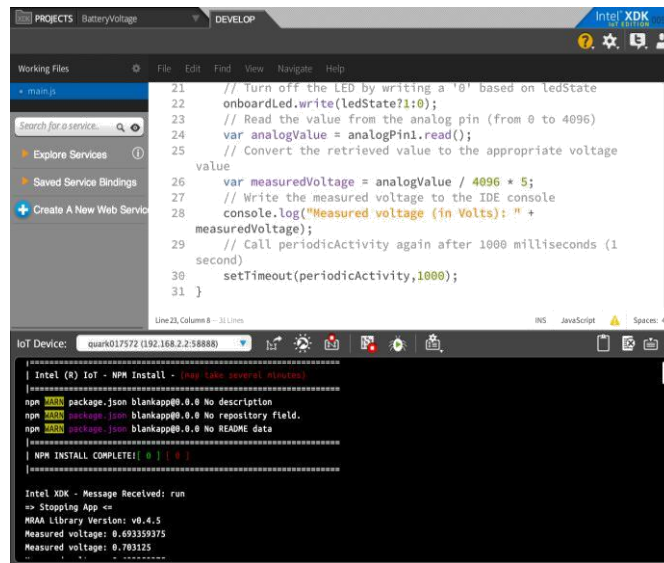


Figura 11. Intel XDK.

2.7.3 Intel System Studio IoT Edition

Este IDE otorga al usuario la capacidad de crear aplicaciones basadas en C o C++. Anteriormente se le conocía como la versión de Eclipse para el Intel IoT Developer Kit, funciona como una extensión de Eclipse que permite al usuario conectar, actualizar y programar proyectos IoT, en una misma placa. En la figura 12 se muestra el entorno de programación [17].

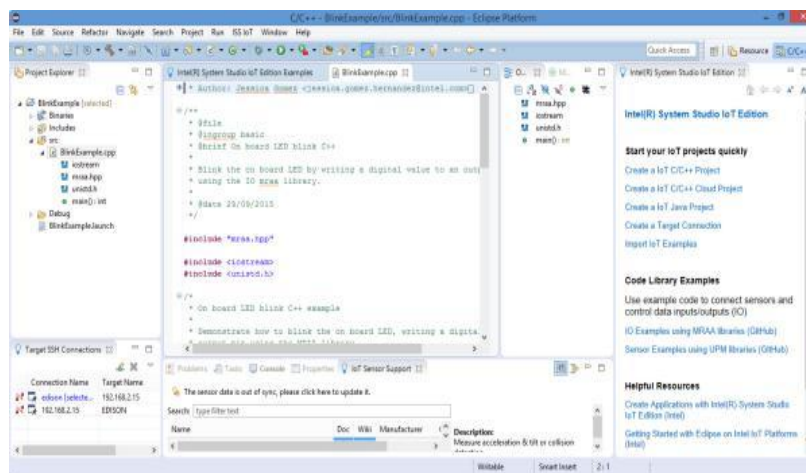


Figura 12. Intel System Studio IoT Edition.

Además de todos los IDE presentados anteriormente, al utilizar la imagen de Linux versión Yocto, se puede manejar el equipo como cualquier sistema basado en Linux, con todas las herramientas que estos ofrecen. Por ejemplo se pueden crear scripts, directamente desde la consola de Linux utilizando lenguajes como Python. Utilizar este método de programación proporciona un mayor control sobre el hardware y las características que ofrece la placa.

2.7.4. Python

Python es un lenguaje de programación de alto nivel diseñado para propósitos generales orientado a objetos. Incorpora módulos, excepciones, programación dinámica, tipos de datos dinámicos de alto nivel y clases. Este lenguaje combina un gran poder con una sintaxis clara; al mismo tiempo que utiliza sistemas de llamadas y librerías que facilitan el desarrollo de aplicaciones; así como varios sistemas de ventanas, adicionalmente es expandible en C o C++. Puede correr en cualquier variante de Unix, Mac y en las versiones de Windows 2000 en adelante.

Este lenguaje viene con una gran variedad de librerías de uso común como el procesamiento de cadenas de datos; protocolos de internet y de comunicación; software dedicados a ingenierías e interfaces de sistemas operativos, así como una gran variedad de extensiones creadas por los usuarios. Debido a que es un software libre Python posee una gran comunidad y documentación de fácil acceso [18].

2.8. MODBUS

La información que se presenta es obtenida de la página web de National Instruments para mayor información dirigirse a [19].

Modbus es un protocolo industrial de comunicación, que trabaja en la capa de aplicación del modelo “Open System Interconnection” (OSI) y está basado en una arquitectura maestro-esclavo o cliente-servidor. En este protocolo el dispositivo que

se encarga de realizar las peticiones se denomina maestro y el dispositivo que se encarga de enviar la información requerida se le llama esclavo. Por lo general los equipos de HMI o SCADA a través de los que se muestra la información son los maestros, mientras que los esclavos pueden ser “Programmable Logic Controller” (PLC), sensores o controladores que intervienen en el proceso y adquieren la información que se desea visualizar. La información que solicita el maestro se almacena en bloques de memorias, que se encuentran dentro del dispositivo esclavo, estos bloques de memoria especifican el tipo de datos y los derechos de acceso y para diferenciar entre ellos el protocolo define un prefijo para cada bloque, en la tabla 5 se muestran las características de los bloques de memoria.

Tabla 5. Bloques de memoria del protocolo Modbus.

Bloque de memoria	Tipo de datos	Acceso de maestro	Acceso de esclavo	Prefijo
Bobinas/Salidas discretas	Booleano	Lectura/Escritura	Lectura/Escritura	0
Entradas Discretas	Booleano	Solo lectura	Lectura/Escritura	1
Registros de entrada	Entero sin signo	Solo lectura	Lectura/Escritura	3
Registros de retención/Registros de salida	Entero sin signo	Lectura/Escritura	Lectura/Escritura	4

Los bloques de datos pueden contener hasta un máximo de 65.536 elementos dependiendo del dispositivo, a cada elemento se le asigna una dirección entre 0 y 65.535. En la tabla 6 se muestra la relación entre el número de referencia del elemento y la dirección en el bloque de memoria.

Tabla 6. Relación entre la dirección de memoria y el número de referencia.

Dirección	Número de Registro	Numero de Referencia
0	1	400.001
1	2	400.002

2	3	400.003
---	---	---------

En las especificaciones del protocolo se recomienda definir las direcciones de memorias con 6 dígitos, para evitar confusiones, ya que por ejemplo, el número de referencia 4.001, podría representar la bobina 4.001 o el registro de salida 1, si se sigue la recomendación la bobina 4.001 se debe expresar como 004.001, mientras que el registro de salida 1 se debe representar como 400.001. El protocolo proporciona un sistema de datos basado en bits y enteros sin signo. En caso de que sea necesario transmitir otro tipo de datos, estos pueden ser divididos en la cantidad de registros necesaria para almacenarlos en la memoria.

Según lo define el modelo de datos, existen códigos definidos para tener acceso a los bloques conceptuales de datos y ejecutar diferentes operaciones. A través de estos códigos el maestro tiene acceso a los diferentes bloques de memoria, aunque también existen códigos que realizan otras funciones. Estos códigos están divididos en tres clases.

Clase 0: son considerados los mínimos requeridos para el uso del protocolo Modbus, ya que permiten a un maestro leer o escribir los bloques de datos.

Clase 1: consisten en el resto de códigos necesarios para tener acceso a todos los tipos de bloque de datos que proporciona el protocolo.

Clase 2: éstos emplean funciones más especializadas que se implementan con menos frecuencia.

En la tabla 7 se muestran los códigos que pertenecen a cada clase y la función que cumplen.

Tabla 7. Códigos de funciones estándares del protocolo Modbus.

Clase	Código	Descripción
0	3	Leer múltiples registros
0	16	Escribir múltiples registros

Clase	Código	Descripción
1	1	Leer bobinas
1	2	Leer entradas discretas
1	4	Leer registros de entrada
1	5	Escribir a bobina individual
1	6	Escribir a registro individual
1	7	Leer Estado de Excepción
2	15	Escribir a múltiples bobinas
2	20	Leer registro de archivo
2	21	Escribir a registro de archivo
2	22	Escribir a registro con máscara
2	23	Leer/Escribir múltiples registros
2	24	Leer FIFO

Modbus puede utilizar 3 protocolos de comunicación para transmitir los datos, estos son Modbus “Transmission Control Protocol” (TCP), Modbus “Remote Terminal Unit” (RTU) y Modbus “American Standard Code for Information Interchange”(ASCII).

2.8.1. Modbus TCP

La transmisión de datos por medio de Modbus TCP consiste en un encabezado como se muestra en la figura 13. El primer campo representa el número de transacción que le otorga el maestro a la petición, el esclavo responde a esta petición con el mismo número. El segundo campo es un identificador de protocolo el cual es normalmente 0, pero se puede utilizar para ampliar el comportamiento del protocolo. El tercer campo se utiliza para definir la longitud del resto del paquete. El cuarto campo se utiliza para determinar la dirección del dispositivo Modbus al cual se

le destina la petición y por último se incluye el mensaje que contiene los códigos de funciones.

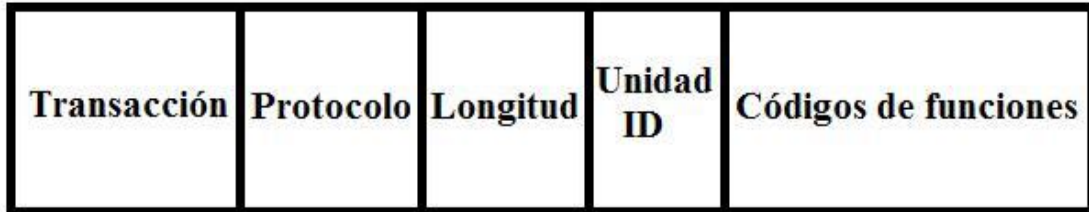


Figura 13. Transmisión de datos Modbus TCP.

2.8.2. Modbus RTU

La transmisión de datos en Modbus RTU es mucho más sencilla que en Modbus TCP, primero se tiene un campo de dirección para definir a cual esclavo se le realiza la petición, normalmente una dirección cero representa un mensaje de difusión, luego se envían los códigos de funciones y por último se tiene un campo de “Cyclic Redundancy Check” (CRC), que se utiliza para asegurar la integridad de los datos. Se debe respetar un intervalo de tiempo antes y después de cada transmisión, como se muestra en la figura 14, el estándar define una cantidad mínima de aproximadamente 2 ms.

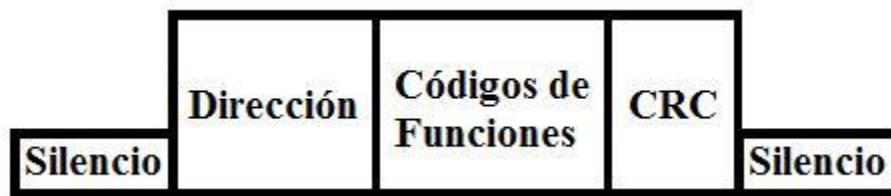


Figura 14. Transmisión de datos Modbus RTU.

2.8.3. Modbus ASCII

En la transmisión de Modbus ASCII cada paquete comienza con la representación en ASCII del símbolo “:”. En el siguiente campo se tiene la dirección

del esclavo seguido de los códigos de funciones y “Longitudinal Redundancy Check” (LRC), también codificados en ASCII y termina con la representación es ASCII de un retorno de carro y una alimentación de línea, tal y como se muestra en la figura 15.



Figura 15. Transmisión de datos Modbus ASCII

2.9. LABVIEW

LabVIEW es un entorno de desarrollo integrado, diseñado específicamente para ingenieros y científicos. Se basa en un lenguaje de programación gráfico, que utiliza un modelo de flujo de datos en lugar de líneas secuenciales de código de texto, lo que permite escribir códigos funcionales utilizando un diseño visual que se asemeja al proceso de pensamiento. Esto significa que se optimiza el tiempo de programación, ya que no se gasta tiempo en la depuración y corrección de la sintaxis del programa [20]. Entre algunas de las ventajas que presenta este software se tienen [21]:

ofrece una perfecta integración en múltiples tipos de instrumentos, buses y sensores;

contiene una extensa colección de controles e indicadores que se pueden utilizar para crear rápidamente una interfaz de usuario personalizada para cualquier aplicación;

ofrece la capacidad de adquirir y analizar datos simultáneamente en un solo entorno;

ofrece acceso fundamental a entradas salidas, puertos IP y programación gráfica de gran alcance, también interactúa con software o bibliotecas de terceros;

versatilidad al usarse con múltiples plataformas y dispositivos;
manejo automático de la asignación de hilos y procesos en paralelo e
incluye funciones que ayudan a registrar datos y crear informes con
facilidad.

LabVIEW es un entorno de programación flexible, que se utiliza para crear aplicaciones únicas de todo tipo. En éste se pueden crear aplicaciones de adquisición y procesamiento de datos; control de instrumentos; automatización y validación de sistemas de pruebas; sistemas embebidos de monitoreo y control; generación de prototipos e incluso es utilizado para ofrecer aprendizaje práctico a nivel universitario [22].

CAPÍTULO III

OPERACIÓN Y DIAGNOSTICO DE LA PLANTA

3.1. DESCRIPCIÓN DE LA PLANTA

La Planta de Reformado de Metano se encuentra en el Laboratorio de Refinación y Petroquímica de la Escuela de Ingeniería Química de la UCV y está formado por [6]:

los gases de reacción, reductor y de arrastre;

mezcladores, para combinar los gases antes de iniciar la reacción;

horno con reactor interno, que está hecho de acero inoxidable para soportar las altas temperaturas que se producen en la reacción;

termocuplas, controlador de temperatura y controladores de flujo, para medir y controlar las variables del proceso;

condensador, utilizado para condensar el vapor de agua y evitar que entre al cromatógrafo;

cromatógrafo de gases, para realizar el análisis de la mezcla antes y después de la reacción y

válvulas y rotámetros, para regular los flujos de los gases.

3.2. PROCESO DE OPERACIÓN DE LA PLANTA

Los elementos que conforman la planta se pueden dividir en bloques según la función que realizan en el proceso. Basados en la información proporcionada por la Escuela de Ingeniería Química a través de trabajos previos realizados por los estudiantes de dicha escuela [6] y [1], se puede decir que estos bloques son:

3.2.1. Bloque de alimentación

Este bloque está formado por un grupo de cilindros que contienen los gases necesarios para realizar el reformado autotérmico de metano, los cuales se dividen en: metano, oxígeno, vapor de agua y dióxido de carbono, para los gases de reacción; hidrogeno, como gas reductor de los catalizadores; y argón, utilizado como gas portador en el cromatógrafo y diluyente en la mezcla de reacción, ver figura 16. Los cilindros de los gases se acoplan al sistema a través de tuberías de cobre de 1/8" de diámetro, al abrir las llaves de los cilindros el gas pasa por un sistema de limpieza el cual elimina las impurezas y la humedad, para finalmente llegar hasta los controladores de flujo.

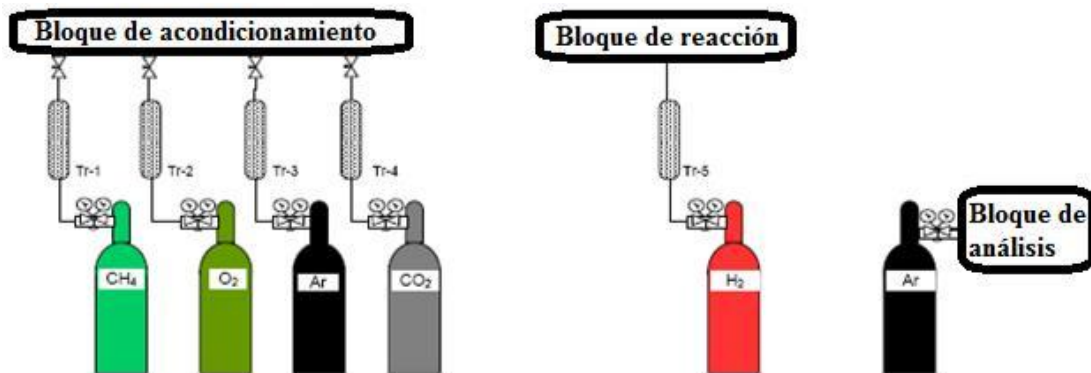


Figura 16: Bloque de alimentación.

3.2.2. Bloque de acondicionamiento

En este bloque se encuentran las válvulas y equipos que se encargan de controlar el paso y la cantidad de flujo de los gases, provenientes del bloque de alimentación, como se puede apreciar en la figura 17. Inicialmente se tienen cuatro controladores de flujo másico para el metano, oxígeno, argón y dióxido de carbono, que se encargan de controlar de forma rápida y precisa el flujo de los gases, cada uno de los controladores tiene acoplado en la entrada un filtro de siete micrones para el filtrado de los gases, asegurando que no entre ninguna impureza al dispositivo y en la

salida poseen válvulas tipo check con el fin de evitar que el flujo circule en sentido contrario. Para controlar el flujo del hidrógeno se emplea un rotámetro con una válvula de diafragma, este gas es utilizado en la reducción del catalizador. Luego del paso de los gases por los controladores de flujo se tiene un sistema de válvulas para permitir el paso al resto del sistema.

Antes de iniciar la reacción es necesario mezclar los gases, para esto se disponen de dos mezcladores de acero inoxidable. Los gases en cada uno de los mezcladores ingresan a estos por la parte superior, para beneficiar el proceso de mezclado haciendo uso de la gravedad. En el primer mezclador se diluyen el dióxido de carbono y el oxígeno en argón, luego de que estos se diluyan exitosamente se agrega el metano en el segundo mezclador. Cada mezclador posee en la salida una válvula check, para evitar que los gases mezclados se devuelvan y así disminuir el tiempo para alcanzar el estado estacionario. Luego de mezclar los gases, se tiene una válvula de tres vías donde se puede escoger entre pasar la mezcla por el saturador o al reactor.

El saturador es un elemento que contiene agua destilada, tiene como función que la mezcla de gases introducida burbujee hasta saturarse con vapor de agua. Este dispositivo está dentro de un baño térmico de glicerina, el cual asegura una fracción de vapor de agua dentro del saturador. En la salida del saturador se encuentra acoplada una tubería de 1/8" de diámetro de acero inoxidable, la cual tiene alrededor una resistencia hecha de un alambre especial que resiste altas temperaturas y está recubierto con cerámica para evitar contacto directo con la tubería. La finalidad de dichas resistencias es evitar que la temperatura de la mezcla baje del punto de ebullición del agua y que el vapor de agua se condense. En este caso se utilizan termocuplas para medir la temperatura de la resistencia y se emplean reóstatos para mantener la temperatura de la resistencia entre los valores aceptables. Luego de todo el proceso realizado en el saturador, el flujo ingresa a una válvula de tres vías, aquí se puede seleccionar entre el reactor o dirigirse hacia otra válvula, en la cual se puede

escoger entre un burbujómetro para medir el flujo o el cromatógrafo para analizar la mezcla antes de la reacción.

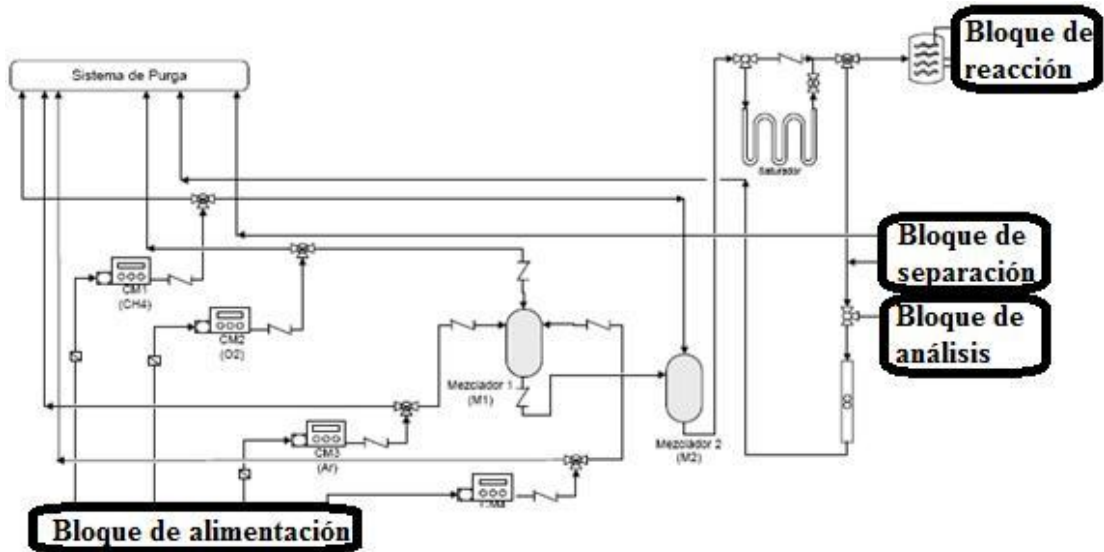


Figura 17: Bloque de acondicionamiento.

3.2.3. Bloque de reacción

Para el bloque de reacción se tiene un horno regulado por un controlador de temperatura multirampa "Proporcional Integral Derivative" (PID), como se muestra en la figura 18. Los gases que salen del bloque de acondicionamiento pasan por un reactor que se encuentra dentro del horno, el cual está hecho de acero inoxidable, con forma tubular y mide 30 cm de largo, está diseñado para soportar altas temperaturas.

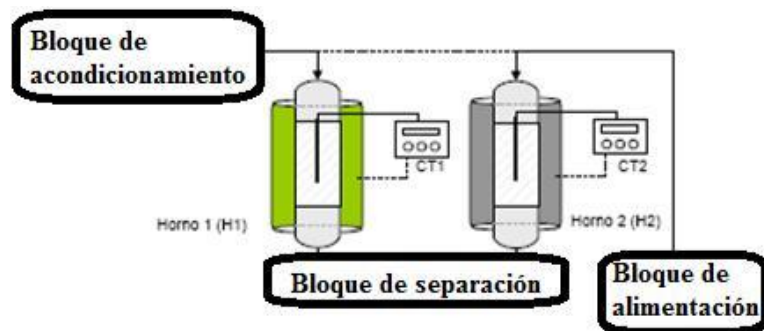


Figura 18: Bloque de reacción

3.2.4. Bloque de separación

El bloque de separación se encuentra a la salida del reactor y está conformado por un condensador con un separador a la salida, como se observa en la figura 19. La función de este bloque es extraer la mayor cantidad posible de agua de la mezcla, para evitar que ingrese al cromatógrafo. El condensador se encarga de enfriar la mezcla, ocasionando que el vapor de agua se condense y se extrae pasando a través del separador. Se utiliza agua como refrigerante en el intercambio de calor, el agua se enfría a través de un baño térmico y es transportada con mangueras de goma de 1/4" de diámetro.

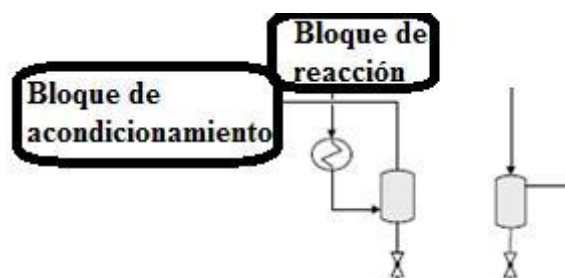


Figura 19: Bloque de separación.

3.2.5. Bloque de análisis

El bloque de análisis está conformado por el cromatógrafo, que se encuentra aislado del calor de irradiación que generan los hornos a través de bloques refractarios. El cromatógrafo tiene a la entrada una válvula de inyección, que se utiliza para dosificar una cantidad exacta de muestra, además posee una columna empacada de Carbosieve SII de 1/4" de diámetro y seis pies de longitud y un "Thermal Conductivity Detector" (TCD). Se utiliza argón de alta pureza como gas de arrastre, y tiene una válvula de inyección automática para realizar un mejor seguimiento de las reacciones. El resultado de los análisis es procesado por una tarjeta de adquisición de datos y se muestra la información en una computadora, ver figura 20.

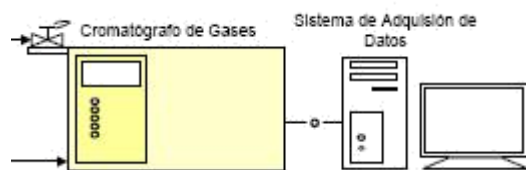


Figura 20: Bloque de análisis.

3.3. DIAGNÓSTICO DEL BLOQUE DE ACONDICIONAMIENTO DE LA PLANTA

Como se mencionó anteriormente, el bloque de acondicionamiento de la planta está conformado principalmente por los controladores de flujo. Estos dispositivos muestran los datos de presión, temperatura, flujo volumétrico, flujo másico, punto de consigna y tipo de gas que controlan, envían dicha información a través de un protocolo de comunicación hacia una computadora, para la visualización de las variables y realizar acciones de control supervisorio del flujo mediante un software dedicado para ello. En esta sección se explicara en detalle las características del controlador, y de los diferentes elementos que se utilizan en el laboratorio para adquirir los datos del caudal y realizar acciones de control supervisorio a través de la computadora.

3.3.1. Controlador de flujo

La medición y control del flujo se realiza a través de controladores de flujo másico, de la marca Cole-Parmer serie 16 modelo MC-1SLPM-D. La información presentada a continuación es extraída del manual de usuario de los controladores de flujo el cual viene con el equipo [23].

3.3.1.1 Principio de funcionamiento

El funcionamiento de los medidores de flujo de gas másico de la serie 16 se basa en la medición precisa del flujo volumétrico. La tasa de flujo volumétrico es determinada creando una caída de presión a través de una restricción interna única (conocida como un elemento de flujo laminar), y midiendo la presión diferencial a través de ella. Esta restricción es diseñada con el fin de que las moléculas de gas sean obligadas a moverse en trayectorias paralelas a lo largo de toda la longitud del paso, por lo tanto, se establece el flujo laminar para todo el rango de operación del dispositivo. A diferencia de otros dispositivos de medición de flujo, en los medidores de flujo laminar la relación entre la caída de presión y el flujo es lineal. El principio fundamental de operación de los medidores de flujo es conocido como la ecuación de Poiseuille (8):

$$Q = \frac{\pi (P_1 - P_2) r^4}{8 \eta L} \quad (8)$$

donde: Q = Tasa de flujo volumétrico;

P_1 = Presión estática en la entrada;

P_2 = Presión estática en la salida;

r = Radio de la restricción;

η = Viscosidad absoluta del fluido y

L = Longitud de la restricción.

Debido a que π , r y L son constantes la ecuación (8) se puede reescribir como:

$$Q = K \frac{\Delta P}{\eta} \quad (9)$$

donde K es un factor constante determinado por la geometría de la restricción. Esta última ecuación muestra la relación lineal entre la tasa de flujo volumétrico (Q), el diferencial de presión (ΔP) y la viscosidad absoluta (η) de una forma simple.

En orden de obtener una tasa de flujo volumétrico precisa, el gas que se está midiendo debe ser seleccionado en el modo de selección de gas del controlador de flujo. Esto es importante debido a que el dispositivo calcula la tasa de flujo basado en la viscosidad del gas, a determinada temperatura. Si el gas siendo medido no es el mismo que se encuentra seleccionado, un valor incorrecto de viscosidad de gas será utilizado en el cálculo del flujo, ocasionando un resultado erróneo en proporción a la diferencia de viscosidad de los dos gases.

La viscosidad del gas, y por lo tanto, la composición del gas, son muy importantes para la precisión de la medición. Cualquier cosa que tenga efecto en la viscosidad del gas, como vapor de agua o aditivos odorantes, afectarán de forma directa la precisión del instrumento. Por ejemplo, si se selecciona metano como gas, y se está midiendo realmente gas natural, la medición resultante será bastante precisa, debido al gran porcentaje de metano que posee el gas natural, pero no será altamente precisa, debido a las pequeñas cantidades de otros gases, como butano y propano, contenidas en el gas natural, los cuales varían la viscosidad a un valor diferente de metano puro.

La viscosidad absoluta varia en pequeñas cantidades con los valores de presión para los cuales están diseñados los medidores de flujo, por lo tanto la lectura de flujo volumétrico no necesita una corrección debido al efecto de la presión. No obstante, cambios en la temperatura del gas si ocasionan variaciones de viscosidad apreciables, por lo tanto, los controladores de flujo poseen un método de compensación interno.

A temperatura ambiente y baja presión el flujo volumétrico y el flujo másico son casi idénticos, sin embargo, esto puede variar con cambios en la temperatura y/o presión debido a que afectan directamente el volumen del gas. Con el fin de poder convertir el volumen a masa, la densidad del gas debe ser conocida. La relación entre el volumen y la masa viene dada por:

(10)

La densidad del gas cambia con la temperatura y la presión, por lo tanto, es necesario saber el cambio en la densidad, para poder convertir de flujo volumétrico a flujo másico. Utilizando la ley de gases ideales, el efecto de la temperatura en la densidad se puede describir como:

$$\rho = \rho_0 \frac{T_0}{T} \quad (11)$$

donde:

- = Densidad en condición de flujo;
- = Temperatura absoluta en condición de flujo;
- = Densidad en condición de referencia y
- = Temperatura absoluta en condición de referencia.

Los cambios en la densidad debido a la presión se pueden representar como:

$$\rho = \rho_0 \frac{P}{P_0} \quad (11)$$

donde:

- = Densidad en condición de flujo;
- = Presión absoluta en condición de flujo;
- = Densidad en condición de referencia y
- = Presión absoluta en condición de referencia.

Por lo tanto para determinar el flujo másico, estos dos factores de corrección deben ser aplicados al flujo volumétrico.

3.3.1.2. Alimentación y conexiones de señales

La alimentación de los controladores de flujo puede ser proporcionada a través de una fuente DC, de 12 a 30 V y sea capaz de entregar una corriente de 250 mA, la fuente debe tener un adaptador jack hembra de 2.1 mm con centro positivo. Otra alternativa para alimentar el dispositivo es utilizar el conector mini-DIN de 8

terminales que posee el controlador, de igual forma se tienen que garantizar las características eléctricas anteriormente mencionadas. En la figura 21 se muestra los conectores jack y mini-DIN.

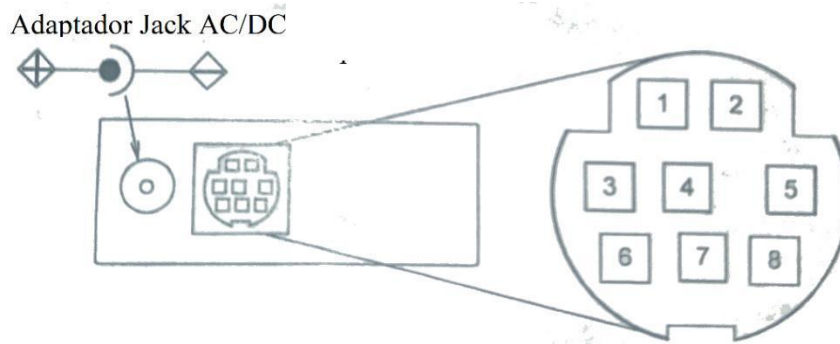


Figura 21: Conectores jack y mini-DIN de 8 terminales.

En la tabla 8 se muestran las funciones que cumplen los terminales del conector mini-DIN.

Tabla 8: Conector mini-DIN de 8 terminales.

Terminal	Función	Color del cable
1	Inactivo o señal de salida primaria 4–20 mA	Negro
2	5.12 Vdc constante o salida analógica secundaria (4-20 mA, 5Vdc, 10 Vdc) o alarma básica.	Marrón
3	Señal de entrada RS-232	Rojo
4	Señal de entrada analógica	Naranja
5	Señal de salida RS-232	Amarillo
6	Señal de salida 0-5Vdc (o 0-10 Vdc)	Verde
7	Alimentación (Según se describió anteriormente)	Azul
8	Tierra (Común para la alimentación, la comunicación y las señales)	Morado

La disponibilidad de diferentes señales de salida depende de las opciones especificadas al momento de realizar la orden de compra del dispositivo. No se debe

conectar la alimentación a ninguno de los terminales del uno (1) al seis (6), ya que puede ocasionar daños permanentes al instrumento. El terminal dos (2) es utilizado como un voltaje constante de 5.12 Vdc, que refleja el voltaje del bus del sistema y puede ser utilizado para transmitir el punto de consigna de forma analógica.

Al menos que se especifique lo contrario al momento de comprar el equipo, la señal de entrada analógica predeterminada es de 0-5 Vdc. Se puede utilizar aplicando una señal de entrada entre 0-5 Vdc, utilizando el terminal ocho (8) de tierra como común, o también se puede utilizar un potenciómetro de 50 k Ω , conectando a uno de sus extremos el terminal dos (2), que genera un voltaje constante de 5.12 Vdc, y al otro el terminal ocho (8) de tierra, el terminal cuatro (4) se conecta a la salida del potenciómetro para obtener una señal de entrada analógica ajustable, tal y como se muestra en la figura 22.

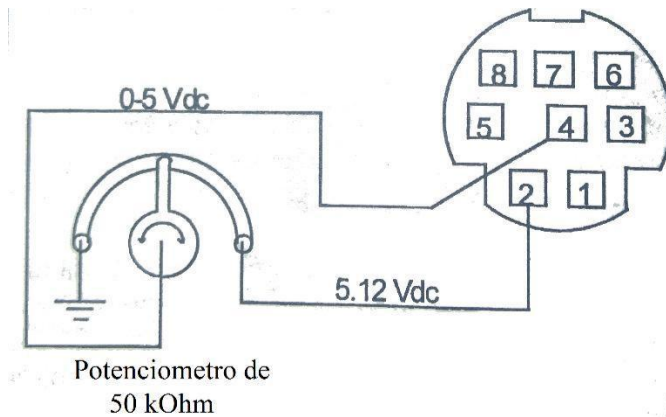


Figura 22: Método analógico para transmitir el punto de consigna a los controladores.

Los controladores de flujo tienen disponible una señal de salida analógica de 0-5 Vdc en el terminal seis (6). Este voltaje se encuentra usualmente en el rango de 0.010 Vdc para flujo nulo y 5.0 Vdc para flujo a plena escala, la tensión es lineal sobre todo el rango de medición y el terminal ocho (8) de tierra es el común para esta señal. Si se especifica al momento de comprar el equipo, esta señal de salida puede

variar a una señal de 0-10 Vdc o de 4-20 mA. De igual forma, si se especifica al momento de comprar se puede hacer uso de una segunda señal de salida analógica en el terminal dos (2). Esta salida puede ser de 0-5 Vdc, 0-10 Vdc o 4-20 mA, la cual puede representar cualquier parámetro medido por el instrumento.

Para la comunicación serial RS-232 se utiliza el terminal tres (3) como señal de entrada, el terminal cinco (5) como señal de salida y el terminal ocho (8) como referencia, los cuales se deben conectar al puerto serial de la computadora o del dispositivo que se encargue de recibir los datos. Es necesario adaptar los terminales de salidas del conector mini-DIN a los de un conector serial DB-9, como se muestra en la figura 23, para realizar esto, el vendedor proporciona un cable adaptador, que posee en uno de sus extremos el conector macho mini-DIN de 8 terminales y en el otro extremo el conector serial DB-9 hembra, representado en la figura 24.

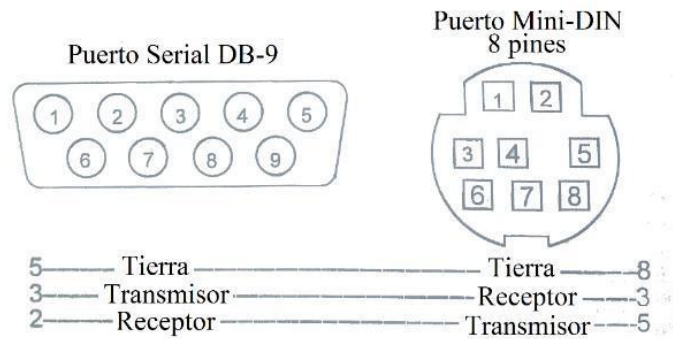


Figura 23: Conexión mini-DIN de 8 terminales a conector serial DB-9

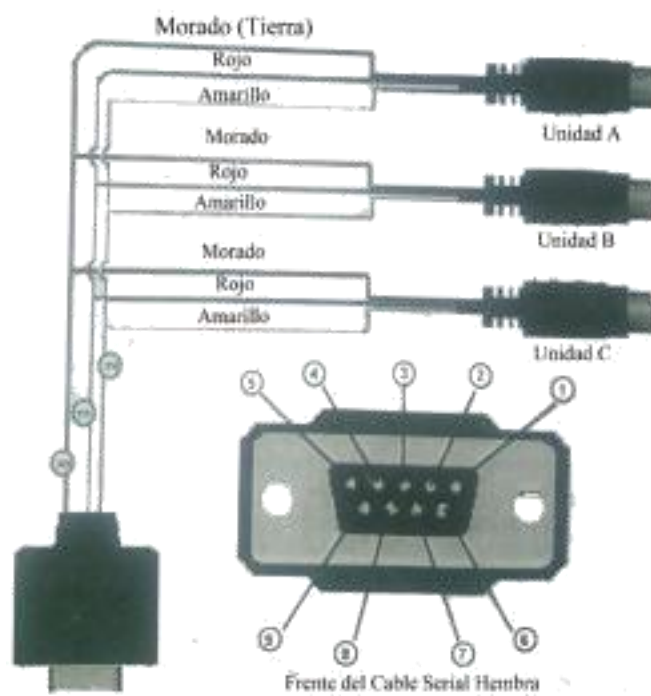


Figura 24: Cable adaptador conector mini-DIN de 8 terminales a conector serial DB-9.

3.3.1.3. Funciones del controlador de flujo

Los controladores de flujo proporcionan en un dispositivo simple y robusto varios datos de interés relacionados con el flujo de los gases. Poseen una “Liquid Crystal Display” (LCD) en la cual se pueden mostrar diferentes pantallas con funciones específicas.

3.3.1.3.1. Pantalla de inicio

Esta pantalla es la primera que se observa al encender el controlador, en la figura 25 se observan los parámetros que se muestran en este modo.

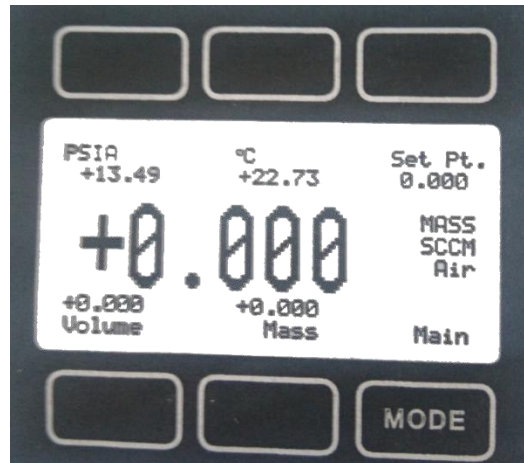


Figura 25: Pantalla de inicio.

Punto de consigna: en la esquina superior derecha de la pantalla, bajo la etiqueta de "Set Pt.", se muestra el valor del punto de consigna. Este parámetro no se puede ajustar a través de este modo, para realizar cambios en él se tiene que dirigir a la pantalla de configuración de control.

Presión: se utiliza un sensor de presión absoluta para medir la línea de presión del flujo de gas que está siendo monitorizado. Este sensor toma como referencia la presión de vacío y lee con precisión la línea de presión, tanto por arriba como por debajo de la presión atmosférica local. Este parámetro se localiza en la esquina superior izquierda de la pantalla, con la etiqueta de "PSIA" y puede ser relocalizado en el centro de la pantalla al presionar el botón que se encuentra encima de la etiqueta.

Temperatura: se utiliza un sensor para medir la temperatura del flujo de gas que está siendo monitorizado. La temperatura se muestra en grados Celsius. Este parámetro se encuentra centrado en la parte superior de la pantalla, bajo la etiqueta de "°C" y se puede relocalizar en el centro de la LCD al apretar el botón respectivo.

Flujo volumétrico: este parámetro se determina mediante el principio de operación descrito anteriormente, se localiza en la esquina inferior izquierda, bajo la etiqueta de "Volume" y puede ser relocalizado al centro, de la misma forma utilizada con los parámetros anteriores.

Flujo másico: este parámetro se obtiene al realizar métodos de corrección al flujo volumétrico, a temperatura ambiente y presión atmosférica. Se encuentra centrado en la parte inferior de la pantalla, bajo la etiqueta de "Mass" y se puede relocalizar con el método mencionado anteriormente.

3.3.1.3.2. Pantalla de selección de menú

Al presionar el botón de “MODE” aparecerá la pantalla que se muestra en la figura 26. En esta pantalla se muestran los diferentes menús con los que cuenta el dispositivo, para seleccionar uno de ellos solo se necesita presionar el botón correspondiente a la opción deseada.

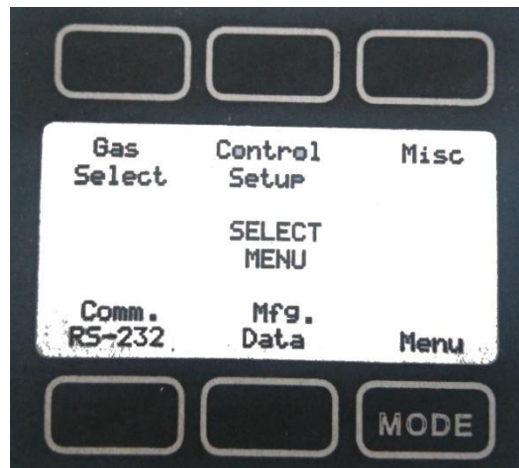


Figura 26: Pantalla de selección de menú.

3.3.1.3.3. Pantalla de configuración de control

Para acceder a esta pantalla se presiona el botón encima de la opción “Control Setup”, en la pantalla de selección de menú. Esta pantalla, que se muestra en la figura 27, permite al usuario configurar la mayoría de los parámetros asociado con el control “Proportional Derivative” (PD). Los controladores de flujo Cole-Parmer, otorgan al usuario la capacidad de escoger como será transmitido hacia el dispositivo el punto de consigna y como se controlará el mismo, además de ajustar los valores de las constantes proporcional y derivativa del lazo de control PD.

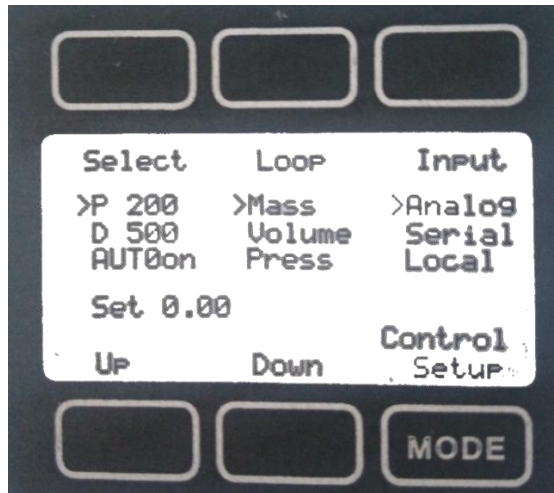


Figura 27: Pantalla de configuración de Control.

“Input”: sirve para cambiar como el punto de consigna será transmitido hacia el controlador, para seleccionar un tipo de transmisión se presiona el botón encima de la etiqueta “Input” hasta que la flecha indicadora este frente a la opción deseada.

“Analog”: este modo se refiere al uso de una señal analógica aplicada en el terminal cuatro del conector mini-DIN. Se puede utilizar un potenciómetro de 50 k Ω para realizar un control manual del punto de consigna, como se mencionó en la sección alimentación y conexiones de señales.

Serial: este modo se refiere al uso del estándar RS-232 para transmitir el punto de consigna al controlador, a través de un computador, PLC u otro dispositivo.

Local: este modo se refiere a seleccionar directamente en el controlador el punto de consigna que se desea, esto se puede realizar a través de la pantalla de configuración de control en la etiqueta de “Select”.

“Loop”: en esta opción se puede seleccionar la variable con la que se desea cerrar el lazo de control. La unidad puede ser configurada para controlar a la salida la presión del gas, la tasa de flujo volumétrico o la tasa de flujo másico. Al presionar el botón encima de la etiqueta “Loop” se selecciona que variable se utilizará.

“Select”: para evitar cambios accidentales de los parámetros del lazo de control PD o del punto de consigna, la posición del seleccionador se coloca de forma predeterminada en una posición nula. Para poder realizar cambios a

las variables primero se debe presionar una vez el botón correspondiente a la etiqueta “Select”.

P: se refiere a la constante proporcional del lazo PD. Antes de cambiar este parámetro se recomienda tomar nota del valor inicial para poder regresar a los valores de fábrica en caso de ser necesario.

D: se refiere a la constante derivativa del lazo PD. Igualmente que la constante P se recomienda tomar nota del valor inicial en caso que sea necesario restaurar los valores de fábrica.

AUTO: si este modo se encuentra encendido, el dispositivo realiza automáticamente un ajuste de cero, cuando recibe un punto de consigna de cero durante más de dos segundos. Un punto de consigna nulo ocasiona el cerrado de la válvula. Esta característica ayuda a la precisión del dispositivo, al remover periódicamente errores acumulativos asociados al flujo residual.

“Set”: se refiere al punto de consigna. Este parámetro solo se puede ajustar a través de este método si se escoge la opción “Local” en la etiqueta de “Input”.

3.3.1.3.4. Pantalla de selección de gas

Para acceder al menú de selección de gas se presiona el botón encima de la etiqueta “Gas Select” en la pantalla de selección de menú.

El gas seleccionado se muestra en la pantalla de inicio, y es indicado por la flecha en la pantalla de selección de gas, ver figura 28. Para cambiar el gas seleccionado, se utilizan los botones bajo las etiquetas “UP” y “DOWN” o encima de “PgUP” y “PgDWN” para posicionar la flecha en frente del gas deseado. Cuando se regresa a la pantalla de inicio, el gas seleccionado será mostrado en la pantalla.

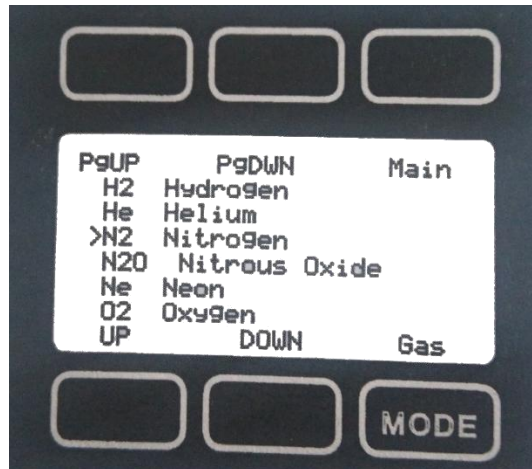


Figura 28: Pantalla de selección de gas.

3.3.1.3.5. Pantalla de selección de comunicación

Para acceder a este menú se presiona el botón debajo de la etiqueta “Comm. RS-232” en la pantalla de selección de menú, ver figura 29.

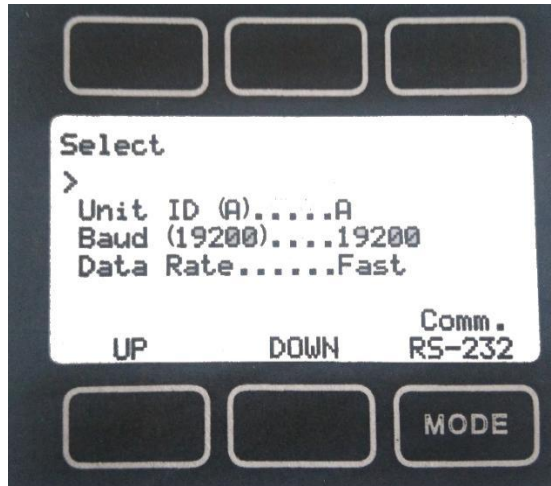


Figura 29: Pantalla de selección de comunicación.

“Unit ID”: los valores aceptados son las letras de la A hasta la Z y el símbolo de @. Este parámetro permite al usuario asignar una dirección única a cada controlador, para que múltiples unidades puedan ser conectadas a un mismo puerto RS-232, utilizando el direccionamiento para descartar la información que lee cada dispositivo. Para cambiar la dirección se presiona

el botón correspondiente a la etiqueta “Select”, al posicionar la flecha en la opción “Unit ID” se presionan los botones correspondientes a las etiquetas “UP” y “DOWN” para escoger la dirección deseada.

“**Baud**”: en esta opción se selecciona la tasa de baudios a la que trabajará la comunicación serial RS-232.

“**Data Rate**”: cambiando este parámetro afecta la tasa a la cual el instrumento envía la información, tiene dos opciones “Slow” y “Fast”, donde la última es el doble de rápida que la primera. La velocidad de la opción “Fast” es determinada según la tasa de baudios seleccionada.

3.3.1.3.6. Pantalla de información del fabricante

Para acceder a esta pantalla se presiona el botón correspondiente a “Mfg. Data” en la pantalla de selección de modo. En la figura 30 se muestra la pantalla “Mfg 1”, la cual muestra el nombre y el número de teléfono del fabricante, y en la figura 31 se muestra la pantalla “Mfg 2”, ésta muestra información importante acerca del controlador, como el modelo, el número de serial y la fecha de fabricación.



Figura 30: Pantalla de información del fabricante “Mfg 1”.

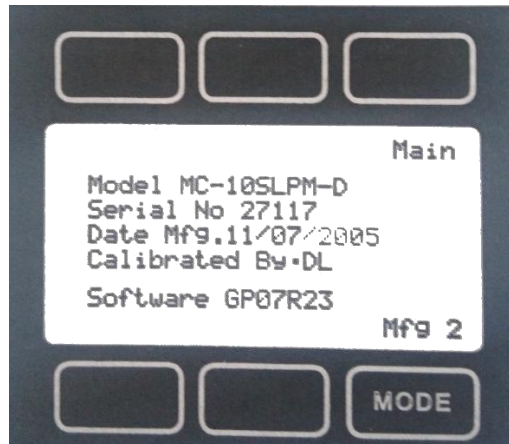


Figura 31: Pantalla de información del fabricante “Mfg 2”.

3.3.1.3.7. Pantalla de misceláneos

Para acceder a esta pantalla se presiona el botón que se encuentra sobre la etiqueta “Misc”, en ella se muestran opciones como el contraste de la LCD, la zona muerta de cero, el flujo promedio y la presión promedio, como se muestra en la figura 32. Para cambiar los valores de cada ítem se presiona el botón correspondiente a la etiqueta “Select” hasta que la flecha se encuentra sobre la opción deseada, luego se presionan los botones “UP” o “DOWN” para hacer el ajuste.



Figura 32: Pantalla de misceláneos.

3.3.1.4. Comunicación RS-232

Para establecer una comunicación exitosa, los parámetros de la comunicación serial deben ser los mismos tanto para el dispositivo que envía como para el que recibe. En este caso el controlador está limitado por los valores de fábrica, los cuales no se pueden alterar. Exceptuando la tasa de baudios, que posee un rango de valores seleccionables mediante la pantalla de configuración de la comunicación. Para los demás parámetros se tiene que asegurar que el otro dispositivo este configurado para trabajar a ocho bits de datos, sin paridad, con un bit de parada y sin control de flujo.

El controlador de flujo posee dos modos para transmitir información a través del estándar de comunicación RS-232. Para escoger el modo de transmisión se utiliza el direccionamiento, proporcionado por la opción de “Unit ID” a través de la pantalla de configuración de la comunicación. Los valores admitidos para el direccionamiento pueden ser las letras de la A hasta la Z o el símbolo de @.

Si se selecciona una letra se usará el primer método de comunicación llamado “Polling Mode” o modo de petición, en el cual la unidad solo envía información cuando recibe una petición, para este caso cada unidad debe tener una dirección diferente. Por otro lado si se selecciona el símbolo de @ se utilizará el segundo método de comunicación llamado “Streaming Mode” o modo difusión, el cual envía los datos constantemente sin necesidad de una petición, a una tasa aproximada de 10 a 60 veces por segundo, dependiendo de la cantidad de datos que se transmitan. Si se utiliza un solo canal de comunicación para intercambiar datos con más de un controlador se debe de utilizar el modo de petición.

Ambos modos de transmisión envían la misma cadena de datos, la cual tiene el siguiente formato:

“+014.70 +025.00 +02.004 +02.004 2.001 Air”,

en dónde cada conjunto de caracteres, separados por un espacio en blanco, representan los parámetros del flujo del gas. En orden de izquierda a derecha estos parámetros son: la presión, la temperatura, el flujo volumétrico, el flujo másico, el punto de consigna y el gas seleccionado.

A través de la comunicación serial se pueden realizar cambios a ciertos valores y opciones que presenta el controlador. Para ajustar el punto de consigna a través de la comunicación serial, primero se debe seleccionar la opción “Serial” en la pantalla de configuración de control, luego se envía un valor entre 0 y 64000, los cuales representan flujo nulo y flujo a plena escala respectivamente, y por último se envía un retorno de carro. Cabe mencionar que en el modo de petición, es necesario colocar la letra correspondiente al direccionamiento antes del valor que representa el punto de consigna.

Para seleccionar el gas utilizando la comunicación serial se deben enviar dos símbolos de \$, seguidos de un número entre 0 y 29 y finalmente un retorno de carro, al igual que en el método anterior, si se utiliza el modo de petición la letra de direccionamiento debe preceder los símbolos \$. El valor del número entre 0 y 29 simboliza las 30 posibilidades de gas que posee el controlador, cada valor representa un gas en específico. En la tabla 9 se muestra una lista de los gases con sus respectivos valores, el nombre y la abreviación con la que se identifica en la LCD del controlador.

Tabla 9: Selección de gas.

#	Gas	Abreviación
0	Aire	Air
1	Argón	Ar
2	Metano	CH4

#	Gas	Abreviación
3	Monóxido de Carbono	CO
4	Dióxido de Carbono	CO ₂
5	Etanol	C ₂ H ₆
6	Hidrógeno	H ₂
7	Helio	He
8	Nitrógeno	N ₂
9	Óxido Nitroso	N ₂ O
10	Neón	Ne
11	Oxígeno	O ₂
12	Propano	C ₃ H ₈
13	Butano normal	n-C ₄ H ₁₀
14	Acetileno	C ₂ H ₂
15	Etileno	C ₂ H ₄
16	Iso-Butano	i-C ₄ H ₁₀
17	Criptón	Kr
18	Xenón	Xe
19	Hexafluoruro de Azufre	Sf ₆
20	75% Argón / 25% CO ₂	C-25
21	90% Argón / 10% CO ₂	C-10
22	92% Argón / 8% CO ₂	C-8
23	98% Argón / 2% CO ₂	C-2
24	75% CO ₂ / 25% Argón	C-75
25	75% Argón / 25% Helio	A-75
26	75% Helio / 25% Argón	A-25

#	Gas	Abreviación
27	90% Helio / 7.5% Argón / 2.5 % CO2	A1025
28	90% Argón / 8% CO2 / 2% Oxígeno	Star29
29	95% Argón / 5% Metano	P-5

También se puede cambiar el modo de transmisión a través de la comunicación serial; para cambiar de modo difusión a modo petición se manda *@= seguido de la letra que se desea establecer como dirección y luego un retorno de carro. Para el caso contrario se debe enviar *@=@ y luego un retorno de carro, es preciso tener cuidado al usar esta característica, ya que si se tiene varios controladores conectados a la misma línea todos serán afectado por igual, para evitar problemas de direccionamiento se tiene que dejar un controlador acoplado a la línea y desconectar los demás.

Por último se pueden ajustar los valores de las constantes “P” y “D”, para ello se disponen de dos registros, 21 y 22 respectivamente. Al mandar R21 o R22 seguido de un retorno de carro el controlador devolverá el valor actual del parámetro solicitado, dicho valor será un numero entre 0 y 64000, y para cambiarlo se envía W21= o W22=, según sea el caso, seguido del valor entre 0 y 64000 que se desee y por último un retorno de carro. Se debe tener cuidado con este método y anotar los valores iniciales de dichas constantes, para poder regresar a los valores de fábrica de ser necesario.

3.3.2. Circuito impreso multiplexor RS-232

A pesar de que los controladores de flujo poseen direccionamiento en la comunicación con el propósito de conectar varios controladores en paralelo directamente hacia una computadora, este método genera colisiones de datos y la información se transmite de forma errónea. Para solucionar este problema, mediante

estudios previos en el laboratorio, se elaboró un circuito impreso que cumple la función de multiplexar los canales de transmisión de los controladores. Se utilizaron para el desarrollo de la placa cuatro relés mecánicos de 24 V y 4 “Light-Emitting Diode” (LED) con sus respectivas resistencias de 1 k Ω , para indicar cuál es el canal que se encuentra conectado. En las figuras 33 y 34 se muestran el diseño esquemático y físico de la placa respectivamente [1].

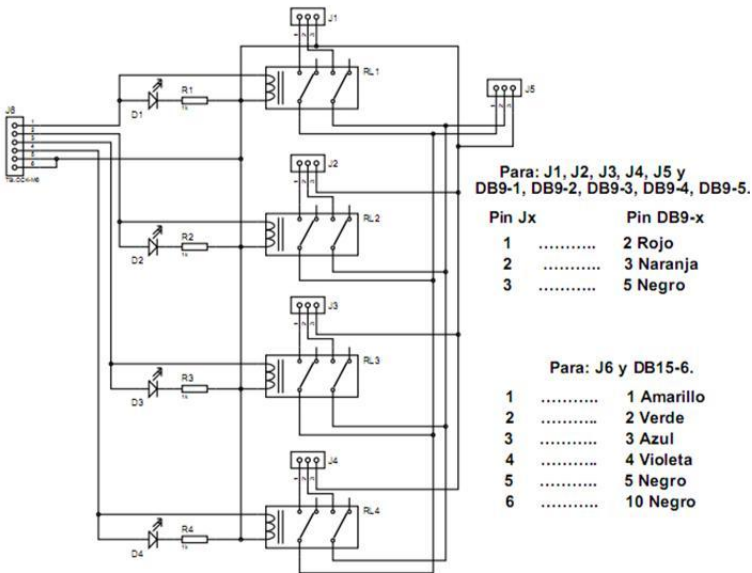


Figura 33. Diseño esquemático del multiplexor.

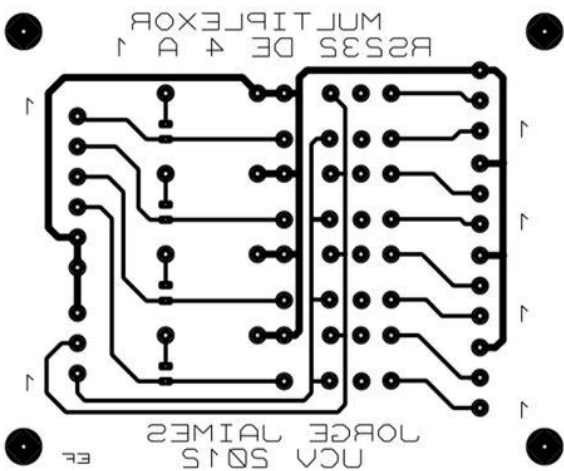


Figura 34. Diseño físico de la placa multiplexor.

En la figura 33 se observa que el diseño esquemático de la placa cuenta con cinco conectores DB-9, cuatro de ellos están destinados para la conexión con los controladores y el último para conectarlo hacia la computadora y establecer la comunicación. También se observa que hay un conector DB-15 con la finalidad de conectar el conmutador de estado de los relés con el equipo que se encarga de realizar el cambio para controlar la multiplexación.

3.3.3. Módulos FP-1001 y FP-DO-401

El equipo FP-1001 de National Instruments es un módulo de interfaz en red RS485 para el sistema de E/S (Entradas/Salidas) FieldPoint, capaz de conectar en un nodo hasta nueve módulos. El equipo administra la comunicación entre el servidor y los módulos de entradas y salidas a través de un bus local de alta velocidad, formado por bases de terminal FieldPoint [24]. En la figura 35 se observa el modulo FP-1001.



Figura 35. Módulo National Instruments FP-1001

El equipo FP-DO-401, que se observa en la figura 36, es un módulo de salidas digitales para FieldPoint, que puede ser utilizado para controlar señales en el rango de 5 hasta 30 V. Se utilizan normalmente para controlar luces indicadoras, relés externos y dispositivos TTL y CMOS [25].



Figura 36. Módulo de salidas digitales FP-DO-401

En el laboratorio se utilizó el módulo de salidas digitales para controlar los relés del circuito impreso multiplexor y el módulo FP-1001 para establecer la comunicación entre el equipo FP-DO-401 y la computadora. En la figura 37 se observan los equipos instalados en el laboratorio.

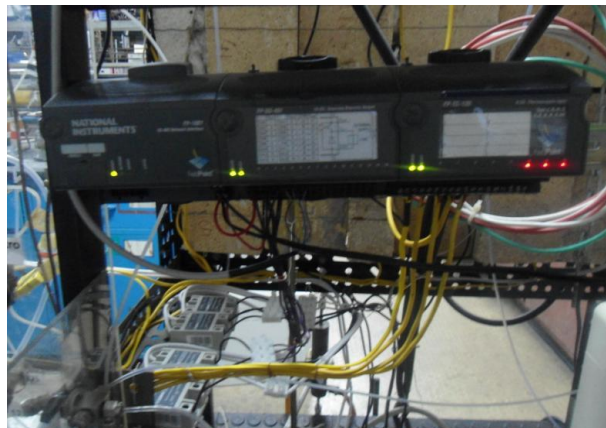


Figura 37. Módulos National Instruments instalados en el laboratorio.

CAPÍTULO IV

DISEÑO DEL SISTEMA SCADA

En este capítulo se presenta el diseño del sistema de control supervisorio y adquisición de datos. La arquitectura del sistema SCADA está dividida en tres partes: primero se tienen los controladores de flujo, que se utilizan para medir y controlar la tasa de flujo de los gases que se involucran en el proceso; luego está el sistema de adquisición de datos, que se comunica con los controladores de flujo para obtener los datos y retransmitirlos hacia una HMI; y por último el sistema de visualización y control, que muestra el estado de los gases y a través del cual se ajusta el punto de consigna de los controladores de flujo.

4.1. DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS

El sistema de adquisición de datos gestiona la comunicación entre los controladores de flujo y el sistema de visualización y control. En la automatización de procesos industriales se suelen utilizar unidades remotas y controladores lógicos programables para realizar esta función, pero para procesos a menor escala, como los que se realizan en laboratorios, no son necesarios esta clase de equipos que están diseñados para trabajar bajo las severas condiciones de un entorno industrial. En el caso de los procesos químicos que se desarrollan en los laboratorios con fines de investigación, se pueden utilizar sistemas embebidos. Estos equipos comúnmente no están diseñados para rangos de temperaturas elevados, inmunidad al ruido eléctrico o resistencia a la vibración y al impacto, características que atribuyen la robustez a los controladores industriales, sin embargo los sistemas embebidos pueden realizar las funciones de un controlador industrial y son más económicos. Existen muchos modelos de estos equipos que son fabricados por compañías como Intel, Texas Instruments, Arduino, entre otros, aunque también se pueden elaborar placas de forma independiente basándose en los requerimientos del proceso que se desea

controlar. Desarrollar un sistema embebido desde el inicio toma mucho tiempo y puede considerarse como un proyecto extenso que no está contemplado en el alcance de este trabajo, es por ello que se decidió utilizar sistemas embebidos comerciales para el sistema de adquisición de datos propuesto.

4.1.1. Selección del sistema embebido

Para la selección del sistema embebido se proponen equipos fabricados por tres compañías reconocidas del mercado, estos son:

Raspberry Pi 1 modelo B+, fabricada por Raspberry Pi Foundation;

Intel Galileo Gen 1, diseñada por Intel Corporation y

BeagleBone Black, elaborada por Texas Instruments.

4.1.1.1. Raspberry Pi 1 modelo B+

Es la versión final de la Raspberry Pi original, reemplazo la tarjeta Raspberry Pi 1 modelo B en julio del 2014. Entre sus características se destacan un microprocesador de 32 bits ARM1176JZFS de 700 MHz; 512 MB de memoria “Random Access Memory” RAM; 4 puertos USB 2.0; 40 terminales “General Purpose In/Out” (GPIO); 1 puerto “High Definition Multimedia Interface” (HDMI); puerto Ethernet; conector de audio de 3.5 mm; “Camera Serial Interface” (CSI); “Display Serial Interface” (DSI); ranura para tarjeta micro SD; un procesador gráfico VideoCore IV; puerto de comunicación serial; y se alimenta a través del puerto micro USB, se debe asegurar que el puerto micro USB suministre 2 A a 5 V. Funciona con una versión del sistema operativo Linux.[26]. En la figura 38 se muestra la placa Raspberry Pi 1 modelo B+.

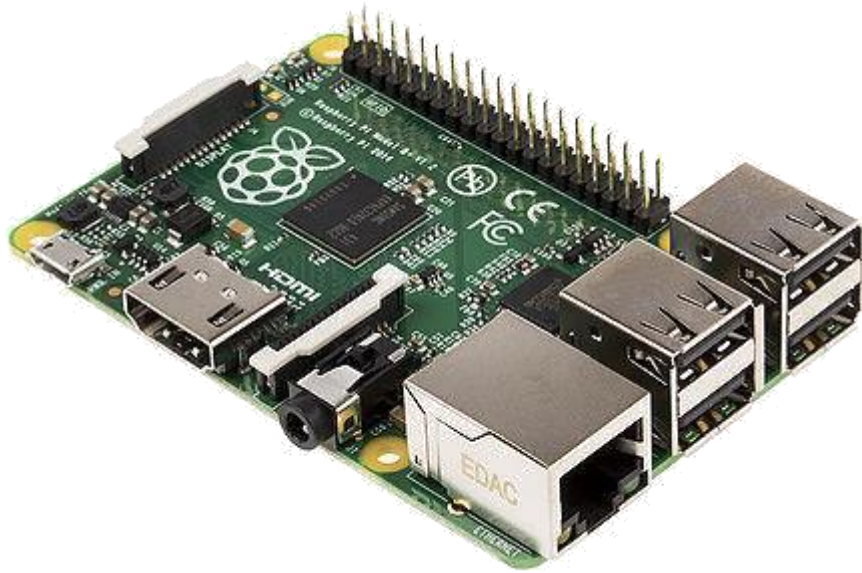


Figura 38. Raspberry Pi 1 modelo B+.

4.1.1.2. Intel Galileo Gen 1

Es una placa de desarrollo basada en un microprocesador Pentium de 32 bits Quark SoC X1000 de 400 MHz. Posee 14 terminales de entradas/salidas digitales (de las cuales 6 proporcionan salida PWM); 6 entradas analógicas; buses de comunicación I²C y SPI; puerto de comunicación serial, puerto Ethernet; conector USB Host 2.0; conector cliente USB; ranura para tarjeta micro SD y se alimenta a través conector de 2.1 mm por medio de un adaptador AC/DC que debe entregar 3 A a 5 V [27]. La placa Intel Galileo utiliza un sistema operativo basado en Linux y puede programarse a través de los métodos explicados en el marco teórico. En la figura 39 se muestran todas las características mencionadas anteriormente.

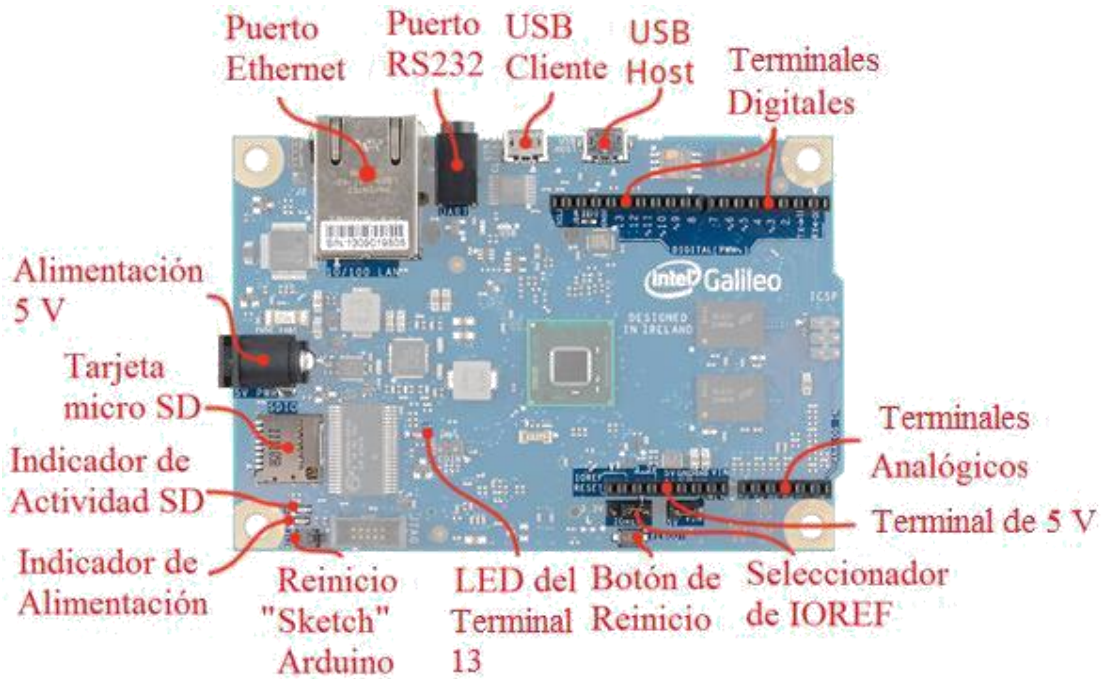


Figura 39. Características del Intel Galileo.

4.1.1.3. BeagleBone Black

Es un sistema embebido diseñado para la comunidad de código libre y aquellos interesados en microcontroladores de bajo costo. Tiene como cualidades un microprocesador de 32 bits Sitara XAM3359AZCZ100 de 1 GHz; 512 MB de memoria RAM; motor gráfico SGX530 3D; puerto Ethernet; puerto serial para depuración; ranura para tarjeta micro SD; memoria de almacenamiento “Embedded Multimedia Card” (eMMC) de hasta 2 GB; puerto host USB; puerto cliente USB; botón de reinicio; botón de encendido; interfaz HDMI; buses de comunicación I²C y SPI; 2 conectores de expansión que poseen 46 terminales cada uno; y puede ser alimentado a través de un conector de 2.1 mm por medio de un adaptador AC/DC que entregue 1 A a 5 V, también se puede alimentar por el puerto mini USB, USB o incluso por los conectores de expansión. Gracias al poderoso microprocesador que posee puede correr sistemas operativos como Linux, Android, Windows, entre otros [28]. En la figura 40 se muestra la placa BeagleBone Black.

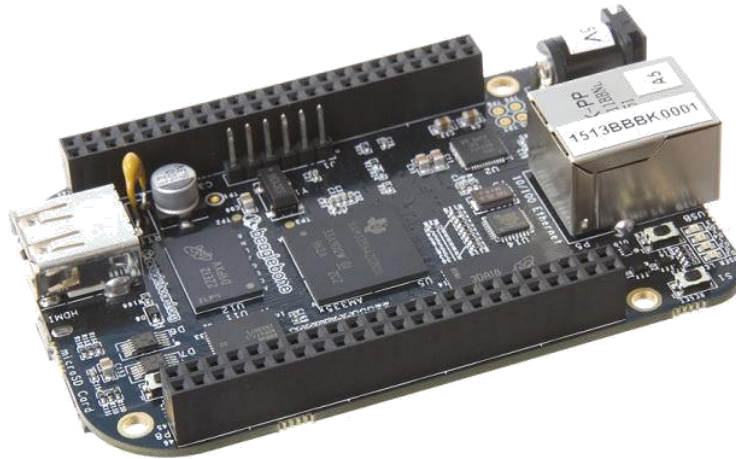


Figura 40. BeagleBone Black.

4.1.1.4. Requisitos del sistema y comparación

El sistema SCADA que se plantea tiene como requisitos mínimos que el dispositivo seleccionado posea una comunicación serial, para adquirir datos de los controladores de flujo; al mismo tiempo deberá ser capaz de comunicarse con una computadora, ya sea por el puerto serial, USB o Ethernet; y debe poseer entradas/salidas de propósito general. El proceso que se está monitorizando es lento, por lo cual las variables que intervienen no necesitan una repuesta de control inmediata, es por ello que las demás características que ofrecen los sistemas embebidos, tales como velocidad de procesamiento, memoria RAM o procesamiento gráfico, no son limitantes para escoger el dispositivo de adquisición de datos. En la tabla 10 se muestra una comparación de las características que ofrecen los equipos.

Tabla 10. Comparación de los sistemas embebidos.

Sistema embebido	Puertos USB/micro USB	Puerto Ethernet	Puerto serial	Microprocesador	Sistema operativo	GPIO	\$
Raspberry Pi 1 modelo B+	4 USB Host 1 Mini USB Host	Sí	Sí.	ARM1176JZFS (32 bits, 700 MHz)	Linux	40 GPIO	25

Intel Galileo Gen 1	1 Micro USB cliente y 1 micro USB Host	Sí	Sí	Quark SoC X1000 (32 bits, 400 MHz)	Linux	14 GPIO	70
BeagleBone Black	1 USB 2.0 Host 1 USB 2.0 Cliente	Sí	Sí	Sitara XAM3359AZCZ 100 (32 bits, 1 GHz)	Linux Android, Windows, etc.	65 GPIOs	45

Como se observa en la tabla 10, los tres equipos propuestos cumplen con los requisitos mínimos del sistema, es por ello que se toma en cuenta como criterio de selección el costo del equipo, dando como resultado que el sistema embebido Raspberry Pi 1 modelo B+ es el más económico y por lo tanto la mejor opción. Sin embargo, en este trabajo se decidió utilizar la placa de desarrollo Intel Galileo Gen 1, debido a que la misma fue adquirida a través de una donación realizada por Intel a la Escuela de Ingeniería Eléctrica, con el propósito de desarrollar trabajos de investigación con dichos equipos.

4.1.2. Selección del protocolo de comunicación entre el sistema embebido y los controladores de flujo

Luego de realizar la selección del sistema embebido, se realizó un estudio de los métodos de comunicación disponibles entre los controladores de flujo y el equipo escogido. Los controladores de flujo poseen un protocolo propietario para la gestión de información interna, pero para comunicarse con otros dispositivos utiliza el estándar de transmisión serial RS-232, el cual está limitado al formato de transmisión de paquetes 8-N-1, la única característica del estándar que puede ajustarse es la tasa de baudios, que está condicionada a los valores incluidos en el dispositivo. Por esta razón, el medio de transmisión de datos que se debe desarrollar está limitado a los estándares de comunicación que son compatibles con RS-232. Algunos de los modos de transmisión de datos que se pueden utilizar son los estándares RS-232, RS-422 y RS-485.

4.1.2.1. RS-232

El RS-232 es la interfaz serial más común y solía estar incluido como un componente característico en la mayoría de las computadoras compatibles con Windows; sin embargo en la actualidad se suele emplear a través de un conversor USB a RS-232. Éste comúnmente se utiliza para conectar los periféricos de una computadora o para automatización industrial, está limitado a conexiones punto a punto, su modo de comunicación es dúplex y utiliza señales desbalanceadas referenciadas a tierra, con una velocidad máxima de 115.2 kbits/s, pero algunos se pueden expandir hasta 1 Mbits/s.

4.1.2.2. RS-422

El estándar de comunicación serial RS-422 utiliza señales eléctricas diferenciales. En contraste con el estándar RS-232, la transmisión diferencial utiliza dos canales para transmitir y recibir, lo cual resulta en una mayor inmunidad al ruido y abarca mayores distancias; puede transmitir hasta velocidades de 10 Mbits/s y se suele utilizar para aplicaciones multipunto, en donde se conecta un solo transmisor a un bus de datos junto con un máximo de 10 receptores, para establecer una comunicación simplex. Gracias a estas características el estándar RS-422 se ajusta mejor a las aplicaciones de nivel industrial.

4.1.2.3. RS-485

El estándar RS-485 es una versión mejorada del estándar RS-422, fue diseñado para mejorar la limitante de multipunto que posee RS-422, permitiendo, en principio, hasta un máximo de 32 equipos comunicarse a través de un mismo bus de datos. Los esclavos pueden comunicarse entre sí sin necesidad de pasar a través de un dispositivo maestro. La inmunidad al ruido y la capacidad multipunto del estándar RS-485, lo convierte en la mejor elección para conexiones seriales a nivel industrial

[29]. En la figura 41 se muestra una comparación entre las arquitecturas multipunto de los estándares RS-422 y RS-485.

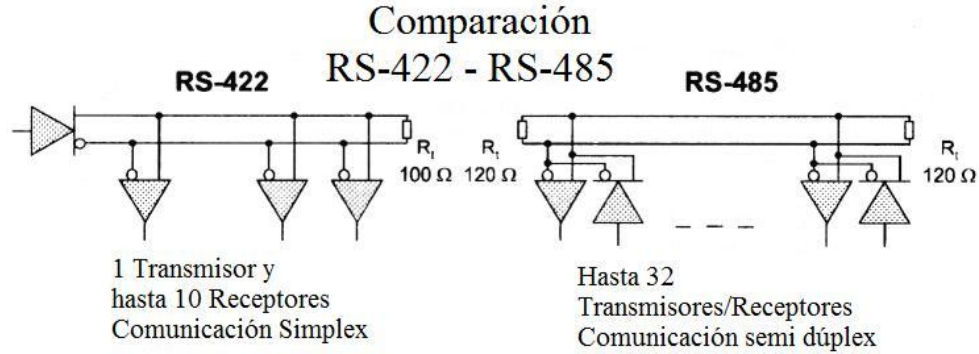


Figura 41. Arquitectura multipunto para RS-422 y RS-485.

4.1.2.4. Requisitos del sistema y comparación

En la tabla 11 se muestra una comparación entre las características más importantes de los tres estándares.

Tabla 11. Comparación entre los estándares RS-232, RS-422 y RS-485.

Especificaciones	RS-232	RS-422	RS-485
Conexión entre emisor y receptor	No equilibrada	Diferencial	Diferencial
Velocidad de transmisión máxima	115.2 kbits/s (En algunos casos hasta 1 Mbits/s)	10 Mbits/s	10 Mbits/s
Número máximo de emisor y receptor	1 Emisor 1 Receptor	1 Emisor 10 Receptores	32 Emisores 32 Receptores
1 Lógico en la salida	-5 a -15 V	-2 a -6 V	-1.5 a -5 V
1 Lógico en la entrada	-3 a -15 V	-0.2 a -7 V	-0.2 a -7 V
0 Lógico en la salida	5 a 15 V	2 a 6 V	1.5 a 5 V
0 Lógico en la entrada	3 a 15 V	0.2 a 7 V	0.2 a 12 V

Especificaciones	RS-232	RS-422	RS-485
Modo de comunicación	Dúplex	Dúplex , semi dúplex y simplex	Dúplex y semi dúplex
Tipo de conexión	Punto a punto	Multi punto o punto a punto	Multi punto

Para el sistema de transmisión de datos es necesario un modo de comunicación a través del cual se reciba y envíe información de los cuatro controladores de flujo y se pueda implementar en el sistema embebido por medio del puerto UART, el cual trabaja a niveles de tensión TTL de 0 a 5 V.

Basándose en esta necesidad, el estándar que destaca de entre los tres es el RS-485, ya que puede establecer una comunicación a 2 hilos semi dúplex multipunto o a 4 hilos dúplex multipunto, además posee alta inmunidad al ruido, la cual puede ser incluso mayor si se utilizan conductores de “Shielded Twisted Pair” (STP), y para implementarlo en el sistema embebido se puede utilizar un MAX485, el cual es un circuito integrado que convierte los niveles de tensión con los que trabaja RS-485 a TTL.

Al contrario que el estándar RS-485, con RS-422 no se puede utilizar la configuración multipunto, ya que es de tipo simplex, es decir, la información fluye en un solo sentido, y el sistema requiere como mínimo una comunicación semi dúplex. Para utilizar el estándar RS-422 se debe utilizar la configuración punto a punto, la cual puede ser dúplex o semi dúplex, y para establecer comunicación con los cuatro controladores de flujo se debe diseñar un circuito que se encargue de multiplexar la comunicación. Además el estándar RS-422 utiliza voltaje diferencial a los mismos niveles de tensión que RS-485 y por lo tanto se puede emplear un MAX485 para convertir los niveles de tensión a TTL.

Para implementar el estándar RS-232 también se debe diseñar un circuito que multiplexe la conexión entre los controladores de flujo, ya que esta comunicación

es punto a punto, y para convertir los niveles de tensión a TTL se utiliza un circuito integrado MAX232.

Para los estándares RS-485 y RS-422 no solo es necesario diseñar un circuito para convertir los niveles de tensión a TTL, sino que también se debe diseñar un bloque de acondicionamiento que transforme la señal proveniente de los controladores de flujo a las características eléctricas de dichos estándares. Estos circuitos pueden ser realizados con transistores, amplificadores operacionales o circuitos integrados, e incluso existen a la venta muchos modelos de circuitos que realizan esta función. La desventaja de estos circuitos es que al transformar la señal se puede perder información, además incrementa los costos de desarrollo y los puntos de posible falla, es por esta razón que se escogió para este trabajo el estándar RS-232 por encima de las otras opciones.

4.1.3. Diseño del circuito impreso

Una vez escogido el estándar de transmisión, se elaboró el circuito impreso que realiza el multiplexado de los controladores de flujo y la conversión de los niveles de tensión a TTL. El diseño se basó en los circuitos integrados MAX232 y CD4052B.

4.1.3.1. MAX232

El integrado MAX232 es un transmisor/receptor dual que incluye un generador de voltaje capacitivo, para proporcionar niveles de tensión que cumplen con el estándar RS-232 a partir de una fuente de alimentación de 5 V. Cada receptor convierte las entradas RS-232 en niveles de tensión TTL de 5 V, estos receptores tienen un voltaje de umbral típico de 1.3 V, una histéresis típica de 0.5 V y puede aceptar entradas desde -30 a 30 V. Cada transmisor convierte entradas TTL en niveles de tensión aceptados por el estándar RS-232. En la figura 42 se muestra la

disposición y el nombre de los terminales del MAX232 y en la tabla 12 se describe la función de cada uno [30].

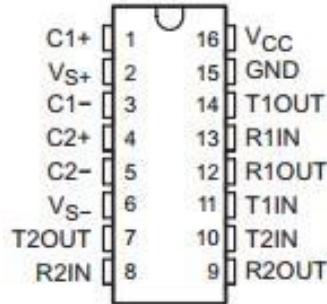


Figura 42. Configuración de terminales del integrado MAX232.

Tabla 12. Funciones de los terminales del integrado MAX232.

Nombre	Nº Terminal	Tipo	Descripción
C1+	1	-	Polo positivo del condensador C1
VS+	2	Salida	Salida positiva de carga solo para condensadores de almacenamiento
C1-	3	-	Polo negativo del condensador C1
C2+	4	-	Polo positivo del condensador C2
C2-	5	-	Polo negativo del condensador C2
VS-	6	Salida	Salida negativa de carga solo para condensadores de almacenamiento
T2OUT, T1OUT	7, 14	Salidas	Línea de datos de salida RS-232 (hacia el sistema remoto RS-232)
R2IN, R1IN	8, 13	Entradas	Línea de datos de entrada RS-232 (desde el sistema remoto RS-232)
R2OUT, R1OUT	9,12	Salidas	Salida lógica de datos (hacia el UART)
T2IN, T1IN	10, 11	Entradas	Entrada lógica de datos (desde el UART)
GND	15	-	Referencia

Nombre	N° Terminal	Tipo	Descripción
Vcc	16	-	Voltaje de alimentación, conectado a una fuente externa de 5 V

El bloque de alimentación aumenta e invierte la señal de 5 V utilizando una bomba de carga que requiere cuatro condensadores externos de 1 μ F. El integrado posee resistencias internas de “pull-up” en los terminales de TIN para garantizar un nivel lógico alto en la entrada cuando la línea tiene alta impedancia y en el caso de una entrada abierta en ROUT resultara en un nivel lógico alto a la salida. Es recomendable colocar un condensador en el terminal 16 de alimentación para disminuir las variaciones de voltaje y evitar un funcionamiento incorrecto del integrado.

4.1.3.2. CD4052B

Los multiplexores y demultiplexores de la familia CD405xB son selectores analógicos controlados digitalmente, los cuales tienen baja impedancia cuando están activos y muy baja corriente de fuga en caso contrario. El integrado CD4052B es un multiplexor diferencial de 4 canales que posee dos entradas binarias de control y un deshabilitador. Las dos entradas binarias seleccionan uno de cuatro pares de canales para establecer la conexión con las salidas. . En la figura 43 se muestra la disposición y el nombre de los terminales del CD4052B y en la tabla 13 se describe la función de cada uno [31].

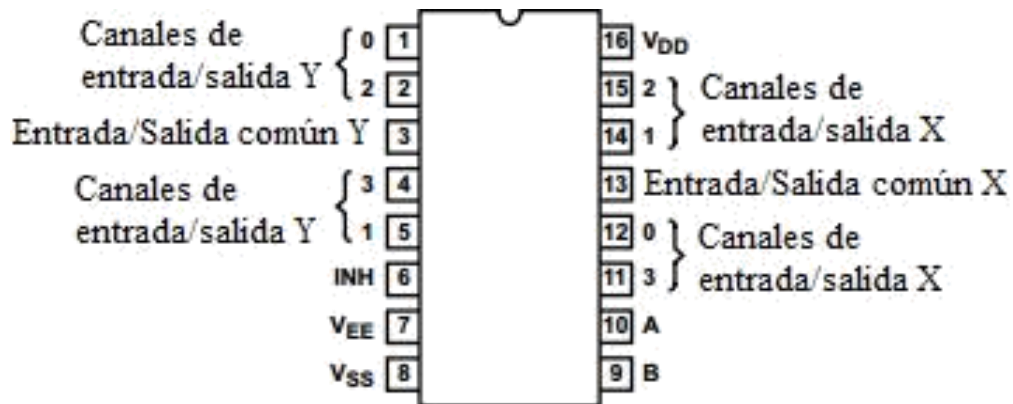


Figura 43. Configuración de terminales del integrado CD4052B.

Tabla 13. Funciones de los terminales del integrado CD4052B

Nombre	Nº Terminal	Tipo	Descripción
Y Canal 0	1	Entrada/Salida	Entrada/Salida Y del canal 0
Y Canal 2	2	Entrada/Salida	Entrada/Salida Y del canal 2
Y Común	3	Entrada/Salida	Entrada/Salida Y común
Y Canal 3	4	Entrada/Salida	Entrada/Salida Y del canal 3
Y Canal 1	5	Entrada/Salida	Entrada/Salida Y del canal 1
INH	6	Entrada	Deshabilita todos los canales
VEE	7	-	Entrada de alimentación negativa
VSS	8	-	Referencia
B	9	Entrada	Selector de canal B
A	10	Entrada	Selector de canal A
X Canal 3	11	Entrada/Salida	Entrada/Salida X del canal 3
X Canal 0	12	Entrada/Salida	Entrada/Salida X del canal 0
X Común	13	Entrada/Salida	Entrada/Salida X común
X Canal 1	14	Entrada/Salida	Entrada/Salida X del canal 1

Nombre	Nº Terminal	Tipo	Descripción
X Canal 2	15	Entrada/Salida	Entrada/Salida X del canal 2
VDD	16	-	Entrada de alimentación positiva

4.1.3.3. Descripción del circuito impreso

Se desarrollaron dos circuitos impresos. El primero contiene los integrados MAX232, con sus respectivos condensadores; el multiplexor CD4052B y “pin headers” macho, necesarios para: alimentar el circuito; controlar el selector del multiplexor; conectar las señales al MAX232 y conectar las salidas que se dirigen al puerto UART del sistema embebido. En el segundo circuito se tienen 12 bornes, para conectar las señales de comunicación que provienen de los controladores de flujo; “pin headers” hembra, para conectarse con la primera placa; y “pin headers” macho, colocados de tal manera que la placa se acople al sistema embebido como si de un “Shield” Arduino se tratase.

Cada integrado MAX232 cuenta con dos grupos de transmisores/receptores duales, por lo que se necesitan dos de ellos para convertir las señales de los cuatro controladores de flujo. Las salidas de estos integrados, a niveles de tensión TTL, se conectan en los canales 0, 1, 2 y 3 del multiplexor. Los selectores A, B y el deshabilitador se controlan a través de tres GPIO del sistema embebido y las salidas comunes X y Y se conectan al puerto UART del equipo. En la segunda placa se tienen acoplados a los bornes cuatro conectores DB-9 macho, para conectar los cables adaptadores de los controladores de flujo.

En las figuras 44 y 45 se muestra el diseño esquemático y físico de la placa de multiplexado; en las figuras 46 y 47 se presenta el diseño esquemático y físico de la placa de acople con el sistema embebido y por último en las figuras 48, 49 y 50 se observa el resultado de los diseños, cada uno por separado y acoplados al equipo.

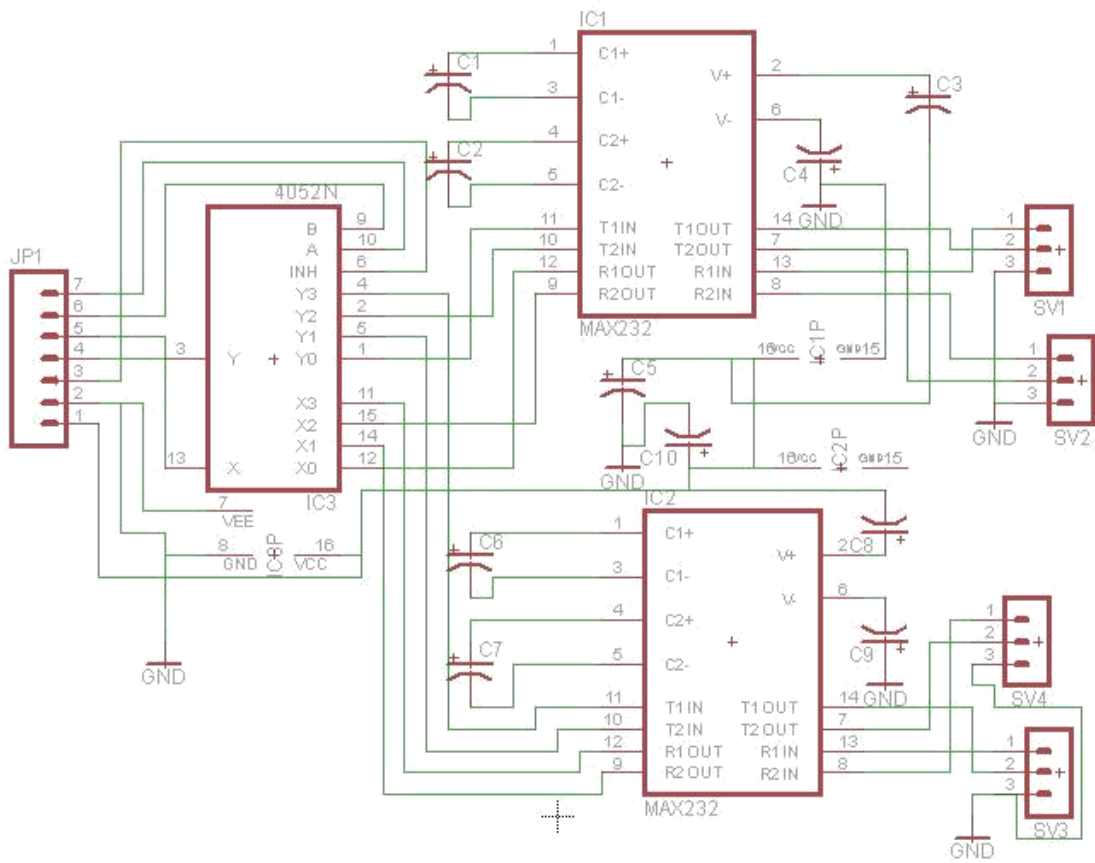


Figura 44. Diseño esquemático de la placa de multiplexado.

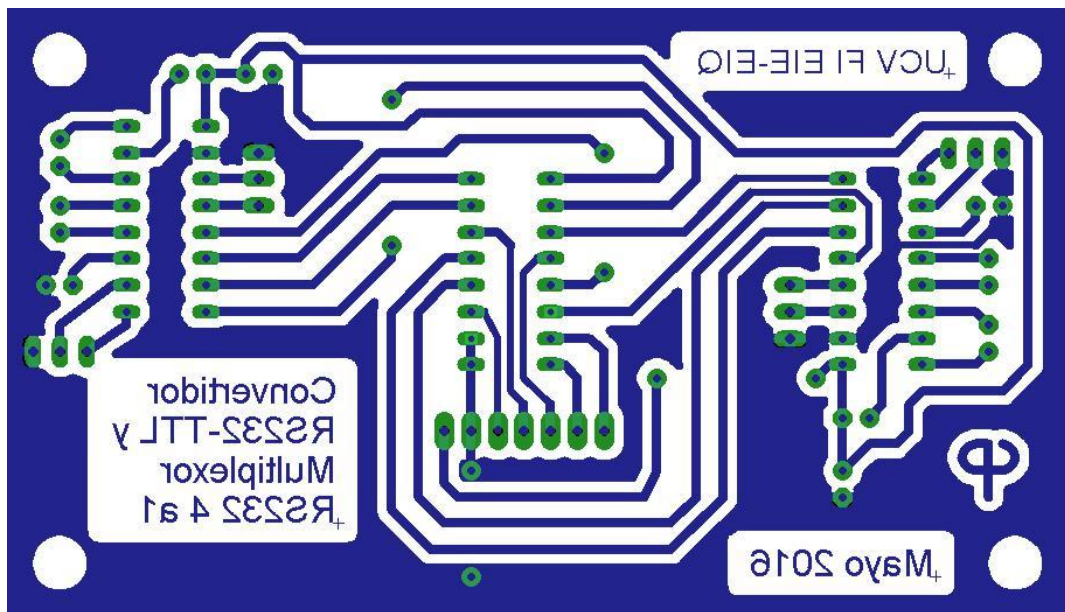


Figura 45. Diseño físico de la placa de multiplexado.

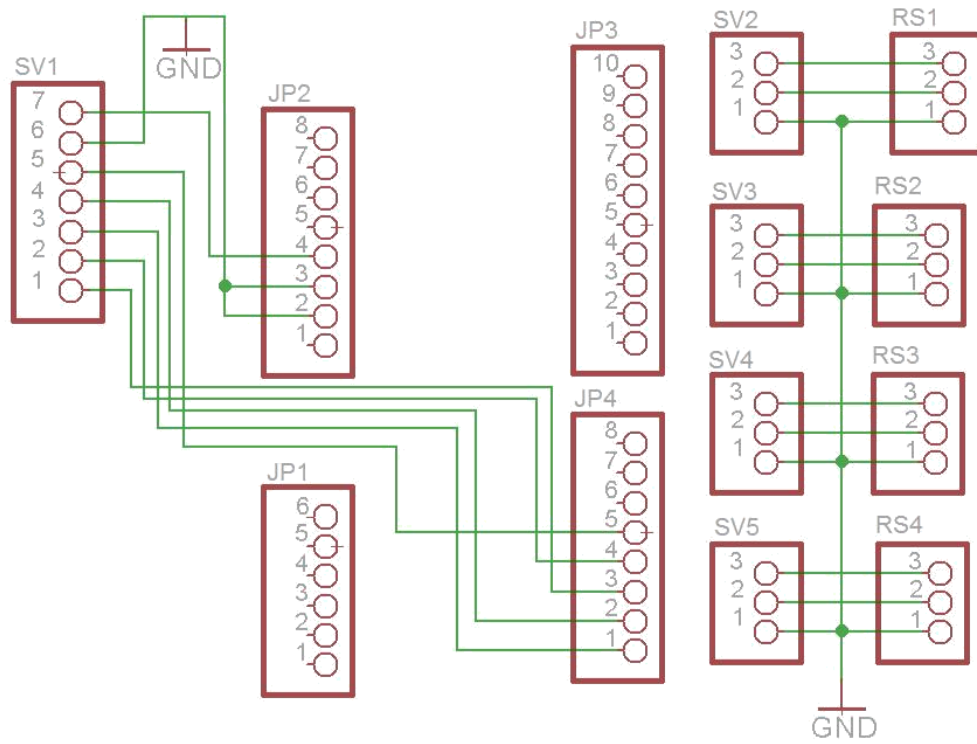


Figura 46. Diseño esquemático de la placa de acople al sistema embebido.

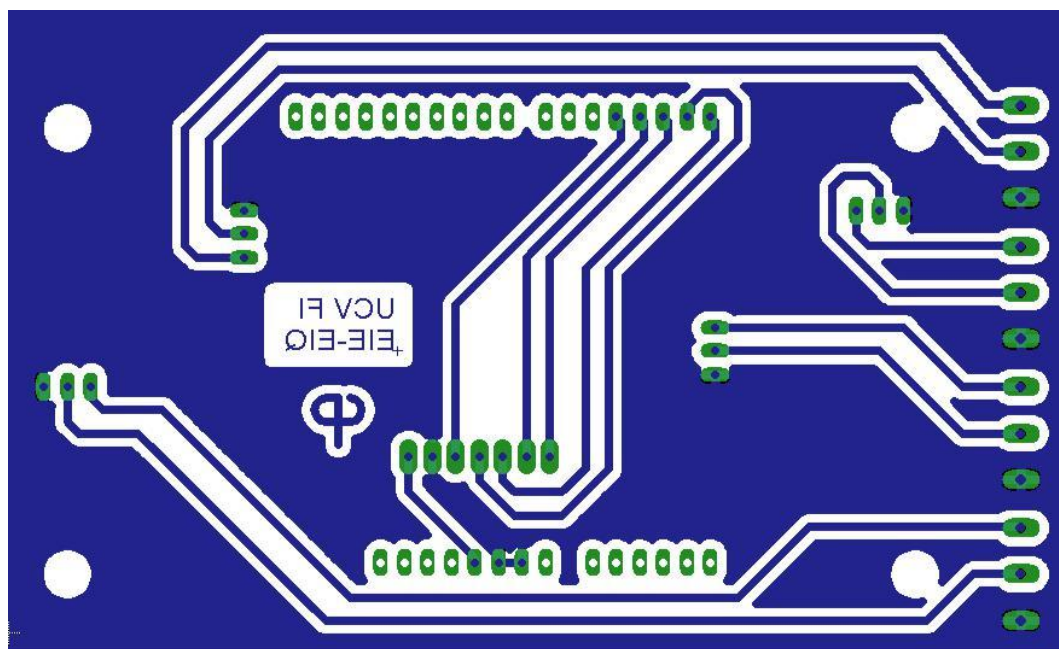


Figura 47. Diseño físico de la placa de acople al sistema embebido.



Figura 48. Circuito impreso con los integrados MAX232 y CD4052B.

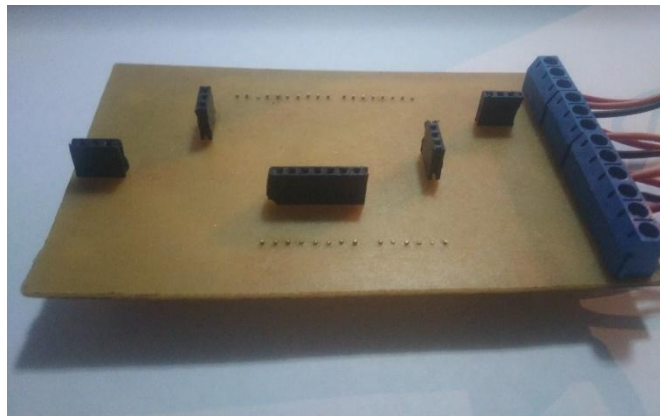


Figura 49. Circuito impreso con los bornes y “pin headers”.

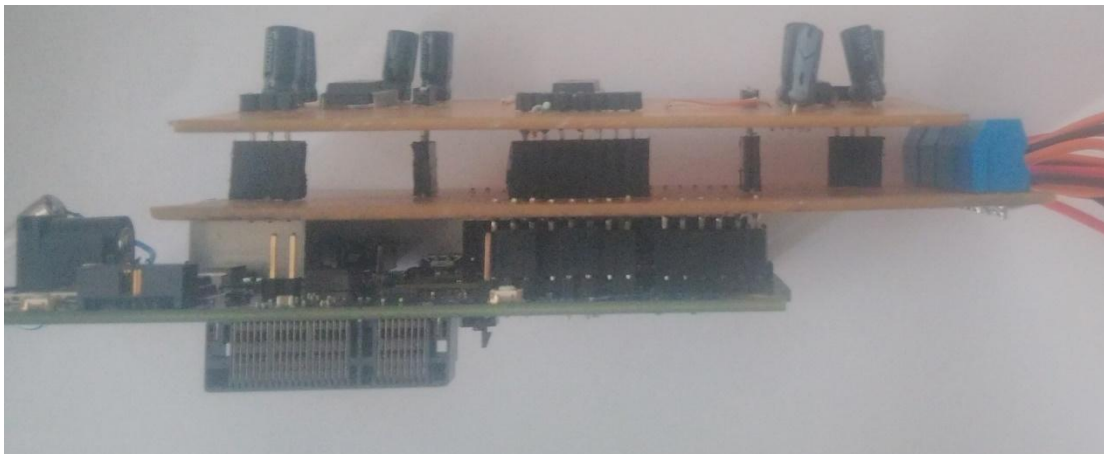


Figura 50. Circuitos elaborados acoplados al sistema embebido.

4.2 DISEÑO DEL SISTEMA DE VISUALIZACIÓN Y CONTROL

El sistema de visualización y control es la forma en que el operador interactúa con el proceso. El usuario monitorea el estado de los gases; registra el comportamiento de las variables en tiempo real; almacena información en un histórico de datos y ajusta el punto de consigna del sistema de control, todo esto a través de una sencilla aplicación HMI que posee control de acceso y alarmas de seguridad.

4.2.1. Selección del software de visualización

En el mercado existe una gran variedad de programas que se utilizan para crear las aplicaciones HMI. La selección del software de desarrollo se realizó a través de un estudio de las características y beneficios de los productos ofrecidos por algunas compañías reconocidas. Entre las opciones que se estudiaron se tienen:

SIMATIC WinCC, de SIEMENS;

Factory Talk View SE, de Rockwell Automation y

LabVIEW, de National Instruments.

4.2.1.1. Simatic WinCC

Es un sistema de visualización de procesos moderno, que presenta una interfaz de usuario atractiva y muchas funcionalidades para controlar procesos automatizados bajo el sistema operativo Windows y para todos los segmentos de la industria. Se pueden desarrollar aplicaciones que van desde sistemas simples de un solo usuario hasta sistemas distribuidos de múltiples usuarios con servidores redundantes que incluyen clientes web. Este software combina ingeniería eficiente con almacenamiento de datos eficaz y máxima seguridad en el manejo de datos [32].

4.2.1.2. Factory Talk View Site Edition

Es un software HMI para monitorizar y controlar aplicaciones distribuidas o multiusuario. Proporciona una imagen completa y precisa de las operaciones, satisfaciendo las demandas de múltiples partes interesadas, incluyendo ingeniería, mantenimiento, operaciones, producción y tecnología de la información. Ofrece una funcionalidad robusta y confiable en un solo software que va desde un sistema HMI robusto a una solución de visualización distribuida [33].

4.2.1.3. LabVIEW

Es un entorno de desarrollo con una sintaxis de programación gráfica que facilita visualizar, crear y codificar sistemas de ingeniería. Está diseñado para incorporarse con otro software, ya sea como métodos alternativos de desarrollo o plataformas abiertas, para garantizar una gran cantidad de herramientas a la disposición del usuario [34].

4.2.1.4. Requisitos del sistema y comparación

Para seleccionar el entorno de desarrollo HMI se tuvo como limitante las especificaciones técnicas del sistema donde se instaló el programa. En el laboratorio se dispone de un computador con procesador Pentium IV, 1 GB de memoria RAM y con un sistema operativo Windows 7. La tabla 14 muestra los requisitos mínimos del computador para utilizar los programas tratados en esta sección.

Tabla 14. Requisitos mínimos del sistema

Requerimientos mínimos	SIMATIC WinCC	Factory Talk View SE	LabVIEW
Procesador	Intel Core 2 Duo o superior	Intel Core i3 3.5 GHz	Pentium III de 866 MHz
Memoria	2 GB o más	3 GB (32 bits) o 8GB (64 bits)	256 MB

Requerimientos mínimos	SIMATIC WinCC	Factory Talk View SE	LabVIEW
Sistema operativo	Windows	Windows	Windows

Al comparar las especificaciones del computador con los requerimientos mínimos de los programas se determina que la única opción que cumple las necesidades es LabVIEW. Es del conocimiento general que para utilizar estos programas es necesario comprar una licencia, que por lo general tienen un coste elevado, no obstante la Escuela de Ingeniería Química posee una licencia de LabVIEW 2011 con el ServiPack 1, por estas razones se utilizará el software de National Instruments para desarrollar el sistema de visualización y control.

4.2.2. Selección del criterio de control

Los controladores de flujo ofrecen un control PD integrado por medio del cual se puede realizar el control a lazo cerrado de la presión, el flujo volumétrico o el flujo másico. Existen diferentes métodos de ajuste de controladores PD, estos se pueden clasificar en métodos de ajuste analíticos y métodos de ajuste empíricos.

Los métodos de ajuste analíticos son procedimientos sistemáticos para la determinación de los parámetros de control, para realizar estos métodos es necesario conocer el modelo de la planta. Un ejemplo de estos procedimientos sería el ajuste en el dominio de la frecuencia de Aström y Hägglung. Al conocer el modelo de la planta en el dominio de la frecuencia se parten de unas características en un estado “A” y se desean llegar a las características en el estado “B”, haciendo uso del margen de ganancia y margen de fase de la planta en ambos estados se determinan los parámetros del controlador, en la figura 51 se observa una ejemplificación de este método. [35]

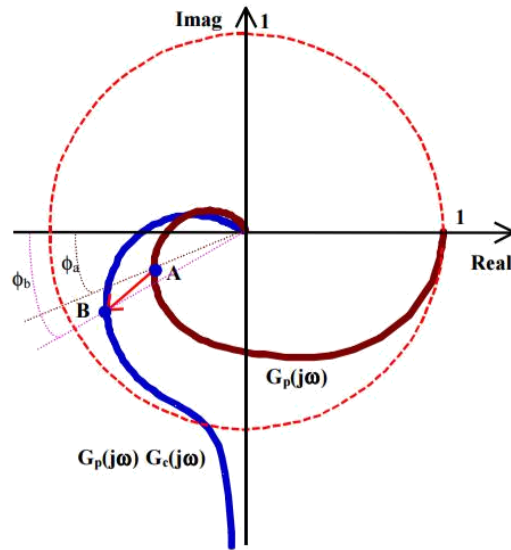


Figura 51. Ajuste en el Dominio de la frecuencia Aström y Hägglung.

Se utilizan las ecuaciones 10 y 11 para determinar los parámetros del controlador PD.

$$\frac{(\quad)}{\quad} \quad (10)$$

$$\frac{(\quad)}{\quad} \quad (11)$$

Los métodos de ajuste empírico son aquellos que calculan los parámetros del controlador a partir de experimentos y el análisis de la respuesta de la planta, Un ejemplo de estos métodos sería el método de la curva de reacción de Cohen y Coon. En éste se analiza la respuesta del sistema ante una entrada escalón y se determinan tres características de la curva de reacción (tiempo muerto, ganancia y constante de tiempo). El controlador es colocado en manual, luego de que el sistema se estabilice se realiza una variación en el controlador y se espera a que el sistema se estabilice nuevamente. La ganancia se calcula como la variación en porcentaje de la variable de proceso dividido entre la variación en porcentaje de la salida del controlador. El tiempo muerto se mide como la diferencia entre el cambio en el controlador y el cambio en la

variable de proceso. Por último la constante de tiempo es la diferencia entre la intersección al final del tiempo muerto y el tiempo en el que la variable de proceso alcanza el 63% del cambio, ver figura 52. [36]

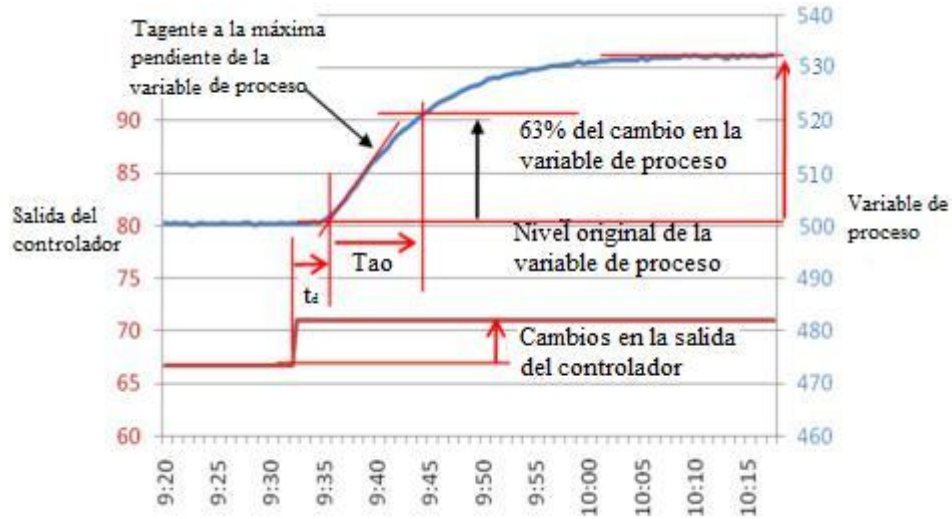


Figura 52. Método de la curva de reacción Cohen y Coon.

En las ecuaciones 12 y 13 se observa la relación entre las características de la respuesta y los parámetros del controlador.

$$\text{---} (\text{---}) \tag{12}$$

$$(\text{---}) \tag{13}$$

En los métodos empíricos también se suelen utilizar métodos de ensayo y error donde se realizan modificaciones sucesivas de los parámetros de control hasta alcanzar el comportamiento deseado del sistema. Este método consiste en disminuir la variable derivativa a 0 y aumentar la variable proporcional hasta observar oscilaciones en la respuesta. Luego se aumenta la variable derivativa hasta que se

alcance el comportamiento deseado, a pesar de que el método manual parece simple requiere mucho tiempo y experticia. [36]

Para el funcionamiento de la planta es necesario controlar el flujo másico de los gases de reacción, es por ello que en estudios anteriores [1] se determinaron los parámetros P y D del lazo de control. Debido a que no se conoce el modelo de la planta se seleccionó un método empírico (método manual) para el ajuste de los parámetros P y D y se fijaron a 89 y 12000 respectivamente. A través del sistema SCADA propuesto se realiza el control supervisorio del flujo ajustando el punto de consigna de los controladores sin alterar el valor de las constantes P y D del lazo de control.

4.2.3. Selección del protocolo de comunicación entre el sistema embebido y el software de visualización

Para la selección del protocolo de comunicación entre la computadora y el sistema embebido se estudiaron las posibilidades de comunicación que existen entre los equipos. El Intel Galileo se puede comunicar con la computadora a través del puerto serial, USB o Ethernet.

Se descarta la utilización del puerto serial ya que el UART 0 se está utilizando actualmente para establecer la comunicación con los controladores de flujo y el UART 1 se utiliza solamente para programación. Así mismo la comunicación por USB también se descarta, en este caso por prevención, ya que el equipo puede sufrir daños severos si falla el suministro de energía y el dispositivo se encuentra conectado a la computadora por dicho puerto. Por lo tanto el protocolo de comunicación que se implementará debe ser compatible con el puerto Ethernet. Entre algunos de los modos de comunicación posibles están:

Modbus TCP,

Ethernet IP y
Secure Shell (SSH).

4.2.3.1. Modbus TCP

Actualmente es uno de los estándares industriales más utilizados debido a su simplicidad, bajo coste, necesidades mínimas de hardware y porque es un protocolo abierto. Se utiliza para la gestión de entradas/salidas distribuidas, el intercambio de información entre dispositivos, así como para monitorizarlos y gestionarlos. Modbus TCP proporciona una red abierta y accesible para el intercambio de datos [37].

4.2.3.2. Ethernet IP

Es un protocolo de red en la capa de aplicación del modelo OSI basado en el protocolo de control e información, utilizado en DeviceNet y ControlNet, ofrece un sistema integrado desde la planta industrial hasta la red central de la empresa. Al basarse en los estándares Ethernet se garantiza un completo funcionamiento con todos los dispositivos Ethernet/IP utilizados en la actualidad [38].

4.2.3.3. Secure Shell

Es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente servidor y permite a los usuarios conectarse a un host remotamente. SSH está diseñado para reemplazar los métodos más viejos y menos seguros que se utilizan como modo de ingreso remoto a otro sistema, a través de la Shell de comando. [39]

4.2.3.4. Requisitos del sistema y comparación

Para establecer la comunicación es necesario que el programa de visualización y el sistema embebido puedan implementar el mismo protocolo de

transmisión de datos. Con respecto al Intel Galileo, el equipo soporta cualquiera de los tres métodos de comunicación mencionados, ya que dentro de la imagen de Yocto de Linux se dispone de un servidor y cliente de SSH, así como también se tiene acceso a varios lenguajes de programación, los cuales poseen muchas herramientas que permiten implementar los otros protocolos. Por otra parte, aunque una de las ventajas de LabVIEW es poseer una alta compatibilidad con software de terceros, en la mayoría de los casos, es necesario adquirir una licencia para hacer uso de estos paquetes, lo cual limita el acceso a parte del público. Con el protocolo Modbus TCP no sucede esto, a través de la versión LabVIEW con la que se cuenta en el laboratorio, se puede crear un servidor de entradas/salidas donde se tiene Modbus TCP como uno de los protocolos típicos, y es por esta razón que se escogió Modbus TCP para establecer la comunicación.

CAPÍTULO V

PROGRAMACIÓN DE LOS ELEMENTOS QUE CONFORMAN EL SISTEMA SCADA

En este capítulo se presenta la programación del sistema embebido y del software de visualización. Se empezó por programar el Intel Galileo, ya que es la interfaz que gestiona la comunicación entre los controladores de flujo y la aplicación HMI.

5.1 PROGRAMACIÓN DEL SISTEMA EMBEBIDO

La programación del sistema embebido se puede realizar a través de diferentes lenguajes de programación, en la mayoría de los casos utilizando alguno de los IDE mencionados en el marco teórico. Para este trabajo se escogió programar el Intel Galileo a través de la consola de Linux utilizando Python como lenguaje de programación.

Programar directamente en la imagen de Linux otorga al desarrollador un gran control de los periféricos y características que ofrece el equipo, además al utilizar Python se puede hacer uso de la gran cantidad de librerías y módulos desarrollados en este lenguaje que facilitan la creación del programa; así mismo Python es un lenguaje que se puede correr en casi cualquier versión de Linux u otros sistemas operativos, lo cual facilita la implementación del programa en diferentes sistemas embebidos.

Para comenzar a programar el Intel Galileo en Python lo primero que se hizo fue construir en una tarjeta micro SD la imagen del sistema operativo Linux.

5.1.1. Imagen del sistema operativo Linux

El tamaño de la tarjeta micro SD que se usó fue de 2 GB, en la página web de Intel se especifica un tamaño mínimo de 2 GB y máximo de 32 GB; luego se descargó la última versión de la imagen del sistema operativo Linux, disponible en la página web de descargas de la placa Intel Galileo, y se extrajo en el escritorio de la computadora; a la tarjeta micro SD se le borraron todas las particiones de memoria existentes y se creó una partición de tipo FAT32, la cual es compatible con todos los sistemas y formatos; para finalizar se grabó la imagen del sistema operativo en la micro SD.

5.1.2. Python y librerías utilizadas

Para acceder a la consola Linux del sistema embebido existen dos métodos; uno es a través del puerto serial de 3.5 mm y el otro es utilizando el puerto Ethernet estableciendo una comunicación SSH con la computadora, en ambos casos al realizar una conexión exitosa, el equipo requiere el nombre de usuario root para acceder al sistema. Para conectarse al sistema usando SSH se puede utilizar un software de comunicación como Putty, si se tiene una computadora con sistema operativo Windows; o directamente desde la terminal si se utiliza el sistema operativo Linux.

La programación de Intel Galileo con Python se realiza a través de archivos de texto denominados scripts, en éstos se escriben instrucciones por medio de las que se maneja el entorno de desarrollo de la placa y son almacenados dentro de la tarjeta micro SD, se recomienda nombrarlos con la extensión .py para diferenciarlos del resto de los archivos de texto. Para ejecutar los scripts se les debe otorgar permisos de ejecución y la primera línea de código escrita debe indicar el directorio en el cual se encuentra instalado Python, esto con el fin de que el sistema reconozca el lenguaje con el cual se interpretan las instrucciones del archivo. Se utilizaron varias librerías

para facilitar el desarrollo del programa, las más importantes se explican a continuación.

5.1.2.1. Wiring-x86

Wiring-x86 es un módulo de Python que permite utilizar la funcionalidad tipo Arduino que poseen las placas de Intel Galileo e Intel Edison. Proporciona una “Application Programming Interface”(API) sencilla, para gestionar los terminales GPIO en la placa, hasta el momento esta librería soporta utilizar dichos terminales como salidas digitales o PWM y leerlos como entrada digital o analógica. Esta librería solo funciona para las placas Intel mencionadas, que tengan el sistema operativo Yocto de Linux [40].

5.1.2.2. Pyserial

Esta librería encapsula el acceso hacia el puerto serial. Proporciona el “backend” para la mayoría de los sistemas operativos en que corre Python; el modulo llamado “serial” automáticamente selecciona el “backend” apropiado. Entre sus características destacan: acceso a la configuración del puerto serial a través de las propiedades de Python; soporta diferentes tamaño de byte, bits de parada, parada y control de flujo; y funciona con o sin límite de tiempo para la recepción [41].

5.1.2.3. Pymodbus

Es una implementación completa del protocolo Modbus. Puede utilizarse sin dependencia de software de terceros, a parte de Pyserial, para crear proyectos más ligeros. Funciona perfectamente en versiones de Python mayores a 2.3. Se puede implementar como cliente o servidor, ofrece características como: escritura y lectura de registros y entradas/salidas discretas; Modbus TCP, “User Datagram Protocol” (UDP), serial ASCII, serial RTU y serial binario; y se puede implementar una comunicación síncrona o asíncrona [42].

5.1.2.4. Expresiones regulares

Este módulo proporciona operaciones de emparejamiento de expresiones regulares, parecidas a aquellas que se utilizan en Perl. Los patrones o caracteres a buscar pueden ser de tipo unicode así como también de ocho bits. Una expresión regular especifica un conjunto de caracteres; la función de este módulo es verificar si un conjunto particular de caracteres coincide con una expresión regular dada [43].

5.1.3. Descripción del programa

El programa diseñado tiene como objetivo principal gestionar la comunicación del sistema SCADA, para cumplir este propósito se desarrollaron tres scripts. El script principal consiste básicamente en la ejecución cíclica de dos funciones; la primera se encarga de recibir la información de los controladores de flujo y almacenarla en los bloques de memoria del servidor Modbus; y la segunda recibe las órdenes de control supervisorio desde la aplicación HMI y las lleva a cabo. Además de las funciones principales se tiene una función que verifica la validez de la trama y descarta aquellas que puedan contener errores y otra que transforma variables tipo flotantes en enteros sin signo, para almacenar la información en los registros del protocolo Modbus. El segundo script configura la dirección IP, el puerto de acceso y los bloques de memoria del servidor Modbus e inicia el servicio; y el último script se utiliza para configurar la dirección IP estática del sistema embebido e iniciar los scripts anteriores, este script se ejecuta en el arranque del equipo. En la figura 53 se observa el diagrama de flujo del script principal.

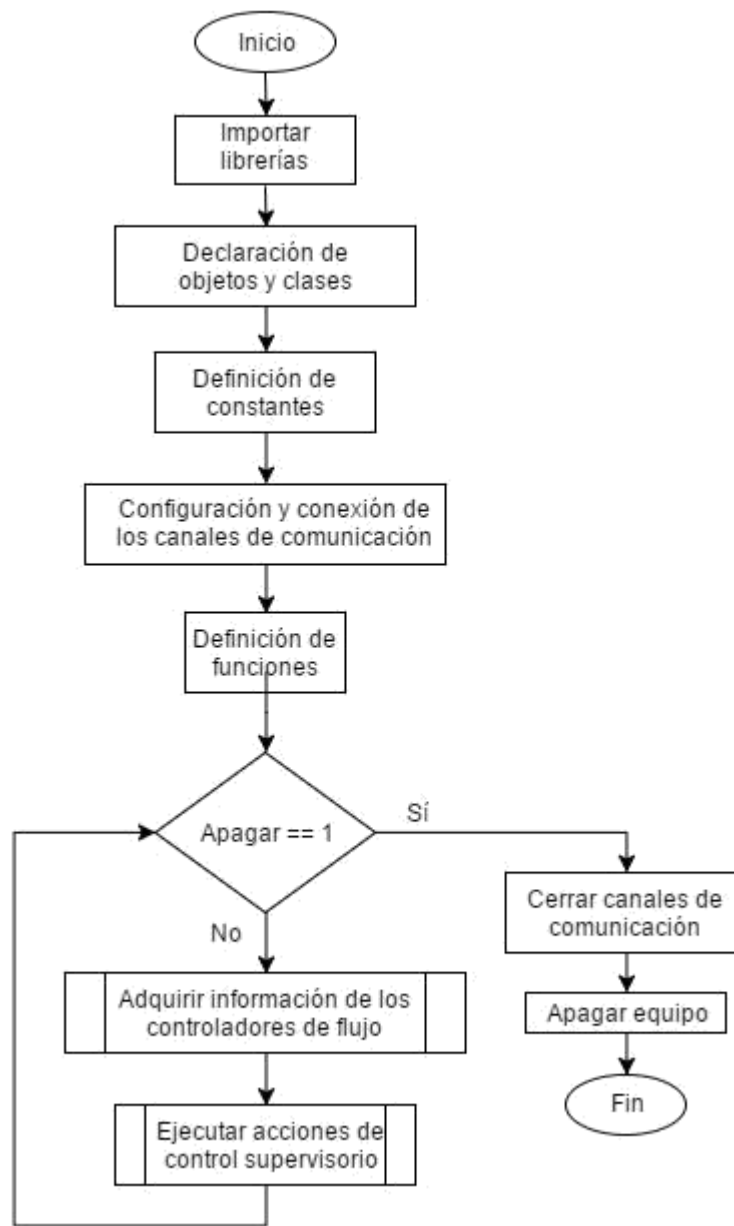


Figura 53. Diagrama de flujo del script principal del sistema embebido.

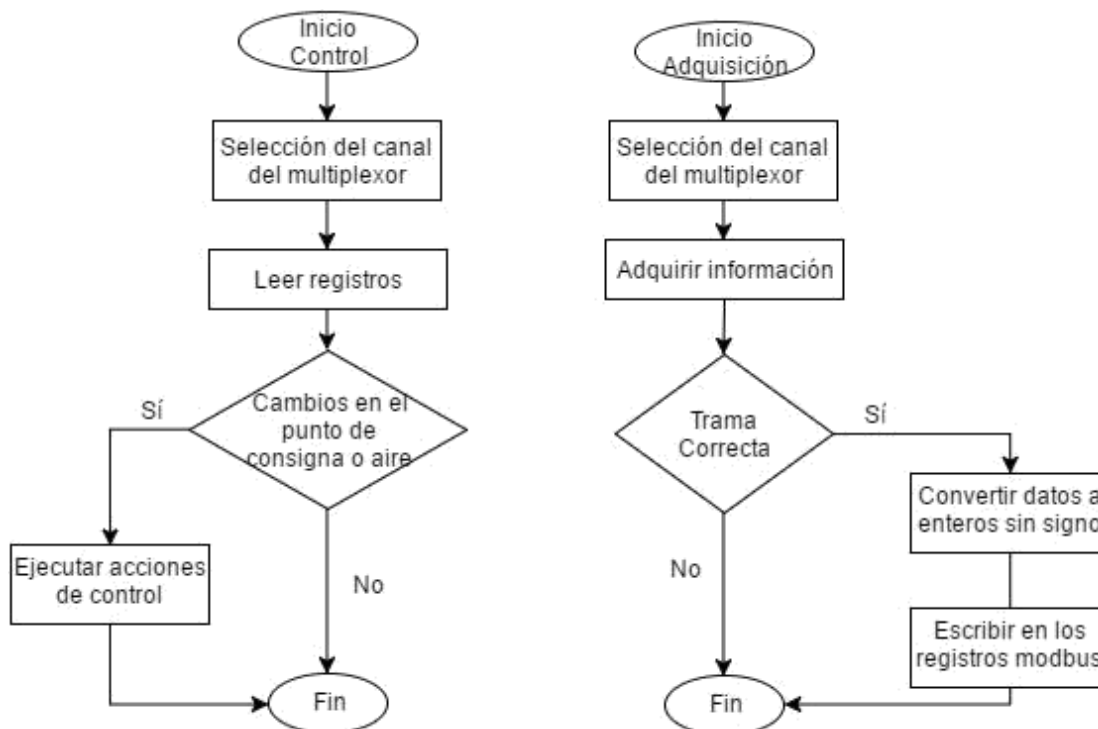


Figura 54. Diagramas de flujo de las funciones principales.

En la figura 54 se muestran los diagramas de flujo de las funciones de adquisición de datos y control supervisorio. La información adquirida de los controladores de flujo posee un formato único definido anteriormente, para la verificación de la trama se creó una expresión regular con el formato de los datos, si los datos recibidos coinciden con la expresión regular, la trama no tiene errores y por lo tanto es apta para almacenar la información en los bloques de memoria del protocolo Modbus, en caso contrario se descarta la trama. Para realizar las acciones de control supervisorio la función compara si hay cambios en el punto de consigna o en el gas que se está controlando, si alguno de los parámetros es diferente al introducido mediante la interfaz HMI este es sustituido por el valor más reciente, en caso contrario no se realiza ninguna acción y se deja el último valor asignado. En la figura 55 se observan los diagramas de flujo de los scripts secundarios.

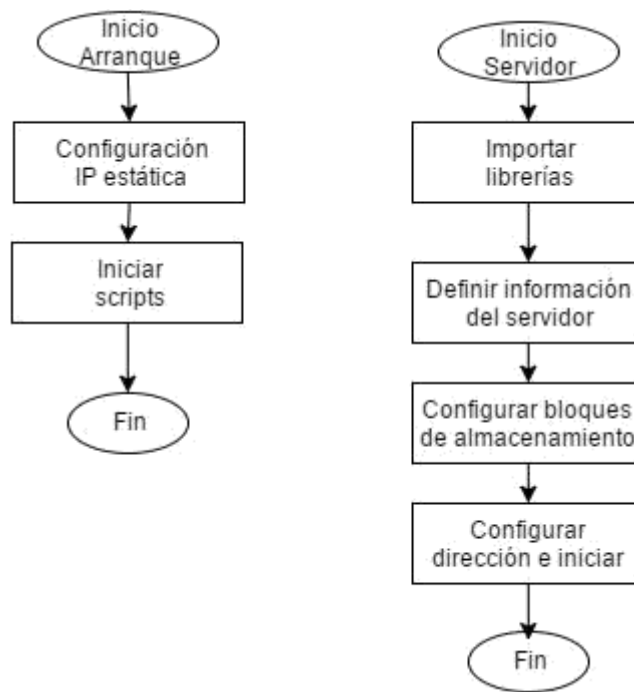


Figura 55. Diagramas de flujo de los scripts secundarios.

5.2 PROGRAMACIÓN DEL SOFTWARE DE VISUALIZACIÓN

5.2.1. LabVIEW y módulos utilizados

Con el software LabVIEW se desarrolló la aplicación HMI a través de la cual se observa el comportamiento de las variables en tiempo real; se realiza el control supervisorio de los controladores; y se activan las alarmas de emergencia. Para facilitar el desarrollo del programa se usó el módulo “Datalogging and Supervisory Control” (DSC) de LabVIEW con el cual se implementó un sistema de control de acceso a través de usuario y contraseña, además de un registro histórico de datos en el cual se almacena la información de los parámetros de los gases.

El módulo DSC de LabVIEW amplía los beneficios de la programación gráfica al desarrollo de aplicaciones de registro de datos y control supervisorio. Se pueden establecer comunicaciones con controladores lógicos programables o controladores de automatización industriales; registrar datos a bases de datos;

administrar alarmas y eventos; y crear interfaces humano máquina con más detalles y herramientas [44].

5.2.2. Descripción del programa

Al iniciar el programa se observa la pantalla de administrador de usuarios, en ésta se solicita un nombre de usuario y una contraseña para poder ingresar al resto de la aplicación, a través de esta ventana también se tiene un botón para apagar el sistema embebido y salir del programa. Al iniciar sesión de forma exitosa se habilitan dos ventanas nuevas, la primera es para monitorear el estado de los gases en tiempo real y realizar las acciones de control supervisorio; y la segunda es para acceder al histórico de datos.

Las acciones de control supervisorio se ejecutan a través de unas herramientas llamadas controles por las cuales se ingresan datos al programa, se tiene un control numérico y uno de selección por cada controlador de flujo. Por medio del control numérico se fija el punto de consigna de cada controlador, este parámetro está limitado entre 0 y 1 “Standard Litters Per Minute” (SLPM) debido al rango de operación del instrumento; para interactuar con el control se pueden utilizar las flechas de incremento y decremento de 0.001 SLPM o ingresar un valor que este entre el rango especificado. El control de selección se utiliza para escoger el gas que se va a monitorear, al pulsar esta herramienta se despliega una lista de los gases a escoger.

El monitoreo en tiempo real se realiza a través de tres gráficas, que se utilizan para la presión, la temperatura y el flujo. En esta sección se disponen de cuatro botones de visibilidad por cada gráfica, a través de los cuales se puede ocultar la variable del controlador que se desee y dos controles numéricos para manejar la escala de la gráfica. Para el registro de datos se tiene en el historiador una interfaz gráfica compuesta de los mismos elementos.

Para las alarmas se tienen dos indicadores LED por cada controlador de flujo, uno rojo y uno amarillo. El indicador rojo se utiliza para la alarma de presión, cuando se tiene una obstrucción o alguna válvula está cerrada a la salida del controlador de flujo, la presión del gas aumentará considerablemente en muy poco tiempo, lo cual puede ocasionar daños en el equipo. La presión a la que debe trabajar el sistema es 1 atm, que equivale a 14,695 PSI, este valor no varía mucho, si la presión supera los 20 PSI indica que ocurre un problema a la salida del dispositivo, es por ello que si el programa detecta que alguno de los controladores posee un valor de presión mayor a 20 PSI, automáticamente fija un punto de consigna cero para evitar que siga fluyendo gas y asegurar que la presión no continúe aumentando. La alarma no desaparecerá hasta que la obstrucción haya sido eliminada o la válvula haya sido abierta de tal manera que la presión baje. El LED amarillo se utiliza para la alarma de temperatura, si al utilizar el equipo, en la entrada no se tiene suficiente presión de gas o la válvula está cerrada la temperatura del equipo comenzará a subir poco a poco, esto se debe a que el equipo tiene un punto de consigna que no logra alcanzar por lo tanto el dispositivo se fuerza para tratar de llegar al valor asignado. Esta alarma funciona de tal forma que si el punto de consigna no es alcanzado durante un cierto periodo de tiempo, automáticamente coloca un punto de consigna cero para evitar que el equipo se siga forzando y se le notificará al operador que revise la entrada del controlador de flujo. En la figura 56 se observa el diagrama de flujo del programa.

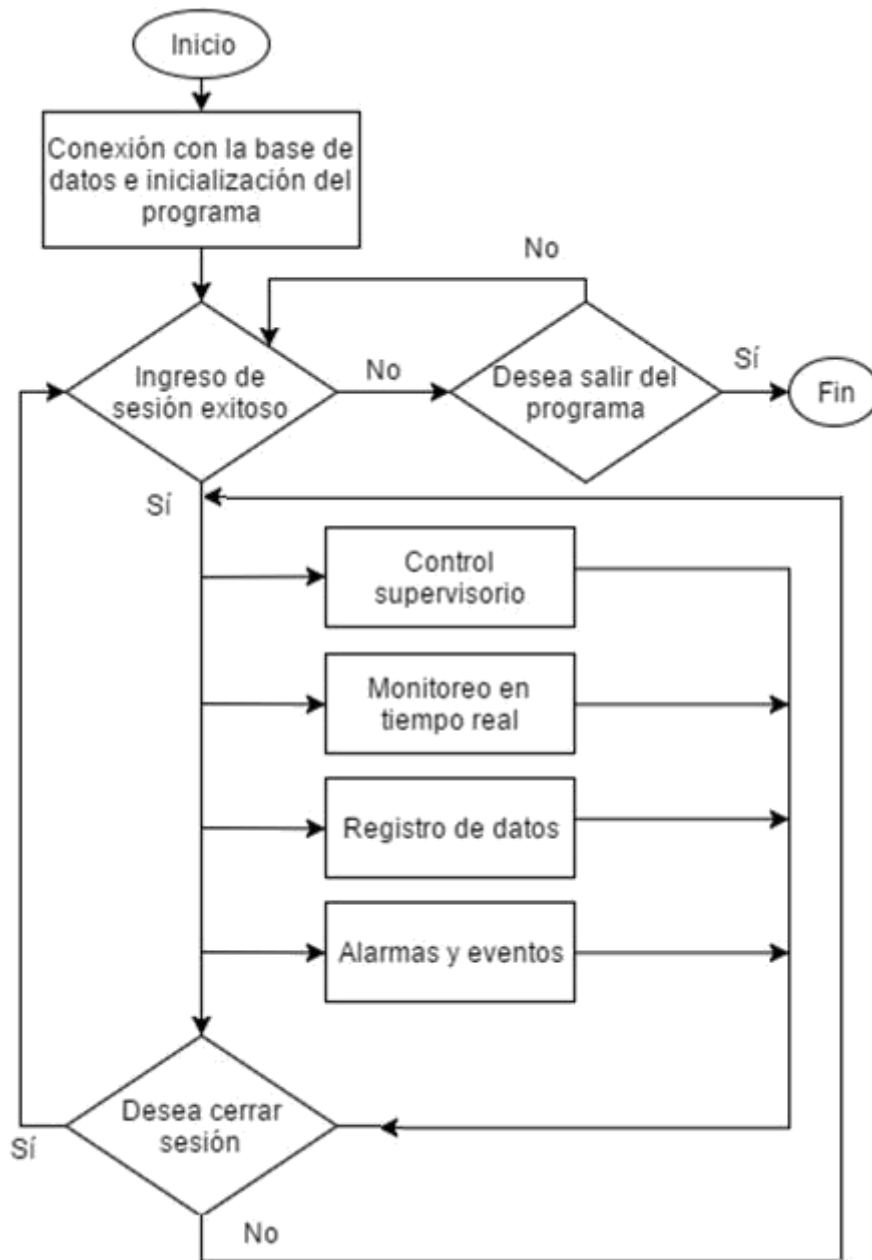


Figura 56. Diagrama de flujo del programa en LabVIEW.

CAPÍTULO VI

VALIDACIÓN Y RESULTADOS

En este capítulo se describen las pruebas realizadas para evaluar el desempeño del sistema SCADA diseñado. La primera prueba que se realizó fue comprobar el funcionamiento del controlador de flujo, para esto se colocó a la salida un flujómetro verificándose que la medición corresponde con el valor de punto de consigna asignado. En la figura 57 se observa la comparación entre el controlador de flujo y el flujómetro.

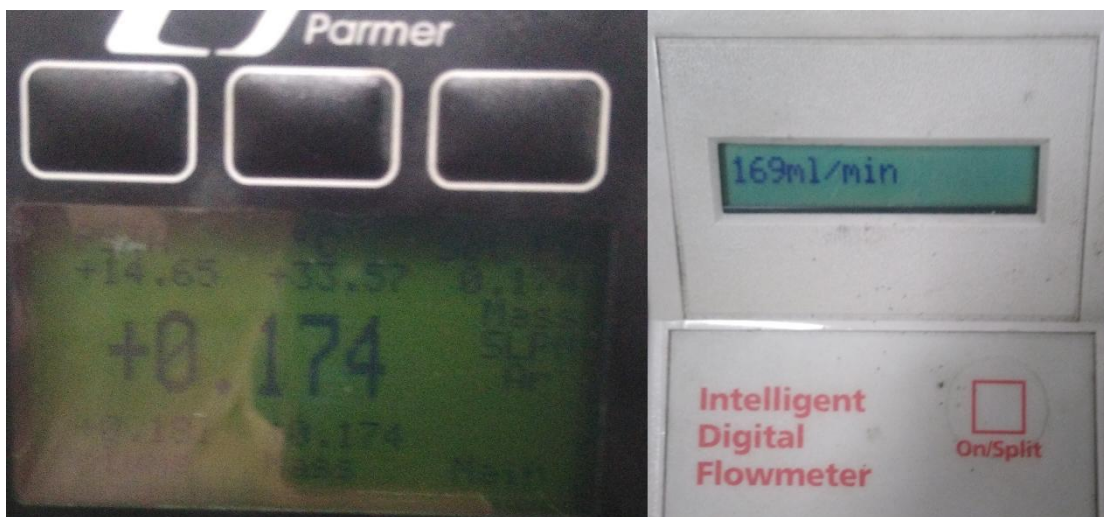


Figura 57. Comparación entre el controlador de flujo y el flujómetro.

Al programa se le realizaron pruebas de funcionamiento aplicadas al sistema embebido y al software de visualización.

6.1. PRUEBAS DE FUNCIONAMIENTO INTEL GALILEO

Para comprobar el funcionamiento del Intel Galileo se crearon varios scripts, los cuales ejecutan partes del programa por separado.

6.1.1. Multiplexor

Este programa realiza el control de los terminales GPIO que seleccionan el canal del multiplexor, haciendo uso de la librería wiring-x86. Para esta prueba se utilizaron dos LED con resistencias de $220\ \Omega$; los LED son conectados a los terminales de salida y encienden siguiendo una secuencia sencilla que simula el multiplexado. En la figura 58 se muestra como los LED encienden para cada caso.

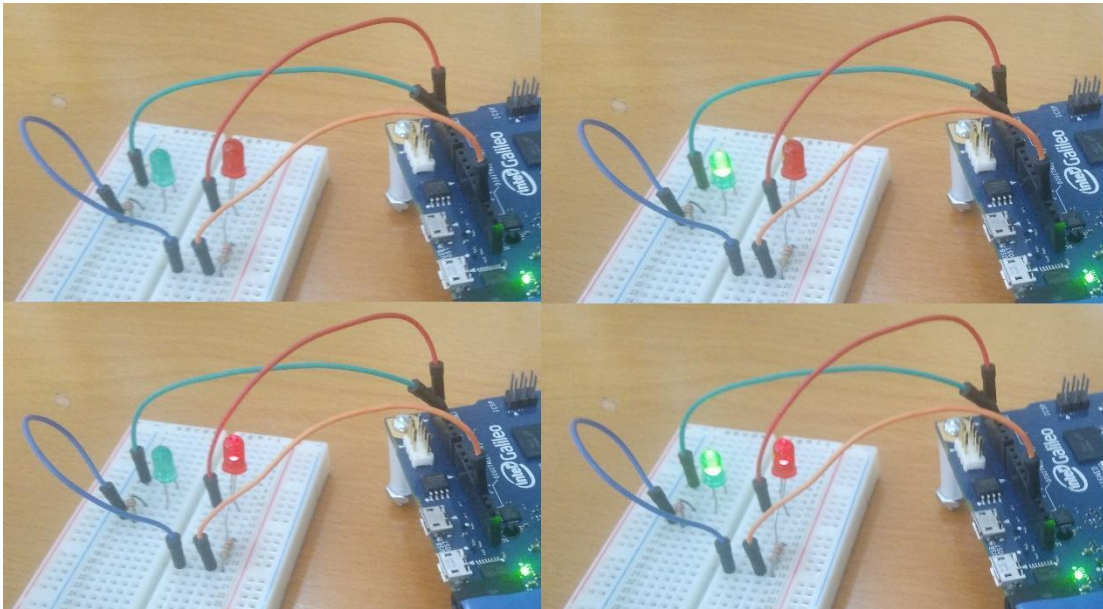


Figura 58. Prueba de funcionamiento de control del multiplexor.

6.1.2. Comunicación serial

El script desarrollado para esta prueba se encarga de configurar los parámetros de la comunicación serial; abrir-cerrar el canal de comunicación; y enviar-

recibir información del Intel Galileo hacia el controlador de flujo, a través de la librería pyserial. La prueba consiste en recibir los datos del controlador de flujo y enviar un punto de consigna cero y cambiar el gas a oxígeno. Como se observa en la figura 59 al recibir los datos del controlador se tiene un punto de consigna de 0,380 y el gas que se monitorea es aire, luego de enviar la información se muestra un punto de consigna 0 y el gas es oxígeno.

```
root@galileo:~# ./pfcomserial.py
Configurando pines
Configurando la comunicacion
Abriendo canal de comunicacion
Seleccin del canal del multiplexor
Leyendo datos
D +014.46 +033.45 +00.000 +00.000 00.380      Air
Escribiendo datos
Confirmando que los datos se escribieron correctamente
D +014.47 +033.45 +00.000 +00.000 00.000      O2
Cerrando canal de comunicacion
```

Figura 59. Prueba de funcionamiento de la comunicación serial.

6.1.3. Flotantes a enteros sin signo

Los datos leídos del controlador de flujo son tipo flotante, para almacenar este tipo de datos se necesita un registro doble Word (32 bits). Los registros de memoria del protocolo Modbus son Word (16 bits), para almacenar una variable tipo doble Word se necesita separarla. Este script ejecuta los comandos necesarios para representar una variable de tipo flotante, como dos variables tipo entero sin signo. Primero se descompone el número flotante en los cuatro bytes que lo conforman; luego se representan los bytes en binario; y al final se agrupan los 16 bits más significativos y menos significativos para formar las dos variables tipo entero sin signo que se almacenarán en los registros Modbus. En la figura 60 se muestran algunas variables tipo flotante que son convertidas en enteros sin signo.

```

root@galileo:~# ./pffloattoint.py
Ingrese numero flotante:1654651.231
Usted ingreso:1654651.231
Flotante dividido en 4 bytes:
[73, 201, 251, 218]
Bytes escritos como binarios:
['01001001', '11001001', '11111011', '11011010']
Enteros sin signo que representan el flotante:
[18889, 64474]
Ingrese numero flotante:30.456
Usted ingreso:30.456
Flotante dividido en 4 bytes:
[65, 243, 165, 227]
Bytes escritos como binarios:
['01000001', '11110011', '10100101', '11100011']
Enteros sin signo que representan el flotante:
[16883, 42467]
Ingrese numero flotante:18.2457
Usted ingreso:18.2457
Flotante dividido en 4 bytes:
[65, 145, 247, 50]
Bytes escritos como binarios:
['01000001', '10010001', '11110111', '00110010']
Enteros sin signo que representan el flotante:
[16785, 63282]

```

Figura 60. Prueba de funcionamiento de la conversión de flotante a entero sin signo.

6.1.4. Comprobación de trama

Para esta prueba se realiza un script utilizando la librería de expresiones regulares. Se crea una expresión regular que mantiene el formato de los datos enviados por los controladores de flujo, si existe cualquier error en la información enviada, se descarta la trama. En la figura 61 se muestra el resultado del script al introducir datos con y sin errores.

```

root@galileo:~# ./pftrama.py
Ingrese trama:B +014.65 +027.95 +00.000 +00.000 00.000 Star29
Usted ingreso:B +014.65 +027.95 +00.000 +00.000 00.000 Star29
Trama correcta
Controlador B
Presion +014.65
Temperatura +027.95
Flujo Volumetrico +00.000
Flujo Masico +00.000
Punto de Consigna 00.000
Gas Star29
Ingrese trama:B +014.65 +027.95+00.000 +00.000 00.000 Star29
Usted ingreso:B +014.65 +027.95+00.000 +00.000 00.000 Star29
La trama se descarto debido a que presenta errores

```

Figura 61. Prueba de funcionamiento de la comprobación de trama.

6.1.5. Modbus

Para llevar a cabo esta prueba se necesitó desarrollar dos scripts. El primero se encarga de la configuración e inicio del servidor Modbus TCP y el segundo se conecta al servidor y realiza acciones de escritura y lectura en los bloques de memoria. La prueba consiste en conectarse al servidor, leer la información del registro 4000001 y la bobina 000001 y sobrescribir dichos datos con 62492 y True respectivamente, al final se cierra la conexión con el servidor. En la figura 62 se observan los resultados.

```

root@galileo:~# ./pfmodbus.py
Conectando al Servidor ModbusTcp
Leer registro 400001
0
Escribir en el registro 400001 el valor de 62492
Confirmar que se escribió correctamente
62492
Leer bobina 000001
False
Escribir en la bobina 000001 nivel logico alto (1)
Comprobar que se escribió correctamente
True
Cerrando la conexión con el Servidor ModbusTcp

```

Figura 62. Prueba de funcionamiento del servidor y cliente Modbus TCP.

6.1.6. Apagar el sistema embebido

Para apagar el sistema embebido a través de la aplicación HMI se utiliza la bobina 000001 de los bloques de memoria del protocolo Modbus. Si se detecta un nivel de True en dicha bobina se ejecuta un comando en la Shell de Linux que apaga el sistema. La ejecución de comandos de Linux a través de Python se realiza por medio de la librería `commands`. En esta prueba se tiene una variable llamada `cont` que se utiliza para escribir True en la bobina 000001 y forzar el apagado cuando esta tiene un valor de 3. En la figura 63 se observa como el sistema se apaga de forma exitosa.

```
root@galileo:~# ./pfapagar.py
Conectando al Servidor ModbusTcp
Escribir True en la bobina 000001 cuando la variable cont sea igual a 3
False
1
False
2
False
3
True
Cerrando conexion con el Servidor ModbusTcp
Apagando sistema embebido

Broadcast message from root@galileo (Wed 2016-08-17 22:23:11 UTC):
The system is going down for power-off at Wed 2016-08-17 22:24:11 UTC!
```

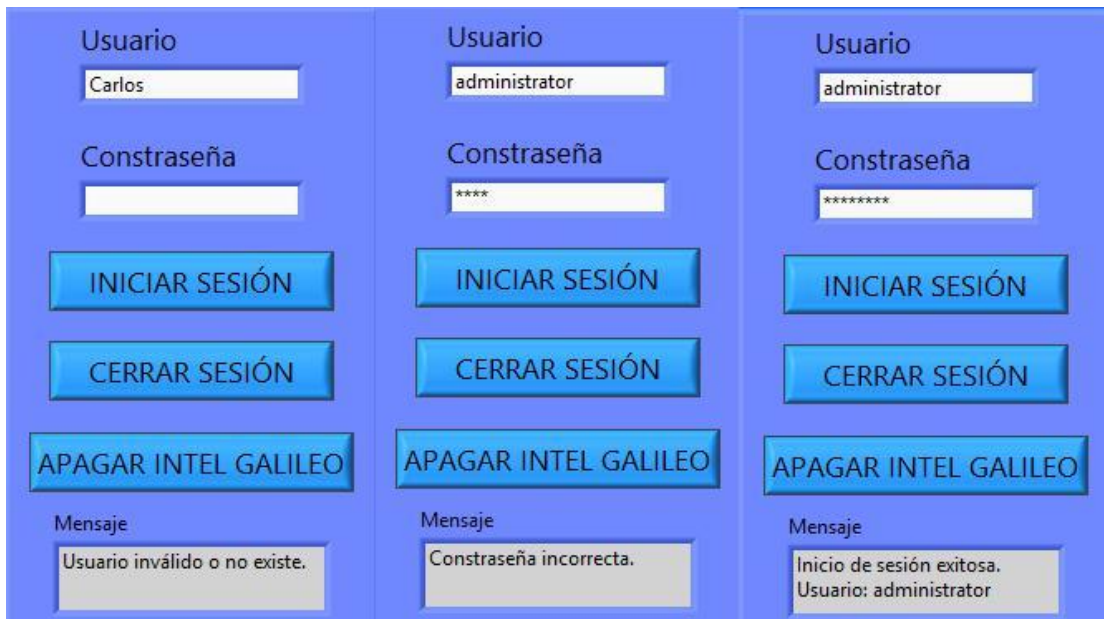
Figura 63. Prueba de funcionamiento apagar el sistema embebido.

6.2. PRUEBAS DE FUNCIONAMIENTO LABVIEW

Una vez desarrollado el programa del Intel Galileo se realizan las pruebas de funcionamiento de LabVIEW. Para estas pruebas no se separó el código en partes tal y como se hizo antes, si no que se integró todo el programa junto con el de Intel Galileo y se comprobó su funcionamiento trabajando en conjunto.

6.2.1. Inicio de sesión

Esta prueba consiste en verificar los casos que pueden ocurrir al iniciar sesión, tales como, usuario inválido o no existente; contraseña incorrecta e inicio de sesión exitoso. En la figura 64 se observan los tres casos mencionados con su respectivo mensaje.



The figure displays three side-by-side screenshots of a login interface with a blue background. Each screenshot shows a form with the following elements: a 'Usuario' (User) input field, a 'Contraseña' (Password) input field, three blue buttons labeled 'INICIAR SESIÓN', 'CERRAR SESIÓN', and 'APAGAR INTEL GALILEO', and a 'Mensaje' (Message) output field. The first screenshot shows the user 'Carlos' and an empty password field, with the message 'Usuario inválido o no existe.' (Invalid user or does not exist). The second screenshot shows the user 'administrator' and a password field with four asterisks, with the message 'Contraseña incorrecta.' (Incorrect password). The third screenshot shows the user 'administrator' and a password field with seven asterisks, with the message 'Inicio de sesión exitosa. Usuario: administrator' (Successful login. User: administrator).

Figura 64. Prueba de funcionamiento de inicio de sesión.

6.2.2. Interfaz humano máquina

En esta prueba se muestra la interfaz gráfica con la que el usuario interactúa para fijar el punto de consigna o cambiar el gas que se controla, también se observa la información adquirida por los controladores de flujo. En la figura 65 se observa la interfaz gráfica mencionada.



Figura 65. Prueba de funcionamiento de la interfaz humano máquina.

6.2.3. Monitoreo en tiempo real

En esta prueba se observó el funcionamiento de la gráfica a través de la cual se monitorea el comportamiento de la presión, la temperatura y el flujo en tiempo real. En la figura 66 se muestran los resultados obtenidos.

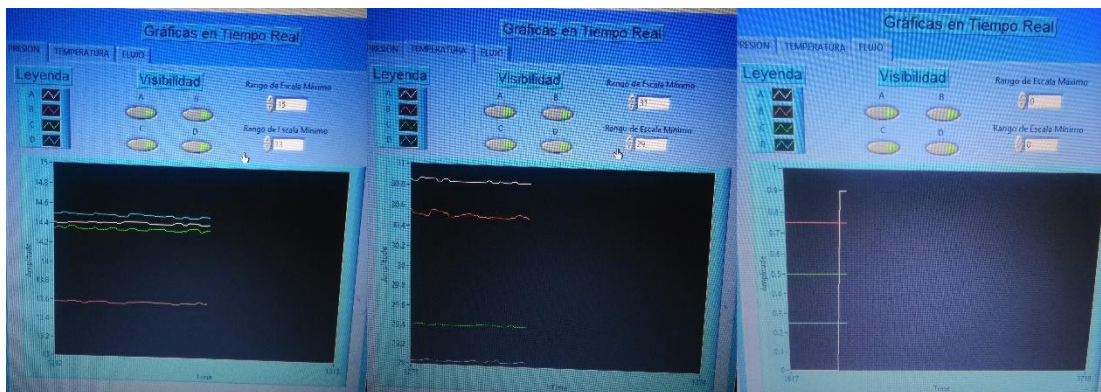


Figura 66. Prueba de funcionamiento de la monitorización en tiempo real.

6.2.4. Alarma de emergencia a la salida del controlador

Para esta prueba se simuló una falla a la salida del controlador fijando un valor de presión mayor a la tolerancia establecida en el sistema. En la figura 67 se observa el correcto funcionamiento de la alarma.

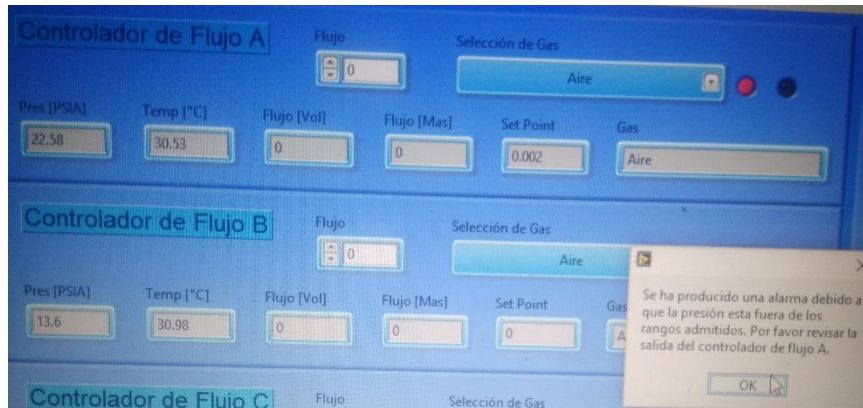


Figura 67. Prueba de funcionamiento de la alarma a la salida.

6.2.5 Alarma de emergencia a la entrada del controlador

Para esta prueba se colocó un punto de consigna de 0.002, con la válvula de entrada cerrada y sin presión de gas, para verificar el funcionamiento de la alarma de entrada. En la figura 68 se observan los resultados obtenidos.

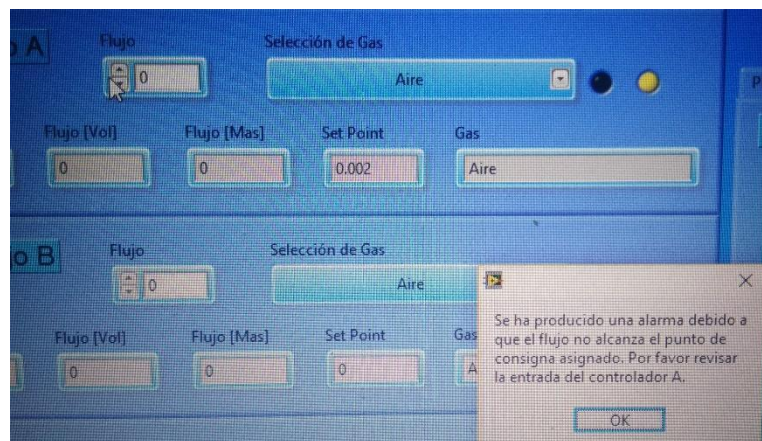


Figura 68. Prueba de funcionamiento de la alarma a la entrada.

6.2.6. Registro de datos

Para esta prueba se colocó el sistema embebido trabajando junto con la HMI durante aproximadamente una hora mientras los datos leídos se registraban en una gráfica. En la figura 69 se observan los resultados obtenidos.

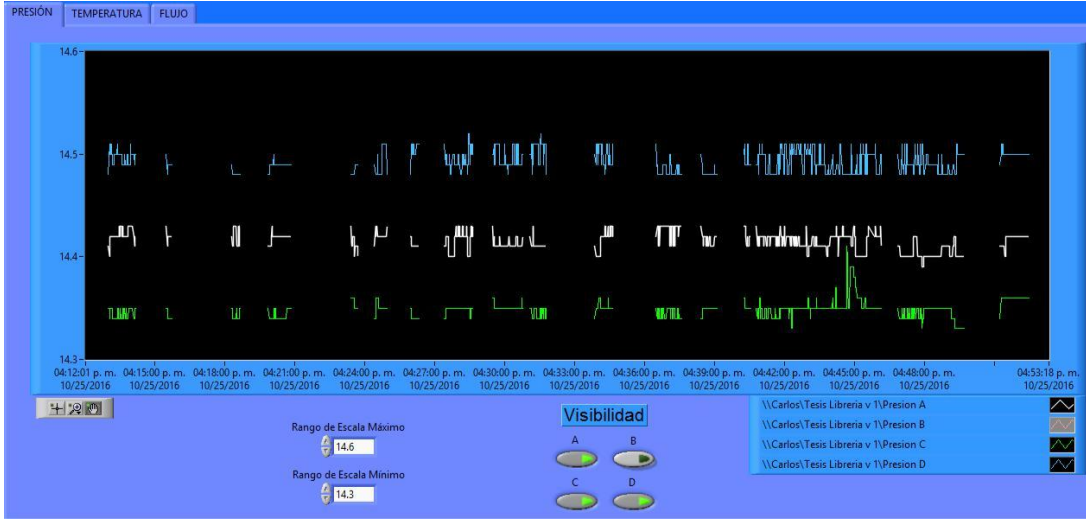


Figura 69. Prueba de funcionamiento del registro de datos.

CONCLUSIONES

En base a los objetivos planteados se obtuvieron las siguientes conclusiones.

El desarrollo del sistema SCADA mejoró la eficiencia, modernización y control de la planta, otorgándole características como monitoreo de variables en tiempo real, control de acceso, alarmas de emergencia y registro de datos, a través de una interfaz humano máquina sencilla.

Se determinó que los sistemas embebidos son una solución viable para realizar la adquisición de datos de un proceso. Representan una alternativa más económica que los controladores industriales y resultan adecuados para el desarrollo de sistemas SCADA de procesos a escala, que comúnmente se llevan a cabo en laboratorios.

Se logró el envío y recepción de datos desde una computadora hacia los controladores de flujo utilizando el protocolo de comunicación Modbus TCP y el estándar de comunicación serial RS-232.

A través del diseño y la elaboración de un circuito impreso se logró establecer la comunicación con los cuatro controladores de flujo.

Se evidenció la capacidad del software LabVIEW para desarrollar una interfaz humano máquina sencilla que proporciona una excelente visualización del proceso y a través de la cual se pueden realizar acciones de control supervisorio.

A partir del módulo DSC de LabVIEW se logró implementar el control de acceso y el registro de datos, características necesarias en cualquier sistema SCADA.

Se demostró el potencial del Intel Galileo para el control y la adquisición de datos y al ser programado con Python ofrece una gran cantidad de herramientas, las cuales facilitaron el desarrollo del sistema y la integración de todas las partes.

Se realizó un manual de usuario del sistema el cual se encuentra en el anexo 1, y los instructivos de código fuente de los programas desarrollados en el Intel Galileo y en LabVIEW, para mayor información dirigirse a los anexos 2 y 3 respectivamente.

RECOMENDACIONES

A continuación se presentan un conjunto de recomendaciones sobre el trabajo realizado.

Se recomienda agregar relés para el control del encendido y apagado de los controladores de flujo.

Diseñar un circuito a través del cual se encienda el Intel Galileo sin necesidad de desconectar y conectar la alimentación.

Implementar un sistema de alimentación ininterrumpida, para apagar el sistema correctamente en caso de que falle el servicio eléctrico.

Se sugiera la capacitación del personal operador sobre el funcionamiento del sistema para lograr un uso adecuado.

REFERENCIAS BIBLIOGRÁFICAS

[1] Jaimes L., Jorge E. “Implementación de los lazos de control de flujo y temperatura del bloque de acondicionamiento de la planta de reformado de metano a escala banco ubicada en el Laboratorio de Petroquímica de la Escuela de Ingeniería Química. Universidad Central de Venezuela”. (Tesis). Caracas: Universidad Central de Venezuela 2013.

[2] National Instruments, “Diseño de Sistemas SCADA para Monitoreo de Procesos Utilizando LabVIEW DSC” [en línea].
<ftp://ftp.ni.com/pub/branches/latam/nidays_2006/Diseno%20de%20Sistemas%20SCADA%20para%20Monitoreo%20de%20Procesos.pdf> [Consultado junio 2015].

[3] Schneider Electric, “¿Cuál es la diferencia entre SCADA y HMI?” [en línea].
<<http://www.indusoft.com/blog/2013/05/31/cual-es-la-diferencia-entre-scada-y-hmi/>> [Consultado junio 2015].

[4] Aquilino Rodríguez Penin,(2012) “Sistemas SCADA”, segunda edición, editorial marcombo [en línea]. <<http://www.marcombo.com/Descargas/8426714188-SCADA/CAP%C3%8DTULO%20I.pdf>> [Consultado junio 2015].

[5] Steven D. Garbrecht, “Los beneficios de las arquitecturas basadas en objetos para los sistemas de supervisión y SCADA” [en línea].
<http://campaigns.wonderware.com/campaigns/Documents/IN-0222-LA_WhitePaper_Invensys_BenefitsOfObject-BasedArchitecturesForSCADA_03-12.pdf> [Consultado junio 2015].

[6] Olejnik A., Ricardo A., “Estudio del reformado autotérmico de metano empleando catalizadores estructurados de níquel soportados sobre alúmina para la producción de gas de síntesis”. (Tesis). Caracas: Universidad Central de Venezuela 2010.

[7] Hollenthoner Stefan, (2007). “Micro-Processors and Micro-Controllers”, editorial IAEA Laboratories Seibersdorf [en línea].

<http://www.inin.gob.mx/mini_sitios/documentos/ATMega8.pdf> [Consultado octubre 2016].

[8] Mazidi Muhammad & Mazidi Janice, (2006). “The 8051 microcontroller and embedded systems”, Séptima edición, editorial Pearson Education Capítulo 1 [en línea]. <<http://ee.sharif.edu/~sakhtar3/books/mazidi/ch1.pdf>> [Consultado octubre 2016].

[9] Heath Steve, (2002). “Embedded Systems Design”, segunda edición, editorial Butterworth-Heinemann [en línea]. <<http://read.pudn.com/downloads158/ebook/707037/Embedded%20Systems%20Design%20-%20ed%20-%200750655461.pdf>> [Consultado octubre 2016].

[10] Axelson Janet, (2007), “Serial Port Complete”, Segunda Edición, Editorial Lakeview Research [en línea]. <https://www.8051projects.net/files/public/1347193247_22104_FT57363_serial-port-complete-com-ports-usb-virtual-com-ports-and-ports-for-embedded-systems-complete-guides-series.9781931448062.31506.pdf> [Consultado octubre 2016].

[11] Intel Corporation, “Intel Galileo Gen 1 Development Board Datasheet” [en línea]. <<http://www.intel.la/content/www/xl/es/embedded/products/galileo/galileo-g1-datasheet.html>> [Consultado octubre 2016].

[12] Intel Corporation, “Set Up Your IDE” [en línea]. < <https://software.intel.com/es-es/get-started-galileo-windows-step5>> [Consultado octubre 2016].

[13] Arduino, “Arduino Software (IDE)” [en línea]. <<https://www.arduino.cc/en/Guide/Environment#toc10>> [Consultado octubre 2016].

[14] Intel Corporation, “Intel Galileo Board Assembly using Arduino” [en línea]. <<https://software.intel.com/en-us/articles/intel-galileo-board-assembly>> [Consultado octubre 2016].

[15] Intel Corporation, “Step 1: Make a bootable micro SD card” [en línea]. <<https://software.intel.com/es-es/get-started-galileo-windows-step1>> [Consultado octubre 2016].

[16] Intel Corporation, “Developing Internet of Things Projects with the Intel XDK” [en línea]. <<https://software.intel.com/en-us/developing-iot-projects-with-intel-xdk>> [Consultado octubre 2016].

[17] Intel Corporation, “Intel System Studio IoT Edition User Guide for C/C++” [en línea]. <<https://software.intel.com/en-us/intel-system-studio-iot-edition-guide-for-c>> [Consultado octubre 2016].

[18] Python, “General Python FAQ” [en línea]. <<https://docs.python.org/3/faq/general.html#id1>> [Consultado octubre 2016].

[19] National Instruments, “Información Detallada sobre el Protocolo Modbus” [en línea]. <<http://www.ni.com/white-paper/52134/es>> [Consultado octubre 2016].

[20] National Instruments, “¿Qué puedo hacer con LabVIEW?” [en línea]. <<http://www.ni.com/labview/esa/>> [Consultado octubre 2016].

[21] ¿Por qué LabVIEW? Consultado en octubre del 2016. Disponible en: <http://www.ni.com/labview/why/esa/>

[22] National Instruments, “Áreas de Aplicación” [en línea]. <<http://www.ni.com/labview/applications/esa/>> [Consultado octubre 2016].

[23] Cole-Parmer. (2005). Precision Gas Flow Controller Operating Manual 16 Series. Caracas. Universidad central de Venezuela. Manual de Operación de los Controladores de Flujo.

- [24] National Instruments, “NI FP-1001” [en línea].
<<http://sine.ni.com/nips/cds/view/p/lang/es/nid/2181> > [Consultado octubre 2016].
- [25] National Instruments “Digital Output Modules for Compact FieldPoint and FieldPoint” [en línea].< http://www.ni.com/pdf/products/us/4dio569_571.pdf> [Consultado octubre 2016].
- [26] Raspberry Pi Foundation, “Raspberry Pi 1 modelo B+” [en línea]. <<https://cdn-shop.adafruit.com/datasheets/pi-specs.pdf>> [Consultado octubre 2016].
- [27] Intel Corporation, “Intel Galileo Gen 1 Development Board Datasheet” [en línea]. <<http://www.intel.la/content/www/xl/es/embedded/products/galileo/galileo-g1-datasheet.html>> [Consultado octubre 2016].
- [28] Gerald Coley, “BeagleBone Black System Reference Manual”, Abril 2013 [en línea]. <https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf> [Consultado octubre 2016].
- [29] National Instruments, “RS-232, RS-422, RS-485 Serial Communication General Concepts” [en línea]. < <http://www.ni.com/white-paper/11390/en> > [Consultado octubre 2016].
- [30] Texas Instruments, “MAX232 datasheet” [en línea].
<<http://www.ti.com/lit/ds/symlink/max232.pdf>> [Consultado junio 2016].
- [31] Texas Instruments, “CD4052B datasheet” [en línea].
<<http://www.ti.com/lit/ds/symlink/cd4053b.pdf>> [Consultado junio 2016].
- [32] Siemens, “Simatic WinCC” [en línea].
<http://www.automation.siemens.com/salesmaterial-as/brochure/en/df_fa_i10077-00-7600_ipdf_wincc_systemoverview_en.pdf> [Consultado octubre 2016].

- [33] Rockwell Automation, “Factory Talk View SE” [en línea]. <<http://www.rockwellautomation.com/rockwellsoftware/products/factorytalk-view-se.page?#overview>> [Consultado octubre 2016].
- [34] National Instruments, “Software de Desarrollo de Sistemas NI LabVIEW” [en línea]. <<http://www.ni.com/labview/esa/>> [Consultado octubre 2016].
- [35] Fernando Morilla García, “Controladores PID”, marzo 2007 [en línea]. <http://www.dia.uned.es/~fmorilla/MaterialDidactico/ajuste_frecuencia.pdf> [Consultada en noviembre 2016].
- [36] Middle East Technical University, “P, PD, PI, PID CONTROLLERS” [en línea]. <file:///C:/Users/Carlps/Downloads/EE402RecitationReport_4.pdf> [Consultada noviembre 2016].
- [37] Javier Barragán Piña, “Modbus TCP”, junio 2013 [en línea]. <<http://uhu.es/antonio.barragan/content/modbus-tcp>> [Consultada octubre 2016].
- [38] Siemon “Ethernet/IP” [en línea]. <https://www.siemon.com/la/white_papers/03-10-13-ethernet-ip.asp> [Consultada octubre 2016].
- [39] Red Hat Enterprise Linux “Protocolo SSH” [en línea]. <<http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>> [Consultada octubre 2016].
- [40] GitHub, Inc, “Wiring-x86” [en línea]. <<https://github.com/emutex/wiring-x86>> [Consultada octubre 2016].
- [41] Chris Liechti, “Pyserial Documentation” Release 3.2.1. [en línea]. <<https://media.readthedocs.org/pdf/pyserial/stable/pyserial.pdf>> [Consultada octubre 2016].
- [42] GitHub, Inc, “Pymodbus” [en línea]. <<https://github.com/bashwork/pymodbus/blob/master/README.rst>> [Consultada octubre 2016]

[43] Python software foundation, “Regular expression operations” [en línea].
<<https://docs.python.org/2/library/re.html>> [Consultada octubre 2016].

[44] National Instruments Corporation, “Módulo LabVIEW . Datalogging and Supervisory Control (DSC)” [en línea]
<<http://www.ni.com/labview/labviewdsc/esa/>>. [Consultada octubre 2016].

ANEXOS