

COMPUTADOR DE FLUJO: DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE
UN PROTOTIPO PARA MEDICIÓN DE FLUJO DE GAS

EDGAR JAVIER BARAJAS HERRERA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
MAESTRÍA EN POTENCIA ELÉCTRICA
BUCARAMANGA
2004

COMPUTADOR DE FLUJO: DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE
UN PROTOTIPO PARA MEDICIÓN DE FLUJO DE GAS

ING. EDGAR JAVIER BARAJAS HERRERA

Proyecto de Grado presentado como requisito para optar el título de
Magíster en Potencia Eléctrica Línea de Investigación Instrumentación
Electrónica

Director:
MSC. JAIME GUILLERMO BARRERO PÉREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
MAESTRÍA EN POTENCIA ELÉCTRICA
BUCARAMANGA
2004

DEDICATORIA

A Dios Todopoderoso, mi protector.

*A mi madre Hortensia, quien fue el verdadero apoyo
en los momentos difíciles de este camino.*

A mi padre Claudio, por su confianza.

*A mis hermanos, César Augusto y Guillermo Andrés,
por su ánimo incondicional.*

A mi familia, razón de mi existencia.

*A mis amigos Gladys, Fábian, y Sergio,
por su paciencia.*

*A Jaime, por su enseñanza, y aporte profesional,
para hacer de este trabajo lo mejor.*

A Isaac y Nelfor, por su colaboración desinteresada.

AGRADECIMIENTOS

Este trabajo fue posible gracias al Centro de Investigación del Gas, en cabeza de su director Julio César Pérez, y al aporte investigativo y profesional del grupo de ingenieros: Nelfor Castelblanco, Ricardo González, Jaime Barrero, Luis Gómez, Mónica Corzo, e Isaac Luque. También fue fundamental la asesoría prestada por el ingeniero Rafael Roncancio, y el instrumentista Pablo Arias de la Empresa Colombiana de Gas.

CONTENIDO

TABLAS.....	8
FIGURAS	9
INTRODUCCIÓN.....	13
1. SISTEMA DE MEDICIÓN DE FLUJO DE GAS.	16
1.1. INSTRUMENTACIÓN PRIMARIA: MEDIDOR TIPO TURBINA.	18
1.1.1. Medidor tipo turbina para gases.	18
1.1.2. Principio de operación.	19
1.1.3. Sensores de la velocidad del flujo.	19
1.1.4. Instalación del medidor.	20
1.1.5. Características de funcionamiento de la turbina.....	23
1.1.6. Ventajas y Desventajas del medidor tipo turbina.	24
1.1.7. Turbina Elster.	25
1.2. NORMAS INTERNACIONALES DE MEDICIÓN DE FLUJO VOLUMÉTRICO DE GAS NATURAL.	25
1.2.1. Medición de gas combustible con medidores de turbina – AGA7.	26
1.2.2. Comprensibilidad y supercomprensibilidad para gas natural – AGA8.	28
1.2.3. Estándar API Capítulo 21.	34
1.3. INSTRUMENTACIÓN SECUNDARIA: SENSORES.	36
1.3.1. Detectores Resistivos de Temperatura.	36
1.3.2. Transmisores de presión.	37
1.4. INSTRUMENTACIÓN TERCIARIA: COMPUTADOR DE FLUJO.	37
1.4.1. Evolución tecnológica de los computadores de flujo.	38
1.4.2. Computador de flujo análogo.	38
1.4.3. Computador de flujo digital.	38
1.4.4. Estado del arte de los computadores de flujo.	41
1.4.5. Generalidades en el diseño de un computador de flujo.	44
2. <i>HARDWARE</i> DEL COMPUTADOR DE FLUJO.	46
2.1. UNIDAD DE MEDICIÓN	46
2.1.1. Señal del medidor primario.	48
2.1.2. Selección de la instrumentación.	48
2.1.3. Acondicionamiento de señales.	52

2.2. UNIDAD DE PROCESAMIENTO, CONTROL Y ALMACENAMIENTO	70
2.2.1. Parámetros de selección.	70
2.2.2. Sistema embebido.	74
2.3. INTEGRACIÓN DEL COMPUTADOR DE FLUJO.	82
2.3.1. Encapsulado.	82
2.3.2. Computador de flujo para medidor turbina.	82
2.3.3. Especificaciones y Costo Total.	83
3. FIRMWARE DEL COMPUTADOR DE FLUJO.	85
3.1. GENERALIDADES DE <i>FORTH</i>.	85
3.1.1. Programación.	86
3.1.2. Experiencias de <i>Forth</i> en la industria.	88
3.1.3. Algunas palabras <i>Forth</i> (comandos).	88
3.1.4. Ejemplos <i>Forth</i> versus C.	91
3.2. IMPLEMENTACIÓN DE LAS NORMAS AGA.	97
3.2.1. AGA8 – GROSS 2.	97
3.2.2. Norma AGA-7.	109
3.3 DESARROLLO DEL FIRMWARE	110
3.3.1. Tarea de Fondo.	113
3.3.2. Tarea de Primer Plano.	115
3.3.3. Rutina de servicio de interrupción.	116
3.3.4. Especificaciones del <i>firmware</i> realizado.	119
4. PRUEBAS DEL PROTOTIPO EN CAMPO.	121
4.1. DETERMINACIÓN DEL ERROR MÁXIMO EN LA INSTRUMENTACIÓN.	121
4.1.1. Variación de una unidad de presión y temperatura.	121
4.1.2. Error máximo del sensor de temperatura.	122
4.1.3. Error máximo del sensor de presión.	122
4.1.4. Error total en la instrumentación.	123
4.2. CALIBRACIÓN DE LOS SENSORES EN LABORATORIO.	123
4.3 PRUEBA EN CAMPO DEL PROTOTIPO.	126
4.3.1. Descripción de la prueba.	126
4.3.2. Resultados de la prueba.	127
4.3.3. Observaciones y Conclusiones de la Prueba.	137
CONCLUSIONES	139
RECOMENDACIONES	140
BIBLIOGRAFÍA	141

TABLAS

Tabla 1. Características medidor tipo turbina.	18
Tabla 2. Ventajas y desventajas del medidor turbina.	25
Tabla 3. Turbina Elster.	25
Tabla 4. Comparación NX-19 vs AGA8.	29
Tabla 5. Rango de Aplicación del Método AGA8 Gross.	30
Tabla 6. Coeficientes viriales B_j para Nitrógeno y Dióxido de Carbono.	32
Tabla 7. Coeficientes viriales C_{ijk} para Nitrógeno y Dióxido de Carbono.	33
Tabla 8. Clasificación de las zonas industriales.	47
Tabla 9. Sensibilidad de la lectura de presión.	66
Tabla 10. Costos de la tarjeta de acondicionamiento.	68
Tabla 11. Resolución vs bits.	70
Tabla 12. Consumo de la tarjeta de acondicionamiento, instrumentación y <i>display</i>	72
Tabla 13. Comparación sistemas de desarrollo.	73
Tabla 14. Especificaciones del Prototipo.	83
Tabla 15. Costo Final del Prototipo.	84
Tabla 16. Secuencia de operaciones de un programa en <i>Forth</i>	96
Tabla 17. Comparación de Resultados del AGA8.	108
Tabla 18. Efectos de los errores de presión y temperatura.	122
Tabla 19. Efectos de la temperatura sobre la medición de volumen.	122
Tabla 20. Efectos de la presión sobre la medición de volumen.	123
Tabla 21. Error en volumen al combinar errores en presión y temperatura.	123
Tabla 22. Resultados calibración de temperatura.	124
Tabla 23. Resultados calibración de presión.	125
Tabla 24. Presión promedio.	128
Tabla 25. Temperatura promedio.	129
Tabla 26. Volumen día 1.	130
Tabla 27. Volumen día 2.	131
Tabla 28. Volumen día 3.	133
Tabla 29. Volumen día 4.	134
Tabla 30. Volumen día 5.	136
Tabla 31. Volumen corregido diario.	137

FIGURAS

Figura 1. Medidor tipo turbina.....	18
Figura 2. Corte transversal de un medidor de turbina.	19
Figura 3. Instalación recomendada para medidor turbina.	21
Figura 4. Detector resistivo de temperatura.....	37
Figura 5. Transductores de presión.	37
Figura 6. RTD PT100, marca <i>FW Murphy</i>	50
Figura 7. Sensor de presión <i>Barksdale</i>	52
Figura 8. Diagrama de bloques de la tarjeta de acondicionamiento de señal.	53
Figura 9. Diagrama de bloques convertidor del LT1054.	54
Figura 10. Alimentación de -12V usando el LT1054.	54
Figura 11. Referencia de tensión REF102 <i>Burr – Brown</i>	55
Figura 12. Circuito para la obtención de la alimentación de +10V.	56
Figura 13. Referencia de corriente REF200 de <i>Burr-Brown</i>	56
Figura 14. Conexión de la RTD con el integrado REF200.	57
Figura 15. Diagrama de bloques del INA118 <i>Burr – Brown</i>	59
Figura 16. Esquemático del 4N26.	60
Figura 17. Circuito para entrada de pulsos.	62
Figura 18. Acondicionador de señal para un RTD Pt100.	64
Figura 19. Bornes típicos de un RTD.....	64
Figura 20. Acondicionamiento de los transductores de presión.	65
Figura 21. Diagrama de bloques del ADS7825 <i>Burr – Brown</i>	67
Figura 22. Conexiones del ADS7825.	68
Figura 23. Tarjeta de acondicionamiento de señal.	69
Figura 24. Diagrama de las entradas y salidas de la tarjeta de acondicionamiento.	69
Figura 25. Sistema de desarrollo TDS2020F y sus partes.	75
Figura 26. Conexión del teclado.....	78
Figura 27. Pines de la tarjeta TDS2020F.	80
Figura 28. <i>Kit</i> de <i>software</i> y <i>hardware</i> del TDS2020F.	81
Figura 29. Vista exterior del computador de flujo.	82
Figura 30. Vista interior del computador de flujo.	83
Figura 31. Diagrama de Flujo para hallar H_{CH}	100
Figura 32. Diagrama de Flujo AGA8 Gross-2.....	101
Figura 33. Diagrama de Flujo Rutinas AGA8 Gross-2.	103
Figura 34. Diagrama de Flujo del <i>Firmware</i> del computador de flujo.	112
Figura 35. Instalación de los computadores de flujo para la prueba.	126
Figura 36. Presión promedio.	128
Figura 37. Temperatura promedio.	129
Figura 38. Volumen corregido día 1.	130
Figura 39. Volumen no corregido día 1.	130
Figura 40. Volumen corregido día 2.	132
Figura 41. Volumen no corregido día 2.	132
Figura 42. Volumen corregido día 3.	133
Figura 43. Volumen no corregido día 3.	134
Figura 44. Volumen corregido día 4.	135
Figura 45. Volumen no corregido día 4.	135
Figura 46. Volumen corregido día 5.	136
Figura 47. Volumen no corregido día 5.	136
Figura 48. Volumen corregido diario.	137

ANEXOS

ANEXO A. CODIGO EN MATLAB DEL AGA8-GROSS2	142
ANEXO B. CODIGO DEL AGA8- GROSS2 EN FORTH.....	152
ANEXO C. SWITCHED-CAPACITOR VOLTAGE CONVERTERS.....	168
ANEXO D. PRECISION VOLTAGE REFERENCE	169
ANEXO E. SISTEMA EMBEBIDO TDS2020F	170
ANEXO F. CODIGO DEL FIRMWARE FINAL EN FORTH.....	179
ANEXO G. MANUAL DE INSTALACIÓN Y OPERACIÓN ECUFLOW2020T	196

RESUMEN

TITULO: COMPUTADOR DE FLUJO: DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN PROTOTIPO PARA MEDICIÓN DE FLUJO DE GAS.

AUTOR: BARAJAS HERRERA, Edgar Javier

PABALABRAS CLAVES: computador de flujo, medición, instrumentación electrónica, medidor turbina, AGA8, AGA7, sistema embebido, forth.

DESCRIPCIÓN:

En el transporte de gas es necesario disponer de un sistema de medición eficiente y confiable de los volúmenes que se entregan y reciben en los procedimientos de compra y venta del producto. La medición realizada por los medidores primarios (turbina) no representa un valor real, debido a las variaciones de temperatura y presión que sufre el gas durante el proceso. Para corregir estos errores, se emplean dispositivos de alta precisión que se colocan en complemento de los medidores primarios. Estos equipos se denominan **computadores de flujo**, los cuales monitorean el cambio en presión y temperatura, calculan la densidad del fluido y proveen la medida de gas en volumen o flujo másico.

El presente proyecto ha construido un Computador de Flujo para medidor primario tipo turbina, basado en un sistema embebido TDS2020F (con microprocesador Hitachi H8/538), tarjeta de acondicionamiento de señal, y transductores de presión y temperatura.

Para la implementación de las normas internacionales de medición de gas (AGA7 y AGA8), y el control del equipo, se utilizó el lenguaje de programación Forth.

Las funciones de la tarjeta de acondicionamiento son:

- aislar la señal proveniente del medidor de turbina, la cual es un tren de pulsos TTL.
- alimentar la RTD a través de un integrado fuente de corriente.
- amplificar la señal de salida de los transductores al margen de 0 a 10V (rango de entrada del conversor análogo-digital)

El prototipo final cumple con las normas de seguridad intrínseca, lo que permitió probarlo en campo donde sus medidas fueron contrastadas con las de una unidad comercial, obteniéndose una diferencia de 0.26%.

ABSTRACT

TITLE: FLOW COMPUTER: DESIGN, CONSTRUCTION AND IMPLEMENTATION OF THE PROTOTYPE FOR FLOW GAS MEASUREMENT

AUTHOR: BARAJAS HERRERA, Edgar Javier

KEYWORDS : computador de flujo, medición, instrumentación electrónica, medidor turbina, AGA8, AGA7, sistema embebido, forth.

DESCRIPTION:

At the gas transporting is necessary to have an efficient and reliable measurement system of the volumes that its deliver and receive at buying and selling process of the product. The measurements made with the primaries meters (turbine), don't represents the right value, due to the temperature and pressure variations that the gas suffers during the process. To correct this errors, it is uses devices with high accuracy that are placed in complement with the primaries meters. This equipments are denominate **flow computers**, that monitoring the pressure and temperature changes, calculate the flow density, and give the gas measurement in volume.

The present Project has built a flow computer for primary meter turbine type, based in an embedded system TDS2020F (with Hitachi H8/538 microprocessor), signal conditioning board, and pressure and temperature transducers.

The implementation of the gas international measurements standards (AGA7 and AGA8), and the device control, was made programming language Forth.

The signal conditioning board tasks are:

- isolate the turbine meter signal, that is a TTL pulses train.
- feed the RTD across of a current source integrated circuit.
- amplify the transducers output signal from a range of 0 to 10V (analog digital conversor input range).

The final prototype comply with the intrinsic security standard, that enabled it to be probed at the field. Its measurements were contrasted with the commercial device, getting a 0.26% difference.

INTRODUCCIÓN

Uno de los aspectos más importantes en el transporte de gas, es el de disponer de un sistema eficiente y confiable de medición de los volúmenes que se están entregando y recibiendo. Para realizar esta labor, la industria de gas emplea dispositivos de alta precisión denominados computadores de flujo, que actúan en complemento a los medidores primarios, tales como turbina, platina de orificio y ultrasonido.

Los computadores electrónicos de flujo, han sido usados hace más de veinte años. Durante este tiempo se han desarrollado desde burdos indicadores de flujo hasta sistemas precisos de medida de energía. El avance en esta tecnología ha permitido que los equipos, además de realizar la corrección de los datos obtenidos en función de la presión y temperatura del fluido, ofrezcan otras funciones como: almacenamiento de información (históricos), conexión a un computador PC portátil para descarga de la información, y transmisión de estos datos remotamente usando un módem.

La industria del gas exige un dispositivo de medida altamente seguro, libre de sabotaje, confiable y exacto, que reduzca el riesgo de errores del sistema sin aumentar su incertidumbre. La experiencia vivida por la industria con algunos dispositivos de campo programables dedicados a controlar bombas, motores y compresores, que han ocasionado la pérdida de datos críticos o señales adquiridas como resultado de un "reset accidental" o de la reprogramación del equipo, hace que se tomen medidas muy exigentes en el diseño y selección de estos dispositivos.

Hoy en día, la buena aceptación por parte del usuario, el auge de la tecnología electrónica y el alto costo de la energía, hacen ver el computador de flujo electrónico, como el primer elemento para la corrección y registro del flujo de gas.

Actualmente los computadores de flujo son construidos en el extranjero, especialmente en Estados Unidos y Europa donde compañías especializadas que poseen la experiencia necesaria en el campo, los fabrican de acuerdo a las normas internacionales AGA (*American Gas Association*), lo que garantiza una alta confiabilidad en la medición. Las unidades se comercializan para uso doméstico o para exportación hacia otros países, como el caso de Colombia.

Al importar este tipo de equipos, se tienen las siguientes desventajas:

- Costos del dispositivo: Por ser un equipo especializado y que emplea lo último en tecnología, su costo es considerablemente alto.
- Costos de los repuestos: Dada la construcción de estas unidades, en caso de avería de alguna tarjeta electrónica el proceso de reparación es complejo. Se debe cambiar la tarjeta y el costo de reposición de este elemento usualmente es similar al de una unidad original.
- Tiempos de entrega: Dado su costo y característica especial, las empresas de gas colombianas no disponen ni de unidades ni de repuestos en *stock*. Es necesario hacer el pedido a fábrica y esperar largos tiempos de despacho y nacionalización para obtenerlos.
- Dependencia tecnológica: Debido a la ausencia de una política de apoyo e incentivo para el desarrollo de nuestro país en este campo, se está supeditado a adquirir los equipos a los costos del mercado, manejarlos a nivel de usuario (sin poder hacer variaciones en la programación interna) y en caso de falla de las tarjetas electrónicas la única solución es sustituirla por otra nueva.

El trabajo de investigación desarrollado por el Ingeniero Edgar Javier Barajas Herrera, para la Maestría en Potencia Eléctrica ha sido titulado “COMPUTADOR DE FLUJO: DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA MEDICIÓN DE FLUJO DE GAS”. El principal objetivo fue diseñar y construir el *hardware* y *firmware* de un computador de flujo para medidores de gas tipo turbina. La idea surgió como una alternativa a la necesidad de suprimir la importación de computadores de flujo y como un aporte de la industria colombiana a la academia, la investigación y la innovación tecnológica de nuestro país.

El proyecto ha creado un prototipo que inicia un proceso de desarrollo de computadores de flujo en Colombia, los cuales podrán operar en las estaciones donde se realice transferencia de custodia, en el momento que obtengan los certificados internacionales de medición. Ante la imposibilidad de contar con un medidor patrón o un laboratorio de calibración de flujo de gas para las pruebas finales del prototipo desarrollado, éste se comparó con un equipo comercial que actualmente utiliza Ecogás (American Meter – AE5000). El AE5000 tiene su propio error, y no es un equipo de última tecnología (pertenece a la década de los noventa), por lo tanto la importancia de las pruebas finales radica en observar el seguimiento de las curvas de medición por parte del prototipo. En el Capítulo Cuatro se exponen y evalúan los resultados obtenidos, al instalar los dos equipos en serie y operarlos en iguales condiciones de flujo. Esta prueba tuvo lugar en el sistema de medición de gas del *city gate** de Barrancabermeja. Esta estación tiene como instrumento primario turbinas Marca Elster.

* Estación reguladora de la cual se desprenden redes que conforman total o parcialmente un sistema de distribución y a partir del cual el distribuidor asume la custodia del gas.

El Proyecto estuvo a cargo del Grupo de Investigación en Exploración y Explotación de Hidrocarburos (GIEH), adscrito al Centro de Investigación del Gas de la Escuela de Ingeniería de Petróleos de la UIS, y del Grupo de Control, Electrónica, Modelado y Simulación (CEMOS) de la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones; y fue cofinanciado por COLCIENCIAS, ECOGAS y UIS. Este trabajo es la continuación de dos monografías desarrolladas en la Especialización en Ingeniería de Gas: “Diseño e implementación del *hardware* y *software* de presentación para un computador de flujo de un medidor de gas tipo turbina” desarrollado por el Ingeniero Rafael Roncancio Rodríguez, y “Diseño e implementación del *hardware* y *firmware* para un computador de flujo tipo turbina usando la norma AGA7” desarrollado por el Ingeniero Ricardo González Vásquez.

La investigación estuvo alimentada por dos proyectos de pregrado desarrollados en la Universidad Industrial de Santander:

- Computador de flujo: diseño e implementación del soporte de hardware, elaborado por la Ingeniera Mónica Corzo; y
- Computador de flujo: diseño e implementación del software para la medición de gas, elaborado por el Ingeniero Luis Carlos Gómez.

El presente libro es el compendio de los resultados obtenidos en el trabajo de maestría, y se encuentra organizado en cuatro capítulos. En el primer capítulo se hace una revisión de un sistema de medición de gas para medidor primario tipo turbina, incluyendo las normas internacionales que rigen este proceso. El segundo capítulo hace referencia a la instrumentación electrónica, la tarjeta de acondicionamiento de señal, y al sistema de desarrollo, empleados en la construcción del computador de flujo; y se explica el desarrollo del *hardware* electrónico. En el tercer capítulo se exponen las principales características del *firmware*, de su implementación en el prototipo, y los algoritmos de control del sistema. Finalmente, el cuarto capítulo muestra los resultados obtenidos durante las pruebas realizadas en campo.

1. SISTEMA DE MEDICIÓN DE FLUJO DE GAS.

Los beneficios económicos que obtiene una compañía de transmisión de gas resultan del transporte de cantidades medidas de gas natural*. Una compañía de transmisión se encarga de proveer este servicio a las compañías de productores y consumidores de gas natural. A su vez, las compañías productoras y consumidoras requieren hacer medición para determinar con exactitud los pagos a recibir o a desembolsar por el gas comercializado.

La medición de gas natural es importante en los siguientes casos:

- Siempre que se presenta un cambio en la responsabilidad o manejo del producto entre propietarios (transportadores o procesadores), proceso que se conoce como transferencia de custodia.
- Cuando el gas se transporta a través de límites nacionales e internacionales.
- Para monitorear el desempeño del sistema de tuberías.

El sistema de medición de transferencia de custodia es considerado la caja registradora en la negociación de compraventa y/o transporte de fluidos. Debido a que se lleva a cabo un proceso de intercambio de dinero, las partes que actúan en la transacción se interesan por tener un sistema de medición de flujo con excelentes propiedades de exactitud.

A nivel internacional, no existe un valor determinado de exactitud exigida para la medición de gas, pero cada país determina el máximo error permitido en sus medidores. En Colombia la exactitud de la medición de volúmenes de gas está determinada por la Empresa Colombiana de Gas (ECOGAS), y actualmente este valor está estipulado en 1%. Por esta razón cualquier equipo que esté dentro del rango de 0-1%, está habilitado para operar dentro del territorio nacional.

La exactitud y la incertidumbre son dos conceptos importantes que se deben tener en cuenta para la instrumentación empleada en los sistemas de medición de gas:

- Exactitud: Se define como la diferencia entre el valor medido y el valor verdadero, y se expresa en porcentaje. En el caso del flujo de gas, el valor medido es leído del medidor, y el valor verdadero es tomado de un patrón. Estos medidores patrones son equipos con exigentes niveles de calibración y

* Mezcla de hidrocarburos livianos, principalmente constituida por metano, que se encuentra en los yacimientos en forma libre ó en forma asociada al petróleo.

aunque naturalmente tienen un ínfimo grado de error ($\pm 0.02\%$), son los puntos de referencia para la calibración de los equipos comerciales de medición.

- Incertidumbre: Desempeño de un medidor bajo condiciones de flujo que pueden ser evaluadas y que determinan la causa de la desviación de la exactitud. Estas causas pueden ser operacionales, de interpretación humana, registros, instalaciones y características del fluido.

Los sistemas de medición donde no existe transferencia de custodia, son utilizados para controlar el proceso de transporte y distribución del gas. En estos sistemas los niveles de exactitud no son tan exigentes y se permiten errores hasta del 4%, dependiendo de la aplicación. Sin embargo, si es de importancia la habilidad del medidor de repetir una medida bajo las mismas condiciones de flujo (repetibilidad).

Un medidor es un equipo que permite conocer directamente el caudal de flujo en un tiempo dado. También es conocido como **instrumentación primaria**. Esta instrumentación depende del principio de medición en que esta basada su operación (medidores de presión, de velocidad, de desplazamiento, etc).

El método ideal¹ para medir gas natural consiste en medir la energía real entregada, pero actualmente no existe un método conveniente para obtener directamente este valor. La forma de calcular la energía entregada, es multiplicar el volumen entregado por el valor de calentamiento del gas. El método real más utilizado para medir gas natural consiste en determinar el volumen o la **velocidad del flujo** que pasa a través de algún tipo de medidor.

El computador de flujo del presente proyecto, esta construido para operar junto a un medidor primario tipo **turbina** (medidor de velocidad del flujo). Estos medidores han tenido gran aceptación en la industria del gas de Latinoamérica, y Colombia; gracias a su buena exactitud² (0.5%), y a su repetibilidad (cercana al 0.05%). En la Sección 1.1., se describe un sistema de medición de gas para este tipo de instrumento primario.

Los gases comerciales varían en su composición química y en sus propiedades físicas, y por ello se comportan de manera diferente que los gases ideales. Esta desviación hace que la medición práctica sea más compleja que la que se podría predecir según la teoría básica. La desviación de la temperatura y presión base implica la necesidad de emplear **transductores** para medir dichas variables. A estos sensores se les conoce como **instrumentación secundaria**.

Los dispositivos encargados de realizar la corrección del flujo medido con base en la presión y temperatura actual, reciben el nombre de **computadores de flujo**, y se convierten en la **instrumentación terciaria** del sistema de medición. Para cumplir su tarea, requieren recibir la información proveniente del medidor (turbina), y de los sensores (presión, temperatura).

¹ MILLAR, Richard. Flow Measurement Engineering Handbook. Mc Graw Hill, 3rd. ed, 1996.

² FLOYD, Angela. Fundamentals of Gas Turbine Meters. Daniel Flow Products, TX 77224.

En el presente capítulo se describirán en detalle la instrumentación mencionada, incluyendo la normatividad que debe cumplir.

1.1. INSTRUMENTACIÓN PRIMARIA: MEDIDOR TIPO TURBINA.

El concepto de medición con turbinas es bastante antiguo. Aunque los primeros medidores se utilizaron para medir agua en los albores del siglo XX, su aplicación a la medición de gas fue desarrollada hasta comienzos de la década del 50. Así, en 1953 entró al mercado el primer medidor de turbina.

1.1.1. Medidor tipo turbina para gases. La turbina (Figura 1), es un medidor lineal que produce una frecuencia (velocidad de rotación) proporcional a la velocidad del fluido en la tubería. Es considerada como uno de los medidores más exactos en medición de gas.

Figura 1. Medidor tipo turbina.



La Tabla 1, muestra las características más representativas del medidor tipo turbina³.

Tabla 1. Características medidor tipo turbina.

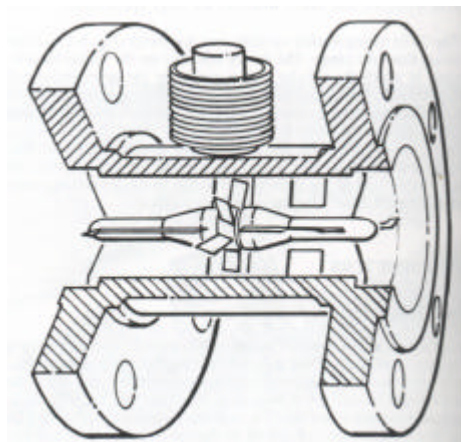
Díámetro interno	6 - 760 (mm)
Rango de volumen	0.015 - 15000 (ACFM)
Repetibilidad	0.1% de lectura
Exactitud y trazabilidad	0.30% de lectura
Número de aspas	12 -24
Máxima frecuencia de pulsos	1 Hz
Temperatura máxima	800 °C
Presión máxima	6000 psi

³ SPITZER, David. Flow Measurement. ISA, 2nd ed, 2001.

Los sistemas de medición de gas que tienen medidor primario tipo turbina, se rigen bajo la norma AGA 7⁴ (*American Gas Association*), la cual ha sido implementada en el prototipo del presente proyecto.

1.1.2. Principio de operación. El medidor turbina consiste de un rotor montado sobre un cojinete y un eje, ubicados dentro del cuerpo del medidor (Figura 2). El fluido a ser medido pasa a través del cuerpo, causando que el rotor gire. La velocidad del rotor se incrementa linealmente con la velocidad del flujo. Así, la rotación de las turbinas es una medida de la velocidad del fluido.

Figura 2. Corte transversal de un medidor de turbina.



El sensor puede ser de dos tipos:

- Un engranaje manejado mecánicamente que permite la visualización del volumen no corregido en un contador montado en la parte externa del medidor.
- Un sensor electrónico que detecta el paso de cada aspa del rotor generando un pulso, que es enviado a un computador de flujo para visualizar la información del volumen no corregido. La frecuencia de los pulsos es proporcional a la rata de flujo volumétrico a través del medidor.

La lectura del sensor, puede estar dada en pies cúbicos o en metros cúbicos.

1.1.3. Sensores de la velocidad del flujo. Los sensores mecánicos están instalados sobre algunos medidores de turbina. Un sensor mecánico consiste de un eje manejado desde un engranaje ubicado sobre el eje del rotor. Este sistema funciona como el medidor de velocidad en un automóvil. Los indicadores pueden mostrar la rata de flujo y el flujo totalizado.

⁴ AMERICAN GAS ASSOCIATION. AGA Report N°7 Measurement of Gas by Turbine Meters. 1996.

El prototipo del presente proyecto ha sido diseñado para recibir pulsos electrónicos. Los sensores electrónicos, usualmente llamados "*pickoffs*", son proporcionados por los fabricantes de turbinas. Existen tres tipos básicos de *Pickoffs*: inductancia magnética, reluctancia magnética, y frecuencia portadora modulada. Estos, están montados en el cuerpo de la turbina y espaciados por una distancia corta de la extremidad del rotor a través de una membrana delgada.

El *pickoff* inductivo magnético requiere que un magneto este embebido en cada aspa del rotor de la turbina. El *pickoff* en su final, posee un arreglo de una bobina en espiral, la cual es usada para producir un pulso eléctrico al paso de cada magneto.

El *pickoff* de reluctancia magnética contiene un magneto dentro de sí mismo, usualmente con una bobina de espiral. Este tipo de sensor requiere que el rotor sea construido de un material que posea alguna permeabilidad magnética. La señal característica producida es del mismo tipo que el *pickoff* inductivo.

El *pickoff* de frecuencia portadora modulada contiene una bobina y un núcleo férrico con un espiral alrededor. Un amplificador adjunto tiene un circuito que oscila electrónicamente con la inductancia del espiral del *pickoff*. La frecuencia de oscilación es mucho más grande que la frecuencia de paso de las aspas, actuando como portadora, y es modulada por el paso de las aspas del rotor. La frecuencia portadora es filtrada en un amplificador (demodulador), dejando solamente la frecuencia del paso de las aspas. La salida del amplificador es transformada en una onda de pulso con una amplitud de voltaje constante (usualmente 5 o 10 voltios pico a pico máximo).

1.1.4. Instalación del medidor. Como el medidor de turbina mide velocidad de flujo, la configuración de la tubería inmediatamente aguas arriba del medidor debe garantizar que el perfil de flujo que ingresa al medidor tenga una distribución uniforme, sin chorros ni remolinos. Puesto que la construcción del medidor de turbina se diseña para dirigir el flujo por el pasaje anular aguas arriba del rotor, este medidor tiende a promediar el perfil de velocidad de la mayoría de condiciones de flujo normales, minimizando así la influencia de las distorsiones de flujo sobre el funcionamiento del medidor.

Como se mencionó anteriormente, el *city gate* de Barrancabermeja es una estación de transferencia de custodia, donde se garantiza que el flujo que pasa por el medidor primario (turbina) presenta el mínimo de distorsiones permitidas por la norma AGA7.

1.1.4.1. Perturbaciones de flujo. Las perturbaciones de flujo afectan el desempeño del medidor. Entre estos efectos se pueden mencionar:

1.1.4.1.1. Efecto remolino. Si el fluido a la entrada del medidor posee un efecto remolino significativo, la velocidad del rotor de la turbina es afectada dependiendo del sentido de orientación del remolino (a favor o en contra de la velocidad de rotación del rotor). La forma de eliminar el efecto remolino es modificando la

instalación de la turbina (cumplir distancias mínimas de aguas arriba y aguas abajo del medidor).

1.1.4.1.2. Efecto perfil de velocidad. El medidor de turbina de gas está diseñado y calibrado bajo condiciones uniformes de perfil de velocidad a la entrada del medidor. En el caso de presentarse una desviación significativa del perfil de velocidad, la medición del volumen de gas es afectada. Para una rata de flujo promedio, un perfil no uniforme de velocidad ocasiona una mayor velocidad del rotor produciendo un mayor registro de volumen medido.

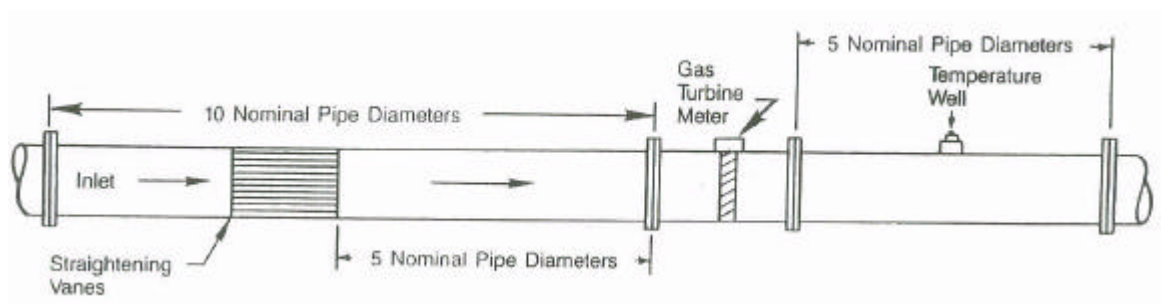
Para disminuir la inexactitud en la medición por no uniformidad del perfil de velocidad se recomienda una instalación apropiada para el medidor. (ver sección 1.1.4.2.).

Las perturbaciones de flujo anteriormente descritas son debidas, a reguladores de presión, accesorios de tubería, filtros, etc.

En la instalación del medidor de turbina se recomiendan los enderezadores de flujo. A pesar de su ubicación en la tubería, ellos no eliminan el efecto de los remolinos fuertes. Los enderezadores de flujo localizados en la tubería aguas arriba del medidor eliminan condiciones de flujo de remolino menores.

1.1.4.2. Instalación recomendada para medidores de turbina en línea. La instalación recomendada requiere una longitud de tubería recta de 10 diámetros nominales, aguas arriba, con la salida de los enderezadores de flujo localizada a cinco diámetros nominales de la entrada al medidor, como se observa en la Figura 3. Aguas abajo del medidor se recomienda una longitud de cinco diámetros nominales. Tanto la tubería de entrada como la de salida deben tener el mismo diámetro nominal del medidor.

Figura 3. Instalación recomendada para medidor turbina.



No se recomienda la instalación de reguladores o válvulas parcialmente cerradas en la proximidad del medidor. En los casos donde se requiera la instalación de estos elementos, el regulador debe ubicarse ocho diámetros nominales adicionales (total 18D) aguas arriba o dos diámetros nominales (total 7D) aguas abajo, respecto a la instalación recomendada en la Figura 3. La instalación de ese dispositivo en la

proximidad al medidor puede dar como resultado una degradación de la exactitud y una reducción en la vida de los cojinetes de la turbina.

El sistema de medición de gas del *city gate* de Barranca cumple con las anteriores recomendaciones, pues la tubería tiene un diámetro de 80mm, y las secciones de tubería recta tienen longitudes de 80cm y 40cm, aguas arriba y aguas abajo, respectivamente. Esto asegura un flujo uniforme en la turbina empleada para la prueba del computador de flujo del presente proyecto.

1.1.4.3. Enderezadores de flujo. El propósito de los enderezadores de flujo (*straightening vanes*) es eliminar los remolinos, y la no uniformidad del perfil de velocidad (chorros) por efecto de accesorios, válvulas y reguladores instalados aguas arriba del medidor.

En la construcción del enderezador (conjunto de tubos de pequeño diámetro) el diámetro de los tubos no debe exceder $\frac{1}{4}$ del diámetro de la tubería de entrada del medidor, además el área transversal de flujo entre tubos del enderezador no debe exceder $\frac{1}{16}$ del área de flujo de la tubería de entrada. La longitud del enderezador debe ser, mínimo 10 veces el diámetro interno de los tubos que conforman el enderezador.

El arreglo de los tubos puede ser cuadrado, hexagonal u otras formas sin que ello incida en el papel del enderezador respecto al medidor de flujo.

El *city gate* de Barrancabermeja posee enderezadores de flujo.

1.1.4.4. Cedazos (*strainers*) o filtros. Los cedazos ayudan a evitar la presencia de sustancias extrañas en la tubería, las cuales pueden causar serios daños a los medidores de turbina. Los cedazos deben diseñarse de tal manera que a un flujo máximo exista una caída de presión mínima y poca distorsión del flujo.

Si se desea un mayor grado de protección se puede usar un filtro tipo seco o un filtro/separador, instalado aguas arriba del medidor. Se recomienda monitorear la presión diferencial a través del filtro, para mantenerlo en buenas condiciones y prevenir distorsión del flujo y posible interrupción del flujo de gas.

1.1.4.5. Requerimientos adicionales de la instalación.

- El medidor y la tubería deben instalarse en una forma tal que se reduzca la tensión que pueda ser ocasionada por la velocidad del fluido, cambios térmicos en el material de la tubería, etc.
- Debe existir alineación concéntrica entre las bridas de la tubería y las del medidor, tanto en la entrada como en la salida. Esta alineación concéntrica eliminará cualquier efecto sobre la exactitud del medidor.
- No se debe permitir la entrada en la tubería de los empaques que sirven para sellar las bridas del medidor.
- El interior del tubo debe ser de rugosidad comercial y el diámetro interno de la brida debe ser igual al de la tubería.

- Las instalaciones donde se puede encontrar líquido deben diseñarse para prevenir la acumulación de ellos en el medidor.
- No se deben realizar soldaduras en los puntos cercanos al medidor, con el fin de evitar distorsión de flujo.
- Los medidores de turbina no deben ser usados donde exista fluctuación en el flujo, interrupciones frecuentes o pulsaciones de presión.

1.1.4.6. Instalación de la instrumentación secundaria. Los sensores que se emplean para integrar el volumen no corregido a las condiciones estándar o para registrar los parámetros de operación, deben instalarse apropiadamente.

1.1.4.6.1. Medidor de temperatura. Puesto que los disturbios aguas arriba deben mantenerse al mínimo, el pozo del termómetro debe ubicarse aguas abajo del medidor. Debe estar localizado entre uno y cinco diámetros nominales de tubería a partir de la salida del medidor y aguas arriba de cualquier accesorio o válvula.

En el *city gate* de Barranca, la conexión del sensor de temperatura esta ubicada a tres diámetros nominales de la salida del medidor (24cm).

1.1.4.6.2. Medidor de presión. El fabricante provee una toma de presión sobre el cuerpo del medidor, que es el punto de conexión de un cable de alta presión que va hasta el computador de flujo donde se encuentra un transductor para la medida de esta variable en libras por pulgada cuadrada o PSI (*pounds per square inch*). El computador de flujo traducirá esta señal y le agregará la presión atmosférica del lugar (dato teórico), para así obtener la presión absoluta necesaria en algunos cálculos. Para efectos de claridad de cálculos se ha tomado por regla general la unidad psig (g - *gauge*) para la presión tomada del transductor y la unidad psia para las presiones absolutas.

1.1.5. Características de funcionamiento de la turbina.

1.1.5.1. Repetibilidad. La repetibilidad hace referencia a qué tan parecidos son los resultados de mediciones sucesivas a las mismas condiciones de operación, llevadas a cabo por el mismo método, por el mismo observador y con las mismas instalaciones para el mismo instrumento. Los medidores de turbina han demostrado la capacidad de repetir dentro de $\pm 0.10\%$ en pruebas sucesivas de periodos de duración muy cortos y dentro de $\pm 0.15\%$ en una base día a día (las pruebas son conducidas bajo condiciones idénticas de operación). La buena repetibilidad en periodos prolongados de tiempo depende de que se mantengan las condiciones físicas del medidor y que se comparen periodos donde el medidor esta operando bajo similares condiciones de flujo.

1.1.5.2. Exactitud. La exactitud de un medidor es el grado de conformidad del valor indicado por un medidor con respecto al valor verdadero de la cantidad medida. En el caso de la medición de flujo de gas natural, la exactitud de un medidor de turbina se especifica dentro del $\pm 1\%$ del volumen real sobre un rango especificado.

1.1.5.3. Linealidad. La linealidad de un medidor de turbina es el máximo porcentaje de desviación de la sensibilidad K (pulsos/pie³) a lo largo del rango lineal. El rango lineal de un medidor de flujo de turbina es el rango de flujo sobre el cual la frecuencia de salida es proporcional al flujo (factor K) dentro de los límites especificados por el fabricante.

1.1.5.4. Caída de Presión. La pérdida de presión que se presenta en una turbina es atribuida a la energía requerida para mover el rotor del medidor y la fricción propia del paso del gas incluyendo cambios en área y dirección. La presión diferencial aproximada, desarrollada a través de un medidor de turbina, en condiciones de operación, se puede calcular con la siguiente formula:

$$\Delta P = \Delta P_{avg} \left(\frac{P_g + P_a}{P_b} \right) \left(\frac{G}{0.60} \right) \left[\frac{Q}{Q_{max}} \right]^2 \quad (\text{ecuación 1})$$

donde:

$\Delta P =$	Presión diferencial, en pulgadas de agua.
$\Delta P_{avg} =$	Presión diferencial promedia (pulgadas de agua), dada por el fabricante, para cada tamaño y modelo de turbina, generalmente @ 0.25psig y Q_{max} .
$P_g =$	Presión de operación, en psig.
$P_a =$	Presión atmosférica del sitio donde esta operando el medidor, psia.
$P_b =$	Presión base o contractual, en psia.
$G =$	Gravedad especifica real del gas.
$Q =$	Rata de flujo real, en condiciones de línea, en pcr/h.
$Q_{max} =$	Máxima rata de flujo, en pies cúbicos reales por hora, a condiciones dadas por el fabricante, generalmente @ 0.25 psig. El valor de Q_{max} lo da el fabricante para cada modelo en particular.

El medidor turbina es un dispositivo que mide velocidad de flujo de gas a partir de una señal proporcional a la velocidad del rotor. Por esta razón, la caída de presión a través del medidor no tiene relación con la medición de la velocidad del fluido, y no es necesaria para los cálculos que se deben realizar en el presente proyecto.

1.1.6. Ventajas y Desventajas del medidor tipo turbina. La Tabla 2, muestra una comparación de las ventajas y desventajas que se disponen al emplear medidor primario tipo turbina.

Tabla 2. Ventajas y desventajas del medidor turbina.

Ventajas	Desventajas
Buena exactitud dentro del rango de operación del medidor.	Revisión periódica de sus partes para garantizar una buena exactitud.
Equipos electrónicos disponibles para lectores de flujo en corto tiempo y alta resolución.	Requiere perfil de velocidad uniforme.
Excelente rango de operación a altas presiones.	Bajo rango de operación a bajas presiones.
Costos medios de inversión comparados con otros tipos de medidores.	

1.1.7. Turbina Elster. El medidor primario no fue objeto de selección para el presente proyecto, pues es un elemento externo al computador de flujo. Una turbina marca Elster, presente en el *city gate* de Barrancabermeja, fue la utilizada para la prueba final de operación del prototipo. Las características principales de este medidor se muestran en la Tabla 3.

Tabla 3. Turbina Elster.

Diámetro Tubería	80 mm
Presión de Trabajo	250 psi
Capacidad Máxima	15000 ACFH
Factor K	0.01
Frecuencia a Qmax	0,041667 Hz

El factor K de una turbina se refiere a la equivalencia en velocidad de flujo de un pulso electrónico. Según la Tabla 3, este factor tiene un valor de 0.01, es decir que en la prueba del prototipo un pulso equivale a 100 ACFH. (Actual Cubic Feed per Hour)

1.2. NORMAS INTERNACIONALES DE MEDICIÓN DE FLUJO VOLUMÉTRICO DE GAS NATURAL.

Como se mencionó en la sección 1.1., el medidor de turbina es un equipo que mide la velocidad, es decir, dependiendo de la rata de flujo del gas, el rotor del medidor se mueve a una velocidad proporcional a la velocidad de flujo. Las revoluciones del rotor se cuentan mecánica o electrónicamente y se convierten a un registro volumétrico continuamente totalizado. Puesto que el volumen registrado por la turbina está a condiciones de presión y temperatura de flujo, este volumen real debe corregirse a las condiciones base para propósitos de venta.

Las normas internacionales para la medición de gas, que debe cumplir un computador de flujo para medidor primario tipo turbina y que han sido implementadas en el equipo desarrollado en el presente trabajo de investigación,

son el AGA7, AGA8⁵ y al Capítulo 21⁶ de la norma API (*American Petroleum Institute*). El seguimiento de estos estándares internacionales por parte del prototipo, hacen de él un equipo actualizado, habilitado para iniciar un proceso de certificación internacional.

1.2.1. Medición de gas combustible con medidores de turbina – AGA7. Esta publicación técnica provee las prácticas actuales y los métodos aceptados para la medición de gas con medidores de turbina de flujo axial. También contiene la información relacionada con su instalación, mantenimiento y calibración, la cual fue presentada en sección 1.1.4. del presente texto. Fue desarrollada por el Grupo de Tareas para Medidores de Turbina de la Asociación Americana de Gas. Este reporte no describe los equipos usados en la determinación de presión, temperatura, densidades y otras variables que deben ser conocidas para la medición exacta de un gas.

1.2.1.1. Caudal a condiciones de flujo. El caudal (rata volumétrica) a las condiciones de flujo se determina por:

$$Q_f = \frac{V_f}{\Delta t} \quad (\text{ecuación 2})$$

donde:

Q_f = Rata de flujo de gas a las condiciones de flujo (equivalente a la frecuencia de pulsos de la turbina).

V_f = Volumen de gas medido a las condiciones de flujo (calculado de la ecuación 2).

Δt = Período de tiempo donde se totaliza el volumen (determinado en el diseño).

K = Factor K dado por el fabricante, pulsos/pie³ ($K=0.01$ para la turbina empleada en el presente proyecto), es decir que 1 pulso de la turbina equivale a 100 pies cúbicos de gas por hora.

1.2.1.2. Rata de flujo a condiciones base. Este caudal se determina según la ecuación 3:

$$Q_b = (Q_f) (F_{pm}) (F_{tm}) (S) \quad (\text{ecuación 3})$$

Factor de presión, F_{pm}

$$F_{pm} = \frac{P_f}{P_b} \quad (\text{ecuación 4})$$

⁵ AMERICAN GAS ASSOCIATION. AGA Report N° 8 Compressibility Factors of Natural Gas and others relates Hydrocarbon Gases. 1992.

⁶ AMERICAN PETROLEUM INSTITUTE. Manual of Petroleum Measurement Standard. Chapter 21 – Flow Measurement Using Electronic Metering Systems, Section 1 – Electronic Gas Measurement. 1st ed, 1993.

donde:

$p_f = P_f + p_a$ presión absoluta del fluido, psia.

P_f = Presión estática manométrica, psig (medida del sensor de presión).

p_a = Presión atmosférica, psia. En Barranca $p_a = 14.65$ psia.

p_b = Presión base (contractual). Para Ecogás $p_b = 14.73$ psia.

Según lo anterior, F_{pm}

$$F_{pm} = \frac{P_f + 14.65}{14.73} \quad (\text{ecuación 5})$$

Factor de temperatura, F_{tm}

$$F_{tm} = \frac{T_b}{T_f} \quad (\text{ecuación 6})$$

donde:

T_b = temperatura base en grados Ranking, (519.67 °R (60°F), para el gas en Colombia).

T_f = temperatura real del gas que fluye, °R. (medida del sensor de temperatura).

Entonces, el factor de temperatura es:

$$F_{tm} = \frac{519.67}{T_f} \quad (\text{ecuación 7})$$

Relación de compresibilidad, s

$$s = \frac{Z_b}{Z_f} \quad (\text{ecuación 8})$$

donde:

Z_b = factor de compresibilidad en condiciones base (calculado por el computador de flujo).

Z_f = factor de compresibilidad en condiciones de flujo (calculado por el computador de flujo).

La relación de compresibilidad "s" se puede determinar a partir del factor de supercompresibilidad (F_{pv}), el cual se define como:

$$F_{pv} = \sqrt{\frac{Z_b}{Z_f}} \quad ; \quad s = F_{pv}^2 \quad (\text{ecuación 9})$$

El valor numérico del factor de supercompresibilidad F_{pv} se obtiene por medio de los métodos de cálculo dados en el "AGA Transmission Measurement Committee Report No.8".

1.2.2. Compresibilidad y supercompresibilidad para gas natural – AGA8.

Dado que el gas natural es un fluido compresible el volumen del gas depende de la presión y temperatura a la que se encuentre. El problema es que no se tiene un solo elemento gaseoso puro, al contrario en el caso del gas natural así como otros gases hidrocarbonados su constitución es una mezcla de diversos gases principalmente metano, nitrógeno, dióxido de carbono, y otros en menor proporción como hidrógeno, monóxido de carbono, etano, propano; y a veces hasta 21 compuestos diferentes haciendo que la rigurosa química asociada deba tener en cuenta los modelos o comportamientos de cada uno de estos gases.

Bajo este orden de ideas, la Asociación Americana de Gas (AGA), que inició trabajos en 1930, sacó su primer manual en 1962 llamado "AGA Reporte NX-19"⁷ para el cálculo del factor de compresibilidad del gas natural. Posteriores investigaciones (1981-1984) mejoraron las ecuaciones y fueron publicadas en el Reporte AGA8 primera edición. Luego numerosos experimentos en asociación con el GRI (Gas Research Institute), aumentaron considerablemente la base de datos y mostraron que las ecuaciones de estado originales debían ser mejoradas aún más.

Así surgió en 1992 el Reporte AGA8 "Compressibility Factors of Natural Gas and Other Related Hydrocarbon Gases", donde se presentan las nuevas ecuaciones mejoradas y una información detallada acerca de la metodología y cálculos precisos para cálculo, ya que mientras el NX-19 puede ser desarrollado con ayuda de una calculadora de mano, el AGA8 requiere el uso de una computadora.

Si bien, este ha sido el último reporte en materia de cálculo de factores de compresibilidad, muchas empresas aún emplean computadores de flujo que traen la NX-19 (comprados desde los '80 o '90), pues cambiar al AGA8 les implicaría un elevado costo de renovación, instalación y actualización de los nuevos equipos. Sin embargo, los nuevos computadores de flujo del siglo XXI tienen incorporadas estas normas en su unidad de procesamiento y han dejado el NX-19 como cosa del pasado. Por esta razón, en el presente proyecto se implementó el AGA8, actividad que se convirtió en la más rigurosa y dispendiosa del proyecto.

Mientras la norma NX-19 fue realizada experimentalmente con una base de datos empírica de 300 experimentos, la norma AGA8 fue efectuada con una base de datos de 12,000 lo cual muestra su supremacía en exactitud respecto a la primera. A continuación, en la Tabla 4⁸ se muestra una comparación del porcentaje de error de cada una de las normas en diversas situaciones.

⁷ AMERICAN GAS ASSOCIATION. NX19 Manual For Determination of Supercompressibility Factors For Natural Gas. 1962.

⁸ CARELL, Bob. Trends in Electronic Flow Computers. Sensors Magazine, Oct. 1999.

Tabla 4. Comparación NX-19 vs AGA8.

Clasificación del Gas	Gravedad Específica	NX-19	AGA8
Baja Gravedad	0.60	0.096	0.044
Moderada Gravedad	0.63	0.335	0.026
Alta Gravedad	0.67	0.819	0.049
Moderado Nitrógeno	0.66	0.066	0.067
Alto Nitrógeno	0.78	0.158	0.056
Moderado CO2	0.71	0.114	0.027
Alto CO2	0.87	0.606	0.079

AGA8 es una norma extensa e involucra gran rigor matemático, tanto en exactitud como en extensión. En ella se proponen dos métodos para deducir el factor de compresibilidad: "Detail"* (detallado) y "Gross" (grueso). La diferencia de estos métodos, estriba en la cantidad de parámetros o características del gas conocidas y el rango de presión y temperatura sobre los cuales se pueden trabajar. En el método "Detail" se requiere la cromatografía completa de aproximadamente 21 componentes del gas y presenta un error de 0.1% en el rango de temperatura de 17 a 143 °F, y de presión de 0 a 1750 psia. Por otro lado, el método "Gross" exige el conocimiento del porcentaje de solo dos gases importantes (N, CO2) y la densidad relativa del gas funcionando con un 0.1% de error en un rango de temperatura de 32-130 °F (0-55°C) y de presión de 0-1200 psia.

En la mayoría de gasoductos colombianos no se cuenta con cromatografía en línea que permita obtener el porcentaje instantáneo de los 21 componentes del gas. En lugar de esto se tienen estimativos del porcentaje de los compuestos principales, lo cual sugiere el empleo del método "Gross" para el cálculo del factor de compresibilidad. Adicionalmente el rango de presión y temperatura de este método es adecuado a las condiciones colombianas de operación de los sistemas de distribución de gas a las ciudades (*city gates*). Por ejemplo, la presión de línea del *city gate* donde se efectuó la prueba es de 250 psi.

1.2.2.1. Método Gross. La ecuación de estado aplicada para hallar el factor de compresibilidad por este método GROSS, sigue un modelo tipo virial. Este método también se conoce como el modelo "SGERG". No requiere una cromatografía detallada, pues utiliza un rango de valor calorífico de gas natural, densidad relativa (o gravedad específica), y contenidos diluidos como parámetros de caracterización generales, en lugar de la composición detallada de un gas natural. El modelo predice con gran precisión factores de compresibilidad de gases naturales que tengan concentraciones de componentes dentro de los rangos dados en la Tabla 5, y con menos de 0.1% molar de agua, y 0.05% molar de sulfuro de hidrógeno.

* Sección 8.1 del Reporte AGA8.

Tabla 5. Rango de Aplicación del Método AGA8 Gross.

CANTIDAD	RANGO
Densidad Relativa (a 60 °F y 14.7 psia)	0.554 a 0.87
Porcentaje Molar de Metano	45 a 100
Porcentaje Molar de Nitrógeno	0 a 50
Porcentaje Molar de Dióxido de Carbono	0 a 30
Porcentaje Molar de Etano	0 a 10
Porcentaje Molar de Propano	0 a 4
Porcentaje Molar total de Butanos	0 a 1
Porcentaje Molar total de Pentanos	0 a 0.3
Porcentaje Molar de Hexanos	0 a 0.2
Porcentaje Molar de Helio	0 a 0.2
Porcentaje Molar de Hidrógeno	0 a 10
Porcentaje Molar de Monóxido de Carbono	0 a 3
Porcentaje Molar de Agua	0 a 0.05

Para el *city gate* de Barranca los 3 datos necesarios para el método "Gross" son:

- Densidad Relativa: 0.565
- Porcentaje Molar de Nitrógeno: 0.01563 %
- Porcentaje Molar de Dióxido de Carbono: 0.00039%

Una ecuación virial de estado es una expansión polinomial en densidad. Cada término de densidad es precedido por un coeficiente virial. Los coeficientes viriales son funciones de la temperatura y la composición. La aplicación de la ecuación virial de estado modelo "SGERG", trunca la ecuación virial del factor de compresibilidad después del tercer coeficiente virial. Este truncamiento provee una alta precisión para transmisión en gasoductos y condiciones de distribución, con límites en presiones y densidades.

El método Gross aproxima una mezcla de gas natural tratándola como una mezcla de tres componentes:

- Un componente equivalente de hidrocarburo –CH–.
- Nitrógeno.
- dióxido de carbono.

El equivalente de hidrocarburo (llamado CH), es empleado para representar colectivamente todos los hidrocarburos encontrados en la mezcla del gas. El nitrógeno y el dióxido de carbono son componentes diluidos. Este modelo emplea unidades internacionales.

El método Gross predice el factor de compresibilidad de un gas natural dadas la presión y temperatura, y si se conocen 3 de los siguientes 4 parámetros:

1. El valor calorífico volumétrico con condiciones de referencia de 77° F, 14.696 psia (25°C, 0.101325 MPa) para valor calorífico molar ideal y 32°F, 14.696 psia (0°C, 0.101325 MPa) para densidad molar.
2. Densidad relativa (gravedad específica) con condiciones de referencia de 32° F 14.696 psia (0°C, 0.101325 MPa).
3. Fracción molar de dióxido de carbono.
4. Fracción molar de nitrógeno.

Según los parámetros anteriores que se empleen, se diferencian dos tipos del método Gross:

- Gross1: El método uno usa valor calorífico, densidad relativa y contenido de dióxido de carbono como entradas.
- Gross2: El método dos usa densidad relativa, contenido de dióxido de carbono y contenido de nitrógeno como entradas. Precisamente, estos son los datos dispuestos por los operadores del gas de Barranca, y por esta razón se empleó el método Gross2 para el cálculo del factor de compresibilidad. La norma AGA8 expone su procedimiento de cálculo en su Anexo B.2 Sec. B.2.3.

La fracción molar del hidrocarburo equivalente –CH– es determinada por la siguiente ecuación:

$$X_{CH} = 1 - X_{N_2} - X_{CO_2} \quad (\text{ecuación 10})$$

El modelo “SGERG” expresa el factor de compresibilidad (Z), tanto para condiciones de flujo como para condiciones base, en términos de la densidad molar (d), el segundo coeficiente virial de la mezcla (B_{mix}), y el tercer coeficiente virial de la mezcla (C_{mix}):

$$Z_f = 1 + B_{mix} d + C_{mix} d \quad (\text{ecuación 11a})$$

$$Z_b = 1 + (B_{mix} \cdot P_{gr}) / (R \cdot T_{gr}) \quad (\text{ecuación 11b})$$

$$B_{mix} = \sum_{i=1}^N \sum_{j=1}^N B_{ij} x_i x_j \quad (\text{ecuación 12})$$

$$C_{mix} = \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N C_{ijk} x_i x_j x_k \quad (\text{ecuación 13})$$

donde,

Z = factor de compresibilidad base (Z_b) o del fluido (Z_f).

B_{mix} = segundo coeficiente virial de la mezcla.

C_{mix} = tercer coeficiente virial de la mezcla.

d = densidad molar (moles por unidad de volumen).

B_{ij} = componente individual de interacción del segundo coeficiente virial.

C_{ijk} = componente individual de interacción del tercer coeficiente virial.

$x_i x_j x_k$ = fracciones molares de los componentes del gas.

N = número de componentes de la mezcla del gas.

Los índices i, j, k, representan componentes moleculares de la mezcla del gas. Los términos B_{ij} y C_{ijk} son los coeficientes viriales de interacción, y dependen de la temperatura. En aplicaciones de gas natural con los límites específicos de esta ecuación, la dependencia de temperatura es de forma cuadrática.

Expandiendo las ecuaciones anteriores para B_{mix} y C_{mix} e identificando todos los términos necesarios para resolver el modelo "SGERG", se tiene:

$$B_{mix} = B_{CO_2-CO_2} x_{CO_2}^2 + B_{N_2-N_2} x_{N_2}^2 + B_{CH_4-CH_4} x_{CH_4}^2 + 2 B_{CO_2-N_2} x_{CO_2} x_{N_2} + 2 B_{CO_2-CH_4} x_{CO_2} x_{CH_4} + 2 B_{N_2-CH_4} x_{N_2} x_{CH_4}$$

(ecuación 14)

$$C_{mix} = C_{CO_2-CO_2-CO_2} x_{CO_2}^3 + C_{N_2-N_2-N_2} x_{N_2}^3 + C_{CH_4-CH_4-CH_4} x_{CH_4}^3 + 3 C_{CO_2-CO_2-N_2} x_{CO_2}^2 x_{N_2} + 3 C_{CO_2-CO_2-CH_4} x_{CO_2}^2 x_{CH_4} + 3 C_{CO_2-N_2-N_2} x_{CO_2} x_{N_2}^2 + 3 C_{CO_2-CH_4-CH_4} x_{CO_2} x_{CH_4}^2 + 3 C_{N_2-N_2-CH_4} x_{N_2} x_{CH_4}^2 + 3 C_{N_2-CH_4-CH_4} x_{N_2} x_{CH_4}^2 + 6 C_{CO_2-N_2-CH_4} x_{CO_2} x_{N_2} x_{CH_4}$$

(ecuación 15)

A continuación se muestran las ecuaciones y métodos necesarios para calcular los términos de interacción del segundo y tercer coeficiente virial de las ecuaciones anteriores.

Coeficientes viriales interactivos

Los valores B_{ij} de los términos para nitrógeno y dióxido de carbono son expresados en (dm^3/mol) y son dados por la ecuación 16.

$$B_{ij} = b_0 + b_1 T + b_2 T^2$$

(ecuación 16)

Donde valores de b_0 , b_1 y b_2 son dados en la Tabla 6. La temperatura T es dada en grados Kelvin.

Tabla 6. Coeficientes viriales B_{ij} para Nitrógeno y Dióxido de Carbono.

Fluid for B_{ij}	b_0 (dm^3/mol)	b_1 (dm^3/mol)	b_2 (dm^3/mol)
N2-N2	-0.144600	0.74091×10^{-3}	-0.911950×10^{-6}
CO2-CO2	-0.868340	0.40376×10^{-2}	-0.516570×10^{-5}
CO2-N2	-0.339693	0.161176×10^{-2}	-0.204429×10^{-5}

De igual modo los valores de C_{ijk} de los términos para nitrógeno y dióxido de carbono son expresados en (dm^6/mol^2) y son dados por la ecuación 17.

$$C_{ijk} = c_0 + c_1 T + c_2 T^2$$

(ecuación 17)

Donde los valores de c_0 , c_1 y c_2 son dados en la Tabla 7, y T es la temperatura en Kelvin.

Tabla 7. Coeficientes viriales C_{ijk} para Nitrógeno y Dióxido de Carbono.

Fluid for C_{ijk}	c_0 (dm^6/mol^2)	c_1 (dm^6/mol^2)	c_2 (dm^6/mol^2)
N2-N2-N2	0.78498×10^{-2}	-0.39895×10^{-4}	0.611870×10^{-7}
CO2-CO2-CO2	0.205130×10^{-2}	0.34888×10^{-4}	-0.837030×10^{-7}
CO2-N2-N2	0.552066×10^{-2}	-0.168609×10^{-4}	0.157169×10^{-7}
CO2-CO2-N2	0.358783×10^{-2}	0.806674×10^{-5}	-0.325798×10^{-7}

Los coeficientes viriales restantes necesarios para el cálculo del factor de compresibilidad de un gas, son las cantidades referentes a los equivalentes hidrocarbonados, CH. Los términos iterativos del segundo y tercer coeficiente virial para los equivalentes hidrocarbonados, CH deben ser calculados del valor calorífico molar ideal Gross del equivalente hidrocarbonado (H_{CH} en KJ/mol a 25°C y 0.101325 MPa). EL valor calorífico molar puede ser determinado por uno de dos métodos que son expuestos en el apéndice B del reporte AGA 8 GROSS. En este proyecto, como se expuso anteriormente se utilizará el método dos.

El segundo y tercer coeficiente virial iterativo del equivalente hidrocarbonado se obtiene de las ecuaciones 18 y 19, respectivamente.

$$B_{CH-CH} = B_0 + B_1 H_{CH} + B_2 H_{CH}^2 \quad (\text{ecuación 18})$$

$$C_{CH-CH-CH} = C_0 + C_1 H_{CH} + C_2 H_{CH}^2 \quad (\text{ecuación 19})$$

Donde B_0 , B_1 , B_2 , C_0 , C_1 , y C_2 son funciones dependientes de temperatura, definidas así:

$$B_i = b_{i0} + b_{i1} T + b_{i2} T^2, \quad i = 0, 1, 2 \quad (\text{ecuación 20})$$

$$C_i = c_{i0} + c_{i1} T + c_{i2} T^2, \quad i = 0, 1, 2 \quad (\text{ecuación 21})$$

Las constantes de las ecuaciones 20 y 21 son dadas en la Tabla 8. La temperatura T esta dada en grados Kelvin.

Tabla 8. Coeficientes viriales B_i y C_i para el equivalente hidrocarbonado, CH.

	i	b_{i0}	b_{i1}	b_{i2}
B_0 (dm^3/mol)	0	-0.425468	0.2865×10^{-2}	-0.462073×10^{-5}
B_1 (dm^3/KJ)	1	0.877118×10^{-3}	-0.556281×10^{-5}	0.881510×10^{-8}
B_2 ($\text{dm}^3 \text{ mol}/\text{KJ}^2$)	2	-0.824747×10^{-6}	0.431436×10^{-8}	$-0.608319 \times 10^{-11}$
	i	c_{i0}	c_{i1}	c_{i2}
C_0 (dm^6/mol^2)	0	-0.302488	0.195861×10^{-2}	-0.316302×10^{-5}
C_1 (dm^6/mol^2)	1	0.646422×10^{-3}	-0.422876×10^{-5}	0.688157×10^{-8}
C_2 (dm^6/mol^2)	2	-0.332805×10^{-6}	0.223160×10^{-8}	$-0.367713 \times 10^{-11}$

El segundo coeficiente virial de interacción para el equivalente hidrocarbonado CH, con nitrógeno N₂ es calculado usando la relación:

$$B_{N_2-CH} = (0.72 + 1.875 \cdot 10^{-5} (320 - T)^2) (B_{N_2-N_2} + B_{CH-CH})/2 \quad (\text{ecuación 22})$$

Donde T está dado en kelvins. Para el equivalente hidrocarbonado CH, con el dióxido de carbono CO₂, la relación es:

$$B_{CO_2-CH} = -0.865(B_{CO_2-CO_2} B_{CH-CH})^{1/2} \quad (\text{ecuación 23})$$

El tercer coeficiente virial iterativo para el dióxido de carbono y nitrógeno con el equivalente hidrocarbonado son calculados así:

$$C_{N_2-CH-CH} = (0.92 + 0.0013 (T - 270)) (C_{CH-CH-CH}^2 C_{N_2-N_2-N_2})^{1/3} \quad (\text{ecuación 24})$$

$$C_{N_2-N_2-CH} = (0.92 + 0.0013 (T - 270)) (C_{N_2-N_2-N_2}^2 C_{CH-CH-CH})^{1/3} \quad (\text{ecuación 25})$$

$$C_{CO_2-CH-CH} = 0.92 (C_{CH-CH-CH}^2 C_{CO_2-CO_2-CO_2})^{1/3} \quad (\text{ecuación 26})$$

$$C_{CO_2-CO_2-CH} = 0.92 (C_{CO_2-CO_2-CO_2}^2 C_{CH-CH-CH})^{1/3} \quad (\text{ecuación 27})$$

$$C_{CO_2-N_2-CH} = 1.10 (C_{CO_2-CO_2-CO_2} C_{N_2-N_2-N_2} C_{CH-CH-CH})^{1/3} \quad (\text{ecuación 28})$$

Si se desconoce la densidad, la ecuación iterativa para su cálculo conociendo la presión del flujo, es obtenida por sustitución de la ecuación 29 en la ecuación 11a.

$$d = P/ZRT \quad (\text{ecuación 29})$$

resultando la ecuación 30:

$$P = dRT(1 + B_{mix}d + C_{mix}d^2) \quad (\text{ecuación 30})$$

La implementación de las normas AGA 7 y AGA 8 en el lenguaje de programación de un sistema digital, hacen parte del *firmware* del computador de flujo desarrollado, el cual se describe en el Capítulo Tres del presente texto.

1.2.3 Estándar API Capítulo 21: Este manual titulado Medición de Flujo empleando sistemas de medida electrónica ("Flow Measurement Using Electronic Metering Systems") fue publicado en 1993 por el Instituto Americano de Petróleo (API). Tiene 2 secciones importantes, la primera trata de la medición electrónica de gas, y la segunda de la medición electrónica de líquidos.

El reporte inicia con una exposición de los elementos que conforman un sistema de medición, haciendo la clasificación de los dispositivos primarios, secundarios y terciarios.

Este estándar ofrece los requerimientos mínimos que debe tener un producto de medición electrónica de gas. Estas reglas se aplicaron al computador de flujo y se resumen como sigue:

- El volumen acumulado en un intervalo dado se calculará como la sumatoria del volumen registrado en intervalos de tiempo más pequeños (dt).
- La mínima frecuencia de muestreo para una variable de entrada dinámica deberá ser 1 vez cada 5 segundos (0.2 Hz). Durante los 5 segundos se pueden acumular varias muestras, para luego ser promediadas por alguna de las técnicas existentes como la ventana móvil. En este proyecto se escogió un período de muestreo de 50ms y una intervalo de promediado de 2s. De esta manera se tiene una frecuencia de muestreo de 0.5Hz, que está por encima del mínimo exigido por el estándar.
- Cuando la salida del generador de pulsos de la turbina tiene una frecuencia menor que 0.2 Hz, las variables de entrada pueden ser muestreadas una vez por cuenta o pulso.
- Cuando no hay detección de flujo (cero pulsos) durante el período de integración de volumen, las variables muestreadas en este período deberán descartarse de los promedios.
- El período de integración de volumen no deberá exceder de 1 hora. En el proyecto se escogió de 15 minutos.
- Los períodos mínimos exigidos para los reportes son de 1 hora y 1 día. Es decir los reportes por lo menos deberán tener el promedio equivalente de cada hora de las variables como volumen, velocidad de flujo, presión, temperatura, etc.
- Deberán existir rutinas de calibración para los transductores de presión y temperatura del equipo. En este proyecto se realizaron estas rutinas y fueron empleadas antes de la instalación final del prototipo, junto con un baño térmico y un peso muerto para la calibración de temperatura y presión respectivamente.
- El teclado del equipo deberá permitir la configuración de las variables teóricas del sistema como la composición del gas, la presión y temperatura base o contractuales, la presión atmosférica del lugar y el factor K de la turbina.
- El visualizador deberá mostrar como mínimo la presión y temperatura actual del fluido, así como el volumen acumulado corregido y sin corregir del día además de la velocidad del flujo de gas. La corrección se refiere a la multiplicación de lo leído en bruto (sin corregir) por los factores de compresibilidad del gas y factores base o contractuales de la operación.
- El estándar especifica un mínimo de tiempo de retención de los datos de medición por 2 años. En nuestro caso, la retención que brinda la batería de litio es de 7 años.

1.3. INSTRUMENTACIÓN SECUNDARIA: SENSORES.

Un sistema de medición de gas con turbina, requiere de medidas adicionales de presión, y temperatura para corregir los volúmenes del medidor. En la industria del gas se emplean RTDs (Detectores Resistivos de Temperatura) para la medida de temperaturas, y transmisores de presión estática para la medida de presiones.

1.3.1. Detectores Resistivos de Temperatura. El termómetro de resistencia es muy usado debido a su exactitud y simplicidad. Se pueden detectar cambios en temperatura de 0.03°F (0.02°C) en procesos industriales. Opera a través del cambio en la resistencia de un metal con la temperatura. Usualmente el metal es platino, cobre o níquel. Para un metal puro, la relación entre la temperatura y la resistencia se expresa como una serie de expansión en temperatura con coeficientes constantes:

$$R_{\text{temp}} = (1 + aT + bT^2 + cT^3 + \dots)R_{\text{ref}} \quad (\text{ecuación 31})$$

donde,

R_{ref} = resistencia en ohms a una temperatura de referencia, usualmente 0°C (32°F).

R_{Temp} = resistencia medida a una temperatura T.

$a = 3.90802 \times 10^{-3}$

$b = -5.802 \times 10^{-7}$

$c = -4.2735 \times 10^{-12}$

El platino es el material más adecuado desde el punto de vista de precisión y extrema linealidad, aunque es el más costoso en comparación con el cobre y el níquel.

En la industria del gas se emplean los RTD Pt100, que son RTDs de platino cuya resistencia a 0°C es 100ohms. Las RTDs de platino se modelan de forma lineal con una precisión de 0.01 °C, empleando la ecuación 31 pero con los siguientes valores de las constantes: $a = 3.85 \times 10^{-3}$ y $b = c = 0$.

La linealidad de las RTD's es mayor en el rango de -100 a +100 °C. Como las estaciones de medición de gas operan en un ambiente promedio de 5 a 45°C, se asegura aún más la linealidad en temperatura.

En la mayoría de aplicaciones se busca evitar la exposición directa del sensor al fluido de proceso. Los termopozos se usan para proteger al sensor contra daños por corrosión, erosión y abrasión en procesos de alta presión; de igual manera protegen al sensor de daño físico durante su manipulación y operación normal. En la Figura 4, se muestra un detector resistivo de temperatura.

Figura 4. Detector resistivo de temperatura.

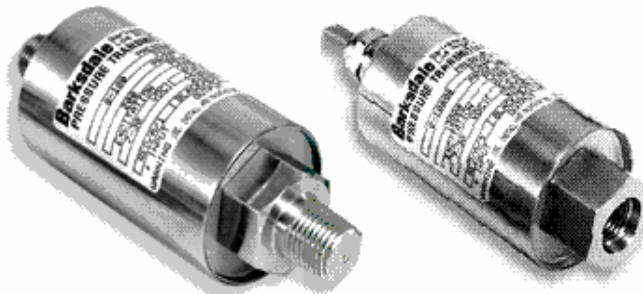


Como se detallará en el Capítulo Dos, en este proyecto se seleccionó una RTD de platino con rango entre 0 y 70°C, el cual es apropiado para el city gate de Barranca cuya temperatura máxima será 45°C y mínima 8°C.

1.3.2. Transmisores de presión. La presión es una de las variables más importantes en instrumentación. Esta definida como fuerza por unidad de área. Para un gas o un líquido, la presión en cualquier punto es igual en cualquier dirección, ésta se expresa usualmente en libras por pulgada cuadrada (psi) en el sistema inglés y en kilo Pascales (kPa) en el sistema métrico.

Para medir presión, se utilizan transmisores (Figura 5) que dependen del tipo de salida que se desee (voltaje o corriente).

Figura 5. Transductores de presión.



Como se detallará en el Capítulo Dos, se seleccionaron sensores de 0-300 psig, pues en el city gate de Barranca se maneja una presión constante regulada de 250 psig, que alcanza un máximo de 270 psig en condiciones extremas. La mayoría de transductores comerciales permiten también una sobrepresión momentánea de dos veces su valor nominal.

1.4. INSTRUMENTACIÓN TERCIARIA: COMPUTADOR DE FLUJO .

En un sistema de medición de gas, el computador de flujo hace parte de la instrumentación terciaria, y en él se desarrollan los cálculos AGA necesarios para la corrección del volumen de gas. A continuación se hace una revisión del estado del arte de estos equipos encontrados en el momento del inicio del presente proyecto.

1.4.1. Evolución tecnológica de los computadores de flujo. El interés original en un sistema de cálculo electrónico de flujo apareció del requisito de proveer datos instantáneos en tiempo real para la industria del gas.

Inicialmente, el objetivo de la industria del gas natural era la eliminación de los problemas asociados con el uso de cartas para medidores de orificio. Estos problemas se debían básicamente a la demora que involucraba (entre cinco y ocho días) el envío postal, el procesamiento de la carta y los cálculos de volumen. Además, se requería un programa de computador sofisticado para desarrollar estos cálculos.

1.4.2. Computador de flujo análogo. Los primeros computadores de flujo fueron dispositivos análogos, donde los voltajes proporcionales al flujo se conectaban a la entrada de amplificadores operacionales. La salida resultante era un voltaje proporcional a la velocidad del flujo que podía ser leído en un medidor o registrador, o enviado por telemetría a otro lugar.

Los computadores de flujo análogos no deben ser interpretados como dispositivos computacionales inferiores. Los circuitos análogos modernos dan excelente exactitud en cómputo y ofrecen estabilidad en un ambiente de temperatura variable. La mayoría de las unidades son de tamaño pequeño y de costo competitivo. Estos equipos constan de un sistema de acondicionamiento de señal de entrada, un circuito multiplicador/divisor y un extractor de raíz cuadrada para proveer el cálculo del volumen, y normalmente están disponibles para platina de orificio con transductores DP (*Differential Pressure*). Algunas unidades tienen una entrada de un medidor de densidad para medida de masa, en lugar de las entradas usuales de presión y temperatura.

1.4.3. Computador de flujo digital. La combinación de circuitos con transistores en un simple dispositivo llamado circuito integrado, fue el hecho que generó el mayor auge en el campo de la electrónica. A partir de esto, muchos circuitos sencillos de transistores, como las puertas lógicas, pueden estar contenidos en una oblea de silicio. Estas obleas son referidas con frecuencia como *chips*. Así, circuitos complejos, tales como amplificadores operacionales, se pueden montar en un solo *chip*. Las puertas lógicas se han combinado para formar *flip-flops* que son usados como dispositivos de memoria o contadores. Existen combinaciones de *flip-flops* que se han puesto en un circuito integrado para desarrollar funciones específicas, lo que principalmente permitió el desarrollo de los computadores de propósito especial.

1.4.3.1. Computador de flujo basado en circuitos digitales. Mientras los computadores de flujo análogos empleaban circuitos integrados lineales como los amplificadores operacionales, los computadores de flujo digitales usan circuitos integrados digitales tales como: puertas lógicas, *flip-flops*, contadores, etc., para controlar el funcionamiento de su calculadora interna.

El advenimiento de esta electrónica digital, mejoró la exactitud de los cálculos de flujo *in situ*. Con los computadores digitales, las señales de corriente o voltaje que

representan los datos de flujo, se colocan a la entrada de convertidores análogo/digitales. Después que las entradas son digitalizadas se convierten a unidades de ingeniería empleadas en la industria del gas ($^{\circ}\text{F}$ para temperatura, y psia para presión), y los cálculos se hacen de forma digital. Los primeros procesadores digitales llevaban a cabo los cálculos como los haría un individuo que usa un portátil y así la secuencia de operaciones se controlaba según la disposición de los componentes electrónicos en los circuitos. La salida del computador digital era la velocidad de flujo expresada en CFH (pies cúbicos por hora) además de una salida de pulsos que representaba unidades volumétricas (pies cúbicos). El valor también se podía convertir a una forma análoga a través de un conversor digital/análogo y así disponer de una señal de corriente o tensión proporcional a las variables de velocidad o de volumen.

1.4.3.2. Computador de flujo basado en microprocesador. La era de los microprocesadores no solo ha introducido muchos cambios en las aplicaciones de medición de gas, sino que ha permitido un gran avance tecnológico en todos los campos. Y no es para menos, pues los microprocesadores son considerados como "un computador en un *chip*".

El advenimiento del microprocesador (μP) mejoró todo el sistema, ya que el μP es controlado por un programa o grupo de instrucciones más que por el diseño del circuito, como ocurría en los primeros calculadores digitales. Esta característica de los computadores de flujo basados en microprocesador, los hace más flexibles que los primeros computadores digitales y los dota de una potencia única para manejar procesos y cálculos más complejos.

Hoy en día, la mayoría de fabricantes de computadores de flujo utilizan microprocesadores. Todos los computadores de flujo de gas están dedicados a resolver las ecuaciones básicas de flujo. La diferencia entre los equipos está en la manera de visualizar resultados, las técnicas de programación, y la forma como se manejan los datos de entrada y salida. Estas características se deben tener en cuenta cuando se quiera comprar un computador de flujo de gas con el fin de seleccionar la unidad más adecuada para las aplicaciones.

Como el microprocesador es programable, su *software* puede ser expandido para incluir otras características además del cálculo del flujo, como rutinas para monitorear entradas análogas. En caso de falla del transductor, se puede iniciar alguna acción, tal como desplegar el estado del instrumento o aproximar la entrada hasta que el dispositivo sea reparado. El microprocesador se puede usar para controlar varios dispositivos en una estación de medición. Por ejemplo, la salida del microprocesador se puede conectar a un actuador de una válvula de control para controlar la velocidad del flujo o la presión.

1.4.3.3. Software y memoria. Tal como los computadores análogos se controlan por el diseño del circuito, el microprocesador debe ser controlado por su programa. El microprocesador debe ser visto como la unidad central de proceso (CPU). El programa o *software* es una secuencia de instrucciones que desarrollan una tarea específica. Los computadores análogos y los primeros computadores digitales

fueron controlados por *hardware*. Es obvio que es más fácil cambiar la secuencia de ejecución de instrucciones que modificar el diseño del circuito de un computador.

Los computadores digitales de flujo pueden ser programados por el fabricante para efectuar muchos programas en milésimas de segundo. Como el microprocesador es controlado por *software*, cada variable puede ser manipulada con su propia ecuación y luego puesta de vuelta en la ecuación de flujo. Los programas pueden estar diseñados solo para una ecuación específica. Esto permite al computador de flujo basado en microprocesador actuar como un instrumento dedicado al propósito de la medición de flujo.

Así como el microprocesador debe tener un programa para controlarlo, también debe tener un medio para almacenarlo. Para ello, frecuentemente se emplea una memoria de solo lectura ROM (Read-Only Memory). Como su nombre lo indica, esta memoria es fija y solo puede ser leída. Hay muchos tipos de ROM's disponibles:

- ROM hecha a la medida, en la cual el programa es incorporado dentro de la estructura de la ROM; éste no puede ser cambiado o alterado de ninguna forma.
- ROM programable o PROM.
- ROM borrable o EPROM, que se programa eléctricamente y puede ser borrada exponiendo el *chip* de silicio a luz ultravioleta. Luego puede ser reprogramada.
- Actualmente se usa también la memoria EEPROM o E²PROM que puede ser borrable eléctricamente.

Otro medio de almacenar el programa es usar una memoria de acceso aleatorio o RAM. Además de leer de la RAM, el microprocesador puede escribir en ella. Los dispositivos RAM no mantienen sus datos cuando se les retira la energía, por esta razón, no son adecuados para el almacenamiento permanente del programa. La RAM es útil para almacenamiento de datos dinámicos usados por el microprocesador durante la ejecución del programa.

1.4.3.4. Consumo de potencia. Un factor en la resistencia al uso de los instrumentos basados en el microprocesador fue la falta de confiabilidad, provocada por muchas fuentes. En la mayoría de casos el dispositivo disipaba significativas cantidades de potencia, con el problema que cuando los usuarios lo ubicaban en ambientes no controlados, la alta disipación de potencia acoplada con la alta temperatura del ambiente ocasionaban fallas.

Actualmente, los desarrollos tecnológicos en este aspecto han hecho que un instrumento que formalmente requería 40 watts para la operación, requiera solo 10 watts. Esta disminución de potencia viene acompañada con un aumento significativo en la capacidad. Además, usando *chips* CMOS (circuitos integrados de bajo consumo de corriente), es posible producir instrumentos basados en microprocesador que requieren mucho menos de 1 watt para operar. Gracias al uso

de tecnología electrónica de potencia extremadamente baja, se han removido muchas objeciones encontradas en los avances tecnológicos alcanzados. Los microprocesadores de tecnología avanzada que utilizan circuitería CMOS, son útiles para medición de flujo de gas natural en sitios donde no hay suficiente suministro de energía.

1.4.4. Estado del arte de los computadores de flujo. Actualmente, en aplicaciones de transferencia de custodia se ha logrado tiempo real, alta exactitud, medición y control. Como el computador de flujo es considerado como “la caja registradora” del sistema de medición de gas, es primordial que mantenga o adquiera la capacidad de interconectarse a diversos dispositivos.

La funcionalidad básica de los computadores de flujo tiende a permanecer constante de generación en generación. Sin embargo, se han hecho avances significativos en los últimos años en las siguientes características:

1.4.4.1. Capacidad de cómputo. Como muchos productos de alta tecnología, los computadores de flujo industriales deben la mayoría de sus mejoras y crecimiento de mercado a la disponibilidad de microprocesadores más rápidos y componentes de memoria de mayor capacidad a costos más bajos. Actualmente las CPUs de 16 bits corriendo a velocidades mayores a los 20 MHz son más que una excepción, la norma para estos dispositivos. Un ejemplo son los computadores de flujo American Meter, que emplean el microprocesador PCF80C552 de 16 bits.

La memoria flash en la tarjeta se mide ahora en megabytes mas que en kilobytes, como era hace algunos años. El incremento en la potencia de los computadores, la velocidad, y su capacidad de memoria son los factores que guían al desarrollo del computador de flujo multipropósito.

Estos avances permiten expandir el diseño básico del computador y extender su vida útil. Seleccionando los microprocesadores y sus periféricos, fácilmente se puede adaptar el computador de flujo básico a nuevas aplicaciones, a medida que ellas aparecen. Actualmente es factible pensar en términos de computadores de flujo hechos al gusto del comprador, producidos en pequeñas cantidades para un usuario o mercado específico.

1.4.4.2. Opciones de entrada/salida (E/S). En la década de los noventa eran mínimos el número y tipo de señales de E/S que soportaba un computador de flujo. Hoy en día, aún existen computadores con opciones de E/S limitadas, pero la tendencia es ir aumentando estas posibilidades. La mayoría de los computadores de flujo nuevos soportan una variedad de entradas directas de sensores de flujo, tales como pulsos de DC, TTL, colector abierto, captación magnética, y contacto seco. Las entradas directas de RTD son características estándar para unidades que compensan temperatura, así como lo son las señales análogas para entradas de presión y densidad. En muchos casos, el *software* de instalación permite asignar y configurar las opciones de E/S sin tener que cambiar el *hardware*.

Actualmente, existen unidades que pueden calcular y medir el flujo de gas de más de un medidor primario, es decir, un computador de flujo llamado “de dos brazos” que puede leer y computar paralelamente la información de dos líneas de gas, cada una con sus respectivos transductores de presión y temperatura. De esta manera se aprovechan las capacidades multitarea de muchas de estas unidades de procesamiento para llevar a cabo la medición simultánea de flujo en un mismo computador de flujo. Naturalmente, estos equipos son más costosos no solo porque requieren el doble de instrumentación, sino también deben tener mayores velocidades de procesamiento, el doble de memoria de almacenamiento, etc.

1.4.4.3. Comunicaciones digitales^{viii}. Históricamente las comunicaciones digitales para computadores de flujo han estado limitadas a RS-232 o RS-485. Los nuevos computadores de flujo pueden tener también interfaces con redes industriales estándar. Hasta la fecha, el protocolo mas común con el que se hace interface es Modbus, pero muchos fabricantes están desarrollando soporte para Profibus, DeviceNet, y Fieldbus. También se ha empezado un trabajo preliminar en redes Ethernet. Estas nuevas opciones representan un gran salto en la capacidad e interoperabilidad de los computadores de flujo.

Además de las opciones de comunicación cableadas, una tendencia creciente es la disponibilidad de enlaces para comunicaciones infrarrojo (IR), que permiten la operación y configuración remota de computadores de flujo. Estos enlaces están aún en desarrollo.

El protocolo empleado comúnmente para transmisión IR es el IrDA, desarrollado por la Asociación de Dispositivos Infrarrojos. Este es el mismo protocolo usado en la mayoría de productos electrónicos. La aplicación típica del IR en el mercado industrial ha sido el uso en áreas inflamables o peligrosas. Un comunicador IR portátil se usa para establecer enlaces con computadores de flujo montados en encapsulados a prueba de explosión. Estos computadores de flujo tienen una ventana IR en su panel frontal a través de la cual reciben y transmiten al dispositivo portátil.

Otra opción que aparece en los nuevos computadores de flujo es el *modem* telefónico para conexiones a través de líneas en tierra, celulares o enlaces de radio. Típicamente estos *modem* se ofrecen como unidades externas, para evitar el aumento del costo de una unidad, ya que es más económico proveer un módem estándar disponible que construir uno que se ajuste a la parte interna del computador de flujo. Otro aspecto que influye especialmente a la hora de comercializar un equipo es la necesidad de cumplir los límites EMI (Interferencias Electro-Magnéticas) y RFI (Interferencias de Radio Frecuencia) dados por ciertos estándares como:

- **CE:** Conformidad Europea (*European Conformity*)
- **CSA:** Asociación de Estándares Canadienses (*Canadian Standards Association*)
- **CENELEC:** Comité Europeo para Estándarización Electrotécnica. (*European Committee for Electrotechnical Standardization*)

- **BASEEFA:** Servicio de Aprobación Británico para Equipo Eléctrico en Atmósferas Inflamables. (*British Approvals Service for Electrical Equipment in Flammable Atmosphere*).
- **UL:** Laboratorios Aseguradores. (*Underwriters Laboratories*)
- **FM:** Fábrica Común. (*Factory Mutual*)

De acuerdo a los requisitos del país o de la empresa distribuidora de gas se exige el cumplimiento de alguna de estas normas. En otros países como Colombia por ahora no se exige alguna en especial y el parámetro de mayor interés es la exactitud del equipo.

1.4.4.4. Adquisición y almacenamiento de datos. Gracias al reducido costo de las memorias de computador, muchos fabricantes de computadores de flujo ofrecen almacenamiento básico de datos como una característica estándar u opcional. Esta memoria usualmente esta en el rango de 256–512 KB. Aunque esta característica es pequeña en comparación con las de almacenamiento industrial, esta capacidad es adecuada para la mayoría de aplicaciones de computadores de flujo. Los datos pueden ser descargados cada cierto tiempo a un sistema de control y/o en una memoria mas grande mediante enlaces realizados por algunas de las opciones de comunicación.

La configuración de la función de almacenamiento de datos se lleva a cabo por medio del *software* de instalación que generalmente se provee con el computador. La mayoría de las entradas, salidas o parámetros calculados pueden ser almacenados, y la frecuencia de almacenamiento es determinada por el usuario. Las funciones especiales (p.e. almacenamiento de valores pico) están disponibles en la mayoría de sistemas de almacenamiento.

1.4.4.5. Tecnología de visualizadores. Uno de los primeros objetos de presentación que se observan en un computador de flujo es el visualizador. Tradicionalmente estos han sido LEDs (diodo emisor de luz) rojos o LCDs (visualizador de cristal líquido). Los mas avanzados tienen iluminación como característica estándar o como una opción. Hoy en día, las opciones para selección de visualizadores ha crecido significativamente. Los LEDs aún están disponibles pero han sido superados por los LCDs con múltiples líneas y cadenas de caracteres mas grandes, así como por los *displays* de fluorescentes de vacío (VFD) y los LCDs totalmente gráficos.

En general, se deben considerar los LEDs para las aplicaciones mas básicas o para situaciones donde la luz ambiente no contribuye a los LCDs. La luz directa y brillante del sol es, con frecuencia, un problema para las unidades LCD y VFD, pero estos son casi una norma en computadores de flujo sofisticados. Estos visualizadores proveen mayor flexibilidad en el tipo, variedad y volumen de información visualizada.

1.4.4.6. Autonomía de operación. Los computadores de flujo están capacitados para emplear la alimentación de la red eléctrica de la estación donde son instalados para su funcionamiento. Además presentan bajo consumo de potencia y

generalmente incluyen una batería recargable en paralelo con la entrada de potencia que les garantiza autonomía de operación ante la pérdida de la energía eléctrica. Adicionalmente, poseen una batería de litio (esta no es recargable) para el respaldo de los datos depositados en la memoria RAM, estas memorias puede almacenar datos históricos por un período de hasta siete años con el soporte de estas baterías.

El computador de flujo diseñado en el presente proyecto, tiene soporte para la conexión de una batería externa, y de una pila de litio de respaldo para la memoria RAM de almacenamiento. No obstante, en las instalaciones donde se realizaron las pruebas finales (Barrancabermeja), poseen UPS´s de respaldo para las interrupciones de energía eléctrica.

1.4.4.7. Métodos de cálculo. Los computadores de flujo estandarizados, normalmente emplean la norma AGA8, la cual contiene las ecuaciones de estado aceptadas comercialmente para el cálculo del factor de compresibilidad (factor Z), y de supercompresibilidad para gas natural y otros gases de hidrocarburos.

Para la corrección de flujo de gas empleando un medidor primario tipo turbina, se utiliza la norma AGA7. Esta publicación técnica provee las prácticas actuales y los métodos aceptados para la medición de gas con medidores de turbina de flujo axial, así como la información relacionada con su instalación, mantenimiento y calibración.

1.4.5. Generalidades en el diseño de un computador de flujo. El diseño de un computador de flujo involucra gran desarrollo electrónico y de *software*. Aunque sus partes están bien definidas, se requiere implementarlas con el mayor rigor electrónico posible, buscando la mejor exactitud y teniendo en cuenta todos los detalles que exige la creación de un sistema de medida totalmente autónomo.

A continuación se describe la estructura básica presente en la mayoría de computadores de flujo:

1.4.5.1. Unidad de medición. Consta de los sensores y transmisores según el instrumento primario de medida, así como de la adecuación de las señales que de estos surjan. Esta unidad debe tener gran resolución y exactitud pues de ella depende un gran porcentaje del error de incertidumbre total del sistema.

1.4.5.2. Unidad de procesamiento y control. A esta unidad se le asigna la tarea de coordinar todo el sistema para que funcione como un dispositivo autónomo de campo. Así mismo, procesa las variables de medición y efectúa los cálculos siguiendo los parámetros dados por los estándares de medición del AGA (*American Gas Asociation*). Consta tanto de *hardware* como de *software*, a este último se le llama *firmware* ya que es un *software* de bajo nivel, y a su vez para diferenciarlo del *software* de visualización y presentación que se desarrolla en un PC (*Personal Computer*) o *laptop* convencional.

1.4.5.3. Unidad de interfase. Ofrece al usuario la posibilidad de leer las variables de interés, e introducir o modificar datos de configuración del sistema. Consta de un visualizador y de un teclado. Estas partes facilitan el uso y brindan comodidad al operador, así como seguridad y larga vida al dispositivo.

1.4.5.4. Unidad de almacenamiento. Tiene la función de registrar y guardar todas las variables de interés por el tiempo que se estime conveniente. Además, evita la pérdida o corrupción de datos, y garantiza el suficiente tamaño de memoria para albergarlos por el tiempo asignado.

La memoria que requiere un computador de flujo para almacenamiento de datos oscila típicamente entre 256KB y 512KB, pues periódicamente (cada mes) los operarios van descargando dicha información a un disco duro. En la sección 2.2.1.3. se presenta el cálculo para determinar la memoria destinada para el computador de flujo del presente proyecto.

1.4.5.5. Unidad de *software*. Esta unidad es externa al computador de flujo pero complementaria. Es la que permite mediante un *software* de comunicación y presentación, descargar a un PC los datos corregidos por el computador de flujo y desplegarlos de una forma amena al usuario. Adicionalmente, puede involucrar una base de datos y un manejo estadístico de la información.

2. *HARDWARE* DEL COMPUTADOR DE FLUJO.

La selección de cada uno de los dispositivos que conforman la estructura básica del computador de flujo, es fundamental para poder obtener un equipo de alta confiabilidad para un sistema de medición de gas natural. En este capítulo se detallan la unidad de medición, y la unidad de control, procesamiento, y almacenamiento (CPU). Además se muestra el encapsulado, donde se integran estas unidades.

2.1. UNIDAD DE MEDICIÓN

La unidad de medición del computador de flujo es la encargada de adquirir y acondicionar las señales físicas que se requieren para determinar el flujo de gas natural en una tubería. Esta unidad trabaja en conjunto con la instrumentación primaria y secundaria.

La unidad de medición del computador para medidor tipo turbina esta conformada por:

- Entrada de pulsos digitales provenientes del medidor primario (turbina, ver sección 1.1.). Este medidor primario hace parte de la línea de gas. La señal de salida de este medidor no requiere acondicionamiento, sólo se realiza aislamiento con el fin de separar las masas del medidor primario y la tarjeta de acondicionamiento.
- Sensores de presión absoluta y temperatura.
- Tarjeta de acondicionamiento de señal.

Antes de detallar la unidad de medición, es necesario identificar el tipo de zona donde los computadores de flujo operan, ya que existen normativas que delimitan el proceso de selección de la instrumentación y componentes en áreas peligrosas o inflamables.

Las plantas o industrias son clasificadas en zonas de diversas clases, divisiones y grupos. Las clases definen el tipo de peligro, la división indica que tan presente está el peligro y el grupo determina la sustancia que causa el riesgo. La Tabla 9, muestra la clasificación de las áreas industriales según su peligrosidad.

Tabla 9. Clasificación de las zonas industriales.

Clases	Clase I: Vapores o gases inflamables Clase II: Polvos combustibles (ejemplo: polvo del carbón) Clase III: Fibras inflamables (ejemplo: papel)
División	División 1: El peligro existe bajo condiciones normales de operación, o >100hrs al año. División 2: El peligro existe en condiciones anormales de operación, o >10 y <100 hrs al año.
Grupo	A, B, C, D : Gases E, F, G: Polvo

Los gases son agrupados basados en la cantidad de energía requerida para la ignición. Cualquier clase de equipos, como los computadores de flujo, deben ser también clasificados y seleccionados (aparte de sus especificaciones de medición) de acuerdo al grupo de gases y vapores que vayan a estar presentes donde el equipo será instalado.

A continuación se da un ejemplo de la energía de ignición de algunos gases:

- Metano Energía de Ignición de 320 Microjulios.
- Propano Energía de Ignición de 300 Microjulios.
- Etileno Energía de Ignición de 160 Microjulios.
- Hidrógeno Energía de Ignición de 40 Microjulios.

En el presente proyecto, el computador de flujo se instaló en el *city gate* de Barrancabermeja, que está clasificado como Clase I, División 1, Grupo D (gas predominante metano). Existen dos tipos principales de protección contra explosiones en estos ambientes: la seguridad intrínseca y el *explosionproof*.

Para que una explosión ocurra deben estar presentes tres elementos: material combustible (gases, vapores, etc.), oxígeno, y energía eléctrica o térmica (de valor suficiente). Si alguno de estas condiciones no están presentes, la explosión no sucederá. La seguridad intrínseca es un método de protección que previene que haya la suficiente cantidad de energía eléctrica o térmica para generar una explosión.

El método de protección de *explosionproof* aconseja encapsular un dispositivo o equipo que no es intrínsecamente seguro de tal forma que si una explosión ocurre permanezca dentro del encapsulado, pero no elimina las causas de la explosión.

Por otro lado, las normas de seguridad intrínseca limitan la energía eléctrica a niveles por debajo del requerido para encender una atmósfera peligrosa específica, de esta forma se hacen innecesarios los encapsulados y conductores a prueba de explosión (*explosionproof*). El único inconveniente es que dado que la seguridad intrínseca opera con baja potencia, sólo instrumentos de reducido consumo eléctrico pueden ser empleados en las áreas peligrosas.

Para equipos de medición o instrumentación, como el que trata el presente proyecto, se hace necesario que el circuito final del computador de flujo sea intrínsecamente seguro, es decir, virtualmente incapaz de generar arcos o chispas que sean capaces de iniciar una explosión de una mezcla peligrosa. Para cumplir esta condición según el estándar europeo para seguridad intrínseca EN50020 ("Electrical apparatus for potentially explosive atmospheres, intrinsic safety"), los aparatos ubicados en zonas altamente peligrosas (Clase I, División 1), como el computador de flujo, deberán tener una alimentación máxima de 30V y un consumo máximo de 100mA. Bajo estas dos condiciones, la instrumentación y componentes electrónicos que se seleccionaron, fueron acompañados del criterio de mínimo consumo de potencia.

2.1.1. Señal del medidor primario. Esta señal, que consiste en un tren de pulsos de frecuencia variable, es generada por el sistema de sensores de flujo instalados junto a la turbina (ver sección 1.1.3.). La frecuencia de este tren de pulsos determina la velocidad en pies cúbicos por hora (CFH) de gas que está pasando por la turbina. Así al multiplicar esta velocidad por un tiempo de medida dado se obtiene el volumen de gas que ha pasado en dicho tiempo. El rango de frecuencias para la turbina Elster instalada en Barranca oscila desde 0 hasta 0.041667 Hz que equivalen de forma lineal a una velocidad de flujo de 0 hasta 15000 CFH (pies cúbicos por hora), respectivamente. El computador de flujo recibe este tren de pulsos en una bornera de la tarjeta de acondicionamiento.

2.1.2. Selección de la instrumentación. Esta sección presenta la instrumentación seleccionada para realizar las medidas de presión y temperatura de la línea. Los sensores se seleccionaron de acuerdo a tres parámetros importantes:

- El rango de la variable que se desea medir.
- El bajo consumo de potencia, según los criterios expuestos al inicio de la presente sección.
- La buena exactitud del sensor, con el fin de cumplir con este parámetro en el equipo final.

Para evidenciar la importancia de la exactitud en los sensores, en el Capítulo Cuatro se presenta una simulación de cómo afecta el error en presión y temperatura en el cálculo del volumen de flujo.

2.1.2.1. Transductores de temperatura. Estos sensores miden la temperatura del gas en la línea. Esta medida es útil para el cálculo del flujo de gas real, ya que permite corregir los valores contractuales base de temperatura (60°F), a los valores de flujo reales en la tubería.

El margen de temperatura que se presenta en la línea de gas del *city gate* de Barranca es desde 32°F a 158°F (0 a 70°C).

De las diferentes opciones para medir temperatura, los sensores más empleados en la industria del gas son los RTD. Comparándolos con las termocuplas, los RTD tienen las siguientes ventajas independientemente del tipo de montaje:

- Mejor linealidad.
- Mayor sensibilidad en el margen de medida (0 a 70°C).
- No presentan problemas de interferencias termoeléctricas cuando trabajan con corriente continua, las cuales sí afectan los pequeños niveles de señal generados por los termopares.

La otra opción de instrumentación, los termistores, se descartaron debido a que su característica no es lineal.

Por otro lado, el tipo de montaje (2, 3, o 4 hilos) determina el grado de influencia de la resistencia propia de los cables de conexión de la RTD, en la exactitud de la medida. La conexión de 2 hilos es la más inexacta al no compensar el efecto de los cables. En cambio, la conexión de 3 hilos es la más empleada en la industria pues corrige este inconveniente. Finalmente, la conexión de 4 hilos es empleada comúnmente en la calibración de patrones de resistencia en laboratorio. La mayoría de RTDs que se encuentran comercialmente, son de 3 hilos (que se pueden emplear en conexión de 2 hilos). Para este proyecto se trabajó con la conexión en 3 hilos alimentada por corriente. (ver sección 2.1.3.3)

El sensor de temperatura seleccionado fue el RTD-345-400-100 de longitud de inserción de 4.5 pulgadas, fabricado por la empresa *Murphy* (costo: US\$240 c/u). Para el presente proyecto se adquirieron dos sensores de este tipo, el segundo es para un segundo brazo de medición en el computador de flujo.

El RTD tiene las siguientes características

- Salida de 100 Ω para cero grados centígrados.
- Exactitud: $\pm 0,12\%$
- Margen de Operación: hasta 400 °F (204.4°C).
- Termopozo de Acero 304 con margen de operación hasta 7000psi a 70°F.
- Conector Metálico NEMA 4X, NEC** Class 2.
- Construida en Acero inoxidable.

Es importante resaltar la buena exactitud ($\pm 0,12\%$) del sensor *Murphy*, que comparada con la de los termopares (aprox. 0,3% a 0,8%) garantiza una mejor medida final del computador de flujo.

El efecto de un error máximo de $\pm 0.12\%$ en la lectura de temperatura del *City Gate* como el de Barranca, es presentado en el Capítulo Cuatro.

* National Electrical Manufacturer's Asociation -USA

** NEC (National Electrical Code - USA)

Figura 6. RTD PT100, marca *FW Murphy*.



El consumo de una RTDs, al igual que en la resistencia, depende de la corriente o tensión con que se alimente. En este proyecto el RTD fue excitado con una fuente de corriente de $200\mu\text{A}$ lo que produce una tensión de 20mV a cero grados centígrados, señal que es acondicionada para dar los niveles de tensión del conversor análogo-digital. (Ver secciones 2.1.3.3. y 2.1.3.4.)

2.1.2.2. Transductores de presión. Estos sensores se requieren para medir la presión actual del gas en la línea. Esta medida es útil para el cálculo del flujo de gas real, ya que corrige los valores contractuales de presión base (14.73 psia) a los valores de flujo reales en la tubería. Para los cálculos que requieren la presión absoluta de la línea, como es el caso del AGA8, se le adiciona por *software* la presión atmosférica a la lectura del transductor. La presión atmosférica es propia del lugar y es un dato introducido por teclado en el computador de flujo.

El primer parámetro a tener en cuenta en la selección de un sensor de presión, es el tipo de señal de salida; ya que esto determina si es necesario acondicionarla o no. Según lo anterior, se encuentran tres tipos de sensores de presión:

Transductores con salida de Milivoltios, que generalmente son los de menor costo. Debido a su bajo nivel de salida, la señal es susceptible a ruido e interferencias eléctricas. Su uso es aconsejable en casos donde la distancia entre el sitio de medición y el lugar donde se espera recibir el dato de presión, es corta. Así se evita pérdida de señal.

Transductores con salida de Voltios, los cuales traen su salida amplificada, por ello son menos sensibles a las perturbaciones pero generalmente su costo es el doble del anterior.

Transmisores de salida de Corriente, donde la señal de salida es transportada sobre los mismos dos cables que alimentan el sensor. Su salida está amplificada y normalizada de 4-20mA, lo que los hace útiles para casos en que la señal debe ser llevada a largas distancias. Su costo es mayor, pero el sistema de acondicionamiento resulta más sencillo de implementar.

En el prototipo desarrollado, los transductores de presión están montados sobre la caja que contiene el computador de flujo, así la distancia entre su salida y las borneras de conexión es muy corta (aprox. 20cm), esto hacía innecesario un transductor con salida en corriente. Un sensor con salida en tensión (Ej: 0-10V) sería óptimo para conectarlo directamente a la entrada del conversor análogo-digital sin necesidad de acondicionamiento. Sin embargo, al revisar diferentes proveedores, se encontró que el consumo de potencia de estos transductores es de 15mA. Como se emplean dos sensores (uno por cada brazo de medición), se estarían consumiendo 30mA, lo cual es el 30% de la corriente máxima que debe consumir el computador de flujo para que sea intrínsecamente seguro.

Por esta razón, se seleccionarán transductores con señal de salida en miliVolts, los cuales al no tener acondicionamiento interno reducen el consumo de corriente a 2 mA máximo. Al poner 2 de estos transductores (uno para cada brazo de medición), se estaría empleando un 4% de la corriente de consumo máxima permitida de todo el sistema.

Después de definir el tipo de salida, la selección depende del rango de operación en los cuales se va a realizar la medición. Por una parte, no es conveniente seleccionar el transductor cerca de los límites de operación, ya que si se presenta una condición de sobrepresión, muy seguramente se afectará el funcionamiento y se podría causar un daño definitivo. Por otro lado, un transductor muy sobredimensionado reducirá la sensibilidad de la medición al desaprovecharse el rango de medida. El *city gate* de Barranca tiene una presión regulada constante de 250psi, la cual se eleva a 270psi en condiciones extremas, por lo que un sensor con un rango de 0 a 300 psi funciona correctamente. Además, el límite de presión de prueba para este transductor es el doble del nominal, así que se estaría protegiendo el equipo hasta presiones de 600psi.

Otra especificación importante del sensor es su margen de operación de temperatura, el cual debe superar los límites propios del sitio donde va a ser instalado de modo que no presente errores en la salida por causa de temperaturas extremas.

Se seleccionaron dos transductores de presión de la serie 422 de *Barksdale* (costo: US\$370 c/u), uno para cada brazo de medición del computador de flujo.

A continuación se presentan las especificaciones de estos sensores:

- Salida No-Amplificada
- Exactitud del $\pm 0,25\%$
- Estabilidad a largo plazo de $\pm 0.5\%$
- Voltaje de entrada: 10 Volts
- Salida *Full* Escala: $100\text{mV} \pm 1\text{mV}$
- Salida de cero: $0\text{mV} \pm 2\text{mV}$
- Sensibilidad: 10mV/V
- Margen de Presión: 0-300 psi

- Margen de Operación de temperatura: -18 a 71 °C
- Potencia disipada: Máxima 20mW, Mínima 6.6mW
- Peso: 130 gramos
- Construcción de Acero inoxidable

Figura 7. Sensor de presión *Barksdale*.



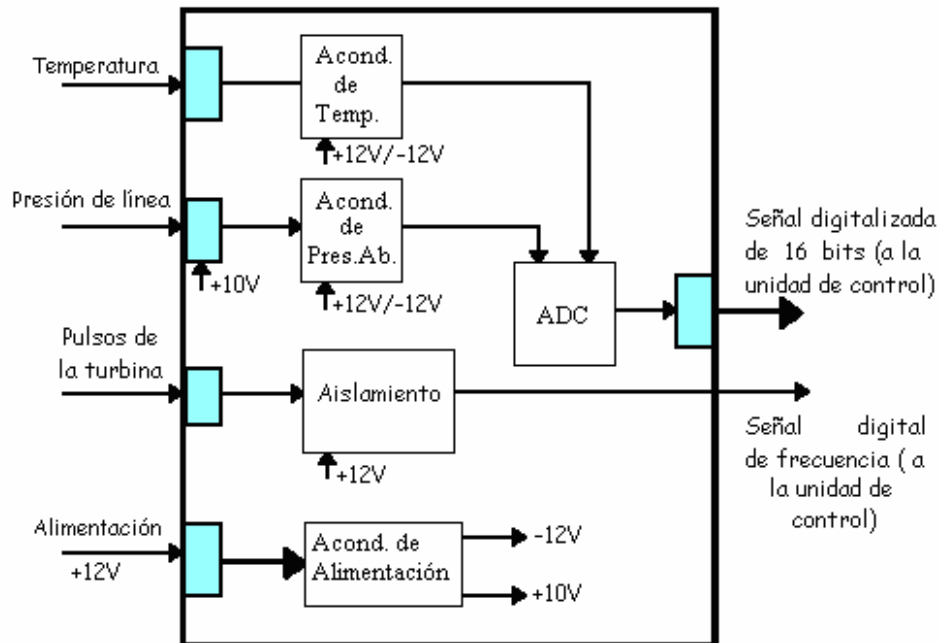
Al igual que se hizo para la RTD, en el Capítulo Cuatro presenta una simulación del error en la medición del volumen ante un error máximo ($\pm 0.25\%$) del transductor de presión Barksdale.

2.1.3. Acondicionamiento de señales. En esta sección se presentan los diferentes circuitos de acondicionamiento empleados para los medidores primarios y para los transductores secundarios. El objetivo de la etapa de acondicionamiento es llevar todas las señales análogas a valores entre 0 y +10V, correspondientes al margen de entrada que maneja el conversor análogo-digital empleado, el cual se describe en la sección 2.1.3.8. Los circuitos de acondicionamiento fueron seleccionados con el requerimiento de bajo consumo de potencia.

La Figura 8 muestra el diagrama de bloques de la tarjeta de acondicionamiento de señal.

La alimentación de los transductores y las tarjetas de acondicionamiento se obtiene a partir de una fuente regulada de +12V, la cual está ya habilitada en las estaciones de Ecogás. Además, se requiere diseñar acondicionamiento de fuente para obtener a partir de ésta, tensiones de +10V para la alimentación de los transductores de presión, y tensiones de -12V para la alimentación negativa de los amplificadores de instrumentación.

Figura 8. Diagrama de bloques de la tarjeta de acondicionamiento de señal.



2.1.3.1. Fuentes de tensión de $-12V$. Para obtener la tensión de $-12V$ a partir de una alimentación de $+12V$, se utiliza un convertidor de tensión.

Entre las múltiples opciones del mercado se seleccionó el convertidor de tensión con reguladores basado en la teoría de condensadores conmutados⁹ LT1054 de *Texas Instruments* (ver Anexo C). Este convertidor presenta características superiores a otros convertidores disponibles en el mercado, entre éstas se encuentran su salida de corriente alta y pérdidas de voltaje significativamente más bajas. Además, posee un esquema adaptativo de manejo conmutado que optimiza la eficiencia sobre un amplio margen de corrientes de salida. Su caída de tensión total a 100mA de corriente de salida es típicamente de 1.1 V. Ésta se mantiene sobre todo el margen de tensión de alimentación de 3.5 V a 15V. Su corriente en estado inactivo es típicamente de 2.5mA.

En la Figura 9 se presenta el diagrama de bloques del circuito LT1054.

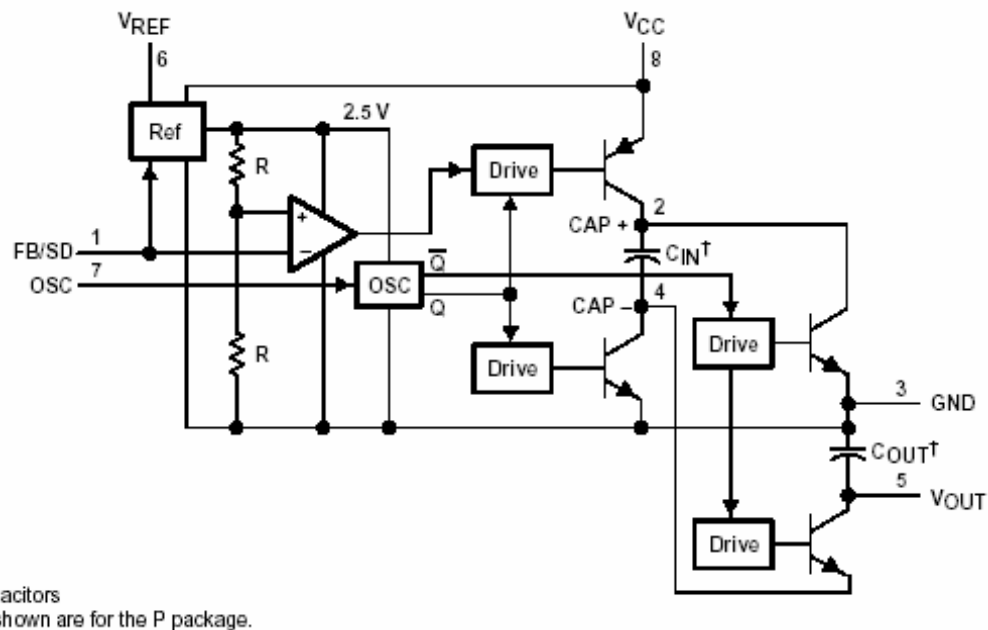
A continuación se presentan las características del LT1054:

- Corriente de salida de 100 mA
- Margen de operación de 3.5 V a 15 V
- Temperatura de operación: -40 a $85^{\circ}C$
- Potencia disipada: Máxima 75mW, mínima 8.75mW
- Amplificador de error y referencia para regulación

⁹ R. Gregorian, K. Martin. "Switched-Capacitor Circuit Design" IEEE Proceedings, Vol. 71, no. 8, pp 941-966, August 1983

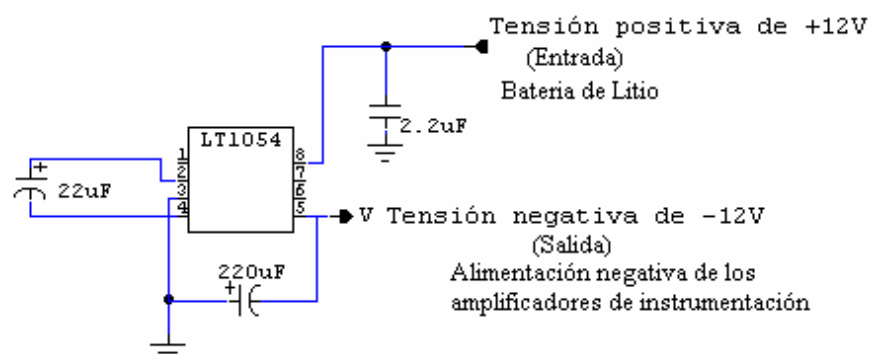
- Sincronización de oscilador externa
- Los dispositivos se pueden conectar en paralelo para aumentar el suministro de corriente.

Figura 9. Diagrama de bloques convertidor del LT1054.



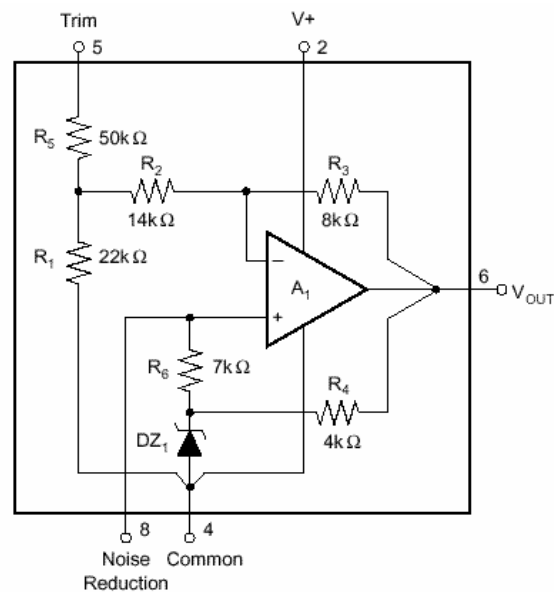
En la Figura 10 se muestra el circuito necesario para obtener una fuente de tensión de -12V a partir de una fuente de +12V. Esta es la configuración de inversor del LT1054, luego la tensión en el pin 5 es de -12V cuando se aplican 12V al pin 8.

Figura 10. Alimentación de -12V usando el LT1054.



2.1.3.2. Fuente de tensión de +10V. Para obtener una tensión de +10V para la alimentación del sensor de presión absoluta, se seleccionó la REF102 (ver Anexo D), que es una referencia de voltaje de precisión de $+10V \pm 0.0025V$. Este garantiza una señal estable de tensión sin apartarse del requisito de bajo consumo. Su salida de voltaje es extremadamente insensible a variaciones en la línea y en la carga y se puede ajustar externamente con un efecto mínimo en las derivas y la estabilidad. Puede operar con fuentes sencillas de 11.4V a 36V.

Figura 11. Referencia de tensión REF102 *Burr – Brown*.

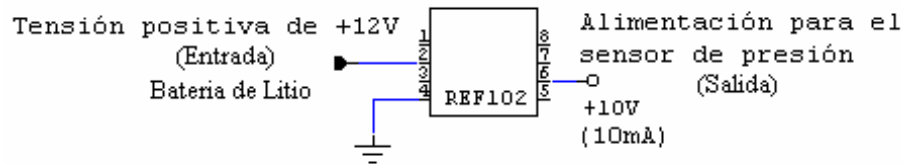


A continuación se presentan las características más importantes de la REF102 de *Burr-Brown* de *Texas Instruments*:

- Salida de $+10V \pm 0.0025V$
- Deriva muy baja: 2.5ppm/°C máx
- Estabilidad excelente: 5ppm/1000hr típica
- Regulación de línea excelente: 1ppm/V máx
- Regulación de carga excelente: 10ppm/mA máx
- Bajo ruido: 5mVpp típico, 0.1Hz a 10Hz
- Amplio margen de alimentación: 11.4VDC a 36VDC
- Baja corriente de reposo: 1.4mA máx
- Potencia disipada: 0.1W
- Opciones encapsuladas: TO-99 Hermético, DIP Plástico, SOIC. Para este proyecto se escogió el circuito integrado DIP.

El circuito necesario para obtener la tensión de +10V a partir de la REF102 es muy sencillo, y se muestra en la Figura 12.

Figura 12. Circuito para la obtención de la alimentación de +10V.

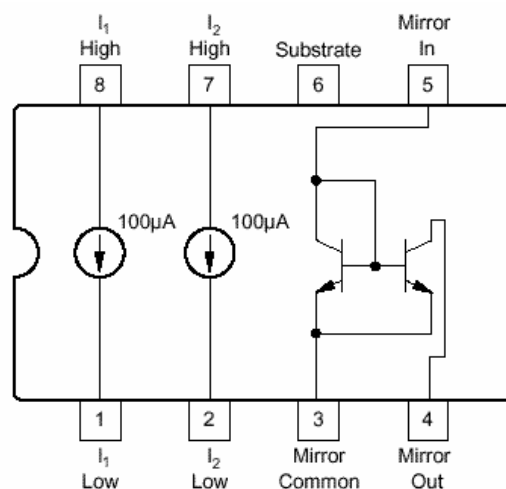


2.1.3.3. Fuente de corriente de 200uA. Debido a que el RTD es un sensor modulador, es decir que produce una variación de un parámetro eléctrico pero que no genera tensión ni corriente, se hace necesario excitarlo para detectar esa variación. La excitación puede ser en tensión o corriente, industrialmente se emplea la alimentación en corriente constante para reducir los efectos de las caídas de tensión propias de la alimentación a tensión constante, efecto que sería bastante notorio en el presente proyecto donde el cable de la RTD alcanza los 3 mts. De esta forma al emplear una referencia estable de corriente se permite, a través de la medida de tensión, obtener el cambio de resistencia en el RTD debido a la variación de la temperatura.

El sensor se debe excitar con una corriente de valor intermedio, no muy pequeño con el fin de que las tensiones en sus terminales no sean demasiado bajas y, por consiguiente, muy susceptibles a interferencias, además se requeriría una ganancia muy alta en el amplificador de instrumentación. Sin embargo, tampoco se puede emplear una corriente muy alta puesto que ésta puede causar el autocalentamiento del RTD, el cual ocasionaría una medida errónea puesto que no sólo se estaría obteniendo la información de temperatura del gas en la línea sino además la variación de temperatura de la resistencia por efecto *Joule*.

Para obtener esta corriente se seleccionó la fuente-sumidero de corriente dual REF200.

Figura 13. Referencia de corriente REF200 de *Burr-Brown*.



Las características más importantes de este integrado son las siguientes:

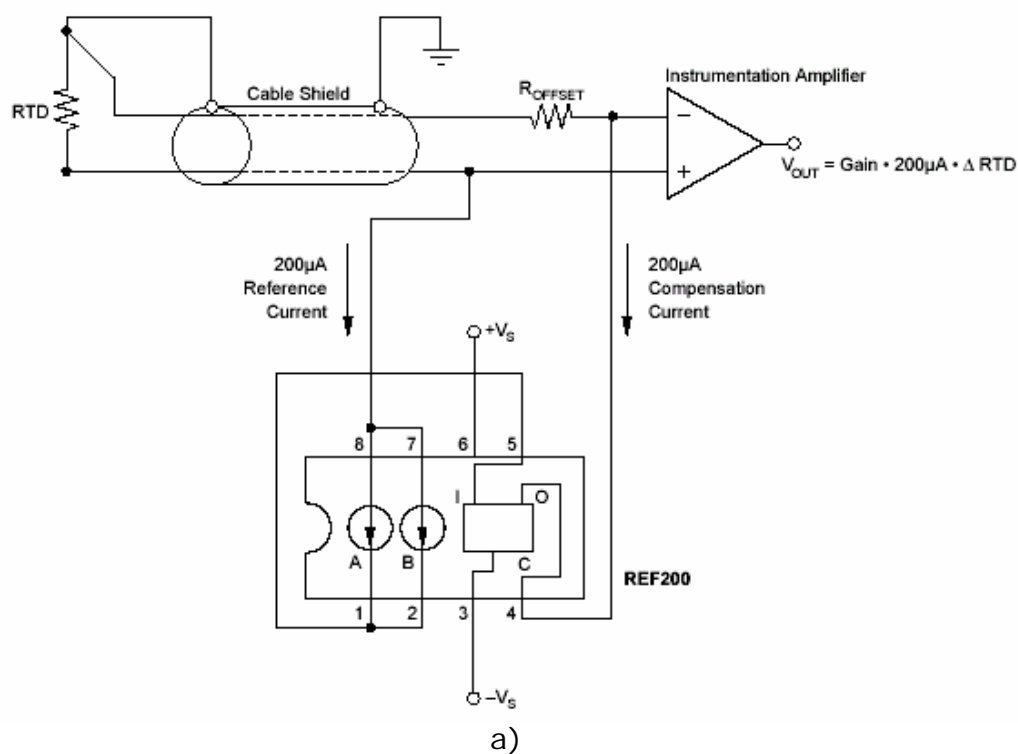
- Completamente flotante: sin conexiones de alimentación o tierra.
- Alta exactitud: $100\mu\text{A} \pm 0.5\%$
- Baja deriva de temperatura: $\pm 25\text{ppm}/^\circ\text{C}$
- Incluye espejo de corriente
- Potencia máxima disipada: 0.15W

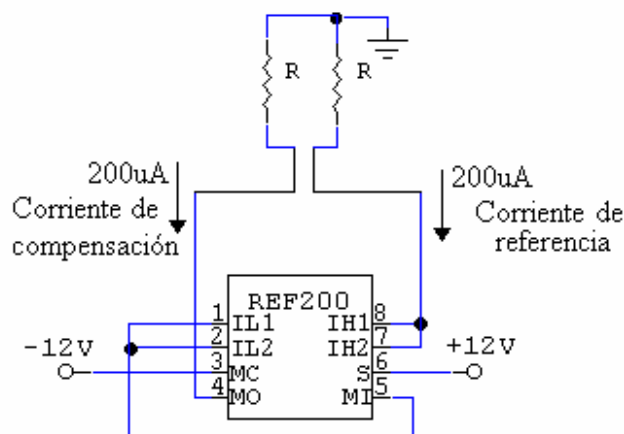
La REF200 combina tres bloques de circuito en un solo *chip* monolítico: dos fuentes de corriente de $100\mu\text{A}$ y un espejo de corriente (Figura 13). Las secciones están aisladas dieléctricamente, lo que las hace completamente independientes. Las secciones se pueden enlazar para obtener corrientes de $50\mu\text{A}$, $100\mu\text{A}$, $200\mu\text{A}$, $300\mu\text{A}$ o $400\mu\text{A}$. Se puede usar circuitería externa para obtener virtualmente cualquier corriente.

La REF200 está disponible en mini-DIP plástico de 8 pines y en encapsulados SOIC. Para este proyecto se escogió la referencia SOIC.

Las conexiones necesarias para obtener una fuente de $200\mu\text{A}$, a partir del circuito integrado REF200, se presentan en la Figura 14.

Figura 14. Conexión de la RTD con el integrado REF200.





b)

2.1.3.4. Amplificador de instrumentación. Las señales de los sensores debe ajustarse al nivel de entrada del conversor análogo-digital (0- 10V); esta función la ejecuta un amplificador que para esta aplicación debe ser de las mejores características, por tanto se emplea un amplificador de instrumentación en vez de un amplificador operacional, ya que este último tiene menores prestaciones. Dentro de las ventajas que se obtienen al emplear un amplificador de instrumentación se tienen: alta impedancia de entrada, alto rechazo del modo común (CMRR), ganancia estable variable con una única resistencia, tensiones y corrientes de desequilibrio bajas, pocas derivas e impedancia de salida baja.

Se escogió de la marca *Burr-Brown* de *Texas Instruments* el circuito integrado referenciado como INA118¹⁰, el cual incorpora en una sola pastilla 3 amplificadores operacionales, red de resistencias y deja al usuario dos bornes libres para que con sólo una resistencia de precisión ($\pm 1\%$), ajuste el valor de la ganancia de 1 a 10000. La protección interna en la entrada puede resistir hasta $\pm 40V$ sin dañarse.

El INA118 tiene las siguientes características que lo hacen idóneo para una aplicación de instrumentación donde se requiere bajo consumo sin sacrificar desempeño.

- Voltaje de offset bajo: $50\mu V$ máx
- Deriva baja: $0.5\mu V/^{\circ}C$ máx
- Baja corriente de polarización de entrada: $5nA$ máx
- CMRR alto: 110 dB mín
- Entradas protegidas a $\pm 40\text{ V}$
- Amplio margen de alimentación: ± 1.35 a $\pm 18V$
- Baja corriente en estado inactivo: $350\mu A$
- Potencia disipada ($V_{IN}=0V$): $5.25mW$
- Temperatura de operación: -40 a $+80^{\circ}C$

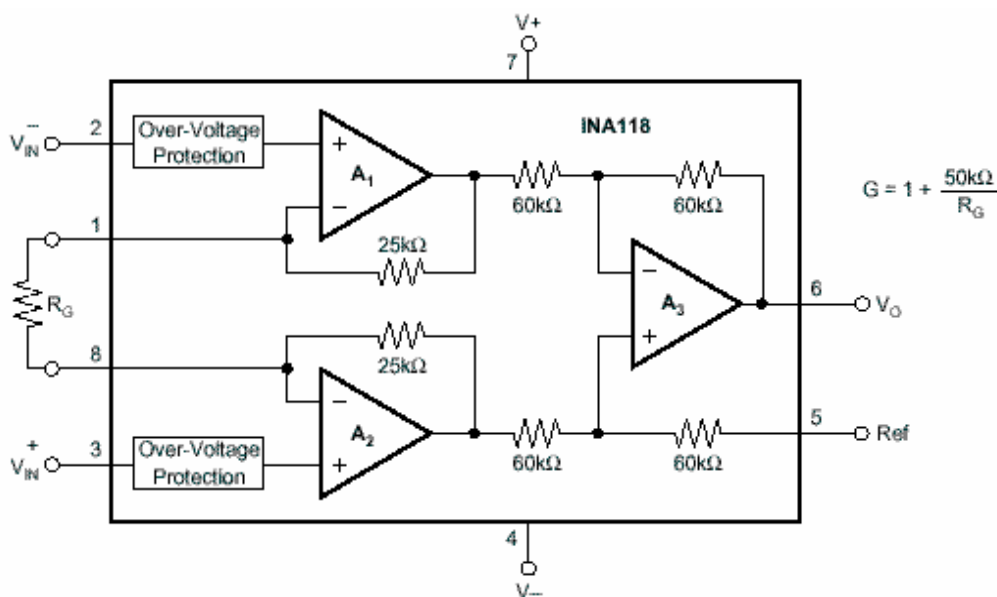
¹⁰ BURR-BROWN CORPORATION. DATASHEET INA118. pp 8, 1994. www.ti.com

- Encapsulado tanto en DIP como en SO-8. Para este proyecto se seleccionó el encapsulado de montaje superficial

La Figura 15 muestra las conexiones básicas requeridas para la operación del INA118. Como se observa, la ecuación 32 determina la ganancia de este amplificador.

$$G = 1 + 50k\Omega/R_G \quad (\text{ecuación 32})$$

Figura 15. Diagrama de bloques del INA118 Burr – Brown.



La salida está referenciada al terminal de salida (llamado "Ref"), que normalmente está aterrizado. Ésta debe ser una conexión de baja impedancia para asegurar buen rechazo al modo común. Una resistencia de 12Ω en serie con el pin "Ref", causará que un dispositivo típico degrade a aproximadamente 80dB CMRR ($G = 1$).

2.1.3.5. Acondicionamiento para la señal del medidor primario tipo turbina. El medidor de turbina genera un tren de pulsos TTL cuya frecuencia es proporcional al flujo por unidad de tiempo que pasa a través del tubo.

Para aislar la masa del tubo y de la turbina con el computador de flujo, es necesario emplear algún tipo de aislamiento. Los acopladores ópticos son dispositivos que resuelven eficientemente este tipo de problemas de aislamiento de tierras, rechazo al ruido de modo común, transformaciones de fuente de alimentación, etc.

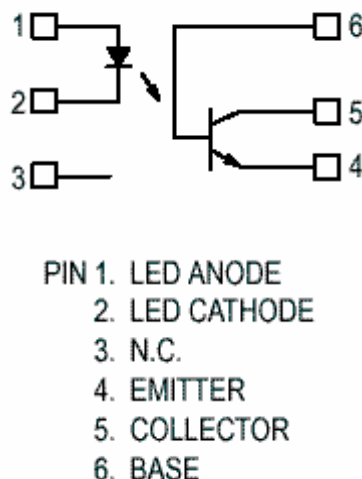
Los optoacopladores típicamente vienen en un pequeño encapsulado de seis u ocho pines, pero son esencialmente una combinación de dos dispositivos distintos: un transmisor óptico, generalmente un LED de Arseniuro de Galio (GaAs) y un receptor óptico tal como un fototransistor o un *diac* disparado por luz. Los dos dispositivos

están separados por una barrera transparente la cual bloquea cualquier flujo de corriente entre ellos, pero si permite el paso de luz. Usualmente las conexiones eléctricas del LED son puestas en los pines de un lado del encapsulado y las del fototransistor al otro lado para separarlos físicamente tanto como sea posible. Esto permite a los optoacopladores proveer una tensión de aislamiento entre la entrada y la salida de 500V a 7500V.

Los optoacopladores son esencialmente dispositivos digitales o de conmutación, así son considerados una excelente forma tanto para transmitir señales de control *on-off* así como datos digitales. Las señales analógicas también pueden ser transmitidas por medio de modulación en frecuencia o modulación por ancho de pulso.

Para este proyecto se escogió el circuito integrado 4N26 el cual es el optoacoplador que realiza el aislamiento óptico de los pulsos, su función es aislar la tierra del medidor y la de la tarjeta de acondicionamiento.

Figura 16. Esquemático del 4N26.



El parámetro más importante para la mayoría de optoacopladores es la eficiencia de transferencia, medida en términos de su relación de transferencia de corriente o CTR. Ésta es simplemente la relación entre el cambio de la corriente de salida del transistor y el cambio de corriente de entrada del LED. Típicamente los valores del CTR están en el rango del 10% al 50% para dispositivos basados en fototransistores y de hasta 2000% para los que están basados en un par de transistores Darlington en su salida.

$$(CTR = I_C / I_F) \quad (\text{ecuación 33})$$

Es importante resaltar que el CTR tiende a variar de acuerdo al nivel de corriente de entrada al LED, así típicamente el CTR tiene su valor máximo en un nivel de

corriente del LED de aproximadamente 10mA y se disminuye tanto para niveles de corrientes más altos o más bajos.

El integrado seleccionado (4N26) posee una eficiencia de acople del 20%, esto significa que para una corriente en el diodo de $I_F = 10\text{mA}$ la corriente de colector en el transistor de salida es de $I_C = 2\text{mA}$. (Con una caída de tensión colector emisor $V_{CE} = 10\text{V}$).

Otros parámetros de los optoacopladores incluyen la tensión máxima colector emisor, $V_{CE(\text{MAX})}$ el cual limita la tensión de la fuente en el circuito de salida ($V_{CE(\text{MAX})} = 30\text{V}$ para el 4N26), y la corriente máxima de entrada al LED $I_{F(\text{MAX})}$ la cual es empleada para calcular el valor mínimo de la resistencia en serie que limitará esta corriente. ($I_{F(\text{MAX})} = 60\text{mA}$ para el 4N26). Además para este integrado el LED soporta un máximo de 3V inversos antes de dañarse ($V_R = 3\text{V}$) y tiene una caída directa (V_F) de 1.15V.

Para la mayoría de las aplicaciones los parámetros básicos del LED (I_F y V_F) son suficientes para definir la entrada. En el 4N26 $V_F = 1.15\text{V}$ e $I_F = 10\text{mA}$.

Por tanto, la resistencia limitadora se puede calcular con la siguiente ecuación:

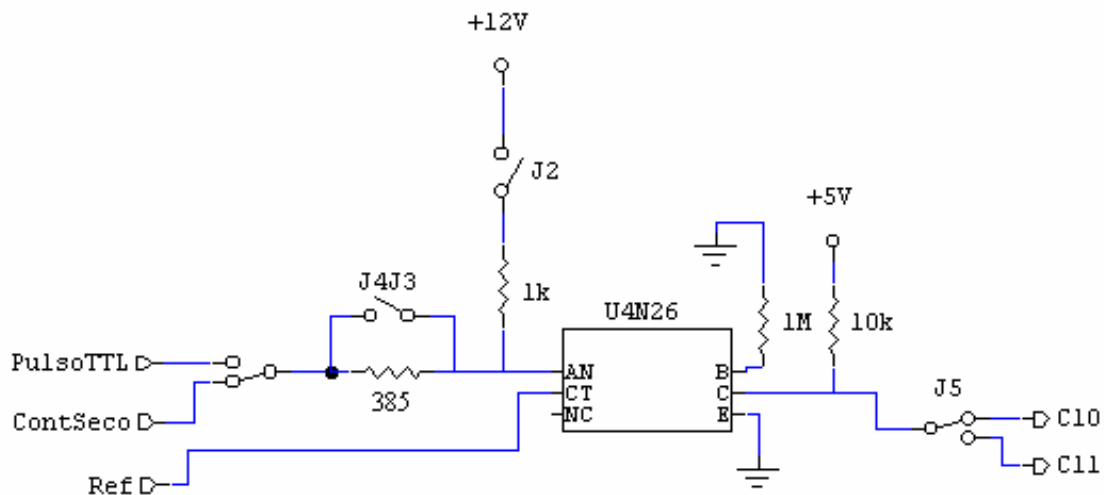
$$R = (V_{in} - V_F) / I_F = (5 - 1.15) / 10\text{mA} = 385\Omega \quad (\text{ecuación 34})$$

Por otro lado, el fototransistor también tiene su base a disposición del usuario, aunque en las aplicaciones generalmente nunca se usa control por esta base. El parámetro que relaciona la corriente de base con la de colector es su ganancia en corriente continua, para el 4N26 es igual a 500 ($h_{FE} = \beta_{cc} = 500$), donde beta es la relación entre la corriente de colector y la corriente de base del transistor ($\beta = I_C / I_B$). Esta se conectó a tierra a través de un resistor de $1\text{M}\Omega$ para polarizarla, esto ayuda a acelerar el proceso de conmutación de corte a saturación al remover la carga almacenada.

El otro parámetro importante, especialmente en aplicaciones de comunicación, es el ancho de banda, el cual determina la frecuencia de la señal más alta que puede ser transferida a través del optoacoplador y está determinada por la construcción interna del dispositivo y el desempeño del fototransistor de salida. Típicamente, los optoacopladores con un simple fototransistor de salida tienen un ancho de banda de 200kHz a 300kHz, mientras que en los basados en pares Darlington se reduce aproximadamente 10 veces quedando en 20kHz a 30kHz.

La frecuencia máxima de trabajo para el 4N26 de acuerdo a su hoja de datos es de 300kHz. La frecuencia máxima de los pulsos de la turbina es de 0.042 Hz. De esta manera, el comportamiento en frecuencia de este circuito no es relevante en el diseño de los computadores de flujo ya que en la práctica ninguna turbina en Colombia genera pulsos con frecuencias superiores a 1 Hz. Lo importante en este caso es la selección adecuada de las resistencias de polarización para que el fototransistor de salida opere como un interruptor, oscilando entre corte y saturación.

Figura 17. Circuito para entrada de pulsos.



Algunos medidores tipo turbina no generan tren de pulsos sino que tienen un contacto seco que cambia de abierto a cerrado de acuerdo a las vueltas que de la turbina. Por esta razón, el prototipo cuenta con un sistema que permite medir la frecuencia del tren de pulsos o la tasa de cambio del estado del contacto seco con el fin de poder detectar la rata de flujo en la tubería. Por lo tanto, la entrada del circuito de acondicionamiento acepta los dos tipos de señal, así que se dejó seleccionable por *jumper* esta configuración.

Si se tiene una señal de pulsos digitales de 5V no se requiere alimentación, de modo que se emplea la configuración de *jumpers* así: J2, y J3, J4 abiertos. Con esta conexión la señal entra directamente al 4N26 pasando a través de la resistencia de 385Ω cuya función es limitar la corriente. Si, por el contrario, la salida de la turbina es un contacto seco, se debe alimentar el sistema para poder observar el estado del contacto, de modo que los *jumpers* J2, y J3, J4 deben estar cortocircuitados. En este caso, el 4N26 verá una señal cuando el contacto esté cerrado y tendrá señal nula cuando esté abierto. La resistencia de 1kΩ limita la corriente.

2.1.3.6. Acondicionamiento para medición de temperatura. La señal del sensor RTD que contiene la información de los cambios en la temperatura es una variación de resistencia, de modo que se hace necesario alimentar el sensor para poder medir esta variación, además se requiere amplificar la señal de tensión obtenida con el fin de llevarla al rango de 0 a 10V.

La Figura 18 muestra el circuito de conexión de la RTD con su excitación en corriente (200μA) dada por el REF200 y con un amplificador de instrumentación trabajando en modo diferencial para amplificar la salida. Al hacer $R_{OFFSET}=R_{RTD}(0^{\circ}C)=100\Omega$, la salida del amplificador sólo mostrará (en tensión) la desviación correspondiente de resistencia de la RTD de su valor de 100Ω (a 0°C). Por tanto, como se indica en dicha figura, la ecuación que relaciona la salida de tensión del amplificador con la temperatura es:

$$V_{out} = G * 200\mu * \Delta R_{TD} \quad (\text{ecuación 35})$$

Donde G es la ganancia propia del amplificador y ΔR_{TD} es la diferencia entre la resistencia instantánea (función de la temperatura) de la RTD con la resistencia de referencia que es de 100Ω ya que se está empleando una Pt100. Así pues:

$$\Delta R_{TD} = (R_{RTD}(T) - 100) \Omega \quad (\text{ecuación 36})$$

Donde la variación de la resistencia de las RTD de platino es lineal en el rango 0-70°C (según lo explicado en la sección 1.3.1) y correspondiente con la siguiente ecuación:

$R_{RTD}(T) = R_0 (1 + \alpha T)$ donde $R_0 = 100\Omega$ y $\alpha = 0,003850 \text{ } ^\circ\text{C}^{-1}$ al remplazar resulta,

$$R_{RTD}(T) = 100 (1 + 3.85 * 10^{-3} T_{(^{\circ}\text{C})}) = 100 + 0.385 T_{(^{\circ}\text{C})} \Omega \quad (\text{ecuación 37})$$

Y combinando con la ecuación inicial, la tensión de salida es:

$$\begin{aligned} V_{out} &= G * 200\mu * (100 + 0.385 T_{(^{\circ}\text{C})} - 100) \\ &= G * 200\mu * 0.385 T_{(^{\circ}\text{C})} \end{aligned} \quad (\text{ecuación 38})$$

$$V_{out} = G * 77\mu * T_{(^{\circ}\text{C})} \quad (\text{ecuación 39})$$

Como se desea que a la temperatura máxima (70°C) exista una tensión de salida de 10V para aprovechar al máximo el rango de entrada del conversor análogo digital, la ganancia se determina como:

$$G = 10 / (77\mu * 70) = 1855.288 \quad (\text{ecuación 40})$$

Ahora, tomando el amplificador de instrumentación de bajo consumo INA118 (sección 2.1.3.4.) se observa que la resistencia en función de la ganancia es:

$$R_G = 50000 / (G - 1) = 50000 / (1855.288 - 1) = 26.965 \Omega \quad (\text{ecuación 41})$$

Comercialmente la resistencia más cercana a este valor es $R_G = 27\Omega \pm 1\%$, es decir, que al adquirir una resistencia su valor podrá oscilar entre 26.73Ω y 27.27Ω . No obstante, el error introducido por este pequeño corrimiento del valor de la resistencia es corregible por *software* por medio de las rutinas de calibración que se presentarán en la sección 3.3.1..

Finalmente el circuito empleado para el acondicionamiento de temperatura se observa en la Figura 18, el cual esquematiza las salidas del sensor RTD Pt100 para su adecuada conexión según la Figura 19.

Figura 18. Acondicionador de señal para un RTD Pt100.

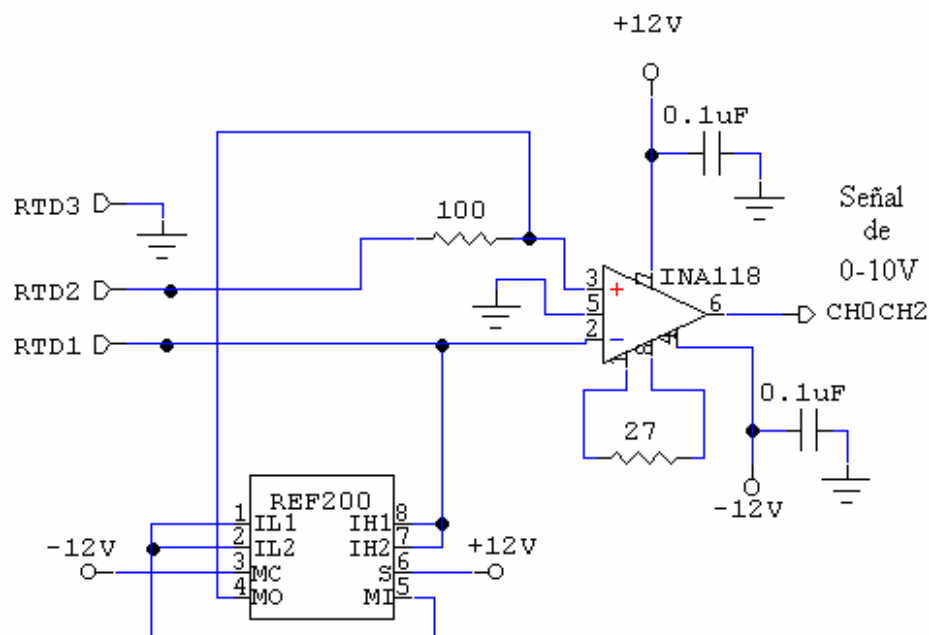
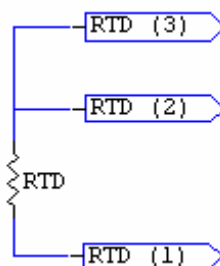


Figura 19. Bornes típicos de un RTD



El circuito de la Figura 18 presenta una ventaja fundamental ya que elimina el error causado por la resistencia de los hilos del RTD. Teniendo garantizado que ésta sea la misma para todos los hilos, el circuito elimina sus efectos debido a que por los brazos circulan corrientes opuestas que harán que las tensiones de error originadas se cancelen entre sí. Este tipo de circuito se conoce como conexión de tres hilos.

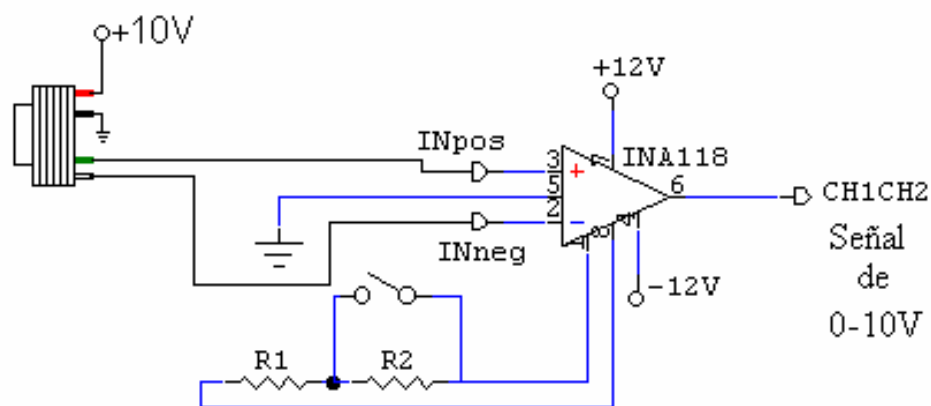
2.1.3.7. Acondicionamiento para medida de presión. La salida de los transductores de presión seleccionados tiene un rango de 0 a 100mV, por lo tanto estos niveles deben ser amplificadas para que alcance los rangos de entrada del conversor análogo-digital (0-10V). Esto determina una ganancia de 100 para el amplificador de instrumentación.

Se implementó el circuito de la Figura 20, donde se emplea el INA118 para la amplificación de la señal. Para una ganancia de 100, la resistencia de ganancia será:

$$\begin{aligned} R_G &= 50000 / (G - 1) \\ &= 50000 / (100 - 1) \\ &= 505.051 \, \Omega = R1 \end{aligned} \quad (\text{ecuación 42})$$

La resistencia comercial de precisión más cercana es de $R1 = 510\Omega \pm 1\%$, así que el valor real podrá oscilar entre 504.9Ω y 515.1Ω .

Figura 20. Acondicionamiento de los transductores de presión.



Como muestra en la Figura 20, en el circuito se adicionó el espacio para una resistencia R2 (la cual está vacía en el prototipo y cortocircuitada por un *jumper*) y está en serie con R1, esto es para modificar ganancias ante proyectos futuros donde se requiera cambiar el rango del transductor de presión.

Igual que en la RTD, el computador de flujo diseñado contiene rutinas de calibración que disminuyen cualquier error de hardware.

2.1.3.8. Conversión análogo/digital de las señales acondicionadas. El conversor A/D (Análogo-Digital) es el dispositivo empleado para convertir las señales análogas acondicionadas de los transductores de presión y temperatura, en señales digitales para ser almacenadas y computadas dentro de la unidad de procesamiento del computador de flujo. La programación del conversor se realiza directamente en la CPU seleccionada para el computador de flujo (ver sección 2.2).

Para esta aplicación, la selección del conversor requiere en especial dos datos: la velocidad y el número de bits. El período de muestreo para el computador de flujo diseñado es de 50ms, por tanto el tiempo de conversión del conversor seleccionado debe ser naturalmente menor, no obstante, este tiempo de muestreo en ms es grande comparado con los tiempos de conversión actualmente disponibles en el

comercio que oscilan en el rango de los μs , así pues este parámetro no es un fuerte limitante.

El otro parámetro de importancia es el número de bits necesarios. Naturalmente entre más bits mejor resolución y sensibilidad tendrá la medida. A continuación en la Tabla 10 se muestra la sensibilidad de la lectura de presión (una de las variables de más peso) obtenida con diferentes resoluciones de conversores y a varias escalas de presión.

Tabla 10. Sensibilidad de la lectura de presión.

Escala Presión -PSIG-	Resolución en PSIG del transductor de presión							
	8 bits PSIG	% Error	10 bits PSIG	% Error	12 bits PSIG	% Error	16 bits PSIG	% Error
0-1500	5.86	0.39	1.46	0.097	0.37	0.025	0.023	0.0015
0-1200	4.69	0.39	1.17	0.097	0.29	0.025	0.018	0.0015
0-700	2.73	0.39	0.68	0.097	0.17	0.025	0.011	0.0015
0-250	0.977	0.39	0.24	0.097	0.06	0.025	0.004	0.0015

De nada sirve tener un transductor de presión de buena exactitud si ésta se ve degradada por la resolución del conversor. Obsérvese que en la escala 0-1200 psig (máxima que se halla en ecogás) con 10 bits se obtiene más de 1 psi de resolución, lo cual representa una gran diferencia en facturación como se verá en las simulaciones que se presentarán en el Capítulo Cuatro. Por tanto, los computadores de flujo deben escoger la mejor resolución que tengan a su alcance pues esto brindará más exactitud en su medida. La resolución de 16 bits es la preferida por el mercado actual^{vi} y fue la empleada en el presente proyecto por su despreciable influencia en la exactitud de la lectura (0.0015%).

Además del número de bits y el tiempo de conversión se tuvieron en cuenta los siguientes aspectos al momento de seleccionar el conversor A/D:

- Interfase: paralela, para mayor velocidad y facilidad de programación.
- Número de canales: 4, para tener la posibilidad de conectar 2 señales de temperatura y 2 de presión en un solo *chip*.
- Temperatura de operación: por lo menos de -10 a 70 °C (rango en Colombia)
- Buena linealidad
- Bajo consumo de potencia.

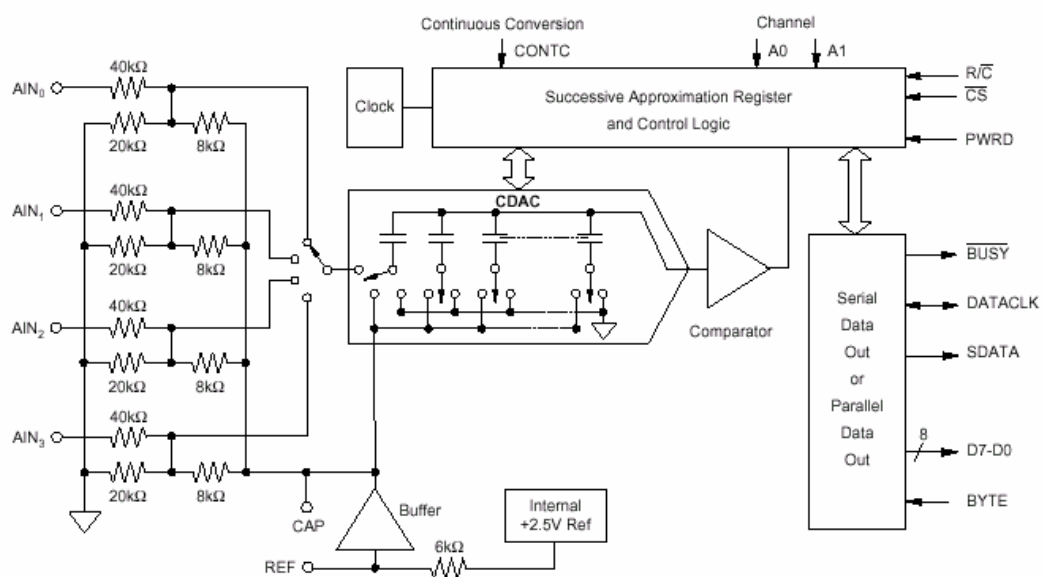
Del mercado se escogió al fabricante *Texas Instruments* y su división con la marca *Burr-Brown*. Como se ha visto, se trabajó con circuitos integrados de esta marca para los diferentes acondicionamientos, esto se debe a que este fabricante es uno de los líderes mundiales en elaboración de semiconductores.

Con estos planteamientos se escogió el circuito integrado referenciado como ADS7825 en encapsulado de montaje superficial. La digitalización la realiza este

solo *chip*, el cual requiere pocos componentes adicionales necesarios para mantenerlo en condiciones estables e inmune al ruido.

En la Figura 21 se muestra un diagrama de bloques de su estructura interna. Sus 4 entradas (AIN1, AIN2, AIN3, AIN4), tienen un margen de $\pm 10V$. Cuenta con circuito de muestreo y retención, referencia y reloj. También se observa que este conversor tiene interfase serial y paralela, dado que esta última es más rápida y da mayor facilidad en la elaboración del *software*, se escogió esta opción.

Figura 21. Diagrama de bloques del ADS7825 Burr – Brown.

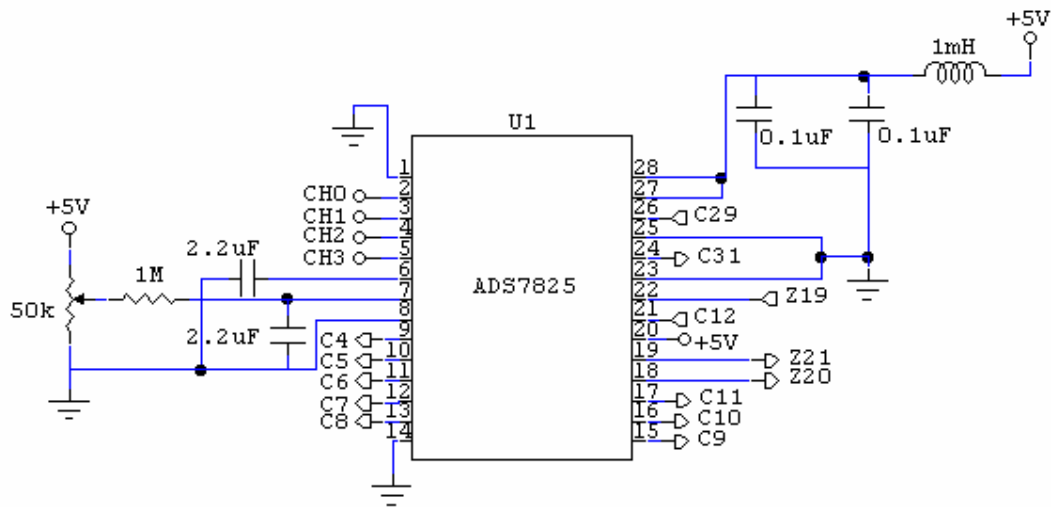


A continuación se presentan las especificaciones más importantes del conversor ADS7825:

- Tiempo máximo de adquisición y conversión: $25\mu s$ (muestreo en el cf 50ms).
- Error máximo de no linealidad integral (INL): $\pm 2LSB$.
- Temperatura de operación: -40 a $85^{\circ}C$.
- Potencia disipada: Máxima 50mW, mínima 50uW.
- Salida Serial o Paralela (en el cf se empleó en modo paralelo).

En la Figura 22 se observa el conexionado necesario para el conversor ADS7825. Las señales a convertir deben estar dentro del margen de 0 a 10V. Se pueden presentar señales de tensión negativa, por ejemplo para el caso de una temperatura inferior a $0^{\circ}C$ en el RTD, pero esta temperatura está fuera del margen de operación de las tuberías de Ecogás en Colombia, aún así, si este valor se diera no representaría un problema para el conversor debido a que su margen de entrada es de -10 a $+10V$.

Figura 22. Conexiones del ADS7825.



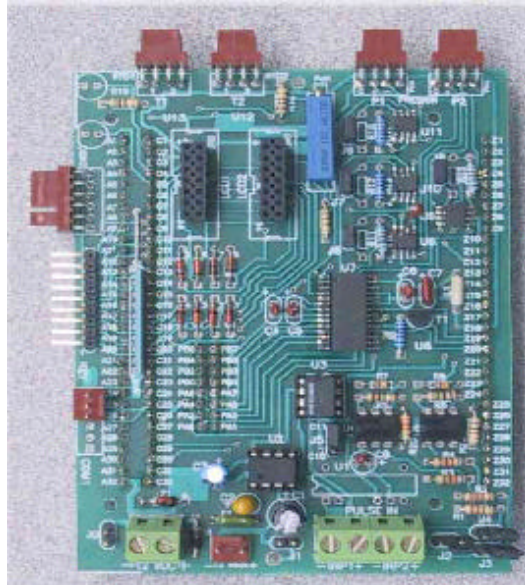
2.1.3.9. Costo de la tarjeta de acondicionamiento. Esta tarjeta contiene los circuitos de acondicionamiento para las señales del medidor de turbina, sensores de presión absoluta y temperatura (Ver Figura 23). Además tiene las fuentes de -12V y +10V obtenidas a partir de la fuente que alimenta al computador de flujo. También incluye el circuito del conversor (ADC) cuyas salidas digitalizadas constituyen las entradas de señal de la unidad de control y procesamiento del computador de flujo.

La Tabla 11, presenta el costo total en pesos colombianos de la tarjeta de acondicionamiento para el computador de flujo tipo turbina.

Tabla 11. Costos de la tarjeta de acondicionamiento.

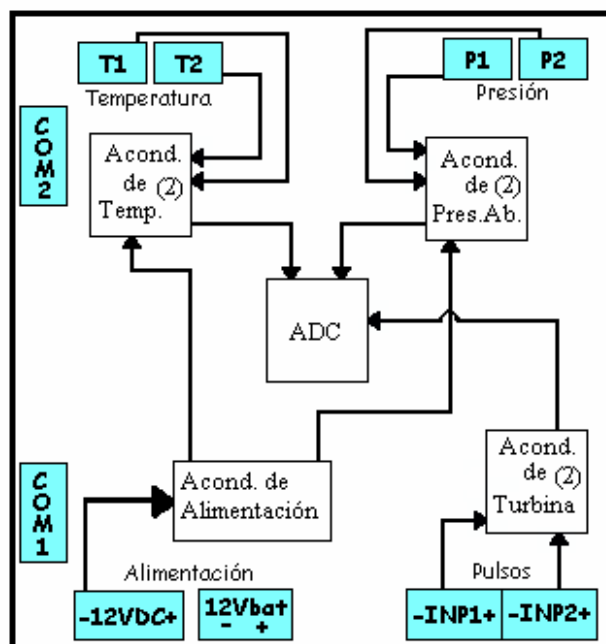
Cantidad	Descripción	V. Unitario (\$)	Valor total (\$)
1	Conversor ADS 7825	150500	150500
2	REF200	16900	33800
5	INA118	34800	174000
2	4N26	1500	3000
1	REF102	16900	16900
1	LT1054	14000	14000
1	Tarjeta de circuito impreso		200000
	Varios (R, C, L, <i>jumpers</i> , conectores)		80000
	Valor Total		672200

Figura 23. Tarjeta de acondicionamiento de señal.



En la Figura 24 se observa un esquema que muestra el plano de conexión de las entradas y salidas de la tarjeta, junto con los circuitos que la conforman. Los bloques en negrilla se encuentran físicamente en las tarjetas implementadas y están en la misma posición que se observa en los diagramas.

Figura 24. Diagrama de las entradas y salidas de la tarjeta de acondicionamiento.



2.2. UNIDAD DE PROCESAMIENTO, CONTROL Y ALMACENAMIENTO

Esta unidad es el corazón del sistema, ya que controla y supervisa las tareas del computador de flujo, tales como:

- Muestreo de señales de presión y temperatura.
- Cálculo del factor Z, a través de la norma AGA8.
- Aplicación de la norma AGA7 (medidor turbina).
- Almacenamiento de datos.
- Control de la interfaz con el usuario.
- Manejo de la comunicación con un computador.

2.2.1. Parámetros de selección. La selección del sistema digital sobre el cual se implementó el computador de flujo, requirió de un análisis de las siguientes características:

- Consumo de potencia
- Número de Bits
- Memoria
- Tamaño
- Punto Flotante

2.2.1.1. Procesador. El cómputo de la norma AGA8-Gross2 exige el manejo de números en punto flotante tanto pequeños como grandes. Como ejemplo, se puede ver los coeficientes de interacción virial para el nitrógeno y dióxido de carbono (ver Tablas 6 y 7, Capítulo Uno) que oscilan entre -0.325798×10^{-7} y -0.144600 ; y los coeficientes viriales para el hidrocarburo equivalente (ver Tabla 8, Capítulo Uno) que varían entre $-0.367713 \times 10^{-11}$ y -0.425468 .

Adicional a esto, los resultados de volúmenes acumulados diarios pueden alcanzar valores de 50000 KCF. Esto indica un rango aproximado de números desde -1×10^{12} hasta 1×10^5 lo cual da una extensión numérica de 1.0000001×10^{12} . En la columna 2 de la Tabla 12, se observa la resolución que se puede alcanzar con un número de bits dado. De esta manera, con registros de 48 bits se podría trabajar con el rango de números requeridos en la norma; sin embargo, las CPU's comerciales son de 8, 16, y 32 bits. En la columna 3 se indican los registros necesarios para representar un número de 48 bits con una base dada de bits (elaborando rutinas de *software*).

Tabla 12. Resolución vs bits.

Longitud del Registro	Extensión numérica	Numero de registros necesarios para 48 bits
8 bits	256	6
16 bits	65536	3
32 bits	4.295×10^9	2
48 bits	2.815×10^{14}	1
64 bits	1.845×10^{19}	1

De la Tabla 12, se deduce que con 8 bits se requerirían trabajar 6 registros simultáneamente para representar un solo número, aumentando así el tiempo de desarrollo. Además se hace obsoleto, pues con una CPU de 16 bits se puede elaborar el mismo trabajo con solo 3 registros. El uso de dos registros en las CPUs de 32 bits daría un sobredimensionamiento de 64 bits el cual es innecesario, y encarece el costo del proyecto. Por tanto, para el computador de flujo se seleccionaron 16 bits para la CPU, lo cual a su vez concuerda con el número de bits de las CPUs de la mayoría de computadores de flujo comerciales¹¹.

Las veloces CPUs de 32 bits se destinan a computadores de flujo de alto trabajo (no solo de medición) pues permiten sobrellevar de manera concurrente una variedad de funciones relacionadas al control de otros equipos, sin afectar las tareas de procesamiento de la medición. Otra aplicación específica de computadores de flujo basados en 32 bits, son los dedicados a medidores de tipo ultrasónico, debido a que el desarrollo y ejecución de la norma de calculo (AGA9) es más compleja.

2.2.1.2. Consumo de potencia. Cuando se construye un equipo electrónico el diseñador siempre debe tener en cuenta que este consuma el mínimo de potencia posible. Para este proyecto, este aspecto cobro mayor importancia ya que el ambiente de trabajo del dispositivo es industrial y su área esta clasificada como peligrosa debido a la presencia de gases inflamables.

El consumo de corriente de todos los componentes hasta ahora seleccionados permite estimar la corriente aún disponible para el hardware que hace falta, el cual consisten en la tarjeta procesadora que contendrá la CPU del sistema, y un *display* para visualizar datos.

En este proyecto se seleccionó un display de 20x4 líneas para tener 4 renglones de visualización de datos y así observar por ejemplo 4 variables importantes; como: presión, temperatura, velocidad de flujo, y volumen acumulado del día. El display escogido fue el OPTREX DMC20434N-B el cual tiene un consumo típico de 2.2mA y 4mA máximo.

La tabla 13 resume el consumo teórico de la tarjeta de acondicionamiento, y adicionalmente el consumo de la instrumentación, y el display. (El consumo de las RTDs está contemplado junto con el REF200, pues este último no presenta consumo sin tener conectada las RTDs).

¹¹ OMNI FLOW. Flow Factor. New Latters, Vol1, Num I-II-III.

Tabla 13. Consumo de la tarjeta de acondicionamiento, instrumentación y *display*.

	Referencia	Cant	Consumo Individual	Consumo Total	Porcentaje Parcial	Porcentaje Total
A C O N D I. I N S T R.	ADS7825	1	10mA	10mA	52.91	
	REF102	1	1.4mA	1.4mA	7.41	
	INA118	4	385µA	1.5mA	7.94	
	4N26	2	500µA	1.0mA	5.29	
	LT1054	1	5mA	5mA	26.46	
	Consumo Total de Acondicionamiento			18.9mA	100%	18.9%
	Transductor de Presión	2	2mA	4mA	47.62	
	RTDs con REF200	2	200µA	400µA	4.76	
	Display OPTREX 20x4	1	4mA	4mA	47.62	
	Consumo Total de Instrumentación			8.4mA	100%	8.4%
	Consumo de Acondicion. + Instrumentación			27.3mA		
	Restante para tarjeta procesadora –CPU–			72.7mA		72.7%
	Consumo Máximo del Comp. de Flujo			100mA		100%

La corriente consumida por la tarjeta de acondicionamiento en una prueba experimentalmente resultó de 18.1mA (menor que la teórica que se trabajó con valores teóricos máximos). Finalmente al restar de 100mA (máximo consumo para satisfacer el criterio de seguridad intrínseca) el valor de 27.3mA (consumo actual) restan 72.66mA como consumo máximo para seleccionar la tarjeta CPU del sistema.

2.2.1.3. Memoria. Existen siete variables importantes que deben promediarse cada 15 minutos: temperatura, presión, velocidad de flujo corregida, velocidad de flujo sin corregir, factor de compresibilidad, volumen corregido, y volumen corregido sin corregir. Cada una de estas variables ocupa 6 bytes de memoria (punto flotante de 48 bits), así pues la capacidad de memoria para almacenar estas 7 variables por 3 meses (90 días) será: $7 \times 4 \times 24 \times 90 \times 6 = 362880$ bytes. Por tanto, una memoria de 512KB es suficiente y permite algún espacio adicional para futuras mejoras.

Generalmente las compañías descargan sus datos cada mes o dos meses, así que el tiempo de almacenamiento de 3 meses dará mayor confianza y tranquilidad de trabajo al operador; pues muchas veces el acceso a los sitios donde está instalado un computador de flujo es remoto o en lugares donde regularmente suceden derrumbes, paros, o situaciones de conflicto armado que conllevan a que el técnico tenga que esperar semanas mientras se solucionan estos inconvenientes para poder finalmente descargar su reporte de datos.

2.2.1.4. Estudio comparativo del mercado. Con las características expuestas en las secciones anteriores, se analizaron varias opciones de diversas marcas de CPU's y sistemas de desarrollo para escoger la más conveniente para este proyecto. En la Tabla 14 se presenta un cuadro comparativo de los sistemas estudiados:

Tabla 14. Comparación sistemas de desarrollo.

	Referencia del Sistema de Desarrollo					
	EV80C196Kx	M681CS08GP	TDS2020F	MC68336	ST10F66	DSP56F807EV M
Procesador	80C196	68HC908GP32	H8/532	MAX791	ST10F166	DSP56F807
Marca	Intel	Motorola	Hitachi	Maxim	ST Micro	Motorola
CPU Bits	16	8	16	16	16	16
Corriente	150mA	100mA	41mA	100mA	200mA	500mA
RAM kbyte	512KB	256KB	512KB	512KB	256KB	256KB
Precio	US\$350	US\$330	US\$399	US\$380	US\$340	US\$340

Según la Tabla 14, se observa que la mayoría de sistemas cumplen con los requisitos de memoria RAM y número de bits en su CPU. El factor determinante en este diseño lo marca el consumo de corriente, el cual sólo es satisfecho por el sistema embebido TDS2020F que con su procesador Hitachi H8/532 tiene un consumo máximo de 41mA, lo cual deja un margen de seguridad de 31mA antes de llegar al máximo permitido para el sistema de desarrollo que era de 72.7 mA.

Al seleccionar y estudiar detalladamente este sistema se encontraron parámetros adicionales que lo convirtieron en la CPU idónea para el desarrollo del presente proyecto. Entre estas características se pueden mencionar:

- Reducido tamaño de 10x8 cm se adapta eficientemente a pequeños espacios.
- Rutinas de Punto Flotante de 48 bits ya listas proveídas por el fabricante.
- Rango de temperatura de -10 a $+70$ °C, el cual cumple ampliamente.
- Soporte para *display* y teclados listos en sus rutinas de software.
- Compilación directa en Flash-EEPROM, lo cual evita el costo y ciclo de desarrollo en un programador PROM.
- Sistema de desarrollo interactivo que permite la rápida implementación del código o *firmware*.
- Rutinas de *software* listas para el manejo de conversores A/D externos.
- Dos puertos de comunicaciones con rutinas para su manejo.
- Buen posicionamiento en el mercado que garantiza la continuidad de producción.

Para este proyecto el número de entradas digitales es sólo dos y corresponde a la entrada de pulsos para la turbina (una por cada brazo de medición), por tanto este parámetro no tuvo peso en el momento de la selección pues todos los sistemas tienen por lo menos de 8 entradas digitales. Por otro lado, las entradas análogas deben ser cuatro y de 16 bits (ver sección 2.1.3.8.), y aunque la mayoría de sistemas de desarrollo tienen conversores A/D incluidos, estos generalmente son de 10 bits o máximo 12 bits. Por esta razón se seleccionó el conversor A/D externo descrito en dicha sección.

Las características del sistema embebido seleccionado, enfatizando exclusivamente en las que se utilizaron en el proyecto, se abordan en la siguiente sección. En el Anexo E se exponen todas las características del sistema embebido.

2.2.2. Sistema embebido. El sistema embebido seleccionado es referenciado como TDS2020F, fabricado por la empresa inglesa *Triangle Digital Services* (TDS). Este sistema embebido es un conjunto de periféricos básicos controlados por un microprocesador, que actúa como un microcomputador de propósito general. Puede ser destinado a cualquier aplicación, pues es un sistema de arquitectura abierta y se ajusta de acuerdo a las necesidades del usuario dependiendo de su aplicación (control, medición, almacenamiento de datos, etc).

La compañía *Triangle Digital Services* tiene una experiencia de 20 años en la fabricación de este tipo de dispositivos con excelente calidad. En especial, el sistema embebido TDS2020F ha sido utilizado para la fabricación de diversos equipos útiles para gasoductos¹², marítimos¹³, *dataloggers*¹⁴, medición y control¹⁵. Además, todos los componentes detallados de la tarjeta se pueden comprar por separado, incluso el solo procesador con su código exclusivo de *Forth* de *TDS Services*. Lo anterior representa un gran soporte en la reparación y mantenimiento de la TDS sin necesidad de tener que adquirir una nueva tarjeta.

El TDS2020F es un poderoso sistema embebido o microcomputador de control de 16 bits (Figura 25), que permite la implementación rápida de aplicaciones debido a su naturaleza interactiva de desarrollo y depuración (facilita la búsqueda y remoción de errores en el programa). Las librerías de *software* proporcionadas proveen soluciones instantáneas a muchos problemas de aplicación. El código fuente, escrito en el lenguaje de alto nivel *Forth* se compila directamente en una memoria no volátil Flash-EEPROM.

El TDS2020F mide solo 10 x 8 cm, tamaño apropiado para el encapsulado de un computador de flujo. Puede ser conectada de tres formas según la aplicación:

- A través de un cable plano
- Insertada en otra tarjeta (de esta forma se empleo en el presente proyecto).
- En un *rack*, usando el conector DIN41612

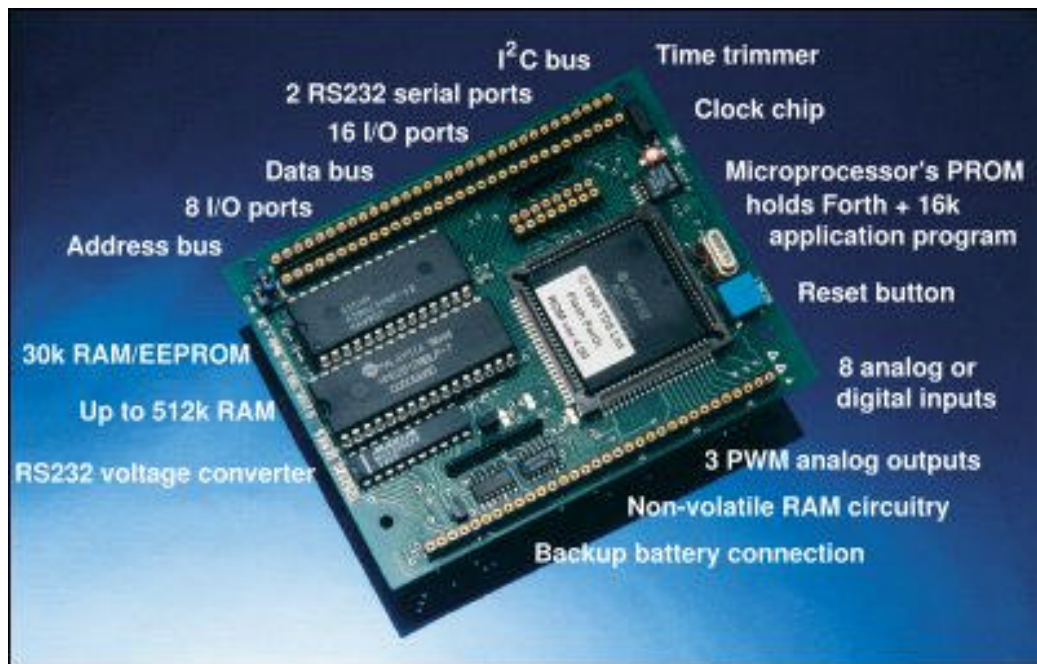
¹² CENPES/PETROBRAS. Autonomous System For Oil Pipelines Inspection. São Paulo, Brazil.

¹³ INDUSTRIAL DEVELOPMENT BANGOR. Oceanographic products ranging from seismic control equipment to drifter buoys.

¹⁴ SAELIG COMPANY. Data Loggers.

¹⁵ Waters Instruments, Rochester, MN. Embedded system for a kidney preservation machine.

Figura 25. Sistema de desarrollo TDS2020F y sus partes.



2.2.2.1. Memoria "Flash". La presencia de esta memoria en la TDS2020F representa ventajas para la aplicación, ya que la compilación directa en una memoria Flash-EEPROM es conveniente pues evita el costo y el tiempo de desarrollo en un programador de memoria PROM. Además, se puede borrar el programa remotamente por medio de un módem y volver a compilarlo. De esta manera, cuando sea necesario realizar modificaciones o actualizaciones en los cálculos y programas de control del computador de flujo, no será necesario extraer de él, la tarjeta TDS2020F.

2.2.2.2. Lenguaje Forth. El TDS2020F acepta dos lenguajes para su programación: *Forth* (alto nivel) y Ensamblador (bajo nivel). *Forth* es un lenguaje óptimo¹⁶ para sistemas embebidos basados en microcontrolador o microprocesador pues trabaja en el orden que realmente (a nivel de *hardware*) se hacen las cosas. No es tan popular en el ambiente académico ya que solo es conocido por dedicados diseñadores de sistemas embebidos.

La razón de programar en *Forth*¹⁷ es porque en sistemas embebidos y microprocesadores, se obtienen resultados más rápidos y eficientes (menos líneas de código) con este lenguaje. Para la máquina, es un tanto de más bajo nivel que 'C' pero es un lenguaje de más alto nivel que ensamblador o *assembler*. Aunque la tarjeta no trae la opción de programación en C, en el Capítulo Tres se exponen las bases de *Forth* y se dan ejemplos comparativos de Forth vs. C que respaldan el

¹⁶ DETTMER, Roger. Go Phase Go Forth. IEE Review, 1998.

¹⁷ BRODIE, L. and FORTH, Inc. Starting Forth. Prentice Hall, Englewood Cliffs, New Jersey. 1987.

porqué muchos diseñadores han escogido a *Forth* como el lenguaje para compilar sus sistemas embebidos.

El *Forth*, escrito especialmente para la TDS2020F, da acceso fácil a todas sus características y permite que el *software* sea escrito rápidamente. Es el mismo *Forth* Estándar Nacional Americano con muchas extensiones o comandos adicionales para explotar las características de *hardware* del TDS2020F.

2.2.2.3. Hardware. A continuación se presentan las características del hardware de la tarjeta:

- El TDS2020F está basado en el microprocesador Hitachi H8/532. El sistema embebido tiene una capacidad máxima de programa de 45kbytes, de los cuales 29kB están en una Flash-EEPROM y 16KB en el microprocesador Hitachi (generalmente con los 29kB es suficiente para cualquier aplicación). De memoria RAM tiene 512kbytes máximo y pueden estar respaldados empleando una batería para mantener los datos vitales mientras la tarjeta está sin alimentación. La memoria para almacenamiento se puede expandir también a más de un gigabyte con tarjetas PCMCIA o tarjetas *flash*.
- Necesita de una fuente de alimentación sencilla con 41mA de salida y además consume solo 155 μ A en un modo especial de baja potencia, usado especialmente en aplicaciones exclusivas de almacenamiento de datos.
- La tarjeta tiene entre 26 y 41 Entradas/Salidas (E/S) paralelas dependiendo de la configuración. Además posee dos puertos seriales RS232.
- Posee un conversor análogo/digital de 8 canales y 10 bits de resolución, el cual al no usarse (por no ser de suficiente resolución) se convierte en líneas de entrada digital. Además tiene tres canales de conversión digital/análogo de 8 bits, estos no se requieren para esta aplicación. Naturalmente se pueden conectar conversores externos de mayor resolución como se aplicó en el presente proyecto (conversor A/D ADS7825).

2.2.2.4. Velocidad. La TDS2020F tiene una frecuencia de reloj de 19.6608MHz y el tiempo de un ciclo es 102 nanosegundos (ns). Las instrucciones se procesan en un promedio de alrededor de 3 millones por segundo (MIPS). La latencia de interrupción es solo de 2.3 μ s (microsegundos), más el tiempo para completar la instrucción actual.

2.2.2.5. Memoria. El sistema embebido TDS2020F tiene la siguiente configuración de memoria.

- **Dentro del microprocesador H8/532:** 32kbyte de PROM OTP (programable una vez): 16kB de lenguaje de alto nivel *Forth* y 16kB de espacio en blanco para programas de aplicación del usuario (estos no se usaron). Además, contiene 1kbyte de RAM, de la cual la mitad se necesita para el sistema dejando 512 bytes libres para ser usados como variables para el programa de aplicación.

Para este proyecto no se usaron los 16kB del microprocesador para programa, sino que se empleó la Flash-EEPROM de 32kB explicada a continuación (El fabricante recomienda solo utilizar estos 16kB si ya se agotaron los 32kB de la memoria Flash).

- **Socket de 28 pines:** En este *socket* de la tarjeta se debe insertar una memoria Flash-EEPROM de 32kB no volátil para el programa final de aplicación del usuario. Debido a que los primeros 3kB son usados para ubicar algunos registros del sistema embebido y E/S (entrada-salida), en realidad solo se tienen disponibles 29kB para el programa del usuario. En la etapa de desarrollo se recomienda emplear una RAM de 32kB y se deja la Flash-EEPROM para la versión final.
- **Socket de 32 pines:** En este *socket* de la tarjeta se pueden insertar 512k, 128k o 32k de RAM, EEPROM o memoria *Flash* para variables adicionales, memoria auxiliar, o almacenamiento de datos. Las memorias EEPROM y *Flash* conservan los datos en ausencia de potencia; mientras las RAM pueden hacerse no volátiles añadiendo una pequeña tarjeta de batería o una celda externa de litio. En este proyecto se empleó una RAM de 512KB soportada por una batería de Litio con duración de 7 años.

2.2.2.6. Entrada salida paralela. La capacidad del puerto paralelo del TDS2020F depende si se usa o no el conversor A/D, el reloj de tiempo no volátil y otras facilidades. En resumen, se disponen entre 26 y 41 bits de entrada o salida de modo paralelo, dependiendo de las opciones empleadas. En el presente proyecto no se emplearon ni el conversor A/D ni el D/A internos de la CPU, y sólo fue requerida una línea digital de interrupción IRQ1 (pin c27) para la entrada de los pulsos de la turbina, más 8 líneas de entrada/salida (pines a14-a21) junto con 4 salidas del puerto B (PB4 a PB7) para manejo de teclado y *display*.

2.2.2.7. Entrada salida serial. La tarjeta tiene dos controladores y receptores seriales que usan formato y voltajes RS232. El Puerto 1 soporta todas las tasas de baudios de 75 a 38.4k y hay un MIDI adicional de 31.25k baudios. El Puerto 2 se puede usar para tasas por encima de 4800 baudios. Aunque el sistema se alimenta con una fuente de alimentación sencilla de +6 a +16V, los puertos serie usan $\pm 8V$ generados internamente en la tarjeta para dar los niveles lógicos de salida que cumplen con las especificaciones RS232. En el proyecto se emplea uno de estos dos puertos para comunicación con el programa de desarrollo del PC TDS-win, suministrado por el fabricante.

2.2.2.8. Microprocesador. El microprocesador que utiliza la TDS es el H8/532 de Hitachi, el cual es un dispositivo de 16 bits. Posee ocho registros de propósito general de 16 bits, *hardware* de multiplicación y división de 16 bits e instrucciones de 8 y 16 bits. Las manipulaciones directas a bit se aplica a registros, toda la memoria y la entrada-salida.

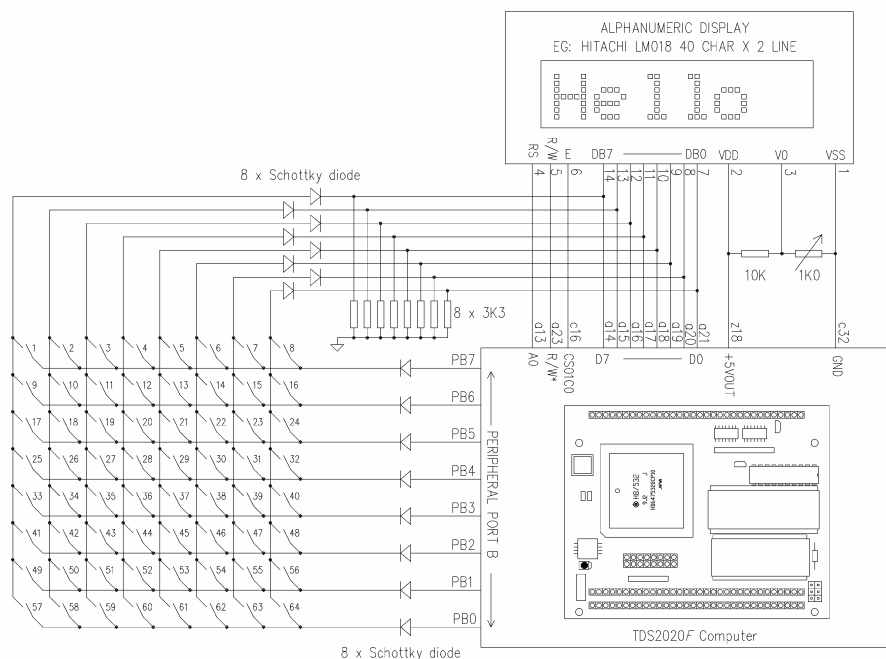
Además de los 4 contadores, el H8/532 posee un puerto serial síncrono y asíncrono, y un sistema de interrupciones muy versátil (65 interrupciones y vectores DTC, 8 niveles de prioridad). El Controlador de Transferencia de Datos (DTC) habilita la transferencia entre E/S y memoria sin necesidad de *software*.

Todas las capacidades del procesador están disponibles al usuario del TDS2020F a través del lenguaje *Forth* de alto nivel, lo que las hace más fáciles de utilizar. El acceso a algunas facilidades de *hardware* del microprocesador ya está construido en el sistema *Forth* pero el usuario es libre de usarlos a su manera por medio de *Forth* o *assembler*.

2.2.2.9. Soporte de teclado. Posee un sistema de *software* y *hardware* que puede habilitar hasta 64 teclas como lo muestra la Figura 26, para ser conectadas al sistema embebido empleando únicamente ocho de las líneas de entrada/salida paralelas. Para ello se conecta un puerto de salida paralelo a un lado de una matriz de teclas 8 x 8 por medio de rápidos diodos *schottky*. El otro lado de la matriz retorna por medio de otros ocho diodos al bus de datos. En este proyecto se emplearon sólo 8 teclas (4x4) que corresponden a las teclas de la matriz formada por las salidas 'PB7' a 'PB4' del puerto B y las líneas 'a14' a 'a17' de entrada/salida.

2.2.2.10. Displays de cristal líquido (LCD). La tarjeta posee el *software* para controlar cualquier LCD alfanumérico basado en el chip HD44780. A través de un chip externo se pueden manejar hasta 8 LCDs. Los LCDs gráficos también se pueden conectar sin ningún *hardware* adicional. La tarjeta dispone del *software* controlador para displays basados en HD61830, T6963C y otros controladores gráficos, como el empleado para el computador de flujo (display OPTREX de 20x4).

Figura 26. Conexión del teclado.



2.2.2.11. Temporizador-contadores. Hay cuatro temporizadores de *hardware*, uno es de 8 bits y los otros tres de 16 bits. Cada uno puede ser manejado internamente o externamente para conteo de eventos. Todos ellos tienen dos registros de captura de salida y los tres temporizadores de 16 bits tienen un modo de captura de entrada. Aunque los temporizadores no fueron usados explícitamente en este proyecto, ellos son empleados internamente por la CPU para el manejo de la multitarea, es decir para conmutar de una tarea a otra en un tiempo determinado. Además son usados por el sistema para llevar cuentas internas de segundos y *ticks* (53.33ms), variables del sistema que sirven para cualquier aplicación.

2.2.2.12. Fuente de alimentación. La TDS requiere de la alimentación de una fuente sencilla de +6V a 16V y su consumo típico de corriente es de 32mA y máximo de 41mA cuando está conectada serialmente a un PC u otro dispositivo. El modo operacional de baja potencia consume solo 155 μ A y no necesita soporte externo de *hardware*. La tarjeta tiene un generador de ± 8 V para ser usado por los puertos serie, el cual se puede apagar por *software* para ahorrar potencia. La alimentación negativa también es útil para algunos periféricos externos. El regulador en la tarjeta puede dar más de 100mA, así que hay corriente extra disponible para alimentar cualquier circuitería externa particular a la aplicación.

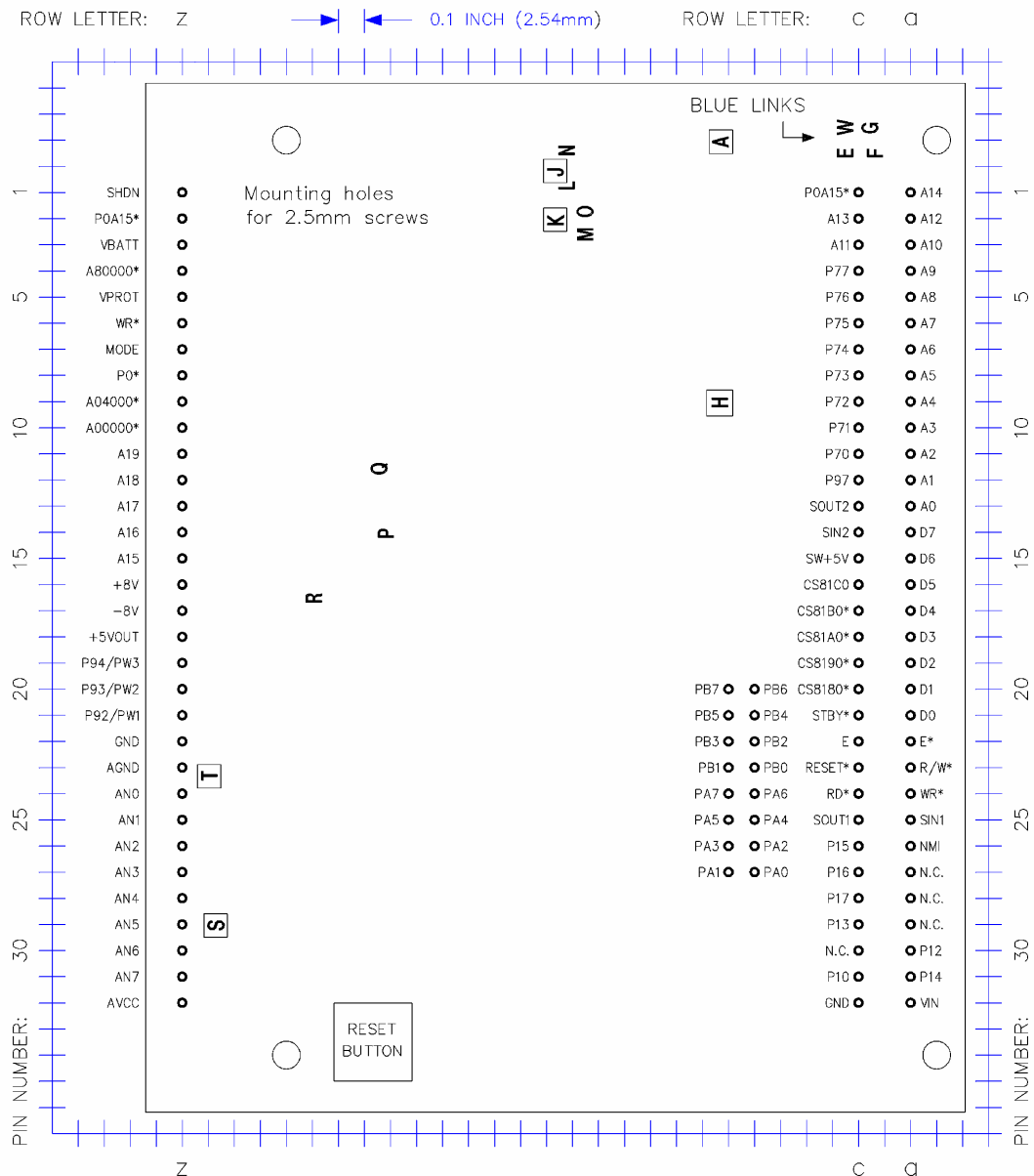
2.2.2.13. Pines de la TDS2020F. En la Figura 27 se detallan todos los pines de la tarjeta TDS2020F y su correspondiente señal. Esta figura sirve de referencia para la conexión de periféricos.

2.2.2.14. Dimensiones. El tamaño de la tarjeta es 100mm x 80mm con huecos de montaje para tornillos de 2.5mm. La máxima altura, excluyendo los conectores para pines es de 15mm. Si se instala una tarjeta de aplicación sobre los pines (modo sándwich), la distancia entre el sistema embebido y la tarjeta de aplicación es de 11 ± 1 mm. La temperatura de operación del sistema es de -10 a $+70^{\circ}\text{C}$. Todos los pines de conexiones coinciden con una matriz de 0.1 pulgadas, de modo que si se requiere, la tarjeta se puede montar fácilmente en una tarjeta prototipo con una matriz de huecos.

2.2.2.15. Sistema *Forth* ANS. El lenguaje es una completa implementación del *Forth* Estándar Nacional Americano que incluye punto flotante más una serie de instrucciones adicionales (desarrolladas por TDS Inc.) que facilitan el uso del hardware y la conexión de periféricos. El lenguaje *Forth* (que es basado en el concepto del trabajo de la pila) ha sido implementado con el objetivo de explotar al máximo el procesador H8/532 el cual tiene físicamente 2 pilas (datos y retorno).

Todos los registros *Forth* están implementados en registros de microprocesador y la pila de parámetros es la misma pila de *hardware* de los procesadores. Las dos pilas de *Forth* (datos y retorno) y las variables de usuario están dentro de la memoria del *chip* del microprocesador para obtener una velocidad de acceso máxima.

Figura 27. Pines de la tarjeta TDS2020F.



2.2.2.16. Extensiones ROM Forth. Entre las extensiones o comandos adicionales incluidos en la ROM Forth se tienen:

- Exploración de teclado.
- Displays de cristal líquido alfanuméricos.
- Soporte de periféricos de bus I²C que incluye RAM y reloj.
- Interrupciones escritas en código *assembler* o *Forth*.
- Multitarea.
- Reloj del tiempo del día.

- Operación a baja potencia.
- *Assembler* simbólico completo.
- E/S serial vectorizada y muchas otras palabras vectorizadas.
- Caracteres Hex, ASCII y de control.
- Aritmética de número doble (32-bit) y manipulación de pila.
- Medida de tiempo de ejecución exacta a un 1 μ s.

2.2.2.17. Multitarea. La tarjeta TDS2020F puede desarrollar impresiones, programas de visualización y comunicaciones, como actividades separadas corriendo al mismo tiempo, a través del uso del sistema multitarea. La depuración de cada tarea se puede hacer independientemente de las otras, permitiendo la corrida de un subprograma solo tecleando su nombre. La etapa final es un proceso mecánico que consiste en enlazar los programas individuales. En el presente proyecto se emplea un modo llamado multitarea por interrupción, en el cual una interrupción periódica es auto-generada para llamar a una tarea paralela.

2.2.2.18. Software de desarrollo. El programa de desarrollo *TDS-PC* para Windows, es el software de PC (computador personal) con facilidad de edición en múltiples ventanas. El código fuente se almacena en el disco del PC, aunque la compilación toma lugar dentro del TDS2020F. Oprimiendo una tecla (F8) se inicia el proceso automático que envía el programa fuente del puerto serie del PC al TDS2020F para su compilación. Luego se depura el código interactivamente en el TDS2020F. La Figura 28 muestra el paquete de desarrollo completo de *hardware*, *software* y manuales que se requieren en un desarrollo inicial.

Figura 28. Kit de *software* y *hardware* del TDS2020F.



2.3. INTEGRACIÓN DEL COMPUTADOR DE FLUJO.

2.3.1. Encapsulado. Para el computador de flujo se seleccionó el encapsulado tipo NEMA 4X¹⁸. Este tipo de encapsulado es el exigido para el montaje de computadores de flujo tanto en intemperie como en interiores.

Según esta selección se escogió la empresa americana ADALET, la cual fabrica encapsulados en acero, aluminio y fibra de vidrio. Este último material es el apropiado para el computador de flujo, dado su peso liviano y las características descritas a continuación.

2.3.2. Computador de flujo para medidor turbina. El sistema embebido, la tarjeta de acondicionamiento, y los sensores de presión, se encuentran ubicados en el interior del encapsulado NEMA 4X, presentado en la sección 2.3.1. El sensor de temperatura es instalado en el interior de la tubería de gas.

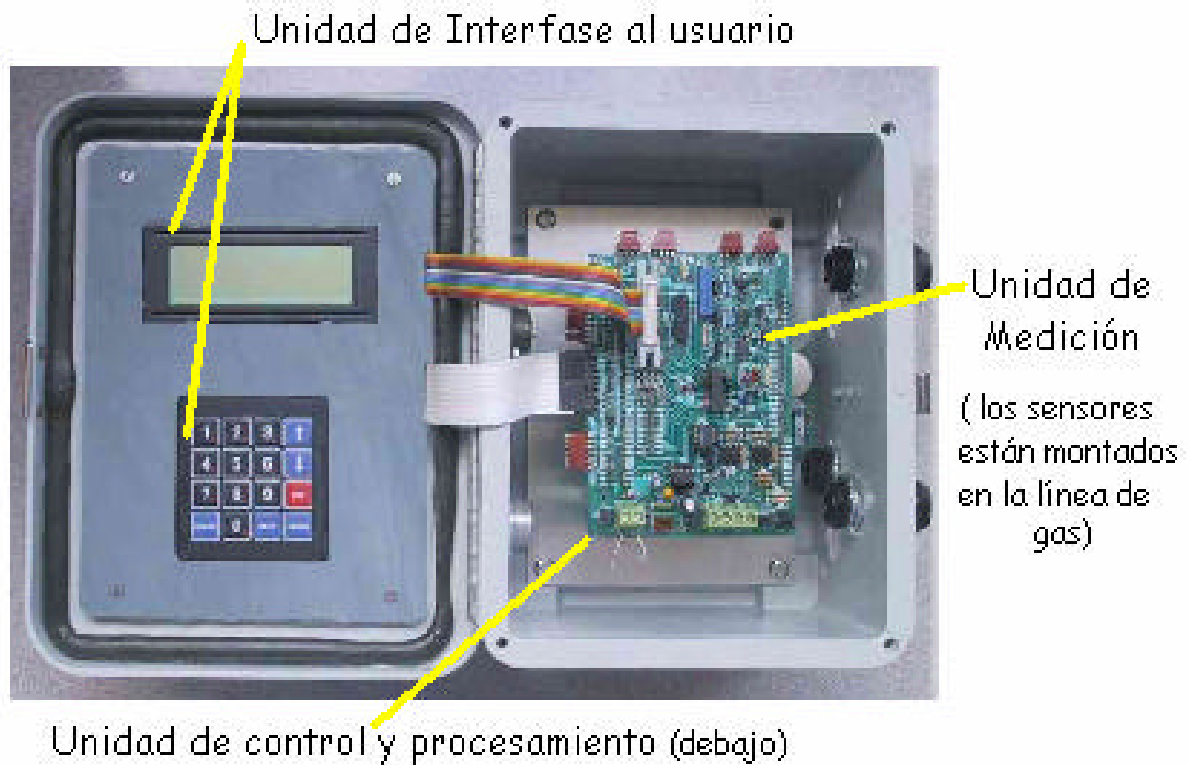
Las Figuras 29 y 30 muestran el exterior e interior del computador de flujo, respectivamente.

Figura 29. Vista exterior del computador de flujo.



¹⁸ NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION. NEMA 4X - Indoor use to provide a degree of protection against lint, dust, seepage, external condensation and spraying of water, oil and noncorrosive coolant.

Figura 30. Vista interior del computador de flujo.



2.3.3. Especificaciones y Costo Total. La Tabla 15, muestra las especificaciones del prototipo final. El costo final de este primer prototipo se estima en la Tabla 16.

Tabla 15. Especificaciones del Prototipo.

Descripción	Especificación
Tipo de Medidor	Turbina
Número de Brazos	2
Factor de Compresibilidad	AGA8-Gross2
Norma de Turbina	AGA7
Presión Máxima del Fluido	300 PSIG
Temperatura Máxima del Fluido	70 °F
Exactitud de Medición de Volumen	<1%
Tensión de Alimentación	9-15V
Consumo de Corriente Máximo	80mA
Entradas Análogas (2 presión, 2 temp.)	4
Resolución de las entrada análogas	16 bits
Entradas Digitales (pulsos de turbina)	2
Temperatura Ambiente de Operación	-10 a 70°C
Dimensiones (alto, ancho, profundo)	11.62 x 9.37 x 6.61 pulg.

Tabla 16. Costo Final del Prototipo.

Descripción	Costo Total \$US
Transductores de Presión (2 unidades)	740
RTDs (2 unidades)	480
Tarjeta de Acondicionamiento	260
Display y Teclado	70
Caja NEMA 4X	180
Sistema Embebido completo.	445
Accesorios	150
COSTO TOTAL	US\$2325

3. FIRMWARE DEL COMPUTADOR DE FLUJO.

El sistema de desarrollo TDS2020F puede programarse en *Forth* o en *Assembler*. Las normas AGA y todo el firmware del computador de flujo fueron implementadas utilizando el lenguaje de programación *Forth* por ser de más alto nivel que *assembler*, facilitar más su comprensión, y requerir menos tiempo de desarrollo. No obstante, dado que el lenguaje *Forth* no es común en los ambientes académicos, en la primera parte de este capítulo se expone una descripción de este lenguaje y algunas comparaciones con el lenguaje C por ser uno de los más difundidos. Posteriormente se tratan los algoritmos del computador de flujo, dedicados a la adquisición de señales y el desarrollo de los cálculos AGA, etc. También se incluye una sección donde se implementó la norma AGA8-Gross2 en Matlab para propósitos de verificación de los algoritmos como constancia previa a su desarrollo en *Forth*.

3.1. GENERALIDADES DE FORTH.

En el desarrollo de programas, el programador de un sistema embebido se enfrenta a una tarea compleja, pues debe tener cuidado en aspectos como el espacio de memoria a ocupar, y el desempeño del procesador de control, el cual debe satisfacer las demandas del sistema de control.

Los requerimientos de un código rápido y compacto han obligado a que muchos sistemas embebidos sean programados en *assembler*, herramienta que hasta para el más experimentado programador, resulta ser una tediosa labor propensa a errores. En muchas aplicaciones *assembler* ha dado origen a lenguajes de alto nivel (*high-level languages*, HLLs). Para sistemas embebidos la opción HLL es a menudo poco práctica, ya que la mayoría de los procesadores que se utilizan son simples recursos de bajo costo con una limitada opción de HLLs. Aún si se tuviesen buenos compiladores de alto nivel, su uso implica una ineficiencia en el código de ejecución.

En los años 60's, el estadounidense Charles Moore estaba escribiendo un *software* para controlar telescopios en la *Kitt Peak National Observatory* en Arizona. Insatisfecho con los lenguajes convencionales se propuso diseñar un nuevo lenguaje que haría su trabajo más fácil y productivo. El resultado fue un lenguaje altamente distintivo, el cual se llamó *Forth*. La razón de desarrollar este nuevo lenguaje fue el deseo de superar la necesidad de que un ingeniero tuviera que aprender el uso de un gran número de herramientas (compiladores, enlazadores, ensambladores y directivas de lenguajes de muy alto nivel).

Forth es un entorno de programación interactivo, originalmente diseñado para programadores que desarrollaban aplicaciones usando mini y microordenadores. Su principal uso está en aplicaciones científicas e industriales tales como instrumentación, robótica, control de procesos, gráficos y procesamiento de imágenes, inteligencia artificial, y aplicaciones de negocios. La principal ventaja de *Forth* es el desarrollo rápido e interactivo de *software* y el uso eficiente del *hardware* del ordenador.

Forth es a menudo referido como un lenguaje porque este es su aspecto más visible. Sin embargo, *Forth* es más que un lenguaje de programación convencional, en el que todas las capacidades normalmente asociadas con un gran número de programas separados (compiladores, editores, ensambladores, etc.) están incluidas dentro del mismo.

Las implementaciones originales de *Forth* fueron sistemas autónomos que incluían funciones normalmente ejecutadas de forma separada por sistemas operativos, editores, compiladores, ensambladores, depuradores y otras utilidades. Se considera especialmente adecuado para sistemas incrustados o embebidos.

La comunidad *Forth* creó el primer estándar a mediados de los 70 por FIG (*Forth Interest Group*). Posteriormente en 1994, tuvo un refinamiento por el Comité Técnico de Estándares Nacionales Americanos (ANSI X3.215-1994) el cual fue el definitivo estándar de *Forth* aceptado mundialmente y conocido como "ANS Forth". Este estándar también fue adoptado como internacional en 1997 por la Organización Internacional de Estándares (ISO/IEC 15145:1997).

3.1.1. Programación. *Forth* cuenta entre sus ventajas el ocupar poca memoria, rápido y extensible. Su punto débil está en tener una sintaxis extraña que lo hace complejo de aprender si no se tienen suficientes conocimientos en programación. Una característica importante es que es un lenguaje basado en el uso de una pila (*stack*). Por ejemplo, la sentencia en el lenguaje *Basic* " PRINT 5 - 3 " es equivalente en *Forth* a " 5 3 - ." donde los números 5 y 3 son puestos en la cima de la pila (3 arriba del 5), el signo menos (-) resta el elemento de la cima del que está debajo, elimina ambos operandos y pone en la pila el resultado, y el punto (.) muestra en pantalla el número de la cima y lo elimina.

Una pila es simplemente un conjunto de números donde la última variable en entrar es la primera en salir (estructura LIFO). Las pilas son usadas por casi todos los lenguajes de programación. Mientras que la mayoría de lenguajes ocultan la operación de la pila para el programador, en *Forth* el programador controla la pila directamente. Por ejemplo al ejecutar: 3 4 + y oprimir 'Enter', el sistema retornará 'OK'. De esta manera *Forth* ha confirmado que la tarea se ha ejecutado correctamente. Al oprimir 'Enter', el interpretador de *Forth* reconoce a 3 y 4 como números y los ubica en la pila. El signo '+' es interpretado como una palabra. De esta manera, *Forth* se caracteriza por ser un lenguaje que trabaja dos elementos básicos: números y palabras.

Para determinar el significado de '+', el interpretador de *Forth* consulta el diccionario *Forth* que contiene detalles de todas las palabras definidas. Como es de

esperarse '+' suma el primero y el segundo número y pone el resultado en la cima de la pila. Para confirmar esto, se puede usar la palabra '.', que muestra el valor de la cima de la pila después de oprimir 'Enter'. En el ejemplo mostrará '7 ok'.

Otra característica importante de *Forth* es el empleo de la RPN (*Reverse Polish Notation*), donde los operandos preceden al operador. En el anterior ejemplo se puede notar que el '+' aparece después de los números que van a ser sumados. Esto parecería ser incómodo, pero lo que trata de hacer el lenguaje es imitar la forma de operar el computador, ya que éste debe conocer los operadores (variables), antes de la operación a desarrollar. Usando RPN, los operadores y los operandos, se presentan automáticamente al computador en la secuencia correcta.

Los programas en *Forth* son diseñados por la definición de nuevas palabras requeridas para implementar el objetivo del programa. Cada nueva palabra puede tener un nombre descriptivo que debe ser corto, claro, y adecuado; haciendo que el programa final sea lo más entendible posible.

Es importante apreciar que todas las palabras en *Forth*, ya sean predefinidas o definidas por el programador, tienen exactamente el mismo estado: son simples entradas en el diccionario. A medida que se crea un programa en *Forth*, se va extendiendo el diccionario según las necesidades, obteniendo así un lenguaje más completo y eficiente. Esta naturaleza extensible de *Forth*, es su más distintiva característica.

Cuando se encuentra una palabra se busca en el diccionario para hallar su definición. La función asociada con la palabra es entonces ejecutada, o se compila una referencia dentro de una nueva definición. Si la palabra no se puede encontrar en el diccionario, el sistema intenta convertirla en un número. Así, si se tiene éxito el número es emplazado en una pila de parámetros. Si falla la conversión de la palabra en un número se muestra la palabra junto a un mensaje de error, indicando que la palabra es desconocida para el sistema.

Forth pertenece a la clase de lenguajes llamada *Threaded Interpretive Languages*. Esto significa que puede interpretar comandos (subrutinas o programas) introducidos por la consola, así como también crear (compilar) nuevas subrutinas y programas. Lo que hace a *Forth* ser tanto un compilador como un intérprete es el conjunto de palabras (subrutinas *Forth*) que, cuando son introducidas y ejecutadas, crean nuevas subrutinas *Forth*.

Forth está caracterizado por cinco elementos principales:

- un diccionario de palabras.
- dos pilas, una para los parámetros, y otra para almacenamiento provisional.
- un intérprete de teclado (canal de entrada).
- un ensamblador.
- un almacenamiento virtual.

3.1.2. Experiencias de *Forth* en la industria. La siguiente es una selección de proyectos con *Forth* a nivel industrial:

- La división de máquinas recreativas de *Atari* usaba *Forth* en muchas máquinas de juegos. La empresa creía que el camino más rápido para un código de alto rendimiento estaba en escribirlo en *Forth*, y codificar los bucles más profundos en ensamblador.
- Según una entrevista con desarrolladores de los famosos juegos de aventuras de Infocom (*Hitch Hikers Guide to the Galaxy*, y otros), los intérpretes de juegos eran escritos en *Forth*.
- *Unison World* produjo más de una docena de juegos para máquinas CP/M, todos escritos en *fig-Forth*. Según Marc de Groot, director técnico, portar los juegos basados en Z80 a microprocesadores 6502 y 6809, normalmente tomaba menos de tres meses.
- Los dispositivos *hand-held* usados por la empresa *Federal Express* para los sistemas de seguimiento de los paquetes están programados en *Forth*. El director de proyectos, *Gene Farrar*, dice que puede actualizar el *firmware* bajo requerimiento en sólo unas semanas; mientras que requiere de al menos 6 meses para el mantenimiento del código basado en C usado por su propio grupo.
- Tres de las cuatro cargas de pago astronómicas en el vuelo del transbordador espacial Columbia de noviembre de 1990 estaban programadas en *Forth*. Conforme a lo declarado por John Hayes, perteneciente al *Johns Hopkins University's Applied Physics Laboratory* (uno de los equipos de diseño), la flexibilidad de *Forth* fue el principal factor que les permitió trabajar, rodeados de los problemas de *hardware* que plagaron el vuelo.
- *Sun Microsystems* usa una ROM de arranque programable basada en *Forth* en sus estaciones de trabajo. Actualmente venden entre 500 y 1000 por día. Dicho sistema fué estandarizado en 1994 como IEEE 1275-1994 *Standard for Boot (Initialization Configuration) Firmware*. Esta normatividad provee la plataforma básica común de los *PowerPC*.
- Los efectos especiales generados por ordenador usados para crear el *Cyberdyne Systems T1000* de la película "Terminator 2: El día del juicio", fueron producidos mediante un paquete de *Morphing* basado en *Forth*.

3.1.3. Algunas palabras *Forth* (comandos). *Forth* contiene cientos de palabras predefinidas en su diccionario, y la mayoría de ellas modifican el estado de la pila. Un ejemplo de ello, es la instrucción DUP que duplica la cima de la pila. Es decir, si (n1) es el estado inicial de la pila antes de la ejecución de la palabra DUP, (n1 n1) será el estado final. Como norma general para entender el funcionamiento de una palabra, ellas vienen declaradas y a continuación entre paréntesis se describe la acción que la palabra realiza sobre la pila. Se usa un doble guión (--) para indicar a

la izquierda el estado previo de la pila y a la derecha el estado posterior a la ejecución de dicha palabra. Siendo así, la palabra DUP se encuentra declarada así:

DUP (n1 -- n1 n1) : donde n1 significa un número que está en la cima de la pila, y seguido del doble guión, se observa que quedó el número ahora esta dos veces sobre la pila. Esta es una forma didáctica para entender lo que le hace cada palabra a la pila pero no es obligación poner este comentario en la definición de las palabras.

Otros ejemplos de palabras de *Forth* predefinidas son:

2DUP (n1 n2 -- n1 n2 n1 n2) : Duplica el primer y segundo ítem de la pila.

DROP (n1 n2 -- n1) : Elimina la cima de la pila.

SWAP (n1 n2 -- n2 n1) : Intercambia los dos últimos números de la pila.

TUCK (n1 n2 -- n2 n1 n2) : Copia la cima de la pila (n2) detrás del segundo ítem de la pila (n1) desplazando los dos primeros (n1 y n2).

+ (n1 n2 -- n3) : suma n1 y n2, los borra, y pone el resultado n3 en la cima.

- (n1 n2 -- n3) : Le resta n2 a n1, y pone el resultado n3 en la pila, borrando los dos operandos n1 y n2.

/ (n1 n2 -- n3) : Divide n1 entre n2, y el cociente entero n3 es puesto en la pila, borrando al dividendo y divisor.

MOD (n1 n2 -- n3) : Divide n1 entre n2, y el residuo n3 es puesto en la pila.

**** (--) : signo usado para poner comentarios en la definición de una palabra. También se utilizan los comentarios entre paréntesis. No le hacen nada a la pila.

i (n1 addr --) : Asigna la cima de la pila a una variable. Por ejemplo: 3 X i equivale a: X=3 .

@ (addr -- n1) : Extrae el valor contenido en una variable y lo pone en la pila. Por ejemplo, X @ pone el valor contenido en X sobre la pila. Si se olvida colocar el @ se mantiene la dirección de la variable X en la pila (lo cual es un error común).

Las palabras predefinidas pueden ser muy útiles, pero el verdadero poder de un lenguaje está en las nuevas palabras que se puedan crear. Muchas implementaciones de *Forth* incluyen la palabra NIP, la cual borra el segundo ítem de la pila, su ficha nemotécnica sería NIP (n1 n2 - n2). Pero si NIP no está predefinida, se podría crear de la siguiente manera:

```
: NIP
  SWAP DROP ;
```


La descripción de las palabras empleadas para la creación de NIP, es la siguiente:

- `:` es una palabra predefinida de *Forth*, que indica el comienzo de la definición de una nueva palabra.
- `'NIP'` es el nombre de la nueva palabra.
- `'SWAP'` y `'DROP'` son palabras predefinidas en *Forth*.
- `;` es una palabra predefinida que indica el final de una nueva definición de palabra.

Según lo anterior, cualquier programa en *Forth* que defina nuevas palabras tiene la siguiente estructura: inicia con `:` seguido por el nombre de la nueva palabra, continuando con la definición de la nueva palabra escrita en palabras *Forth* previamente definidas, y termina con `';`.

La definición de una palabra que eleve al cuadrado, se podría implementar de la siguiente manera: se duplica el número con `DUP`, se multiplica por sí mismo y finalmente se muestra en pantalla con punto `(.)`.

```
: SQUARED DUP * . ;
```

Para elevar 8 al cuadrado se escribiría en la ventana interactiva de *Forth* así:

```
8 SQUARED [enter]
64 ok (mostrará la pantalla)
```

Virtualmente todas las modernas CPU están diseñadas en torno a pilas. *Forth* usa eficientemente la CPU porque refleja en su sintaxis dicha arquitectura subyacente basada en pilas.

Los números se añaden siempre a la cima de la pila, y se remueven desde la cima de la pila. La línea de entrada *Forth*

```
2 5 73 -16 <Enter> ok
```

deja la pila de la siguiente forma:

Celda#	Contenido
0	-16 (Cima de la pila)
1	73 (Siguiendo en la pila)
2	5
3	2 (Fondo de la pila)

Supongamos que después de lo anterior se copia la siguiente línea:

```
+ - * . <Enter> -104 ok
```

El resultado de `-104` se genera de las operaciones sucesivas que se presentan en la pila cuya descripción se muestra a continuación:

celda#	inicial	+	-	*	.
--------	---------	---	---	---	---

0	-16	57	-52	-104	
1	73	5	2		
2	5	2			
3	2				pila vacía

La operación final punto ".", muestra el resultado -104 en la pantalla, dejando vacía la pila.

3.1.4. Ejemplos Forth versus C. Programar en *Forth* consiste esencialmente en la creación de nuevas palabras necesarias para la implementación de un código.

Como una ilustración del estilo de programación *Forth*, se considerarán tres programas: el primero eleva un número entero a una potencia dada entera, el segundo halla el máximo común divisor de dos números empleando el método de Euclides, y el tercero calcula la raíz cuadrada de un número con el método de Newton. Los ejemplos muestran su contraparte en C para efectos de comparación.

3.1.4.1. Elevar a una potencia dada. Elevar un número entero a una potencia entera dada es simplemente multiplicar el número por sí mismo un número determinado de veces. En este ejemplo desarrollado en C, la función *main* invoca a la función *power* para mostrar las potencias de 0 a 9 de los números 2 y -3.

```
main()      /* función principal */
{
    int i;
    for (i=0; i<10; ++i)
        printf("%d %d %d\n", i, power(2,i), power(-3,i));
}
```

La versión *Forth* de este programa es:

```
: main ( - ) \ palabra principal, puede ser cualquiera
  10 0       \ La variable I es autocreada en los ciclos
  DO I . 2 I power . -3 I power . CR
  LOOP
;
```

Naturalmente *Forth* enfoca la programación desde un punto de vista diferente, por ejemplo "4 5 power" pone los números 4 y 5 en la cima de la pila y *power* eleva 4 a la 5 y pone el resultado (1024) en la cima de la pila borrando los operandos anteriores (4 y 5). La palabra *CR* significa retorno de carro y es para saltar una línea después de cada ciclo del *LOOP*.

La versión en C de la función *power* sería:

```
power(x,n)  /* eleva x a la n-ava potencia; n > 0 */
{
    int x,n;
```

```

int i,p;
p = 1;
for (i=1; i<=n; ++i)
    p = p * x;
return(p);
}

```

En *Forth* esto puede ser escrito como:

```

: power ( x n -- p ) \ eleva x a la n-th potencia; n > 0
  1 SWAP 0 DO OVER * LOOP NIP ;

```

Los argumentos (x y n) de la función en C se han convertido en un comentario en *Forth* (el cual es opcional). El comentario indica las modificaciones de la pila (x n -- p) al decir que la palabra *power* toma los dos últimos valores de la pila (x n) y deja sólo valor en ella (p, el resultado). Este comentario es ignorado por el compilador y sirve para documentación al programa.

La simplicidad de *Forth* tiene su precio. No siempre es fácil entender lo que le sucede a la pila, y para alguien no experimentado le requiere un poco de práctica pensar en términos de la pila. A continuación se expone explícitamente el estado de la pila después de cada palabra o comando de *power*.

: power	(x n -- p)	\ -- x n
1		\ -- x n 1
SWAP		\ -- x 1 n
0		\ -- x 1 n 0
DO		\ -- x 1
OVER		\ -- x 1 x
*		\ -- x 1*x
LOOP		\ finalmente: -- x x^n
NIP		\ -- x^n
;		

No hay reglas estrictas para programar en *Forth*, y lo óptimo está en la creatividad del programador para encontrar el camino más eficiente de implementar un algoritmo en términos de un manejo rápido y adecuado de la pila.

Otra diferencia de los dos lenguajes es que C usa variables locales como argumentos para la función llamada, tanto para el momento de llamar como en la función llamada.

En *main* : int i ; power(2,i)
 En *power* : power(x,n) ; int x,n;

En *Forth* simplemente se ponen datos sobre la pila y se evita así la creación de variables extras. La pila actúa como un *block* de notas (*notepad*). En principio no se necesitan, aunque es posible adicionar variables locales a un sistema *Forth*.

La interface de *Forth* con el usuario es un intérprete de línea de comandos. Se puede llamar y ejecutar cada palabra en el intérprete de comandos y así probarlas por separado. Mientras que en C se necesita la función *main* para probar la función *power*, en *Forth* se puede probar instantáneamente escribiendo directamente en la línea de comandos "2 3 power ." y verá el resultado de 2 a la 3. Así se puede probar una palabra con diferentes conjuntos de parámetros de forma instantánea.

3.1.4.2. Máximo Común Divisor (Euclides): Euclides en su libro "Elementos", dio este método para calcular el máximo común divisor (MCD) de dos números, con el que no es necesario descomponer el número en factores primos ya que esta descomposición puede ser muy difícil cuando los números son muy grandes. El algoritmo dice: sean a y b los números de los que queremos calcular el máximo común divisor. Hacemos las siguientes divisiones hasta que el residuo de una de ellas sea cero.

$$a = bq_1 + r_1$$

$$b = r_1q_2 + r_2$$

$$r_1 = r_2q_3 + r_3$$

$$r_2 = r_3q_4 + r_4$$

$$\dots\dots\dots$$

$$r_{n-1} = r_nq_{n+1} + 0$$

Siendo q_1, q_2, \dots los cocientes y r_1, r_2, \dots los residuos. El máximo común divisor es r_n . Por ejemplo, al calcular el MCD de 200 y 162 se tiene lo siguiente:

$$200 = 162 \times 1 + 38$$

$$162 = 38 \times 4 + 10$$

$$38 = 10 \times 3 + 8$$

$$10 = 8 \times 1 + 2$$

$$8 = 2 \times 4 + 0 \quad \text{El MCD de 200 y 162 es 2.}$$

Una forma de implementarlo algorítmicamente es la siguiente:

Introducidos 2 números, 18 y 12 por ejemplo:

18 12 | Se toma el valor mayor y se le resta el menor (18-12=6)

6 12 | El resultado se pone debajo del numero mayor, el otro pasa igual.

Y queda así:

6 12 | Se toma de nuevo el valor mayor y se le resta el menor (12-6=6)

6 6 | y el resultado se pone debajo del número mayor.

El algoritmo termina cuando los 2 números finales sean iguales, y este número igual será el MCD (en este caso 6).

El código en C para realizar esta tarea solicitando los dos números de pantalla es:

```
#include<stdio.h>
#include<conio.h>

long int MCD(long int a, long int b)
{
    while(a!=b)
    {
        if(a>b)
            a=a-b;
        else
            b=b-a;
    }
    return(a); /* también se puede return(b) pues serán iguales */
}

void main()
{
    long int num1,num2; clrscr();
    printf("\n Primer numero ==>"); scanf("%ld",&num1);
    printf("\n Segundo numero ==>"); scanf("%ld",&num2);
    printf("\n El maximo comun divisor es ==> %ld",MCD(num1,num2));
    getch();
}
```

Con la sentencia *if* dentro del *while* se garantiza que siempre se resta el número mayor del menor, por esta razón no interesa el orden de entrada de los números.

Ahora en *Forth* el método de Euclides se resume en una línea de código trabajando con los residuos, al definir una palabra llamada MCD la cual antecedita de los 2 números a trabajar dejará en la pila el MCD de ellos.

Empleando los residuos el método se resuelve así, nuevamente con 18 y 12:

18 12 | Se divide y se toma el residuo de la operación (Res 18/12=6) y se mira si es cero, si el residuo no es cero se continua el procedimiento.

12 6 | Se divide el último número (12) de la operación anterior entre el último residuo (6) y se toma el residuo (0). El MCD es el dividendo de la última división, MCD=6. En el caso que el residuo no sea cero, se continúa el procedimiento.

Si se hubiera entrado primero 12 y luego el 18 quedaría así:

12 18 | Al dividir 12 entre 18 queda como residuo 12.

18 12 | Se divide el último número (18) de la operación anterior entre el último residuo (12) y se toma el residuo (6).

12 6 | Se divide el último número (12) de la operación anterior entre el último residuo (6) y se toma el residuo (0). Como el residuo es cero entonces MCD=6.

Se puede observar, que el método es independiente del orden de entrada de los números.

La definición de MCD en *Forth* es la siguiente:

```
: MCD ( n1 n2 - n3 ) BEGIN DUP WHILE TUCK MOD REPEAT DROP ;
```

Después de introducir esta definición en el diccionario (compilarla), podemos interactuar así con el intérprete:

```
12345 32325 MCD .  
15 ok (mostrará la pantalla)
```

Al desglosar esta definición observamos el comportamiento de la pila:

```
: MCD ( n1 n2 - n3 ) \ -- n1 n2  
  BEGIN              \ -- n1 n2  
  DUP                \ -- n1 n2 n2  
  WHILE              \ -- n1 n2 Si n2=0 salta a luego del REPEAT  
    TUCK              \ -- n2 n1 n2  
    MOD               \ -- n2 nr  
  REPEAT             \ -- n2 nr Regresa a BEGIN  
  DROP ;             \ -- n1
```

Por ejemplo, el segundo ciclo de esta palabra conllevaría a :

```
  BEGIN              \ -- n2 nr  
  DUP                \ -- n2 nr nr  
  WHILE              \ -- n2 nr Si el residuo es cero sale, sino sigue  
    TUCK              \ -- nr n2 nr  
    MOD               \ -- nr nr2 Este será el segundo residuo  
  REPEAT             \ -- nr nr2  
  DROP ;             \ -- n2
```

Es importante aclarar que para *Forth*, un *verdadero* es cualquier número diferente de cero y un *falso* es un número igual a cero, esto es importante a la hora de la evaluación de banderas o condiciones. Por ejemplo, la palabra WHILE toma la cima de la pila y si es verdadera (diferente de cero) ejecuta la siguiente palabra o instrucción, pero si es falsa (=0) el control del programa salta a ejecutar la siguiente palabra que haya después del REPEAT más cercano.

3.1.4.3. Raíz cuadrada (Newton). Este método se basa en el principio que si m es un estimado de la raíz cuadrada de N , entonces $\frac{1}{2}(m + N/m)$ es una mejor estimación. Tomando $N/2$ como el valor inicial de m y sucesivamente aplicando $(m+N/m)/2$ se pueden generar raíces cuadradas de gran exactitud.

En lenguaje C aplicando 20 iteraciones (que es suficiente para números hasta 10^7) esto se pudiera resolver como:

```
#include <stdio.h>
#include<conio.h>

main(void)
{
    float x, n;
    int i;
    printf("\n Raiz de :");    scanf("%f",&N);
    x = N/2;

    for (i = 0; i < 20; i++) {
        x = (x + n/x ) / 2;
        printf("%f\n", x);
    }
}
```

En lenguaje *Forth* se puede crear el programa del método de Newton para hallar la raíz cuadrada de un número N, generando dos nuevas palabras: "1erEstima" la cual calcula $N/2$, y "MejorEstima" que calcula $(m + N/m)/2$. Repitiendo "MejorEstima" iterativamente las veces que el usuario desee se llega al valor de la raíz.

La primera estimación y la mejor estimación quedan definidas por:

```
: 1erEstima ( N -- N N/2 )
DUP 2 / ;

: MejorEstima ( N m -- N (m+N/m)/2 )
2DUP / + 2 / ;
```

El detalle de las operaciones en la pila de cada palabra se muestran en la Tabla 17.

Tabla 17. Secuencia de operaciones de un programa en *Forth*.

Palabra	Sub - Palabras	Efecto en la Pila
Estado inicial de la pila		-- N (el número a sacar raíz)
1erEstima	DUP	-- N N
	2	-- N N 2
	/	-- N (N/2)
Estado inicial de la pila		-- N m
MejorEstima	2DUP	-- N m N m
	/	-- N m (N/m)
	+	-- N (m+N/m)
	2	-- N (m+N/m) 2
	/	-- N (m+N/m)/2 (nueva estimación)

La palabra “MejorEstima” tiene que ser aplicada sucesivamente, lo cual se logra mediante un ciclo DO LOOP, el cual tiene el siguiente formato:

Límite Inicio DO ... palabras forth LOOP

Límite e *Inicio* son dos números que controlan el número de *LOOPS* realizados, cuyo total es simplemente la resta (*Límite-Inicio*) número de veces. Si optamos por desarrollar 20 interacciones sucesivas de “MejorEstima”, entonces el programa raíz cuadrada cuyo nombre se escogió como RAIZ2, tomará la siguiente escritura:

```
: RAIZ2      ( N -- vN )
1erEstima
20 0 DO MejorEstima LOOP
NIP . ;
```

El NIP final simplemente pone en orden la pila, eliminando el segundo elemento no deseado y el punto muestra el resultado. El anterior ejemplo realiza un número fijo de iteraciones, acepta solo números enteros mayores que 1, y retorna una respuesta entera.

3.2. IMPLEMENTACIÓN DE LAS NORMAS AGA.

Como se explicó en el Capítulo 1, las normas AGA empleadas en este proyecto son el AGA8-G2 y el AGA7. Con la primera se halla el factor de compresibilidad Z que será necesario en los cálculos del AGA7, la cual da como resultado el volumen de gas corregido a la temperatura y presión contractuales. No obstante, el rigor del cálculo está en el AGA8-G2 pues maneja un número grande de variables y constantes, además es un algoritmo iterativo en comparación con el AGA7 que, como se verá más adelante, es simplemente una multiplicación explícita de 5 factores, uno de ellos el Z.

3.2.1. AGA8 – GROSS 2. Esta norma tiene en sus anexos la forma de implementarse en *software*, las rutinas que deben crearse y todo un completo código en FORTRAN para su desarrollo. También tiene una tabla para validar los resultados, y comparar si el *software* desarrollado por el usuario cumple con este estándar. El primer paso en el presente proyecto, fue pasar este código FORTRAN a un lenguaje de fácil implementación y más entendible como MATLAB para así verificar los resultados, la validez del código fuente y apropiarnos más del funcionamiento de los algoritmos dados por AGA. El paso siguiente fue trasladar el código de Matlab ya probado, al lenguaje *Forth*.

El procedimiento del método AGA8-Gross 2 para calcular Z fue explicado en la sección 1.2.2.1. Por conveniencia repetimos las ecuaciones 11, 12, y 13, ya que son la base del método. Para las demás ecuaciones mostradas en los siguientes diagramas de flujo remitirse a la misma sección.

$$Z_f = 1 + B_{mix} d + C_{mix} d \quad (\text{ecuación 43a})$$

$$Z_b = 1 + (B_{mix} \cdot P_{gr}) / (R \cdot T_{gr}) \quad (\text{ecuación 43b})$$

$$B_{mix} = \sum_{i=1}^N \sum_{j=1}^N B_{ij} x_i x_j \quad (\text{ecuación 44})$$

$$C_{mix} = \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N C_{ijk} x_i x_j x_k \quad (\text{ecuación 45})$$

Z = factor de compresibilidad base (Z_b) o del fluido (Z_f).

B_{mix} = segundo coeficiente virial de la mezcla.

C_{mix} = tercer coeficiente virial de la mezcla.

d = densidad molar (moles por unidad de volumen).

B_{ij} = componente individual de interacción del segundo coeficiente virial.

C_{ijk} = componente individual de interacción del tercer coeficiente virial.

$x_i x_j x_k$ = fracciones molares de los componentes del gas.

N = número de componentes de la mezcla del gas.

Las Figuras 31 y 32 representan el diagrama de flujo del método Gross 2 para el cálculo del factor de compresibilidad (Z). El procedimiento está dividido en dos grandes secciones:

En la primera sección (Figura 32) se hace la entrada de información de caracterización del gas correspondiente. Para el prototipo del presente proyecto, los datos de entrada son densidad relativa del gas (G_r), contenido de dióxido de carbono (X_{CO_2}) y contenido de nitrógeno (X_{N_2}). Esta sección tiene como objetivo determinar H_{CH} (calor molar del hidrocarburo equivalente CH) y un factor de compresibilidad base (Z_b) a condiciones base o de referencia.

El procedimiento supone inicialmente de un factor de compresibilidad a condiciones de base de uno ($Z_b = 1$) y deduce nuevos Z_b hasta que la iteración produzca una diferencia menor que $e = 5 \times 10^{-11}$.

El ρ_{air} es la densidad de masa del aire que se calcula según la ecuación 46, T_{gr} y P_{gr} son la temperatura y presión de referencia (para Ecogás estas son $T_{gr} = 60^\circ F$ y $P_{gr} = 14.73 \text{ psia}$), R es la constante del gas (8.31451 J/mol-K) y G_1 con G_2 son constantes químicas de relación entre masas molares. ($G_1 = -2.709328$ y $G_2 = 0.021062199$).

$$\rho_{air} = 28.96256 / (R \cdot T_{gr} / P_{gr} + VIR) \quad (\text{ecuación 46})$$

Donde el término VIR se calcula con la siguiente ecuación 47:

$$VIR = -0.12527 + 5.91 \times 10^{-4} T_{gr} - 6.62 \times 10^{-7} T_{gr}^2 \quad (\text{ecuación 47})$$

Primero se calculan la masa molar total (M_r) y la masa del hidrocarburo equivalente (M_{rCH}), esta última como la resta a la masa molar total (M_r) de la masa del dióxido de carbono (M_{rCO_2}) y del nitrógeno (M_{rN_2}). Seguidamente con esos datos se calcula

el H_{CH} (calor molar del hidrocarburo equivalente CH) con la relación de constantes G_1 y G_2 .

Ahora con este primer resultado de H_{CH} , se calcula el coeficiente de interacción B_{CH-CH} (ecuación 17, Capítulo Uno) y el segundo coeficiente virial B_{mix} (ecuación 14, Capítulo Uno). Los anteriores se hallan a una temperatura de referencias de 60°F.

Teniendo B_{mix} , finalmente se calcula $Z_b = 1 + (B_{mix} \cdot P_{gr}) / (R \cdot T_{gr})$ y se compara con el valor inicial supuesto y así se repite el procedimiento de esta sección hasta que las diferencias de resultados sea menor a $e = 5 \times 10^{-11}$.

La segunda sección (Figura 32) realiza el siguiente procedimiento para determinar finalmente el factor de compresibilidad a condiciones del fluido (Z_f):

- Se toman los valores de H_{CH} y Z_b previamente hallados, y se lee la temperatura (T) y presión (P) del fluido al cual se desea hallar el factor de compresibilidad (en el computador de flujo estas son las lecturas de los transductores).
- Se hallan los coeficientes de interacción binario (B_{CH-CH}) y terciario ($C_{CH-CH-CH}$) con la temperatura del fluido (dada por la RTD), empleando las ecuaciones 18 y 19 del Capítulo Uno, respectivamente.
- Se calculan el segundo y tercer coeficiente virial (B_{mix} y C_{mix}) en la ecuación de estado (ecuaciones 14 y 15 Capítulo Uno, respectivamente). Estos coeficientes son función de la temperatura del fluido, el calor molar gross del hidrocarburo y la fracción molar de componentes no hidrocarburos del gas.
- A continuación, se calcula la densidad molar (d) empleando la ecuación de estado: $P = d R T [1 + B_{mix}d + C_{mix}d^2]$. Si se conociera esta densidad por algún método (ej: cromatografía externa) se saltaría este paso, pero en nuestro caso no se sabe. Así pues, conociendo P y T de los sensores, la constante R y los coeficientes viriales hallados, la única incógnita es la densidad y se resuelve iterativamente con un criterio de convergencia de 1×10^{-6} (dado en norma).
- Finalmente se halla el factor de compresibilidad Z con la ecuación 43a: $Z_f = 1 + B_{mix}d + C_{mix}d^2$, y también otras cantidades como el factor de supercompresibilidad F_{pv} que simplemente es la raíz cuadrada de la relación entre Z_b a condiciones base (hallada en la Figura 32) y el Z a condiciones del fluido. También se puede visualizar la densidad molar calculada.

Figura 31. Diagrama de Flujo para hallar H_{CH} .

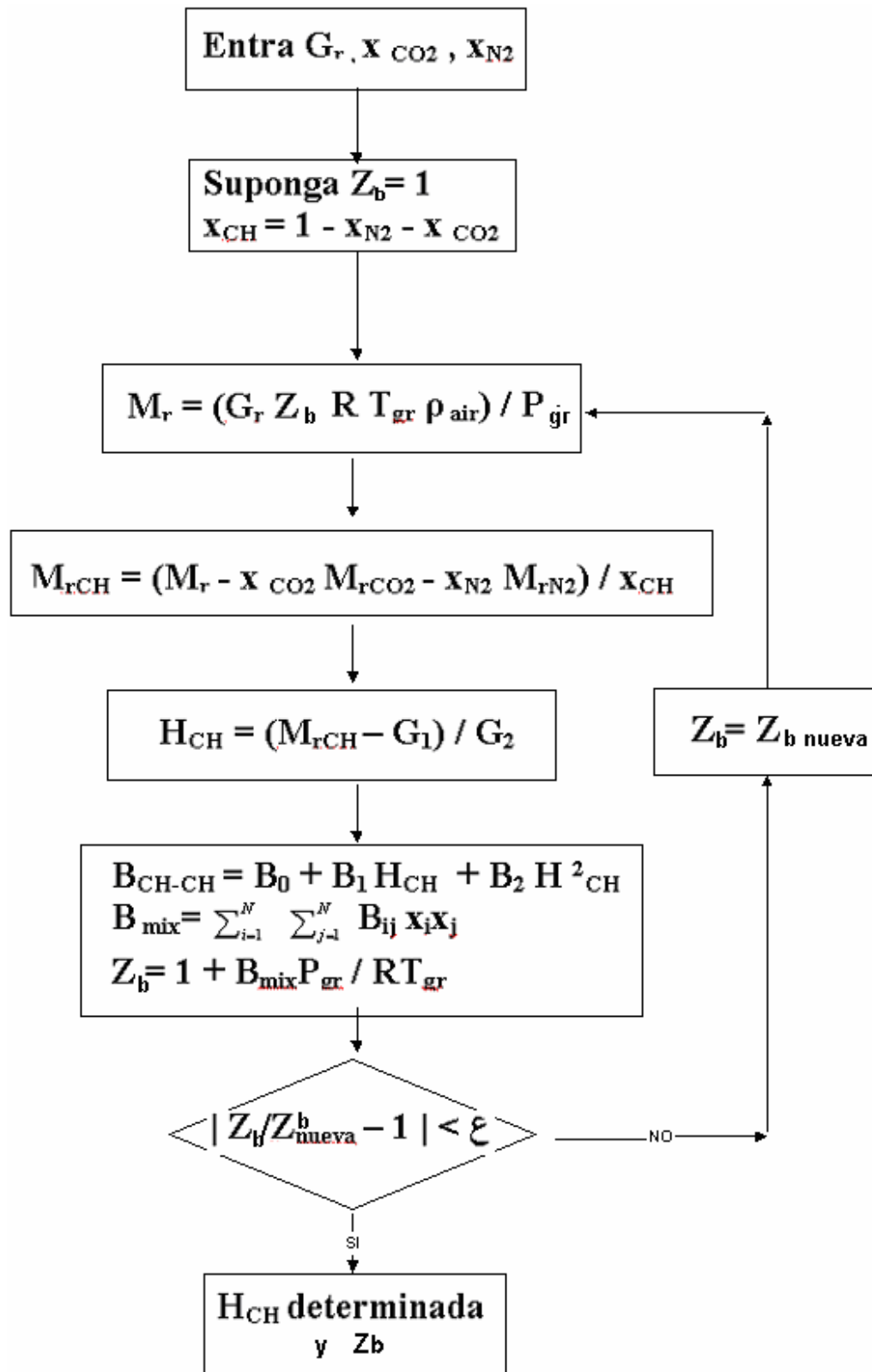
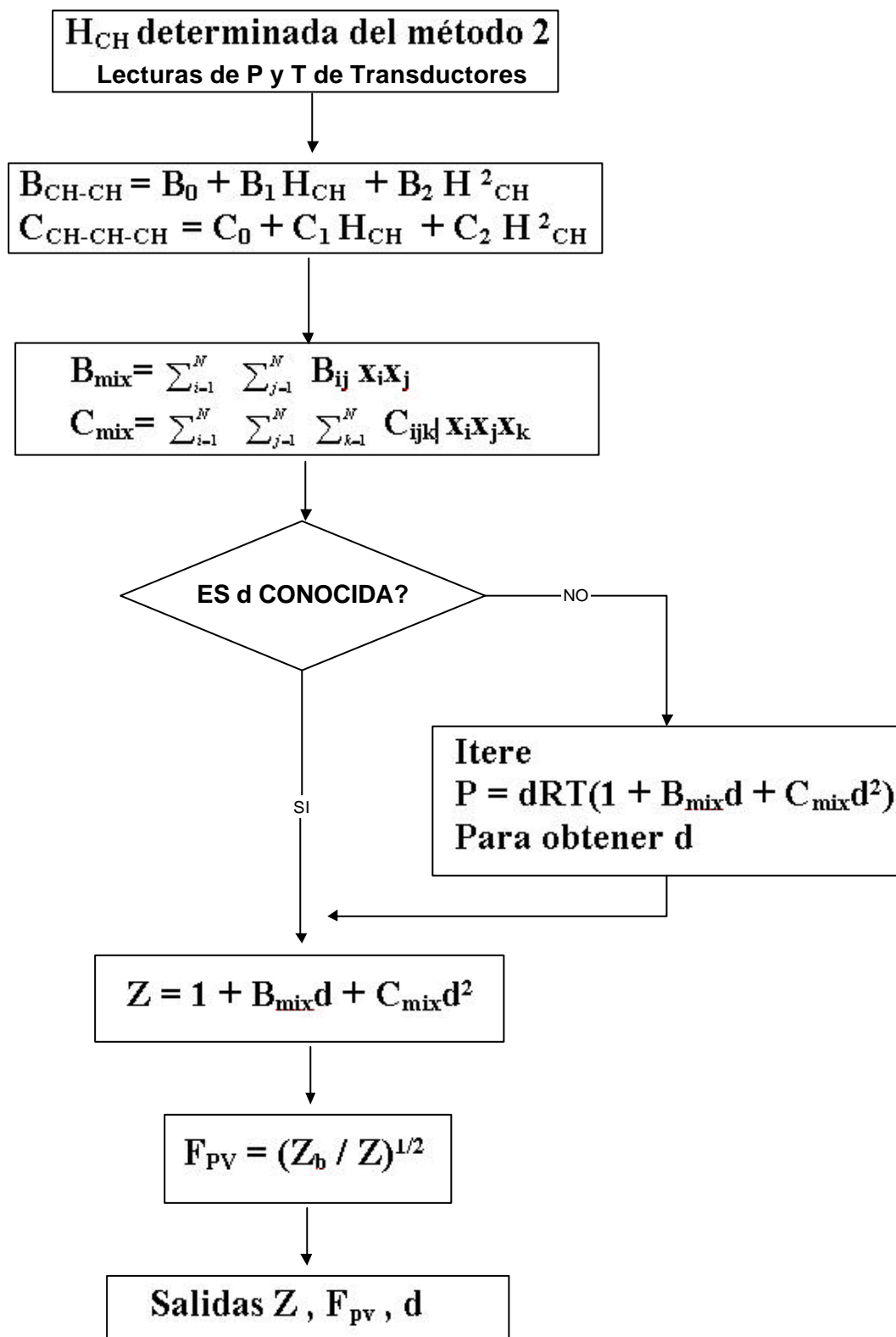


Figura 32. Diagrama de Flujo AGA8 Gross-2.



3.2.1.1. AGA8 en MATLAB. El reporte AGA8 sugiere el desarrollo de varias rutinas para la implementación de los diagramas de flujo mostrados. Es importante aclarar que el código del AGA8 desarrollado en Forth y en Matlab que se presenta en esta y la siguiente sección fue una transcripción del código en *Fortran* dado por la misma norma. El comité AGA8 decidió desarrollar por sí mismo el modelo del código para que las compañías de computadores de flujo o de medición de gas aplicaran el método propuesto por AGA8, pues al principio nadie lo implementaba dada la complejidad de programarlo y usaban algoritmos anteriores como el NX-19.

Es así como la norma AGA8 crea seis rutinas de software en FORTRAN para el desarrollo de los algoritmos y a cada rutina le asocia el cálculo de cantidades que dependen de algún factor en especial.

Los nombres de estas 6 subrutinas son:

1. PARAMGS: Inicialización de constantes.
2. CHARGS: Cálculo de cantidades que dependen de la caracterización del gas.
3. VIRGS: Cálculo de las cantidades que dependen de la temperatura.
4. PGROSS, DGROSS, ZGROSS: Cálculo de las cantidades que dependen de la densidad.

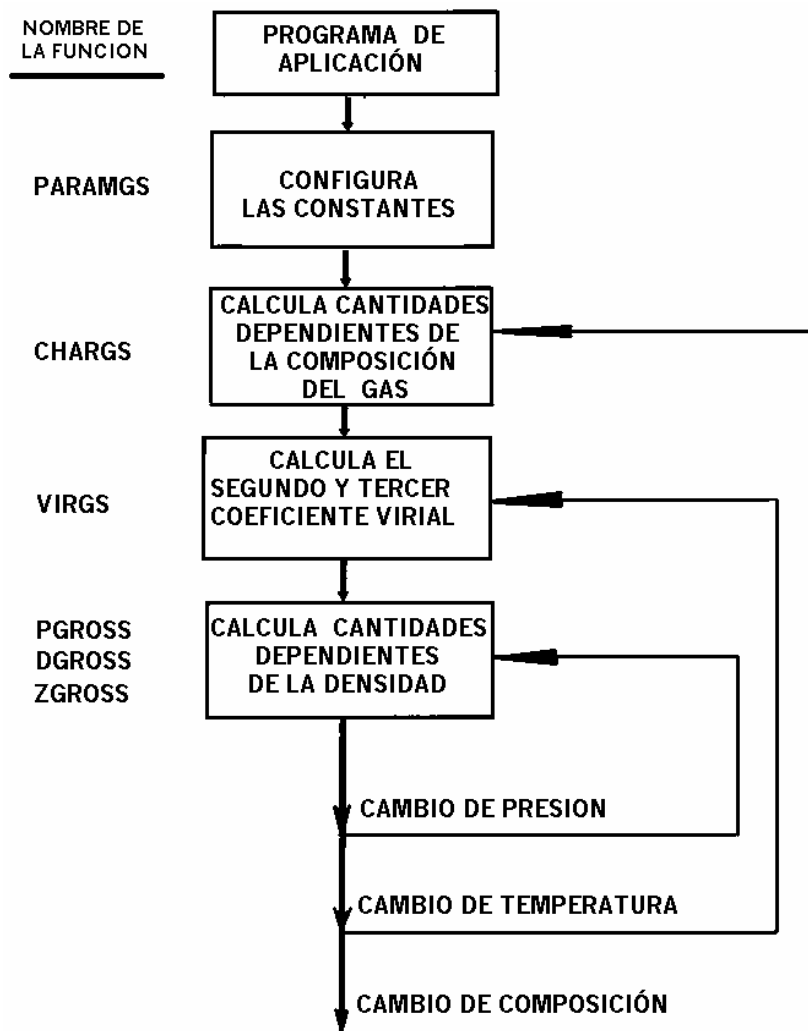
Estas subrutinas se encadenan como se muestra en la Figura 33 para así obtener un código eficiente que sólo calcula lo necesario ante el cambio de algún parámetro de entrada en especial.

Se implementaron en Matlab estas seis subrutinas utilizando los mismos nombres dados por el AGA y con un nombre del programa de aplicación llamado "aga8.m". Igualmente al reporte se tomó la temperatura y presión base como 60°F y 14.73 PSIA respectivamente. En cuanto a la composición del gas y su densidad relativa (o gravedad específica) el reporte propone 5 tipos de gases para probar. Así en Matlab se creó una matriz con la característica de estos 5 gases para que el usuario escoja con cual desea trabajar para hallar el Z. La matriz es la siguiente:

```
%%          Grav.Espc. Tb    Pb      XN2      XCO2      XH2 XCO
XK(:,1) = [0.581078; 60; 14.73; 0.002595; 0.005956; 0; 0; ];
XK(:,2) = [0.608657; 60; 14.73; 0.031284; 0.004676; 0; 0; ];
XK(:,3) = [0.649521; 60; 14.73; 0.010068; 0.014954; 0; 0; ];
XK(:,4) = [0.644869; 60; 14.73; 0.134650; 0.009850; 0; 0; ];
XK(:,5) = [0.686002; 60; 14.73; 0.057021; 0.075851; 0; 0; ];
XK(:,6) = [0.565000; 60; 14.73; 0.015630; 0.000390; 0; 0; ];
```

Las 5 primeras son del AGA y la sexta fila corresponde al gas de Barrancabermeja.

Figura 33. Diagrama de Flujo Rutinas AGA8 Gross-2.



A continuación se detalla lo que desarrolla cada función, indicando los argumentos de entrada y de salida así como las unidades en que trabaja. Si bien, se ha venido trabajado hasta entonces en °F y PSI, algunas rutinas hacen una conversión a °K (kelvin) y Mpa (megapascuales) para procesar sus cálculos ya que algunas constantes del método dadas por tablas vienen con esas unidades.

Función PARAMGS. Inicializa las constantes utilizadas por la ecuación de estado SGERG. Estas constantes incluyen el peso molecular del nitrógeno y de dióxido de carbono, la constante del gas y las constante usadas en la ecuación de estado. Esta rutina PARAMGS es llamada solo cuando empieza la aplicación.

Función CHARGS. Determina el valor calorífico del hidrocarburo equivalente, usando el método dos del apéndice B.2 del AGA 8. Los valores de entrada de la subrutina son: el número del método (2); el calor volumétrico (KJ/dm³), la

densidad relativa y cinco elementos de un arreglo que contiene las fracciones molares de N₂, CO₂, H₂ y CO. CHARGS retorna el factor de compresibilidad base y la densidad base a las condiciones de referencia 60 °F y 14.73 psia. Esta subrutina es llamada de nuevo, solo si hay cambios en la caracterización de la mezcla del gas. En un computador de flujo ya instalado en un sitio generalmente se conservan constantes la composición del gas y demás características propias del lugar (Ej: Patm); por tanto PARAMGS y CHARGS no cambian con el tiempo.

Función VIRGS. Esta subrutina calcula el segundo y tercer coeficiente virial de la mezcla del gas a una temperatura dada. Los coeficientes constantes de la tabla 7 y 8 del AGA8 (pgs. 31 y 32) son almacenados en arreglos (matrices) en PARAMGS y combinados usando reglas definidas por las ecuaciones 30 al 42 del AGA 8 (pg. 30 en adelante). En este tercer bloque de cálculo, la temperatura debe ser especificada. VIRGS es llamada por DGROSS cuando se sucede un cambio en la temperatura del gas.

El último bloque de la Figura 33 determina las cantidades que dependen de la densidad.

Función DGROSS. Determina la densidad molar de una mezcla de gas natural dadas presión y temperatura. La presión debe estar en Mpa y la temperatura debe estar en Kelvin. La densidad es hallada en mol/dm³. DGROSS usa el método Chamber (Referencia 4, apéndice B.8 AGA 8) para hallar la densidad de la mezcla del gas a una temperatura T y presión absoluta P. DGROSS llama la función PGROSS que calcula la presión para una temperatura y densidad específicas. La función PGROSS llama a ZGROSS que calcula el factor de compresibilidad para una temperatura y densidad específicas.

Función PGROSS. PGROSS calcula la presión de la mezcla del gas natural como una función de densidad y temperatura. La densidad debe estar en mol/dm³ y la temperatura debe estar en kelvin. La presión es retornada en Mpa.

Función ZGROSS. ZGROSS calcula el factor de compresibilidad de la mezcla del gas natural como una función de la densidad y la temperatura. La densidad debe estar en mol/dm³ y la temperatura debe estar en kelvin.

En Matlab todas estas subrutinas fueron implementadas como funciones, cada una en un archivo .m independiente. El encabezado de estas funciones es de la siguiente forma para indicar que entra y que retorna de la función.

- **PARAMGS**

No solicita ni arroja datos, solo inicializa variables y coeficientes constantes.

- **function [BMIX,CMIX,ERRNUM]=VIRGS(TGR,X,BCH)**

La entrada es la temperatura de referencia en Kelvin, un vector de composiciones del gas seleccionado X, el coeficiente de interacción binaria BCH. La función retorna

el primer y segundo coeficiente virial y una bandera (ERRNUM) que indica algún error numérico o ausencia de convergencia iterativa.

- **function [ZB,DB,ERRNUM] = CHARGS(METHOD,HV,GR,X,TH,TD,PD,TGR,PGR)**

La entrada consiste en el Método=2 (Gross2), valor calorífico de la mezcla del gas en kJ/dm³ (HV); la gravedad específica (GR); el vector de composición del gas (X); temperatura de referencia para valor calorífico en °K (TH) y para densidad molar en °K (TD), y finalmente la temperatura y presión de referencia para la densidad relativa (TGR y PGR respectivamente).

- **function [DGROSS] = DGROSS(P,T)**

La entrada es un dato de presión en Megapascuales (Mpa) y una temperatura en Kelvin (°K) y retorna la densidad molar en mol/dm³.

- **function [PGROSS] = PGROSS(D,T)**

Calcula la presión en Mpa dada una densidad en mol/dm³ y una temperatura en °K.

- **function [ZGROSS] = ZGROSS(D,T)**

Retorna el coeficiente Z dada una densidad en mol/dm³ y una temperatura en °K.

El código de cada una de estas funciones y el programa principal se encuentra en el Anexo A. Empleando dicho código se obtuvo la Tabla 18 (mostrada al final de la siguiente sección),

3.2.1.2. AGA8 en Forth. Después de comprobar los buenos resultados del código en Matlab, lo cual confirmaba que se habían entendido e implementado correctamente la norma, se procedió a su realización en lenguaje *Forth*. Esta tarea era más dispendiosa que la de Matlab pues era necesario estar muy pendiente del manejo de la pila en todas las operaciones.

A continuación se muestra la comparación de 2 códigos que realizan lo mismo en Matlab y en *Forth*:

1. Matlab: TBDR = 5/9 * (TBASE + 459.67); % Ecuación que convierte °F a °K

1. Forth : TBASE F@@ 459.67E0 F+ 5E0 9E0 F/ F* TBDR F!!

En *Forth*: se saca el valor flotante de TBASE y se pone en la pila con F@@, seguidamente se pone 459.67E0 en la pila, luego se suma en formato flotante con F+ y queda el resultado en la cima de la pila. Luego se pone 5E0 y 9E0 en la pila para dividirse con F/ y seguidamente con F* se multiplica el resultado de esta división con la suma anterior que había quedado en la pila, finalmente se almacena con F!! el resultado de la multiplicación en la variable flotante TBDR y la pila vuelve a quedar vacía.


```

2.Matlab:  F2 = PGROSS (X2,T) - P;  % Llamada a la función PGROSS y resta P
2.Forth:   X2 F@@ T F@@ PGROSS
           PGROS F@@ PRE F@@ F-   F2 F!!

```

En Forth: se sacan los valores de X2 y T y se ponen en la pila, seguidamente se llama la palabra PGROSS (esta palabra cogerá estos dos valores de pila, calcula una presión y la guarda en PGROS), luego con "PGROS F@@" se obtiene el valor de esta variable y se lleva a la pila, luego se pone en pila el valor de PRE para finalmente restarlo con F- y almacenarlo en F2 con F!!.

Para interacción con el usuario se elaboraron 3 palabras llamadas P, T, y GAS, las cuales almacenarán la presión y temperatura de la línea y con un número del 1 al 6 se escogerá un tipo de gas para probar. (Del 1 al 5 son los dados por la norma y el sexto es la composición del gas de Barrancabermeja). La definición de estas palabras es la siguiente:

```

: P
PDATA F!!
14.73E0 PATMO F!! ;

: T
TDATA F!! ;

: GAS
CASE
1 OF
  0.581078E0  SPECG F!!  \ Gravedad Especifica
  0.002595E0  %NI    F!!
  0.005956E0  %CO2   F!!
ENDOF
2 OF
  0.608657E0  SPECG F!!  \ Gravedad Especifica
  0.031284E0  %NI    F!!
  0.004676E0  %CO2   F!!
ENDOF
3 OF
  0.649521E0  SPECG F!!  \ Gravedad Especifica
  0.010068E0  %NI    F!!
  0.014954E0  %CO2   F!!
ENDOF
4 OF
  0.644869E0  SPECG F!!  \ Gravedad Especifica
  0.134650E0  %NI    F!!
  0.009850E0  %CO2   F!!
ENDOF
5 OF
  0.686002E0  SPECG F!!  \ Gravedad Especifica
  0.057021E0  %NI    F!!
  0.075851E0  %CO2   F!!
ENDOF
6 OF
  14.65E0      PATMO F!!  \ Presion Atmo de Barranca
  0.565000E0  SPECG F!!  \ Gravedad Especifica
  0.015630E0  %NI    F!!
  0.000390E0  %CO2   F!!

```

```
ENDOF  
ENDCASE ;
```

Ahora bien, si en la pantalla interactiva de *Forth* se desea hallar el factor de compresibilidad a una presión absoluta de 250 PSIA, una temperatura de 60 °F y con una composición de gas correspondiente al gas número 3 se escribiría así:

```
250E0 P 60E0 T 3 GAS AGA8 [enter]  
| ZETA= 0.95449822 ok
```

y seguidamente el programa arrojará en pantalla el factor de compresibilidad Z.

El código completo de AGA8-Gross2 en *Forth* con sus rutinas se encuentra en el Anexo B.

A continuación se lista la implementación en *Forth* de las dos ecuaciones más largas que son las concernientes al cálculo del segundo y tercer coeficiente virial (Bmix y Cmix) de la Figura 32. En las ecuaciones se cuentan 10 sumas y 23 multiplicaciones flotantes para calcular Bmix y 50 multiplicaciones, 12 sumas y 6 potenciaciones flotantes para el cálculo de Cmix.

El código en *Forth* para el cálculo de los coeficientes viriales B_{mix} , es:

```
X11 F@@ BCH F@@ F*  
X22 F@@ BN2 F@@ F* F+  
X33 F@@ BCO2 F@@ F* F+  
X44 F@@ BH2 F@@ F* F+  
X55 F@@ BCO F@@ F* F+  
  
XIN2 F@@ XIN1 F@@ F*  
B12 F@@ F* 2E0 F* F+  
  
XIN3 F@@ XIN1 F@@ F*  
B13 F@@ F* 2E0 F* F+  
  
XIN4 F@@ XIN1 F@@ F*  
B14 F@@ F* 2E0 F* F+  
  
XIN5 F@@ XIN1 F@@ F*  
B15 F@@ F* 2E0 F* F+  
  
XIN3 F@@ XIN2 F@@ F*  
B23 F@@ F* 2E0 F* F+  
  
XIN4 F@@ XIN2 F@@ F*  
B24 F@@ F* 2E0 F* F+  
  
BMIX F!!
```

El código en *Forth* para el cálculo de los coeficientes viriales C_{mix} es:

```

C11 F@@ X11 F@@ F* XIN1 F@@ F*
C22 F@@ X22 F@@ F* XIN2 F@@ F* F+
C33 F@@ X33 F@@ F* XIN3 F@@ F* F+
C44 F@@ X44 F@@ F* XIN4 F@@ F* F+
C23 F@@ X22 F@@ F* XIN3 F@@ F* F+
C32 F@@ X33 F@@ F* XIN2 F@@ F* F+
C15 F@@ X11 F@@ F* XIN5 F@@ F* F+

C11 F@@ C11 F@@ F* C22 F@@ F* FFF F@@ F**
XIN2 F@@ F* X11 F@@ F* EEE F@@ F* 3E0 F* F+

C11 F@@ C22 F@@ F* C22 F@@ F* FFF F@@ F**
XIN1 F@@ F* X22 F@@ F* EEE F@@ F* 3E0 F* F+

C11 F@@ C11 F@@ F* C33 F@@ F* FFF F@@ F**
XIN3 F@@ F* X11 F@@ F* 0.92E0 F* 3E0 F* F+

C11 F@@ C33 F@@ F* C33 F@@ F* FFF F@@ F**
XIN1 F@@ F* X33 F@@ F* 0.92E0 F* 3E0 F* F+

C11 F@@ C11 F@@ F* C44 F@@ F* FFF F@@ F**
XIN4 F@@ F* X11 F@@ F* 1.2E0 F* 3E0 F* F+

C11 F@@ C22 F@@ F* C33 F@@ F* FFF F@@ F**
XIN1 F@@ F* XIN2 F@@ F* XIN3 F@@ F* 1.1E0 F* 6E0 F* F+

CMIX F!!

```

De esta forma se obtuvo la Tabla 18, la cual para una presión y temperatura muestra el factor de compresibilidad obtenido tanto en Matlab como en Forth y lo compara con el dado por el reporte AGA8 (pg.199 del reporte AGA8).

Tabla 18. Comparación de Resultados del AGA8.

Temp. (°F)	Presión (psia)	Tipo Gas	Factor de Compresibilidad		
			Reporte AGA8	AGA8 en Matlab	AGA8 en Forth
32	100	1	0.982375	0.98237479992543	0.98237480
32	400	1	0.929268	0.92926794978275	0.92926795
32	800	2	0.853785	0.85378533463339	0.85378534
32	1200	2	0.787484	0.78748355351708	0.78748355
50	200	3	0.961069	0.96106897147342	0.96106897
50	800	3	0.843550	0.84355009686887	0.84355010
100	14.73	4	0.998557	0.99855726413009	0.99855726
100	600	4	0.944627	0.94462660878975	0.94462661
130	400	5	0.961865	0.96186507985394	0.96186508
130	800	5	0.927215	0.92721504364583	0.92721505

En la Tabla 18 se observa que en ambos casos al redondear los resultados a 6 dígitos se obtiene un error de 0% respecto al resultado dado por el reporte AGA8, verificando la validez de los códigos implementados.

3.2.2 Norma AGA-7. Las ecuaciones dadas por AGA7, dadas en sección 1.2.1 del presente texto, se concatenan en una sola, cuya salida es la velocidad de flujo a condiciones base o contractuales. Al multiplicar esta velocidad de flujo por un intervalo de tiempo dado se obtiene el volumen que gas que ha pasado durante ese lapso de tiempo. De esta forma se obtiene la ecuación:

$$Q_b = K * Q_f * \frac{P_f + P_a}{P_b} * \frac{T_b + 459.67}{T_f + 459.67} * \frac{Z_b}{Z_f} \quad (\text{ecuación 48})$$

Donde:

Q_b = Velocidad de Flujo de Gas a condiciones base (pies cúbico por hora, CFH)

K = Factor de calibración. Número dado por la hoja de datos de la turbina.

Q_f = Velocidad de Flujo de Gas a condiciones del fluido, dada por la curva de la turbina. A cada frecuencia de pulsos le corresponde una velocidad del fluido.

P_f = Presión del fluido en PSIG. Dada por el transductor de presión.

P_a = Presión atmosférica del lugar en PSIA.

P_b = Presión base o contractual. La suministra la empresa vendedora.

T_f = Temperatura del fluido en °F. Dada por el RTD.

T_b = Temperatura base o contractual. La suministra la empresa vendedora.

Z_b = Factor de compresibilidad a condiciones base. Hallado por el AGA8.

Z_f = Factor de compresibilidad a condiciones del fluido. Hallado por el AGA8

El último término también es igual a $F_{PV}^2 = Z_b/Z_f$.

3.2.2.1. AGA7 en Matlab. Esta ecuación se implementa sencillamente en Matlab como sigue:

```
% VELOCIDAD DEL GAS EN PIES CUB/HOR
QB= K*(QF/1000)*(PDATA/PB)*((TB+459.67)/(TDATA+459.67))*(ZB/Z)
```

Si se multiplica este valor por el número de horas ($E_j=24h$) y dividido en 1 millón, se obtendrá el volumen de gas que pasó en un día en Megapiés cúbicos.

```
HORAS = 24
VOL = QB * HORAS / 1e6 % VOLUMEN EN MEGA PIES CUBICOS MCF
```

3.2.2.2. AGA7 en Forth. La palabra en *Forth*, implementada para hallar la rata de flujo corregida a condiciones contractuales recibe el nombre de AGA7. Esta palabra se ejecuta después de haber hallado el factor de compresibilidad con AGA8. Su implementación es sencilla y solo requiere cuidado de escritura. Es de notar que en *Forth* todo número en flotante deberá escribirse con exponente, por ejemplo el número 4 para operarlo como flotante se escribe 4.0E0. Igualmente las

palabras para asignar, traer de la memoria y operaciones generales son las mismas que para números enteros excepto que van precedidas de la letra F.

La palabra AGA7 implementada es la siguiente:

```
: AGA7
TBASE F@@ 459.67E0 F+      \ Suma Tb con 459.67 dejando resultado en pila
FPV F@@ F* FPV F@@ F*      \ Lo multiplica dos veces por Fpv y deja en pila
TEMPR F@@ 459.67E0 F+      F/ \ Suma la Tf con 459.67 ,divide la pila en esto
PREPR F@@ PATMO F@@ F+      F* \ Suma Pf + Patm y multiplica con lo anterior
PBASE F@@                  F/ \ Divide el resultado anterior entre Pb
QFNC F@@ F* KFL F@@ F*      \ Multiplica por Qf y luego por la constante K
QFC F!!                    \ Finalmente almacena resultado en QFC = Qb
;
```

3.3 DESARROLLO DEL FIRMWARE

El programa o *firmware* que controla todo el computador de flujo es un conjunto de código que incorpora muchas funciones (o palabras en *Forth*), todas estas funciones se pueden agrupar en las siguientes:

- Monitorear datos de presión y temperatura
- Recibir y procesar los pulsos de la turbina
- Efectuar los cálculos AGA8 y AGA7
- Almacenar y promediar datos de la hora y el día
- Mantener una interfase visual con el usuario
- Proporcionar la posibilidad de calibración

Para efectuar todo esto de una forma “simultanea” fue necesario utilizar una capacidad de la tarjeta llamada multitarea periódica por interrupción de software. De esta forma se dividió el programa en dos grandes tareas: una de fondo y otra de primer plano.

La tarea de fondo (*background*) mantendrá la interfase teclado -display y actualizará cualquier cambio de las constantes y una rutina de calibración, su palabra en Forth asociada fue *ALL*. La tarea de primer plano (*foreground*) realizará la adquisición continua de señales de presión y temperatura así como el promedio y almacenamiento de las mismas, su palabra en Forth asociada fue *INT*. Esta tarea es llamada cada 50ms por una interrupción interna del TDS2020 asociada con el trabajo de multitarea.

Existe una tercera rutina, llamada en Forth *IRQ1*, que no está en ninguna de las dos tareas planteadas. Esta tarea corresponde a una rutina de servicio de interrupción de hardware, la cual se invoca exclusivamente cuando llegue un pulso de la turbina. La línea digital de entrada de pulsos de la turbina es conectada a la interrupción de hardware (*IRQ1*) la cual tiene asociada esta rutina para calcular el volumen de gas correspondiente al pulso que ha generado la turbina y seguidamente corregirlo a condiciones contractuales efectuando AGA8 y AGA7.

La figura 36 muestra el diagrama de flujo del sistema completo. En esta figura se observan dos flujos de código, el de la izquierda corresponde a la tarea de fondo (*background*) y el de la derecha a la tarea de primer plano (*foreground*). Entre las dos tareas se observa la rutina de servicio de interrupción la cual es independiente de ellas y sólo se ejecuta ante la entrada de un pulso de la turbina.

En las siguientes secciones se muestran los detalles más importantes de las dos tareas y de la rutina de servicio a la interrupción. Pero antes de entrar en las dos tareas, como se observa en la Figura 34, existe una inicialización del sistema la cual fue contenida en la palabra INITIALISE.

La palabra que engloba todo el firmware se llama WORK, la cual contiene a MAIN (programa principal donde se llama a INITIALISE) y un CATCH de errores. Este "CATCH ERROR" es una sentencia predefinida en Forth para la TDS2020 que atrapa cualquier error que llegue extemporáneamente a suceder, ya sea matemático, algún fallo en punteros, desbordamiento, etc. y obligaría al programa a retornar a inicializarse automáticamente garantizando la continuidad del trabajo.

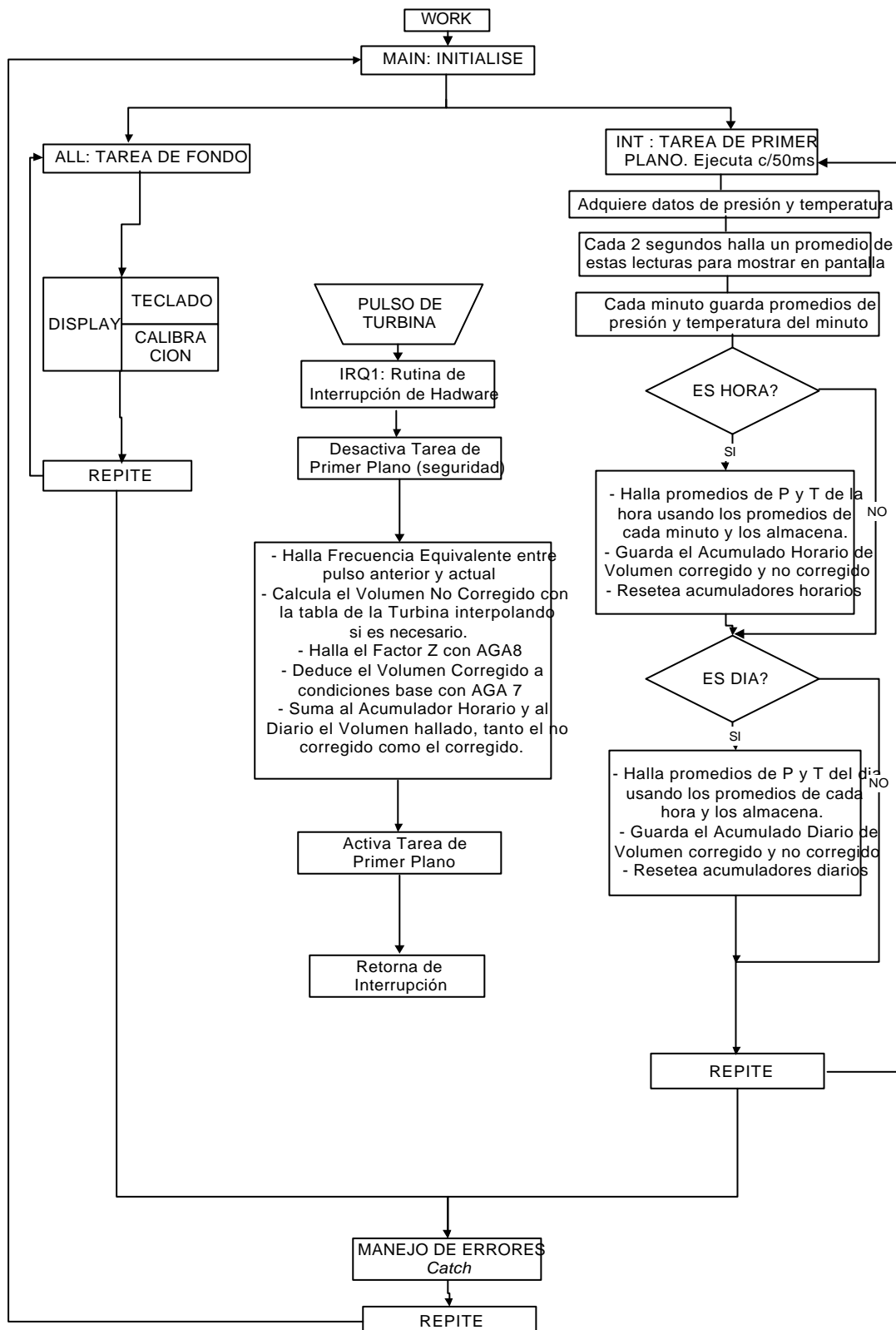
La definición de las 3 palabras que dan comienzo a todo el sistema WORK, MAIN e INITIALISE es corta y se muestra a continuación:

```
: WORK          \ Palabra que engloba todo el firmware
  BEGIN [']      \ Comienzo del ciclo global
    MAIN         \ Ejecuta MAIN y si llega a haber un error casual...
    CATCH ERROR  \ CATCH lo atrapa y vuelve a ejecutarse MAIN
  AGAIN ;

: MAIN
  18.10.04 NOW   09.00.00 HRS W! W@ \ Inicializa la fecha y hora(opcional)
  INITIALISE    \ Inicializa el sistema
  ALL ;        \ Llama la palabra ALL que tiene la tarea de fondo o background.

: INITIALISE ( -- )
  DIS          \ Deshabilita cualquier interrup. que llegara durante INITIALISE
  $FFB0 6 ZERO \ Deshabilita la multitarea hasta salir de esta palabra
  ADS7825-INIT \ Inicializa los registros del conversor A/D
  CEROS        \ Inicializa en cero las variables
  INIT-IRQ     \ Habilita la interrupción de hardware para recibir pulsos
  EIS          \ Vuelve a habilitar todas las interrupciones
  START-LOG ;  \ Habilita la tarea de primer plano (empieza la multitarea)
```

Figura 34. Diagrama de Flujo del *Firmware* del computador de flujo.



3.3.1. Tarea de Fondo. La tarea de fondo está contenida en la palabra ALL la cual tiene a su cargo manejar el teclado y el display. De esta forma mientras no haya alguna tecla oprimida ALL sólo refrescará el display continuamente.

```
: ALL          \ Tarea de Fondo (Background)
BEGIN          \ Comienzo del ciclo de la tarea de fondo
INILCD        \ Inicializa el LCD de 20x4
MEN REQU      \ Con MEN se muestra el menú en display y REQU solicita tecla
CASE          \ De acuerdo a la tecla oprimida se bifurca según el caso
  "1 OF WIPE L1- MENDI0 1 MODO ! 1000 MS MODO1 ENDOF \ Entra al Modo1
  "2 OF WIPE L2- MENAL1 2 MODO ! MODO2 ENDOF        \ Entra al Modo2
  "3 OF WIPE 3 MODO ! MODO3 ENDOF                    \ Entra al Modo3
  "4 OF WIPE 4 MODO ! MODO4 ENDOF                    \ Entra al Modo4
  "0 OF WIPE START ENDOF                             \ Da control al PC
ENDCASE
WIPE          \ Limpia pantalla
KEY? UNTIL    \ Retorna al BEGIN si no se oprimió alguna tecla
;
```

El menú principal mostrado en el display es llamado por MEN y es el siguiente:

```
: MEN1 <LCD ." ==== CIG-ECOGAS ==== LCD> ; \ Primer reglón del display
: MEN2 <LCD ." === ECOFLOW2020T === LCD> ; \ Segundo reglón del display
: MEN3 <LCD ." 1-DISPLAY 2-ALARMAS" LCD> ; \ Tercer reglón del display
: MEN4 <LCD ." 3-CALIBRA 4-CONFIGU" LCD> ; \ Cuarto reglón del display
: MEN WIPE MEN1 MEN2 MEN3 MEN4 ; \ MEN limpia LCD y muestra los 4 reglones.
```

Si se oprime 1, el LCD mostrará las variables importantes cuales son temperatura, presión, velocidad de flujo y volumen acumulado, cada una en un reglón.

Al oprimir 2, el LCD entrará a solicitar los límites máximos de presión y temperatura para posibles alarmas.

Si se oprime 3, se entra al modo calibración, donde el computador de flujo hallará las constantes de calibración para corregir las lecturas de presión y temperatura con base a un patrón que se le ponga como un peso muerto o un baño térmico.

Al oprimir 4, se ingresa al modo configuración, en el cual se pueden introducir las constantes propias de trabajo como presión atmosférica, composición del gas, temperatura y presión base o contractuales, etc.

Las opciones 1, 2 y 4 consisten en puro manejo de variables bien sea para mostrarlas en el display o para asignarles un valor desde el teclado. Mientras que la opción 3 de calibración si tiene un tratamiento matemático que se explicará a continuación. Al oprimir 3 se invoca un menú (*MODO3*) que pregunta cuál transductor va a calibrar y de esta forma se selecciona del 1 al 4 según corresponda:

```
: MODO3
BEGIN
INKEY ?DUP 0> IF
TRANSLATE CASE
```



```

"1 OF WIPE L1- <LCD ." * PRESION Braz1 *" LCD> 500 MS CALI-P1 END OF
"2 OF WIPE L1- <LCD ." * TEMPERA Braz1 *" LCD> 500 MS CALI-T1 END OF
"3 OF WIPE L1- <LCD ." * PRESION Braz2 *" LCD> 500 MS CALI-P2 END OF
"4 OF WIPE L1- <LCD ." * TEMPERA Braz2 *" LCD> 500 MS CALI-T2 END OF
ENDCASE
THEN
50 MS
INKEY TRANSLATE "C = UNTIL
;

```

Dada la linealidad de los transductores adquiridos, amplificadores de instrumentación y conversor A/D se trabaja con una calibración lineal, es decir, una recta que ajustará los valores digitalizados del conversor a los valores dados por una referencia externa como un peso muerto o un baño térmico.

El rango de presión de calibración es de 0-300 psig y el de temperatura es de 32-135°F (0°C-57.22°C). El procedimiento seguido por la mayoría de computadores de flujo y por el desarrollado en este proyecto, consiste en calibrar el cero y el máximo de estos valores y el software se encarga de calcular las constantes (M y B) de la recta que se ajusta a estos dos puntos ($Y = MX + B$).

Por ejemplo, para calibrar el cero del transductor de presión se pone en el peso muerto un valor de 0 psig. El computador de flujo mostrará quizás algún valor diferente de cero (*un offset*), el negativo de este valor será la constante B de la ecuación de la recta de presión. Ahora para calibrar el máximo, se pone el peso muerto a generar 300 psig y se le dice al computador de flujo que está calibrando el máximo (*Full scale*). De esta forma, ya teniendo la constante B , el valor leído por el computador (X) y el valor de referencia (Y) se halla la pendiente de la recta con la siguiente ecuación 49:

$$M = (Y - B) / X \quad (\text{ecuación 49})$$

La forma de indicarle al computador de flujo que se está calibrando el cero o el máximo es oprimiendo la tecla correspondiente según como muestra el *display*. Con la palabra (CA-ALL) se muestra en pantalla la lectura actual y seguidamente se solicita que se oprima cero (0) si se trata de la calibración del cero, o se oprima nueve (9) si se trata de la calibración del máximo (*Full*).

```

: CA-ALL
WIPE L1- <LCD ." LEE= " LCD>
L2- <LCD ." PRE:0-300 TEM:32-135" LCD>
L3- <LCD ." Cer[0]-Full[9]-OK[1]" LCD>
L4- <LCD ." Opcion: " LCD>
500 MS ;

```

Al volver a entrar al menú calibración el usuario podrá constatar que la lectura mostrada en el display coincida con la referencia que haya puesto, y de esa forma oprimiría uno (1=OK) para indicar que está bien y así podrá salir de este menú sin necesidad de volver a calibrar.

3.3.2. Tarea de Primer Plano. La tarea de primer plano, está configurada en la multitarea para que se realice cada 50ms. De esta manera, el sistema internamente genera una interacción de software periódica de 50ms, la cual provoca un salto de la tarea de fondo a esta de primer plano y cuando acaba retorna a la tarea de fondo.

La palabra que gobierna esta tarea se llama INT y tiene a su cargo la toma de las señales de presión y temperatura, el cálculo de promedios y el almacenamiento de dichos promedios en la memoria cuando lleguen instantes especiales como una nueva hora o un nuevo día.

```
: INT \ Tarea de Primer plano, ocurre cada 50ms
STOP-LOG
LOG
#INT @@ 1+ DUP #INT !! \ Adquiere señales de presión y temperatura
                        \ Incrementa en 1 #INT, el cual es un contador
                        \ que sirve para saber cuantos periodos de 50ms van acumulados

\ HALLA UN PROMEDIO DE CADA 2 SEGUNDOS DE P y T
40 MOD 0= IF          \ Cada 2 segs entra a este ciclo, 2s = 40 veces 50ms
  PROM2              \ hace el PROMED de P y T de 2 segundos
  FIRST @@ 1 = IF    \ Si NO es la primer vez que se enciende el CF
  AVERAGE THEN      \ entonces hace el promedio acumulativo
  FIRST @@ 0= IF     \ Si se acaba de encender el CF
    PREPR F@@ PREAV F!! \ Establece condiciones iniciales
    TEMPR F@@ TEMAV F!! \ que consiste en hacer el promedio acumulativo
                        \ igual al promedio de los 2 segundos
    1 FIRST !!       \ quita bandera de primera vez
    INIT-IRQ         \ habilita interrupción de hardware (pulsos turbina)
  THEN \
THEN

\ HALLA UN PROMEDIO DE CADA 60 SEGUNDOS DE P y T

#INT @@ 1200 MOD 0= IF \ Cada 60 segundos guardo la PR y TEM en vector
PREAV F@@ PRES-W #MIN @@ FLOATSD D+ 2DUP POINT-P 2!! EF!
TEMAV F@@ TEMP-W #MIN @@ FLOATSD D+ 2DUP POINT-T 2!! EF!
#MIN @@ 1+ #MIN !!
0 #INT !!
THEN

MINS @ 60 MOD        \ SI ES CAMBIO DE HORA, HALLA EL PROMEDIO DE LA HORA
0= IF
  YAH @@ 0= IF
    1 YAH !! PROMED-HR
  THEN
ELSE 0 YAH !!
  0. #IRQ1-H 2!!
THEN

MINS @ 0= IF         \ SI ES CAMBIO DE DIA, HALLA EL PROMEDIO DEL DIA
  YAD @@ 0= IF
    1 YAD !! PROMED-DY THEN
  ELSE 0 YAD !!
```

```

0. #IRQ1-D 2!!
THEN

```

```

EDELAY @@ LATER RETURN; \ estas 2 líneas configuran el tiempo periódico
-27 +ORIGIN ASSIGN INT \ de esta palabra y su asociación con multitarea

```

De aquí la palabra más importante es LOG la cual captura las señales de presión y temperatura y las va sumando en un acumulador llamado SLOGP y SLOGT; cada 2 segundos se halla el promedio de estas señales y se borra el acumulador. En este punto si se pasa el resultado de la conversión (un número entero entre 0-65536) a unidades de presión (psig) y temperatura (°F) usando las constantes de calibración MP1 y BP1 para presión y MT1 y BT1 para temperatura. De esta forma las variables finales PREPR y TEMPR, contienen el dato actualizado de presión y temperatura para refrescar en el display cada 2s.

```

: LOG \ Toma y acumula muestras de presión y temperatura
SLOGP 2@@ 1 A-D16 M+ SLOGP 2!! \ Lee Presión del canal 1 del ADC
SLOGT 2@@ 0 A-D16 M+ SLOGT 2!! \ Lee Temperatura del canal 0 del ADC
#LOG2 @@ 1+ #LOG2 !! ;

: PROM2 \ Promedia cada 2s las muestras acumuladas y convierte a unidades
SLOGP 2@@ D>F #LOG2 @@ S>F F/ MP1 F@@ F* BP1 F@@ F+ PREPR F!!
SLOGT 2@@ D>F #LOG2 @@ S>F F/ MT1 F@@ F* BT1 F@@ F+ TEMPR F!!
0 #LOG2 !! 0. SLOGP 2!! 0. SLOGT 2!! ;

```

3.3.3. Rutina de servicio de interrupción. Esta rutina se ejecuta solo cuando entra un pulso de la turbina. Es activada por el flanco descendente del pulso en la entrada IRQ1 de la tarjeta (pin C27) y su palabra asociada fue llamada IRQ1. Esta palabra tendrá como principal cometido hallar y acumular el volumen de gas que va pasando y a su vez llamar a las rutinas AGA7 y AGA8 para corregir este volumen a condiciones contractuales.

Es importante aclarar cómo se halla la frecuencia entre dos pulsos contiguos y la velocidad de flujo correspondiente a esta frecuencia. Siendo *Tnew* el tiempo en el cual sucede un pulso y *Told* el tiempo en el cual ocurrió el pulso anterior, la frecuencia asociada es $f = 1 / (T_{new} - T_{old})$. Si los tiempos fueran dados en segundos la frecuencia estaría directamente dada en Hz.


En el TDS2020 hay una variable interna que automáticamente se incrementa en 1 cada 53.33ms, este tiempo es llamado 1 *Tick* = 53.333ms. Así pues, si se resta el valor de esta variable en el momento actual y en el instante del pulso anterior se obtendría el número de ticks transcurrido entre los dos pulsos y así la frecuencia sería:

$$f = 1 / [(T_{new} - T_{old}) * 0.05333] \text{ Hz. } \quad (\text{Siendo } T_{new} \text{ y } T_{old} \text{ en número de ticks})$$

En Forth durante un pulso previo se guarda el valor actual del número de *ticks* (dado por la palabra de Forth *@TIME*) en la variable *Told*, y tan pronto ocurre el nuevo pulso guardo el número de ticks actual en la variable *Tnew*. Luego se efectúa la resta y el resultado se pasa a flotante (*D>F*) para sacarle el inverso y se

multiplica por 18.75 que es el inverso de 53.333ms y así se obtiene finalmente la frecuencia en Hz correspondiente a ese pulso entrante. En *Forth* se escribe así:

```
@TIME TNEW 2!! \ Obtiene los ticks para el pulso actual
1E0 TNEW 2@@ Told 2@@ D- D>F F/ \ Realiza la resta de los ticks e invierte
18.75E0 F* FREQ F!! \ Multiplica por (53.33ms)-1 y guarda en FREQ
```

ELSTER

Turbine Meter Calibration Data

Serial No.	80036543		J1/J2	32/41
Cartridge No.			Working Pressure	250 PSIG
Meter Type	G 250		Qmax	15000 ACFH
Pipe Diam.	80	mm	1 Pulse	100 ACFH
Housing ANSI	150	RF	Hz at Qmax	0,041667

Calibration Pressure	250	PSIG
----------------------	-----	------

% of Meter Qmax	5	7	10	15	20	30	50	75	100
Hz	0,002083	0,002917	0,004167	0,006250	0,008333	0,012500	0,020833	0,031250	0,041667
Meter Error %	-0,1	-0,3	-0,3	-0,3	-0,3	-0,3	-0,1	0,1	0,2
K-Factor									
Meter Accuracy	99,9	99,7	99,7	99,7	99,7	99,7	99,9	100,1	100,2
Coefficient									
100/ K-Factor	1,001001	1,003009	1,003009	1,003009	1,003009	1,003009	1,001001	0,999001	0,998004

Date / Signature

Figura 35. Tabla de calibración de la turbina empleada.

Seguidamente, la frecuencia instantánea hallada se ubica en la tabla de la turbina (Figura 35) en la segunda fila, de allí se extrae la velocidad de flujo equivalente (primera fila). Y esta velocidad se calibra por el coeficiente dado en la última fila.

De la tabla de la turbina se extraen los siguientes datos importantes:

Velocidad de Flujo Máxima (Qmax) = 15000 ACFH (pies cúbicos por hora)

Frecuencia Máxima = 0.041667 Hz

Presión de Trabajo = 250 PSIG

Diámetro del Tubo = 80 mm

Ejemplo: Suponiendo que la frecuencia calculada por la rutina mostrada resulta de 0.0125 Hz, se ubica en la tabla y se ve que corresponde a una porcentaje de velocidad máxima (% of Meter Qmax) de 30%. Como Qmax= 15000 ACFH, entonces el 30% de esto será 4500 ACFH. Luego este valor se multiplica por el coeficiente de calibración (Coefficient 100/K-Factor) que en este caso es 1.003009 y resulta finalmente que la frecuencia de 0.0125 Hz nos indica que el gas está pasando con una velocidad de 4513.5405 ACFH (actuales pies cúbicos por hora).

Si se multiplica este valor por el período de tiempo $(0.0125\text{Hz})^{-1} = 80\text{s} = 1/45 \text{ hora}$ se obtiene el volumen de gas que atravesó la turbina en el tiempo comprendido entre los dos pulsos. En este caso resulta de $4513.5405 \times (1/45) = 100.3$ pies cúbicos de gas.

Cuando la frecuencia calculada no resulta igual a ninguna de las frecuencias dadas por la tabla, se debe interpolar linealmente entre las dos columnas que la contengan. En Forth esta interpolación la realiza la palabra INTERP-C.

A continuación se muestra el desarrollo de la palabra IRQ1 en la cual están contenidas todas las funciones realizadas cada vez que llega un pulso de la turbina, esta palabra no finaliza con punto y coma (;) sino con (RETURN;) pues se trata del final de una palabra que sirve de rutina de servicio a una interrupción de hardware.

```
: IRQ1 \ Esta interrupción se activa con cada flanco de caída en el pin C27

@TIME TNEW 2!! \ Saca el # de TICKS que van en el día y lo guarda
STOP-LOG      \ Deja de solicitar muestras hasta que haga este cálculo

\ Calcula el tiempo transcurrido entre el pulso anterior y el actual...
\ y al sacarle el invertido se halla la frecuencia correspondiente

1E0 TNEW 2@@ Told 2@@ D- D>F F/
18.75E0 F* FREQ F!!

\ Reemplaza el numero de TICKS actual TNEW en Told para el próximo pulso.

TNEW 2@@ Told 2!! \ Told será el punto de referencia para el siguiente.

\ Según la tabla de la turbina se halla el %
\ porcentaje de velocidad de flujo equivalente a la frecuencia hallada

360E3 1PULSE F@@ F* RQMAX F@@ F/ FREQ F@@ F* %RQFI F!!

\ Con este % se interpola en la tabla de calibración de la turbina
\ y se halla la velocidad de flujo calibrada y el volumen de gas asociado

INTERP-C

\ Se suma este volumen en el acumulado horario y el acumulado diario

QFNC-HR F@@ QFNC F@@ F+ QFNC-HR F!!
CDAY-NC F@@ QFNC F@@ F+ CDAY-NC F!!

\ Corre el AGA8 y AGA7 para corregir resultados a condiciones contractuales

AGA8-G2
AGA7

\ Se suman a los acumulados horarios y diarios de valores de volumen
\ corregidos a condiciones contractuales

QFC-HR F@@ QFC F@@ F+ QFC-HR F!! \ acumulador horario
```

```

CDAY-C F@@ QFC F@@ F+      CDAY-C F!!      \ acumulador diario

#IRQ1 2@@ 1. D+ #IRQ1 2!!    \ incrementa el contador de número de pulsos
START-LOG                   \ habilita la toma de señales de P-T
RETURN;                      \ regreso de rutina de interrupción
-60 +ORIGIN ASSIGN IRQ1     \ Asigna esta palabra a la interrupción IRQ1.

```

Los acumulados de hora y de día no se van a desbordar pues fueron definidos como un flotante, los cuales tienen capacidad de manejar hasta un valor de 3×10^{12} y los acumulados están en el rango de 1×10^4 por mucho, ya que se dan en unidades escaladas como megapies cúbicos y además al final de cada hora y día son vaciados para empezar un nuevo conteo.

3.3.4. Especificaciones del *firmware* realizado. En el apéndice F se exponen todas las rutinas de Forth que conforman el *firmware* del computador de flujo. De los 29Kb disponibles en el TDS2020 para el desarrollo de aplicaciones se emplearon 26Kb en todo el desarrollo y están distribuidos de la siguiente forma:

Rutinas de punto flotante	8.80 KB
Rutinas de AGA8	7.20 KB
Rutinas de manejo de teclado y LCD	5.53 KB
Rutinas de AGA7	0.46 KB
Rutinas de manejo de memoria externa	0.32 KB
Rutinas del manejo del conversor A/D	0.31 KB
Programa Principal y demás subrutinas	3.38 KB
TOTAL	26.0 KB

Las rutinas de manejo de memoria externa hacen referencia a una librería que tiene el sistema de desarrollo para manejar la memoria RAM que se le ponga al sistema, la cual en este caso fue de 512KB.

Los tiempos de ejecución de las operaciones en flotante de 48 bits son:

Suma Flotante F+	0.49ms
Resta Flotante F-	0.45ms
Multiplicación Flotante F*	0.60ms
División Flotante F/	5.40ms

Los tiempos de ejecución de operaciones en Forth con números enteros de 16 bits como contadores, punteros, etc son:

Suma +	4.1 μ s	
Multiplicación UM*	7.3 μ s	(multiplicación sin signo)
División UM/MOD	6.7 μ s	(deja cociente y residuo en pila)

El tiempo de *scan* del programa total no es constante, pues es un sistema multitarea y a la vez con rutina de interrupción aleatoria, también depende si hay corte de hora o de día, pues en esos instantes se realizan tareas de promedios y almacenamiento. Sin embargo, en forma general se promedia un tiempo de 0.05s

cuando nada especial ocurre, 0.25s cuando ocurren cortes de hora o de día, y 2.05s cuando llega un pulso de la turbina y se ejecuta AGA8, AGA7, interpolación y toda la rutina de interrupción. Correspondiendo 1.98s a AGA8, y lo demás a AGA7 con interpolación y almacenamiento del volumen hallado.

Hablando del entorno eléctrico, los sistemas donde se instalan los computadores de flujo siempre están respaldados por UPS´s, banco de baterías y/o paneles solares. De todas formas, si se le quita la energía eléctrica al computador de flujo, él deja de contabilizar durante el tiempo que ésta permanezca ausente y vuelve a iniciar cuando llegue la energía eléctrica. Los punteros, contadores, promedios, etc. no se pierden pues están en memoria externa la cual está siempre respaldada por la batería de litio de 3V. Sin embargo, es tarea del operador saber el tiempo que estuvo ausente la alimentación para marcar este intervalo como no medido.

4. PRUEBAS DEL PROTOTIPO EN CAMPO.

4.1. DETERMINACIÓN DEL ERROR MÁXIMO EN LA INSTRUMENTACIÓN.

Con el objetivo de evidenciar la importancia en la exactitud de los sensores de presión y temperatura, se realizaron unas pruebas simuladas aprovechando los algoritmos desarrollados en Matlab durante el presente proyecto.

4.1.1. Variación de una unidad de presión y temperatura. A continuación se presenta un ejemplo de como influye un error de presión y temperatura en la medida final del volumen de gas y el impacto económico que esto representa.

El ejercicio consiste en modificar los parámetros de presión y temperatura en 1psig y 1°F, respectivamente, para determinar las diferencias de facturación que existirían en cada caso. Se debe tener en cuenta que 1000 pies cúbicos (1 KCF) de gas cuestan un máximo de \$US1.5. Además, se suponen tres escenarios típicos de funcionamiento: alta, media, y baja presión, con velocidades típicas del flujo de gas (en CFH).

Para los cálculos, se definió la presión y temperatura base en 14.73psia y 60°F respectivamente, la presión atmosférica en 14.65psia (Barranca) y una concentración de gases de 0.01563% de N y 0.00039% de CO₂ con una densidad relativa de 0.565 según el gas de Barranca. Aplicando un programa de AGA7 y AGA8-Gross2 desarrollados en Matlab durante el proyecto, y que se describen en el Anexo A, se obtienen los resultados que se muestran en la Tabla 19. Como se puede ver, la lectura errada de un solo psig o un solo grado Fahrenheit ejerce un gran impacto en la facturación del gas. Se observa que en altas presiones 1°F produce mayor alteración en la medición, pero en presiones relativamente bajas como la de 250 tiene más impacto la diferencia de 1 psig que la de 1°F. No obstante para todos los casos la exactitud del sensor de presión es más relevante que la del sensor de temperatura, pues un pequeño porcentaje de error en presión representa más PSIGs que el mismo porcentaje de error en °F.

Como se ha mencionado, en Colombia la empresa Ecogás acepta hasta un $\pm 1\%$ de error total en la medición. Si la parte matemática se hace con rigurosa precisión, el error será exclusivo de la instrumentación, el acondicionamiento y el conversor análogo digital. Bajo este contexto de exactitud se seleccionarán los componentes de instrumentación.

Tabla 19. Efectos de los errores de presión y temperatura.

Error% P - T	Pres. PSIG	Tem. °F	Velocid. CFH	Factor de Compres. Z	Volumen Anual Mega pies MCF	Diferencias Millones US\$
	1200	90	30000	0.88380007	3'355,615.76	
	1201	90	30000	0.88372473	3'358,664.70	
0.083	+1	0	Diferencia en Volumen:		3,048.94	4.573
	1200	90	30000	0.88380007	3'355,615.76	
	1200	89	30000	0.88287435	3'365,256.57	
-1.11	0	-1	Diferencia en Volumen:		9,640.81	14.461
	700	80	30000	0.92086818	1'929,944.39	
	701	80	30000	0.92076556	1'932,860.32	
0.143	+1	0	Diferencia en Volumen:		2,915.93	4.374
	700	80	30000	0.92086818	1'929,944.39	
	700	79	30000	0.92026826	1'934,787.63	
-1.25	0	-1	Diferencia en Volumen:		4,843.24	7.265
	250	70	15000	0.96738078	346,590.147	
	251	70	15000	0.96725918	347,943.501	
0.400	+1	0	Diferencia en Volumen:		1,353.35	2.030
	250	70	15000	0.96738078	346,590.147	
	250	69	15000	0.96714790	347,329.350	
-1.43	0	-1	Diferencia en Volumen:		739.20	1.109

4.1.2. Error máximo del sensor de temperatura. El efecto de un error máximo de $\pm 0.12\%$ en la lectura de temperatura en un *City Gate* como el de Barranca, donde la temperatura promedio es de 70°F y con la turbina a velocidad máxima (15000 CFH), se puede cuantificar empleando los algoritmos de Matlab. Como se puede observar en la Tabla 20, el error producido en la medición del volumen de gas cuando existe el máximo error en el sensor de temperatura Murphy, es aproximadamente $\pm 0.018\%$, evidenciando la buena exactitud del transductor seleccionado. También se concluye que al aumentar la temperatura se disminuye el volumen.

Tabla 20. Efectos de la temperatura sobre la medición de volumen.

Error % -T-	Presión PSIG	Temp. °F	Factor Z	Vol. Diario KCF	Error Volumen %
Referencia →	250	70	0.96738077	949,562.05	
+0.12	250	70.084	0.96740026	949,392.36	
Diferencia →	0	+0.084	0.002%	-169.69	-0.01787%
-0.12	250	69.916	0.96736128	949,731.80	
Diferencia →	0	-0.084	-0.002%	169.75	0.01788%

4.1.3. Error máximo del sensor de presión. La Tabla 21 muestra el error en la medición del volumen ante un error máximo ($\pm 0.25\%$) del transductor de presión Barksdale. Allí se observa que el error en presión ocasiona un error casi de la

misma magnitud en el cálculo del volumen, luego la exactitud de este sensor es un parámetro crítico a la hora de seleccionarlo. También se concluye que al aumentar la presión, se incrementa el volumen.

Tabla 21. Efectos de la presión sobre la medición de volumen.

Error % -P-	Presión PSIG	Temper. °F	Factor Z	Vol. Diario KCF	Error Volumen %
Referencia →	250	70	0.96738077	949,562.05	
+0.25	250.625	70	0.96730478	951,879.33	
Diferencia →	+0.625	0	-0.007855%	+2317.28	+0.24404
-0.25	249.375	70	0.96745679	947,245.13	
Diferencia →	-0.625	0	+0.007858%	-2316.92	-0.24399

4.1.4. Error total en la instrumentación. El efecto total en la medición empleando el peor de los casos, el cual se obtiene al combinar los errores máximos de temperatura y presión para los transductores seleccionados en el presente proyecto, se observa en la Tabla 22. La temperatura produce el efecto contrario que la presión, por eso la combinación se hace con los errores máximos pero de signo contrario.

Tabla 22. Error en volumen al combinar errores en presión y temperatura.

Error % -P-	Error % -T-	Presión PSIG	Temper. °F	Factor Z	Vol. Diario KCF	Error Volumen %
Referencia →		250	70	0.96738077	949,562.05	
+0.25	-0.12	250.625	69.916	0.96728523	952,049.54	
Diferencia →		0.625	-0.084	-0.009876	2,487.49	0.26196%
-0.25	+0.12	249.375	70.084	0.96747622	947,075.90	
Diferencia →		-0.625	+0.084	+0.009867	-2486.15	

Según la Tabla 22, el máximo error en la medición del volumen de gas, generado por los transductores de presión y temperatura, es de $\pm 0.262\%$ en un escenario como el *city gate* de Barranca, donde la presión y la temperatura promedio son de 250 psig y 70 °F respectivamente. Este error permanece aún por debajo del máximo permitido por Ecogás (1%), haciendo más flexible el trabajo de otras fuentes de error del *hardware* del equipo.

4.2. CALIBRACIÓN DE LOS SENSORES EN LABORATORIO.

Antes de la prueba en campo se realizó una comprobación en laboratorio con la simulación de señales de presión y temperatura con potenciómetros.

De esta forma para simular la RTD, un simple potenciómetro de 200 Ω variándolo entre 100 Ω y 122.03 Ω correspondería a un intervalo de temperatura de 32-135°F (0°C a 57.22°C). En la práctica de la medición de gas en Colombia se usan los °F

para visualización de datos, por esto se escogió esta unidad. El computador de flujo puede medir hasta 158°F (70°C) pero para acotar más el rango de calibración al saber que la temperatura en Barranca no iba a sobrepasar los 130°F, se determinó un rango máximo de 32-135°F para calibración.

Las fórmulas dadas en la sección 2.1.3.6. permiten determinar el rango de variación del reóstato y la tensión en los terminales de la RTD para una temperatura dada. Retomando de esa sección y reacomodando términos se deducen lo siguiente:

$$R_{RTD}(T) = 100 + 0.385T_{(^{\circ}C)} \Omega = 100 + 0.385(5/9)(T_{(^{\circ}F)} - 32) [\Omega] \quad (\text{ecuación 50})$$

Para 32°F (0°C) resulta R=100Ω y para 135°F (57.22°C) resulta R=122.03Ω

y despejando T se obtiene,

$$T_{(^{\circ}F)} = 4.67532 R_{RTD} - 435.53247 \quad [^{\circ}F] \quad (\text{ecuación 51})$$

Además como por la RTD pasa siempre una corriente constante de 200μA, la caída de tensión en la RTD será:

$$V_{RTD} = 200\mu \cdot R_{RTD} \quad [V] \quad (\text{ecuación 52})$$

que remplazando en la ecuación 51 resulta la ecuación 53,

$$T_{(^{\circ}F)} = 23376.6 V_{RTD} - 435.53247 \quad [^{\circ}F], \quad (\text{ecuación 53})$$

con V_{RTD} en V.

Esta es la fórmula teórica de la temperatura que debería mostrar el display en °F dado un valor de resistencia de la RTD. En la Tabla 23 se muestran las lecturas del *display* reales obtenidas (sin calibración) contra los cálculos de esta ecuación teórica, y la lectura después de la calibración.

Tabla 23. Resultados calibración de temperatura.

R_{RTD}	V_{RTD} [mv]	T(°F) Teórico	T(°F) Práctica No Calibrada	T(°F) Práctica Calibrada	Error % final
100	20.0	32.00	30.8	32.0	0
102.5	20.5	43.69	40.7	43.7	-0.02
105.0	21.0	55.38	50.6	55.4	-0.04
107.5	21.5	67.06	59.5	67.1	-0.06
110.0	22.0	78.75	70.6	78.7	0.06
112.5	22.5	90.44	80.5	90.4	0.04
115.0	23.0	102.13	90.1	102.2	-0.07
117.5	23.5	113.82	99.9	113.9	-0.07
120.0	24.0	125.51	109.1	125.4	0.09
122.03	24.41	135.0	116.1	135.0	0

El rango de error está entre $\pm 0.12\%$ la cual es la exactitud dada por el RTD y valorada en la sección 2.1.2.1.

Para el transductor de presión (0-300psig $V_{out}=0-100\text{mV}$) se hizo un divisor de tensión que ofreciera un rango de 0-100mV con una entrada de tensión de 10V fijos (que es la alimentación que ofrece el regulador de referencia REF102), de esta forma se simulará un intervalo de presión de 0-300psi que corresponderá a una tensión de salida de 0-100mV del transductor. La ecuación teórica que relaciona la tensión de entrada con la presión leída es:

$$P = 3 * V ; \quad \text{(ecuación 54)}$$

con P en psig y V en mV

Midiendo la tensión en el potenciómetro con un multímetro Fluke-187 y comparándola con la presión que muestra el *display* tanto antes como después de la calibración, como resulta en la Tabla 24.

Tabla 24. Resultados calibración de presión.

V_{in} (mV)	P(psig) Teórico	P(psig) Práctica No Calibrada	P(psig) Práctica Calibrada	Error % final
0	0	0.2	0.00	0
10	30	28.3	29.2	2.67
20	60	57.9	58.9	1.83
30	90	87.5	88.9	1.22
40	120	117.2	119.6	0.33
50	150	146.6	149.5	0.27
60	180	176.3	179.6	0.22
70	210	206.1	210.1	0.095
80	240	235.6	239.8	0.083
90	270	265.4	269.8	0.070
100	300	294.8	300.0	0

El error a bajas presiones es mayor, y sólo el intervalo de presiones de 180-300psig cabe en el rango de exactitud de $\pm 0.25\%$ dado por el transductor. El *city gate* de Barranca tiene una presión promedio de 250psig y oscila un máximo de $\pm 20\text{psi}$ (230-270psig) así pues en este intervalo el error es menor de 0.09% y es admisible tal como fue valorado en la Sección 2.1.2.2. En general, se observa que la calibración tiene buenos resultados lineales y aunque no es 100% exacta, los resultados son muy buenos, especialmente en el rango de operación donde estará el equipo. El error mayor en presión se debe al total rango de entrada (0-100mV), en comparación con (20-24.41mV) para el de temperatura, el cual es más lineal dentro de sus cortos límites.

Ahora si, luego de verificar el funcionamiento en laboratorio se llevó el prototipo a campo y para observar su comportamiento frente a condiciones reales de proceso, el prototipo final del computador de flujo fue instalado en el *city gate* de Barrancabermeja, donde se obtuvieron los resultados que se describen en las siguientes secciones.

4.3 PRUEBA EN CAMPO DEL PROTOTIPO.

4.3.1. Descripción de la prueba. A partir de estas pruebas, el prototipo se empezó a denominar ECUFLOW2020T. Estas siglas tienen el siguiente significado:

E: ECOGAS

C: COLCIENCIAS

U: UIS (CIG – CEMOS)

FLOW: Medidor de flujo de gas

2020: Basado en el sistema TDS2020

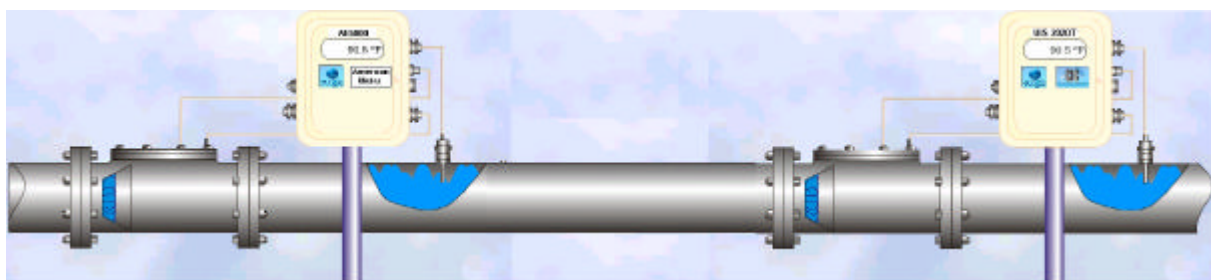
T: Para medidor primario tipo turbina

Para probar y constatar la exactitud real en la medición de volúmenes de gas el ECUFLOW2020T debería probarse con un computador de flujo patrón empleando a su vez turbina patrón. Ante la ausencia de disponibilidad estos dos equipos, la prueba más real que se podía hacer era la conexión en serie del ECUFLOW2020T con un computador de flujo comercial ya instalado en el *City Gate* de Barranca.

Aunque el computador de flujo ya instalado tiene su propio error ($<1\%$) y además es de la década de los '90 y emplea NX-19 para el cálculo de Z, Ecogás consintió esta prueba para comprobar la validez general del prototipo construido siendo concientes que la prueba real para determinar la exactitud del equipo debería hacerse en un laboratorio certificado de patrones de flujo, los cuales no hay en Colombia y cuyo proceso estaría contemplado en un proyecto futuro de certificación internacional. Por ahora se validará el desempeño del ECUFLOW2020T mostrando resultados coherentes y similares a los que arroje el computador ya instalado.

El ECUFLOW2020T se instaló en serie con el computador de flujo American Meter (AE5000), equipo con el cual opera el *city gate* de Barranca. Ver Figura 36. Las pruebas se realizaron durante cinco días, donde se tomaron los registros de presión, temperatura, volumen corregido, y no corregido, de los dos equipos.

Figura 35. Instalación de los computadores de flujo para la prueba.



Los parámetros de configuración para el computador de flujo suministrados por el operador del gasoducto fueron:

- Concentración de Nitrógeno: 0,01563%.
- Concentración de CO₂: 0,00039%.
- Gravedad Específica: 0,565.
- Presión Atmosférica: 14,519 psia.
- Temperatura Base: 60 °F.
- Presión Base: 14,65 psig.

Seguidamente al conocimiento de estos datos, se introdujeron en el computador de flujo para que quedara así configurado. Luego se procedió a la calibración por *software* del prototipo con un baño térmico y un peso muerto que había en campo, se calibro el cero y la máxima escala (presión: 0-300psig , temperatura: 32-135°F). La prueba se realizó por 5 días seguidos, aunque el primer día fue sólo en la tarde, los días 2 y 3 completos, y las mañanas de los días cuarto y quinto.

4.3.2. Resultados de la prueba. Las variables de lectura más importantes del computador de flujo son: presión, temperatura, volumen corregido y volumen no corregido. En las siguientes secciones se presentarán los promedios diarios de dichas variables. Es de aclarar que durante la prueba se alimentó el computador de flujo aisladamente con un batería de 12V prestada por el operador, esto con el fin de no emplear el sistema eléctrico de la compañía para evitarle riesgos ante alguna eventualidad que sucediera con el ECUFLOW2020T.

A continuación se muestran los promedios de presión y temperatura de cada día dados por los dos computadores de flujo. Si bien no se sabe la forma en que el AE5000 realiza sus promedios, el ECUFLOW los desarrolla así:

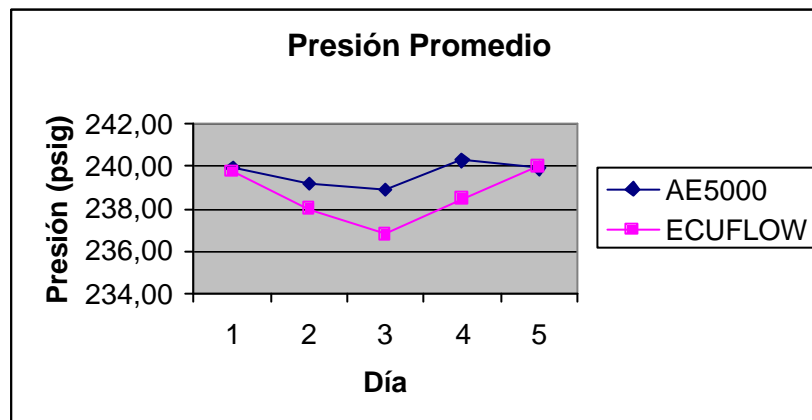
- Cada 50ms toma muestras de presión y temperatura
- Cada 2 segundos promedia los 40 datos de presión y 40 de temperatura y estos son los que refrescan la lectura del *display*.
- A su vez halla un promedio acumulado, que consiste en la media entre el promedio actual y el calculado los 2 segundos anteriores.
- Al llegar al minuto guarda el promedio acumulado en un vector.
- Cuando es corte de hora se hace el promedio del vector anterior (60 datos de presión y 60 de temperatura), se guarda en otro vector de promedios horarios y se reinician los promedios de minuto y acumuladores
- Cuando llega corte de día se promedia el vector de promedios horarios (24 datos de presión y 24 de temperatura), se guarda en un vector de almacenamiento de promedios diarios y reinicia el ciclo explicado. Este último promedio del día es el que se mostrará en las siguientes secciones.

4.3.2.1. Presión Promedio. Este promedio es obtenido de sumar los datos horarios de presión. La Tabla 25 y la Figura 37 muestran las lecturas de la presión promedio día, obtenidas de los computadores de flujo.

Tabla 25. Presión promedio.

Lectura Presión (psig)Promedio Día			
Día	AE5000	ECUFLOW	% Error
1	239,92	239,75	0,07
2	239,16	237,99	0,49
3	238,88	236,80	0,87
4	240,27	238,46	0,75
5	240,00	239,90	0,04

Figura 36. Presión promedio.



Se observa un error mayor en los datos en los días 2, 3, y 4; esto se debió al descargue inesperado de la batería prestada (al parecer no estaba completamente cargada como se presumía), cuando se revisó al cuarto día la tensión en bornes era de 9.5V, habiéndose empezado y calibrado con 12V. Ese mismo día se cambió la batería por otra y se continuó así el resto del cuarto día y el quinto día.

Es de aclarar, que si bien el ECUFLOW2020T acepta un rango de tensión de alimentación de 9-15V, la calibración debe efectuarse con la tensión a la cual vaya a quedar operando el equipo. Por tanto, las medidas más confiables son día 1 y 5. Por otro lado las medida del ECUFLOW2020T siempre está un tanto por debajo, esto es debido a una ínfima caída de presión que hay a través de la tubería, especialmente en los codos presentes entre los dos computadores de flujo.

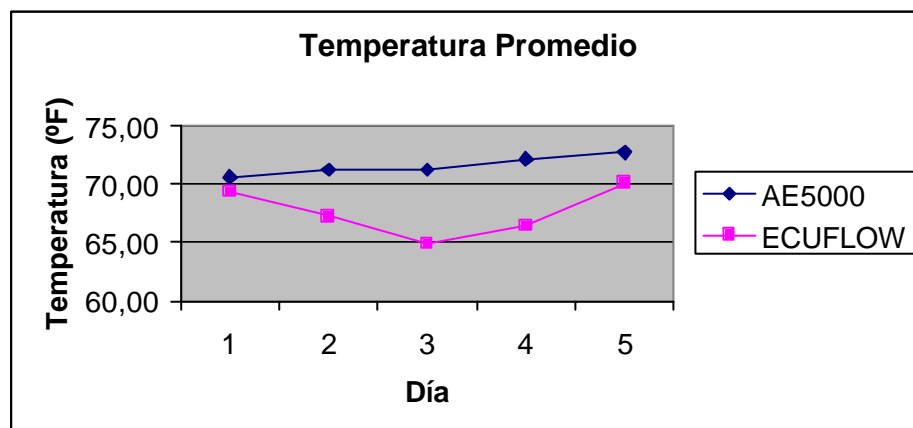
4.3.2.2. Temperatura Promedio. Este promedio es obtenido de sumar los datos horarios de temperatura. La Tabla 26 y la Figura 38 muestran las lecturas de la temperatura promedio día, obtenidas de los computadores de flujo.

Tabla 26. Temperatura promedio.

Lectura Temperatura (°F) Promedio Día			
Día	AE5000	ECUFLOW	% Error
1	70,64	69,38	1,78
2	71,27	67,26	5,63
3	71,21	64,86	8,92
4	72,16	66,47	7,89
5	72,72	71,30	1,95

Se presenta el mismo fenómeno de error explicado anteriormente para los días intermedios. Adicionalmente se observa que la lectura de temperatura siempre está levemente por debajo, dado que este *city gate* es un punto donde se pasa el gas de alta presión a baja presión ocurriendo un enfriamiento del gas y a medida que se avanza por la tubería las temperaturas son diferentes, por esta razón la toma de la RTD del ECUFLOW2020T está a una temperatura un poco por debajo de la toma de la RTD del American Meter.

Figura 37. Temperatura promedio.



4.3.2.3. Volumen corregido y no corregido. El volumen corregido es uno de los parámetros más importantes en la industria del gas, puesto que con dicho valor se realiza la transferencia de custodia, es decir, con este valor es que se factura a los clientes. El volumen corregido proviene del ajuste realizado con la norma AGA7 del volumen no corregido estimado por la turbina.

A continuación se presentan los reportes hora de los volúmenes corregido (CorVol) y no corregido (UnCorl) tomados de los computadores de flujo durante los días de la prueba. Al final de las tablas de los 5 días se muestra un resumen de los errores de medición de volumen de gas que son los que finalmente determinan la exactitud del prototipo.

Tabla 27. Volumen día 1.

Día 1 Time	AE5000		ECOFLOW	
	CorVol MCF	UnCor MCF	CorVol MCF	UnCor MCF
07:00PM	45	2	46,65	2,66
08:00PM	28	2	28,87	1,64
09:00PM	21	1	21,65	1,23
10:00PM	14	1	14,40	0,82
11:00PM	11	1	10,80	0,61

Figura 38. Volumen corregido día 1.

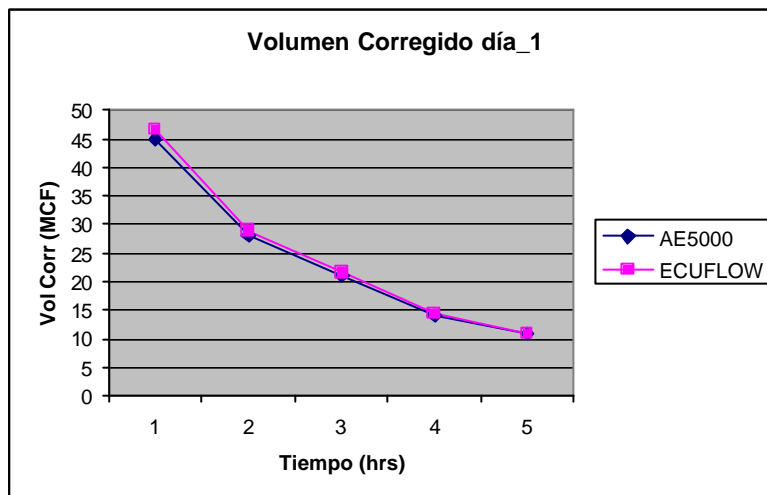
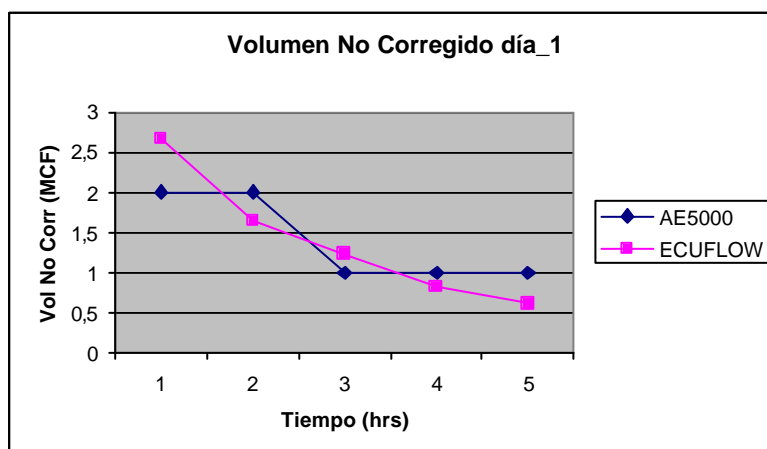


Figura 39. Volumen no corregido día 1.



De la Tabla 27 se observa la similitud de las medidas, sin embargo resalta que el AE5000 sólo da valores enteros y se ignora la forma en que aproxime (si por encima, por debajo, o por .5). De hecho, algunos operarios han mostrado su insatisfacción ante estas aproximaciones pues son muy gruesas y al parecer no tan coherentes con la realidad.

En la primera fila de la Tabla 28, donde el AE5000 reporta un volumen no corregido de cero (0) y a su vez un volumen corregido de 10; hecho que no tiene lógica pues con volumen cero no habría tampoco volumen corregido pues indica que la turbina no registró flujo de gas. No obstante, como la facturación se hace con el volumen corregido los operarios han ignorado este fenómeno o incoherencia y mostraron agrado por la resolución decimal y concordancia mostrada por el ECUFLOW2020T.

Tabla 28. Volumen día 2.

Día 2 Time	AE5000		ECOFLOW	
	CorVol MCF	UnCor MCF	CorVol MCF	UnCor MCF
12:00AM	10	0	10,79	0,61
01:00AM	12	1	11,79	0,81
02:00AM	11	0	10,80	0,61
04:00AM	15	1	16,23	0,92
05:00AM	51	3	52,59	2,98
06:00AM	67	4	69,00	3,90
07:00AM	74	4	74,51	4,21
08:00AM	82	4	85,29	4,82
09:00AM	106	7	107,99	6,15
10:00AM	143	8	145,45	8,39
12:00PM	83	4	83,31	4,81
01:00PM	44	3	43,62	2,51
02:00PM	34	2	36,66	2,11
03:00PM	30	2	33,08	1,91
04:04PM	39	2	39,97	2,31
05:00PM	47	3	49,06	2,81
06:00PM	60	3	59,82	3,41
07:00PM	45	3	43,90	2,51
08:00PM	30	1	28,15	1,60
09:00PM	22	2	21,12	1,20
10:00PM	16	0	15,82	0,90
11:00PM	12	1	12,30	0,70

Figura 40. Volumen corregido día 2.

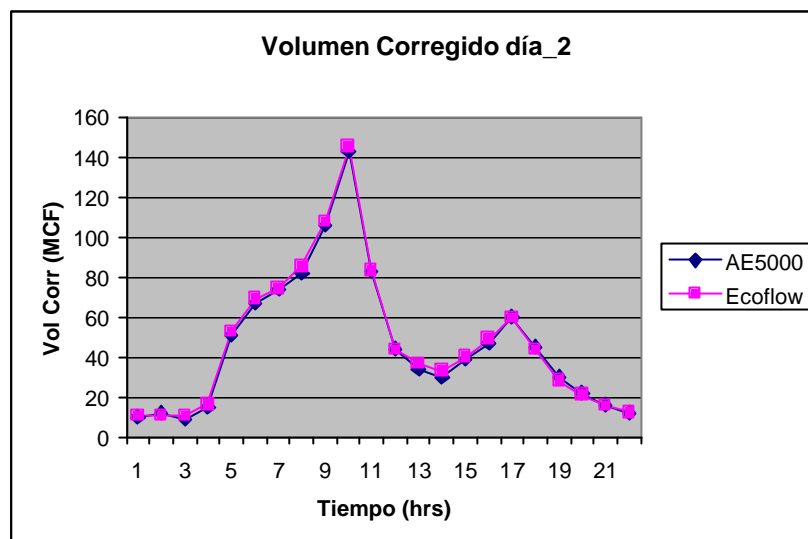


Figura 41. Volumen no corregido día 2.

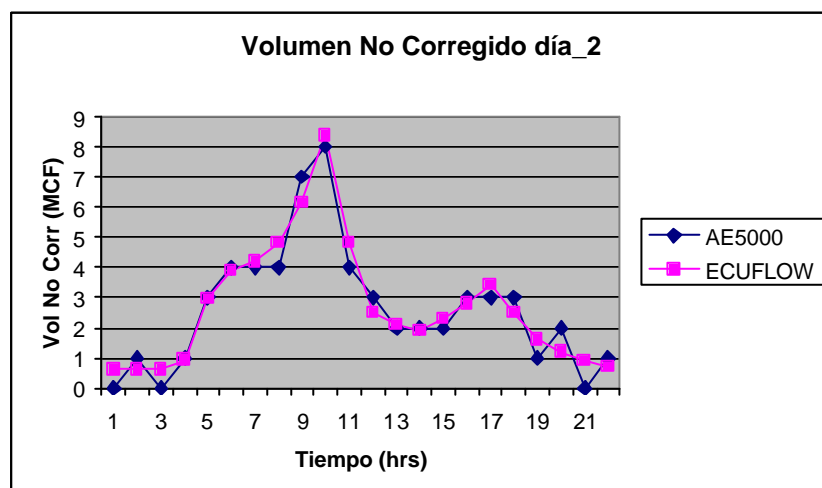


Tabla 29. Volumen día 3.

Día 3 Time	AE5000		ECOFLOW	
	CorVol MCF	UnCor MCF	CorVol MCF	UnCor MCF
12:00AM	11	1	10,55	0,60
01:00AM	10	0	9,55	0,50
02:00AM	11	1	10,54	0,60
03:00AM	10	1	10,55	0,60
04:00AM	14	0	15,83	0,90
05:00AM	46	3	49,47	2,81
06:00AM	67	4	69,09	3,91
07:00AM	75	4	76,10	4,31
08:00AM	82	5	82,85	4,71
09:00AM	104	6	110,30	6,31
10:00AM	143	8	145,10	8,38
11:00AM	148	9	143,26	8,29
12:00PM	88	5	82,27	4,71
01:00PM	45	3	42,01	2,41
02:00PM	36	2	36,70	2,11
03:00PM	37	2	36,68	2,11
04:00PM	41	2	43,64	2,51
05:00PM	52	3	54,46	3,11
06:00PM	60	4	59,94	3,41
07:00PM	45	2	44,00	2,51
08:00PM	30	2	28,22	1,60
09:00PM	23	1	22,94	1,30
10:00PM	15	1	14,11	0,80
11:00PM	13	1	12,33	0,70

Figura 42. Volumen corregido día 3.

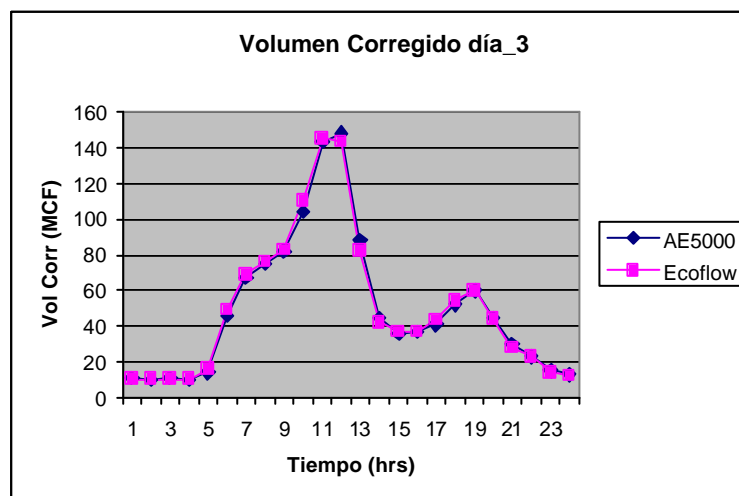


Figura 43. Volumen no corregido día 3.

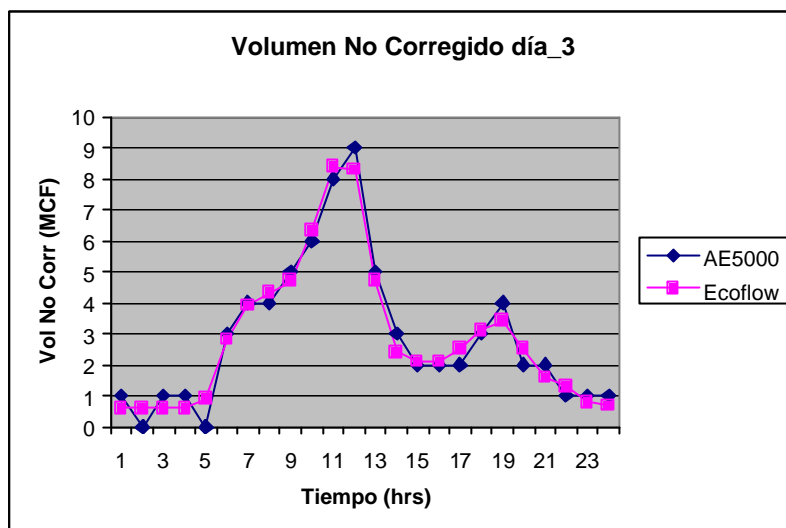


Tabla 30. Volumen día 4.

Día 4 Time	AE5000		ECOFLOW	
	CorVol MCF	UnCor MCF	CorVol MCF	UnCor MCF
12:00AM	10	0	10,56	0,60
01:00AM	11	1	10,55	0,60
02:00AM	10	1	10,55	0,60
03:00AM	11	0	10,55	0,60
04:00AM	15	1	21,12	1,20
05:00AM	53	3	53,09	3,01
06:00AM	68	4	70,82	4,01
08:00AM	31	1	31,66	1,81
09:00AM	24	2	22,92	1,30
10:00AM	18	1	17,63	1,00
11:00AM	14	1	14,08	0,80

Figura 44. Volumen corregido día 4.

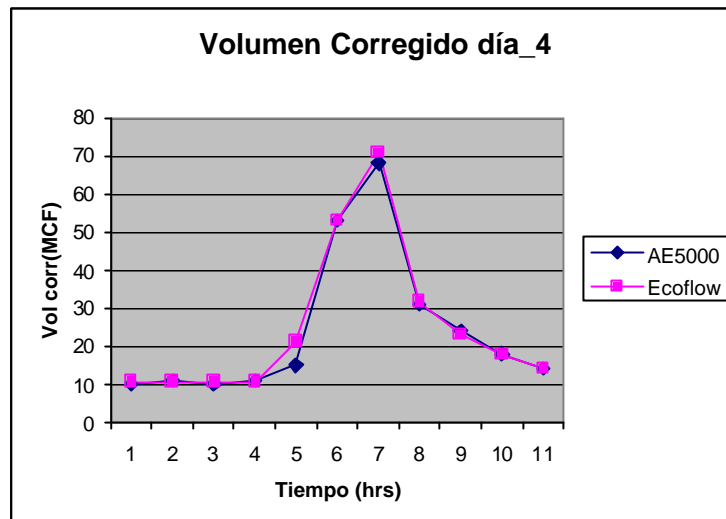


Figura 45. Volumen no corregido día 4.

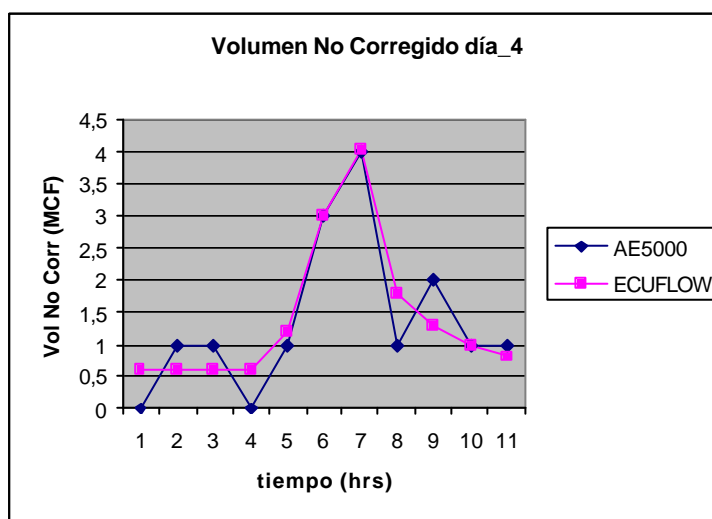


Tabla 31. Volumen día 5.

Día 1 Time	AE5000		ECOFLOW	
	CorVol MCF	UnCor MCF	CorVol MCF	UnCor MCF
12:00AM	10	0	12,33	0,70
01:00AM	11	1	10,58	0,60
02:00AM	10	0	10,58	0,60
03:00AM	11	1	10,58	0,60
04:00AM	12	1	10,57	0,60
05:00AM	33	2	32,23	2,01
06:00AM	63	3	61,73	3,51
07:00AM	74	5	75,56	4,31
08:00AM	75	4	75,58	4,31
09:00AM	83	5	84,20	4,81
10:00AM	106	6	106,13	6,11

Figura 46. Volumen corregido día 5.

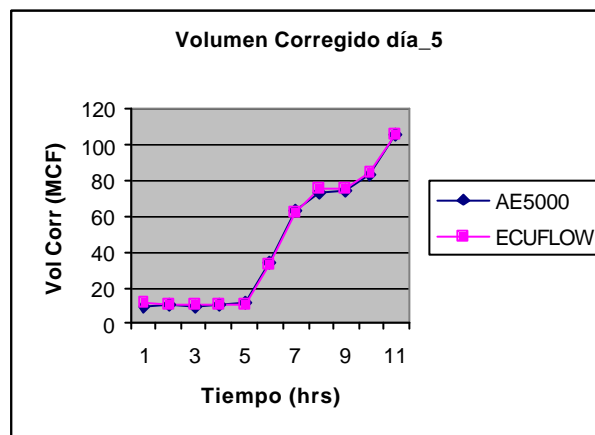
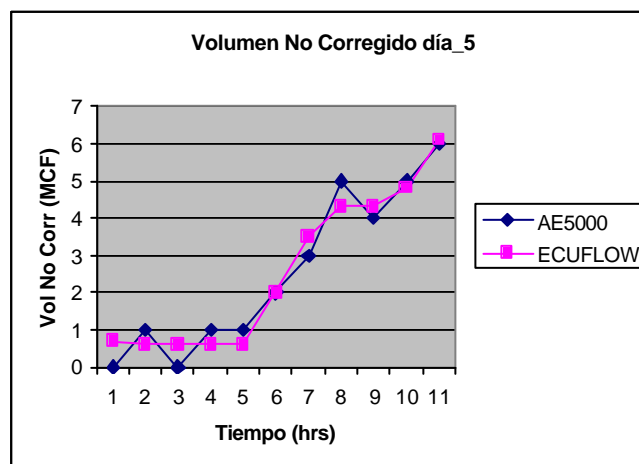


Figura 47. Volumen no corregido día 5.



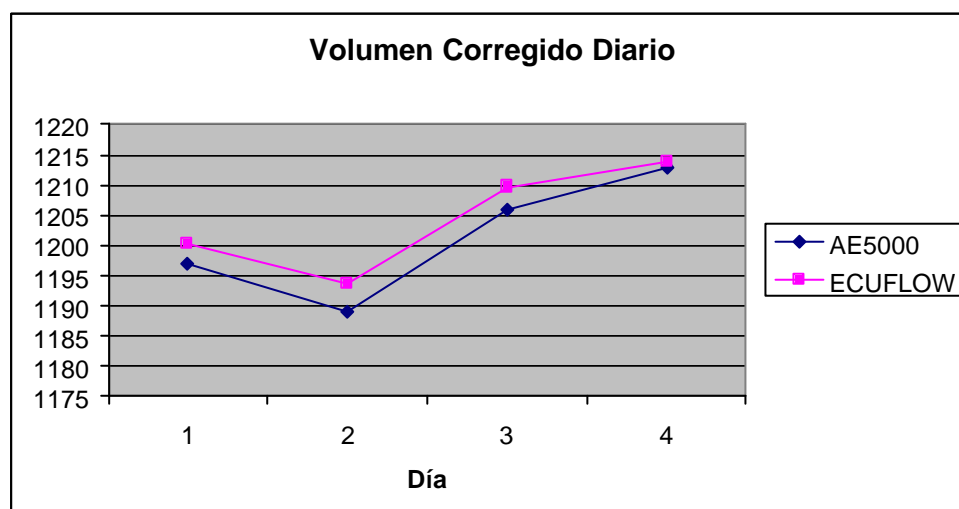
4.3.2.4. Volumen corregido diario. La acumulación de los reportes horarios del volumen corregido en el día es el consumo total diario.

Tabla 32. Volumen corregido diario.

LECTURA DE VOLUMEN DIARIO CORREGIDO					
Día	AE5000	ECUFLOW	Error de Volumen Diario	Costo Anual \$US	% Error
1	1197	1200.38	3.38	1850.55	0.28
2	1189	1193.73	4.73	2589.68	0.40
3	1206	1209.69	3.69	2020.28	0.31
4	1213	1213.76	0.76	416.10	0.06

La Tabla 34, muestra el consumo diario registrado por ambos computadores. También se especifica el error en volumen diario experimentado y un estimativo del costo anual de este error. (mil pues cúbicos valen US\$1.5). En la columna derecha el % de error del ECUFLOW frente al AE5000.

Figura 48. Volumen corregido diario.



4.3.3. Observaciones y Conclusiones de la Prueba.

- La prueba no se realizó de manera continúa, debido a que en ocasiones fue necesario extraer el ECUFLOW de la línea para evitar interferir con las labores diarias de los operadores del *city gate* de Barrancabermeja. Además como se comentó en la sección 4.2.1 hubo un cambio de batería en el día cuarto, que justifica las diferencias de presión y temperatura dadas antes del cambio debido a una alimentación no adecuada para calibración realizada.

- No obstante, se observa un buen seguimiento de las señales de presión y temperatura y finalmente del volumen registrado. Es así como considerando el computador de flujo AE5000 como un patrón, el ECUFLOW tiene un error promedio del 0,26 % en la medición del flujo.
- Una causa de la diferencia entre los dos equipos se debe a que el AE5000 emplea la antigua norma NX-19 y el ECUFLOW la norma AGA8-Gross2 y como se mostró en la Tabla 4 Sección 1.2.2, las diferencias pueden ser muy significantes.
- En Colombia el error máximo permitido para este tipo de equipos es de un 1% lo cual muestra la aceptación del ECUFLOW dentro de la normativa nacional.
- El computador de flujo ECUFLOW suministra los valores de volumen con mayor resolución decimal, en comparación con el AE5000 que realiza aproximaciones enteras. Esto mostró resultados más consistentes y de mayor aceptación por el operador.
- El ECUFLOW ha mostrado una confiabilidad y seguridad de operación de al menos 100 horas en campo.

CONCLUSIONES

- El prototipo obtenido, ECUFLOW2020T, es un computador de flujo habilitado para corregir el flujo de gas, utilizando como medidor primario una turbina.
- ECUFLOW esta basado en un sistema embebido (TDS2020F) importado. Además, contiene una tarjeta de acondicionamiento (diseñada y construida en la UIS), una batería de respaldo, transductores de presión estática, y de temperatura, encapsulado, "display" y teclado.
- En el ECUFLOW se ha implementado la norma AGA 8 GROSS 2 (actualmente en Colombia se utiliza el NX-19), a través del lenguaje de programación Forth, con el cual se ha obtenido un error del cero por ciento en el cálculo del factor de compresibilidad.
- El equipo cumple con la seguridad intrínseca necesaria para operar en áreas peligrosas, pues su consumo de corriente esta por debajo de los 100mA y la alimentación que requiere para funcionar no es mayor a 16 V. Gracias a lo anterior, se posibilitó la prueba en campo donde trabajo aproximadamente 80 horas.
- Durante la prueba se comprobó el excelente seguimiento que realiza el ECUFLOW2020T en la medición, ya que los resultados fueron contrastados con los arrojados por un equipo comercial (AE5000), obteniéndose una diferencia del 0,26 %.
- El ECUFLOW dispone de un *software* de control que se ejecuta bajo Windows, para descargar los reportes desde un computador portátil, y permitir mayor amigabilidad al usuario.
- El encapsulado del ECUFLOW es tipo NEMA 4X, (protección contra corrosión, polvo en el viento, lluvia, salpicaduras de agua, y agua directa sobre la carcasa, y formación de hielo).
- A través de este proyecto se ha incentivado el trabajo interdisciplinario dentro de la universidad y la cooperación interinstitucional, logrando un gran avance en la investigación y el desarrollo tecnológico de Colombia.

RECOMENDACIONES

- Los resultados obtenidos en este proyecto han sido exitosos, pero se requiere de mas apoyo a las siguientes fases del proyecto para llevar a cabo el desarrollo e implementación final de computadores de flujo hechos en Colombia con excelentes garantías de funcionalidad y operabilidad.
- Las perspectivas económicas del equipo se muestran alentadoras, ya que su costo comercial podría llegar a ser de solo un 60 % del que actualmente poseen los equipos importados.
- Los costos del computador de flujo se pueden reducir aun más si se proyecta la construcción de la tarjeta CPU en nuestro país con la colaboración de la academia.
- Las siguientes etapas del proyecto se deben dirigir al seguimiento de las normas internacionales de emisión e inmunidad electromagnética, y de seguridad intrínseca, para incrementar la confiabilidad de operación del ECOFLOW.
- Se debe contactar un laboratorio de metrología internacional que pueda garantizar y certificar las mediciones que realiza el equipo, con el fin de obtener una entrada rápida al mercado de computadores de flujo en una futura etapa de comercialización.

BIBLIOGRAFÍA

- AMERICAN GAS ASSOCIATION. (1996). AGA report n° 7: Measurement of Gas by Turbine Meters. Arlington, Virginia.
- AMERICAN GAS ASSOCIATION. (1992). AGA report n° 8: Compressibility Factors of Natural Gas and others Relates Hydrocarbon Gases. Arlington, Virginia.
- AMERICAN PETROLEUM INSTITUTE. (1993). Manual of Petroleum Measurements Standards. Chapter 21- Flow Measurement using Electronic Metering Systems. Washington, D.C.
- BAKER, R. (2000). Flow Measurement Handbook: Industrial Designs, Operating, Principles, Performance an Applications. Cambridge University.
- BRODIE, L. and Forth, Inc. (1987). Starting Forth: An Introduction to the Forth Language and Operating System for Beginners and Professionals. Prentice Hall, Englewood Cliffs, New Jersey.
- CORZO, M. (2003). Computador de Flujo: Diseño e Implementación del Soporte de Hardware y Selección de la Instrumentación. Universidad Industrial de Santander, Bucaramanga.
- GÓMEZ, L. (2004). Computador de Flujo: Diseño e Implementación del Software para la Medicion del Flujo de Gas. Universidad Industrial de Santander, Bucaramanga.
- GONZÁLEZ, R. (2000). Diseño e Implementación del Hardware y Firmware para un Computador de Flujo Tipo Turbina usando la Norma AGA 7. Universidad Industrial de Santander, Bucaramanga.
- LOY, E. y LANASA, P. (2002). Fluid Flow Measurement: A Practical Guide to Accurate Flow Measurement. Gulf Professional Publishing, United States of America.
- RONCANCIO, R. Diseño e Implementación del Hardware y Software de Presentación para el Computador de Flujo de un Medidor de Gas tipo turbina. Universidad Industrial de Santander, Bucaramanga.

ANEXO A. CODIGO EN MATLAB DEL AGA8-GROSS2

AGA8.m

```
% PROGRAMA PARA CALCULAR EL FACTOR DE COMPRESIBILIDAD CON EL METODO AGA8-
GROSS2

clear all, clc

global MR MW VIR DOAIR HCH BCH BCO2 X X1 X2 F1 F2 BMIX CMIX C11 C22 C33 C0 C1
C2

TODOVAR; % definición de más variables globales

disp ('-----
-----')
disp (' - - PROGRAMA PARA HALLAR EL COEFICIENTE DE COMPRESIBILIDAD - -')
disp (' - - DEL GAS NATURAL CON EL METODO AGA8 GROSS-2 - -')
disp ( ' ' );

TDATA = input('Entre la TEMPERATR. en °F = '); % TEMPERATURA EN °F
PDATA = input('Entre la PRESIÓN en PSIA = '); % PRESIÓN EN PSIA

% Escoge el tipo de gas segun la pag117 del AGA8
ICOMP = input('Tipo de GAS: 1=Coast 2=Amarillo 3=Ekofish 4=HighN2 5=HighCO2
6=Ecogas = ');

METHOD = 2 ; % Número del método: en este caso es el Gross-2

% De aqui en adelante es traslación literal a Matlab de las líneas de código
% dadas en Fortran y diseñadas por el comité AGA8. La estructura del programa
% nombre de variables y subrutinas fueron creados por ese comité.
% Remítase al Apéndice B4 pag.100 de la norma AGA8 para verlo en Fortran.

%          Grav.Esp. TB      PB          %N2          %CO2          %H2      %C2
XK(:,1) = [0.581078; 60; 14.73; 0.002595; 0.005956; 0; 0; ];

XK(:,2) = [0.608657; 60; 14.73; 0.031284; 0.004676; 0; 0; ];

XK(:,3) = [0.649521; 60; 14.73; 0.010068; 0.014954; 0; 0; ];

XK(:,4) = [0.644869; 60; 14.73; 0.134650; 0.009850; 0; 0; ];

XK(:,5) = [0.686002; 60; 14.73; 0.057021; 0.075851; 0; 0; ];

XK(:,6) = [0.565000; 60; 14.73; 0.015630; 0.000390; 0; 0; ];

for J = 1 : 7
    XD(J) = XK(J,ICOMP);
```

```

        XDC(J) = XK(J,ICOMP);
end

PARAMGS;

TBDR = 5/9 * (XD(2) + 459.67);
PB = XD(3) * 0.006894757;

SPECG = XD(1);
HS = 0;
X(1) = 0;
for I = 2 : 5
    X(I) = XD(I+2);
end

[ZB,DB,ERRNUM] = CHARGS(METHOD,HS,SPECG,X,TBDR,TBDR,PB,TBDR,PB);

TK =(5/9) * (TDATA + 459.67);
PMP = PDATA * 0.006894757;

D = DGROSS (PMP,TK);
Z = ZGROSS (D, TK); % Esta es la respuesta Z
FPV = sqrt (ZB/Z); % Este es el factor de supercompresibilidad
Df = D* MWX / 16.01846 ; % Esta es la densidad molar

% Gross-2 COMPARAR RESULTADOS de Z CON LA PAG. 119 DEL AGA8
disp ('-----')
disp ('COEFICIENTE DE COMPRESIBILIDAD Z = ')
disp(Z)

```

TODOVAR.m

```

global PROG RGAS MWX MW % CONSTANT

global XX % GROSSCOMP

global DHIGH PLOW PHIGH TLOW THIGH %LIMIT

global BB B0 B1 B2 C0 C1 C2 TOLD BBMIX CCMIX % VIRIAL2

```

PARAMGS.m

```

TODOVAR;

RGAS = 8.31451e-3;

MW(2) = 28.01350;

```

```

MW(3) = 44.010;
MW(4) = 2.0159;
MW(5) = 28.01;

B0(2,2) = -0.144600;
B1(2,2) = 0.740910e-3;
B2(2,2) = -0.911950e-6;
B0(3,3) = -0.868340;
B1(3,3) = 0.403760e-2;
B2(3,3) = -0.516570e-5;
B0(4,4) = -0.110596e-2;
B1(4,4) = 0.813385e-4;
B2(4,4) = -0.987220e-7;
B0(5,5) = -0.130820;
B1(5,5) = 0.602540e-3;
B2(5,5) = -0.644300e-6;
B0(2,3) = -0.339693;
B1(2,3) = 0.161176e-2;
B2(2,3) = -0.204429e-5;
B0(1,4) = -0.521280e-1;
B1(1,4) = 0.271570e-3;
B2(1,4) = -0.250000e-6;
B0(1,5) = -0.687290e-1;
B1(1,5) = -0.239381e-5;
B2(1,5) = 0.518195e-6;
B0(2,4) = 0.012;

BB(1,1) = -0.425468;
BB(2,1) = 0.877118e-3;
BB(3,1) = -0.824747e-6;
BB(1,2) = 0.286500e-2;
BB(2,2) = -0.556281e-5;
BB(3,2) = 0.431436e-8;
BB(1,3) = -0.462073e-5;
BB(2,3) = 0.881510e-8;
BB(3,3) = -0.608319e-11;

C0(2,2) = 0.784980e-2;
C1(2,2) = -0.398950e-4;
C2(2,2) = 0.611870e-7;
C0(3,3) = 0.205130e-2;
C1(3,3) = 0.348880e-4;
C2(3,3) = -0.837030e-7;
C0(2,3) = 0.552066e-2;
C1(2,3) = -0.168609e-4;
C2(2,3) = 0.157169e-7;
C0(3,2) = 0.358783e-2;
C1(3,2) = 0.806674e-5;
C2(3,2) = -0.325798e-7;
C0(4,4) = 0.104711e-2;
C1(4,4) = -0.364887e-5;
C2(4,4) = 0.467095e-8;
C0(1,5) = 0.736748e-2;
C1(1,5) = -0.276578e-4;
C2(1,5) = 0.343051e-7;
-----

```

CHARGS.m

```
function [ZB,DB,ERRNUM] = CHARGS(METHOD,HV,GR,X,TH,TD,PD,TGR,PGR)

global MR MW VIR DOAIR HCH BCH X C0 C1 C2

TODOVAR;

TOLD = 0 ;
ERRNUM = 0;
VIR = -0.12527 + 5.91e-4*TGR - (6.62e-7)*(TGR^2);

DOAIR = 28.96256/(RGAS*TGR/PGR + VIR);

G1 = -2.709328;
G2 = 0.021062199;
HTV4 = 285.83;
HTV5 = 282.98;

%% METODO 2

if (METHOD == 2)
    Z0 = 1;

    X(1) = 1 - X(2) - X(3) - X(4) - X(5);

    MR = GR*Z0*RGAS*TGR/PGR*DOAIR;
    MW(1) = (MR - X(2)*MW(2) - X(3)*MW(3) - X(4)*MW(4) - X(5)*MW(5))/X(1);
    HCH = (MW(1) - G1)/G2;
    BCH = BB(1,1) + TGR*(BB(1,2)+BB(1,3)*TGR) +
    (BB(2,1)+TGR*(BB(2,2)+BB(2,3)*TGR))*HCH ...
    + (BB(3,1)+TGR*(BB(3,2)+BB(3,3)*TGR))*HCH^2 ;

    [BMIX,TEMP,ERRNUM]=VIRGS1(TGR,X,BCH,1);

    ZONEW = 1 + BMIX*PGR/RGAS/TGR;

    while ((abs(Z0/ZONEW - 1))>(0.5e-10))
        Z0 = ZONEW;
        MR = GR*Z0*RGAS*TGR/PGR*DOAIR;
        MW(1) = (MR - X(2)*MW(2) - X(3)*MW(3) - X(4)*MW(4) - X(5)*MW(5))/X(1);
        HCH = (MW(1) - G1)/G2;

        BCH=BB(1,1)+TGR*(BB(1,2)+BB(1,3)*TGR)+(BB(2,1)+TGR*(BB(2,2)+BB(2,3)*TGR))*HCH
        ...
        + (BB(3,1)+TGR*(BB(3,2)+BB(3,3)*TGR))*HCH^2 ;

        [BMIX,TEMP,ERRNUM]=VIRGS1(TGR,X,BCH,1);
        ZONEW = 1 + BMIX*PGR/RGAS/TGR;

    end
end

if (ERRNUM ~= 0)
    return
end
```



```

B0(1,1) = BB(1,1) + HCH*(BB(2,1)+BB(3,1)*HCH);
B1(1,1) = BB(1,2) + HCH*(BB(2,2)+BB(3,2)*HCH);
B2(1,1) = BB(1,3) + HCH*(BB(2,3)+BB(3,3)*HCH);

C0(1,1) = -0.302488 + HCH*(0.646422e-3 - 0.332805e-6*HCH );
C1(1,1) = 0.195861e-2 + HCH*(-0.422876e-5 + 0.223160e-8*HCH );
C2(1,1) = -0.316302e-5 + HCH*(0.688157e-8 - 0.367713e-11*HCH );

XX(1) = X(1);
XX(2) = X(2);
XX(3) = X(3);
XX(4) = X(4);
XX(5) = X(5);

TLOW = 263;
THIGH = 338;
PLOW = 0.5e-9;
PHIGH = 12;
DHIGH = 8 ;

DB = DGROSS(PGR,TGR);
MWX = GR*DOAIR/DB;

TB = (60 + 459.67)/1.8;
PB = 14.73*6894.757/1000000;
DB = DGROSS(PB,TB);
ZB = ZGROSS(DB,TB);

```

PGROSS.m

```

function [PGROSS] = PGROSS(D,T)

TODOVAR;

PGROSS = D*RGAS*T*ZGROSS(D,T);

```

DGROSS.m

```

function [DGROSS] = DGROSS(P,T)

TODOVAR;
global X1 X2 F1 F2

TOL = 1e-6;
X1 = PGROSS (0,T);
X2 = PGROSS (DHIGH,T);
F1 = PGROSS (X1,T) - P;

```

```

if (abs(F1) <= TOL )
    DGROSS = X1;
    return;
end

F2 = PGROSS (X2,T) - P;

if (abs(F2) <= TOL)
    DGROSS = X2;
    return;
end

if ((F1*F2) > 0)
    disp ('RAIZ NO HALLADA');
    DGROSS = 0;
    return;
end

for IC = 1:100
    X3 = (X1*F2 - X2*F1)/(F2 - F1);
    F3 = PGROSS (X3,T) - P ;

    if (mod (IC,6) == 0)
        DGROSS = ( X1 + X2 )/2;

    else
        if (((F1-F2)*(F1-F3)*(F2-F3)) == 0 )
            return;
        end
        DGROSS = X1*F2*F3/((F1-F2)*(F1-F3))+X2*F1*F3/((F2-F1)*(F2-
F3))+X3*F1*F2/((F3-F1)*(F3-F2));

        if (((DGROSS - X1)*(DGROSS-X2))>= 0)
            DGROSS = (X1 + X2)/2;
        end
    end

    F = PGROSS(DGROSS,T) - P ;

    if ( abs(F) <= TOL)
        return;
    end

    if ((F*F3)<0)
        disp ('dgross es x1')
        X1 = DGROSS;
        F1 = F;
        X2 = X3;
        F2 = F3;
    elseif ((F3*F1)> 0)
        X1 = DGROSS;
        F1 = F;
    else
        X2 = DGROSS;
        F2 = F;
    end
end

```

```

end

disp('DGROSS NO CONVERGE');
DGROSS = 0 ;
return ;

```

VIRGS.m

```

function[BMIX,CMIX,ERRNUM] = VIRGS(T,X,TEMP,OPT)

TODOVAR;

global BCO2 CMIX BMIX C11 C22 C33

ERRNUM = 0;

if (T == TOLD)
    BMIX = BBMIX;
    CMIX = CCMIX;
    return
end

X11 = X(1)*X(1);
X22 = X(2)*X(2);
X33 = X(3)*X(3);
X44 = X(4)*X(4);
X55 = X(5)*X(5);

if (OPT == 0 )
    BCH = B0(1,1) + T*(B1(1,1)+B2(1,1)*T);
end
BN2 = B0(2,2) + T*(B1(2,2)+B2(2,2)*T);
BCO2 = B0(3,3) + T*(B1(3,3)+B2(3,3)*T);
BH2 = B0(4,4) + T*(B1(4,4)+B2(4,4)*T);
BCO = B0(5,5) + T*(B1(5,5)+B2(5,5)*T);

if (BCO2*BCH < 0)
    disp ('raiz negativa');
    pause(2);
    ERRNUM = 1 ,
    return
end

B12 = (0.72 + 1.875e-5*(320-T)*(320-T))*(BN2 + BCH)/2;
B13 = -0.865*sqrt(BCO2*BCH);
B14 = B0(1,4) + T*(B1(1,4) + B2(1,4)*T);
B15 = B0(1,5) + T*(B1(1,5) + B2(1,5)*T);
B23 = B0(2,3) + T*(B1(2,3) + B2(2,3)*T);
B24 = B0(2,4);
BMIX = BCH*X11 + BN2*X22 + BCO2*X33 + BH2*X44 + BCO*X55 ...
    +2*B12*X(1)*X(2) + 2*B13*X(1)*X(3) ...
    +2*B14*X(1)*X(4) + 2*B15*X(1)*X(5) ...
    +2*B23*X(2)*X(3) + 2*B24*X(2)*X(4) ;

```

```

if (OPT == 1)
    disp('paso por el opt 1 de virg');
    return
end

TOLD = T;
BBMIX = BMIX;

E = 0.92 + 0.0013*(T-270);
F = 1/3;
C11 = C0(1,1)+ T*(C1(1,1)+C2(1,1)*T);
C22 = C0(2,2)+ T*(C1(2,2)+C2(2,2)*T);
C33 = C0(3,3)+ T*(C1(3,3)+C2(3,3)*T);
C44 = C0(4,4)+ T*(C1(4,4)+C2(4,4)*T);

if ((C11 < 0) | (C33 < 0))
    disp('invalido termino en virgs');
    ERRNUM = 1 ;
    return
end

C15 = 3*(C0(1,5) + T*(C1(1,5) + C2(1,5)*T));
C23 = 3*(C0(2,3) + T*(C1(2,3) + C2(2,3)*T));
C32 = 3*(C0(3,2) + T*(C1(3,2) + C2(3,2)*T));

CMIX = C11*X11*X(1) + C22*X22*X(2) + C33*X33*X(3) + C44*X44*X(4) ...
    + C23*X22*X(3) + C32*X33*X(2) + C15*X11*X(5) ...
    + E*3*X11*X(2)*(C11*C11*C22)^F ...
    + E*3*X22*X(1)*(C11*C22*C22)^F ...
    + 0.92*3*X11*X(3)*(C11*C11*C33)^F ...
    + 0.92*3*X33*X(1)*(C11*C33*C33)^F ...
    + 1.20*3*X11*X(4)*(C11*C11*C44)^F ...
    + 1.10*6*X(1)*X(2)*X(3)*(C11*C22*C33)^F ;

CCMIX = CMIX;

```

VIRGS1.m

```

function[BMIX,CMIX,ERRNUM] = VIRGS1(T,X,BCH,OPT)

TODOVAR;

global X BMIX CMIX

OPT=1;

ERRNUM = 0;

if (T == TOLD)
    BMIX = BBMIX;
    CMIX = CCMIX;
    return
end

```

```

X11 = X(1)*X(1);
X22 = X(2)*X(2);
X33 = X(3)*X(3);
X44 = X(4)*X(4);
X55 = X(5)*X(5);

BN2 = B0(2,2) + T*(B1(2,2)+B2(2,2)*T);
BCO2 = B0(3,3) + T*(B1(3,3)+B2(3,3)*T);
BH2 = B0(4,4) + T*(B1(4,4)+B2(4,4)*T);
BCO = B0(5,5) + T*(B1(5,5)+B2(5,5)*T);

if ((BCO2*BCH)< 0)
    disp ('raiz negativa');
    ERRNUM = 1 ,
    return
end

B12 = (0.72 + 1.875e-5*(320-T)*(320-T))*(BN2 + BCH)/2;
B13 = -0.865*sqrt(BCO2*BCH);
B14 = B0(1,4) + T*(B1(1,4) + B2(1,4)*T);
B15 = B0(1,5) + T*(B1(1,5) + B2(1,5)*T);
B23 = B0(2,3) + T*(B1(2,3) + B2(2,3)*T);
B24 = B0(2,4);

BMIX = BCH*X11 + BN2*X22 + BCO2*X33 + BH2*X44 + BCO*X55 ...
    +2*B12*X(1)*X(2) + 2*B13*X(1)*X(3) ...
    +2*B14*X(1)*X(4) + 2*B15*X(1)*X(5) ...
    +2*B23*X(2)*X(3) + 2*B24*X(2)*X(4) ;

return;

```

ZGROSS.m

```

function [ZGROSS] = ZGROSS(D,T)

TODOVAR;

TEMP=0;
global X

X = XX;

ZGROSS = 0;

[BMIX,CMIX,ERRNUM]=VIRGS(T,X,TEMP,0);

if (ERRNUM ~= 0)
    return;
end

ZGROSS = 1 + BMIX*D + CMIX*D*D ;

```


ANEXO B. CODIGO DEL AGA8-GROSS2 EN FORTH

AGA8.TDS: RUTINA PARA HALLAR Z (FACTOR DE COMPRESIBILIDAD)

```
INCLUDE FPA.TDS
\
INCLUDE #EXTVAR2.TDS

FEVAR PDATA \ Presion de Entrada en **PSIA** para trabajo de AGA
FEVAR TDATA \ Tempera de Entrada en **F** para trabajo de AGA

1000E0 PDATA F!!
50E0 TDATA F!!

EVAR DFLAG 0 DFLAG !! \ Usada en DGROS

FEVAR PATMO 14.73E0 PATMO F!!
FEVAR TBASE 60E0 TBASE F!! \ Temp de Referencia F
FEVAR PBASE 14.73E0 PBASE F!! \ Presion de Referencia PSIA
FEVAR SPECG 0.581078E0 SPECG F!! \ Gravedad Especifica
FEVAR %NI 0.002595E0 %NI F!!
FEVAR %CO2 0.005956E0 %CO2 F!!
FEVAR %HI 0E0 %HI F!!
FEVAR %CO 0E0 %CO F!!

FEVAR TBDR 0E0 TBDR F!!
FEVAR HS 0E0 HS F!!
FEVAR FPV 0E0 FPV F!!
FEVAR Df 0E0 Df F!!
FEVAR TK 0E0 TK F!!
FEVAR PMP 0E0 PMP F!!
FEVAR ZETA 0E0 ZETA F!!

INCLUDE PIDE-PTG.TDS

INCLUDE #COMMON.TDS
INCLUDE #VIRGS2.TDS
INCLUDE #ZGROS2.TDS
INCLUDE #PGROS.TDS
INCLUDE #DGROS2.TDS
INCLUDE #CHARGS2.TDS

\ *****
: AGA8

." ...CALCULANDO... "

\ PREAV F@@ PDATA F!!
```

```

\ TEMAV F@@ TDATA F@@

\ PREAV F@@ PATMO F@@ F+ PDATA F!!
\ TEMAV F@@ TDATA F!!

0E0 B011 F!! 0E0 B111 F!! 0E0 B211 F!!
0E0 C011 F!! 0E0 C111 F!! 0E0 C211 F!!

\ TBDR = 5/9 * (TBASE + 459.67) de F a K
TBASE F@@ 459.67E0 F+ 5E0 9E0 F/ F* TBDR F!!

\ PB = PBASE * 0.006894757 Pasa de 'psia' a 'MPa'
PBASE F@@ 0.006894757E0 F* PB F!!

0E0 HS F!!

0E0 XIN1 F!!
%NI F@@ XIN2 F!!
%CO2 F@@ XIN3 F!!
%HI F@@ XIN4 F!!
%CO F@@ XIN5 F!!

\ ASIGNACION DE VARIABLES
HS F@@ HV F!!
SPECG F@@ GR F!!
TBDR F@@ TH F!!
TBDR F@@ TD F!!
PB F@@ PD F!!
TBDR F@@ TGR F!!
PB F@@ PGR F!!

0E0 MW1 F!!

CHARGS

TDATA F@@ 459.67E0 F+ 5E0 9E0 F/ F* TK F!!
PDATA F@@ 0.006894757E0 F* PMP F!!

PMP F@@ PRE F!!
TK F@@ TEM F!!
DGROS
DGROSS F@@ DEN F!!

TK F@@ TEM F!!
ZGROS
ZGROSS F@@ ZETA F!!

\ FPV = sqrt (ZB/Z);
ZB F@@ ZETA F@@ F/ FSQRT FPV F!!

\ Df = D* MWX / 16.01846
DEN F@@ MWX F@@ F* 16.01846E0 F/ Df F!!

." | ZETA = "
ZETA F@@ F.
;

```


COMMON.TDS: RUTINA PARA HALLAR Z (FACTOR DE COMPRESIBILIDAD)

FEVAR MWX
EVAR PROG

FEVAR B011
FEVAR B111
FEVAR B211
FEVAR C011
FEVAR C111
FEVAR C211

FEVAR TOLD
FEVAR BBMIX
FEVAR CCMIX

FEVAR DHIGH
FEVAR PLOW
FEVAR PHIGH
FEVAR THIGH
FEVAR TLOW

FEVAR XIN1
FEVAR XIN2
FEVAR XIN3
FEVAR XIN4
FEVAR XIN5

FEVAR XX1
FEVAR XX2
FEVAR XX3
FEVAR XX4
FEVAR XX5

FEVAR MW1

\ **** GENERALES

EVAR IC
FEVAR TEM
FEVAR PRE
FEVAR TOL
FEVAR X1
FEVAR X2
FEVAR X3
FEVAR FFF
FEVAR F1
FEVAR F2
FEVAR F3
FEVAR DGROSS
FEVAR PGROSS

FEVAR DEN
FEVAR ZGROSS

EVAR ERRNUM
FEVAR BMIX

FEVAR CMIX
FEVAR TEMP

FEVAR FLAG
FEVAR HCH
FEVAR HV
FEVAR GR

FEVAR BCH
FEVAR DOAIR
FEVAR Z0
FEVAR Z0NEW
FEVAR Z0TDPD
FEVAR G1
FEVAR G2
FEVAR HNO
FEVAR MR
FEVAR HTV4
FEVAR HTV5

FEVAR TB \ TEMP BASE
FEVAR PB \ PRESION BASE

FEVAR DB
FEVAR ZB
FEVAR TH
FEVAR TD
FEVAR PD
FEVAR TGR
FEVAR PGR
FEVAR VIR
FEVAR OPT
FEVAR EEE
FEVAR X11
FEVAR X22
FEVAR X33
FEVAR X44
FEVAR X55

FEVAR C11
FEVAR C22
FEVAR C33
FEVAR C23
FEVAR C32
FEVAR C15
FEVAR C44
FEVAR BN2
FEVAR BCO2
FEVAR BH2
FEVAR BCO
FEVAR B12
FEVAR B13
FEVAR B14
FEVAR B15
FEVAR B23
FEVAR B24

CHARGS.TDS:

EVAR FLAKC

: CHARGS

0 FLAKC !!

0E0 TOLD F!!

0 ERRNUM !!

-0.12527E0

5.91E-4 TGR F@@ F* F+

6.62E-7 TGR F@@ TGR F@@ F* F* F-

VIR F!!

28.96256E0

8.31451E-3 TGR F@@ F*

PGR F@@ F/ VIR F@@ F+ F/

DOAIR F!!

-2.709328E0 G1 F!!

0.021062199E0 G2 F!!

285.83E0 HTV4 F!!

282.98E0 HTV5 F!!

1E0 Z0 F!!

1E0 XIN2 F@@ F- XIN3 F@@ F- XIN4 F@@ F- XIN5 F@@ F- XIN1 F!!

Z0 F@@ GR F@@ F*

8.31451E-3 F* TGR F@@ F*

PGR F@@ F/ DOAIR F@@ F*

MR F!!

MR F@@ XIN2 F@@ 28.01350E0 F* F-

XIN3 F@@ 44.010E0 F* F-

XIN4 F@@ 2.0159E0 F* F-

XIN5 F@@ 28.01E0 F* F- XIN1 F@@ F/

MW1 F!!

MW1 F@@ G1 F@@ F- G2 F@@ F/

HCH F!!

-0.425468E0

TGR F@@

0.286500E-2 -0.462073E-5

TGR F@@ F* F+ F* F+

0.877118E-3

TGR F@@

-0.556281E-5 0.881510E-8

TGR F@@ F* F+ F* F+

HCH F@@ F* F+

-0.824747E-6

TGR F@@

0.431436E-8 -0.608319E-11

TGR F@@ F* F+ F* F+

HCH F@@ 2E0 F** F* F+

BCH F!!

TGR F@@ TEM F!!

1 OPT !!

VIRGS

BMIX F@@ PGR F@@ F*

8.31451E-3 F/ TGR F@@ F/ 1E0 F+

ZONEW F!!

BEGIN

Z0 F@@ ZONEW F@@ F/ 1E0 F-

FABS 0.5E-10 F>

WHILE

ZONEW F@@ Z0 F!!

Z0 F@@ GR F@@ F*

8.31451E-3 F* TGR F@@ F*

PGR F@@ F/ DOAIR F@@ F*

MR F!!

MR F@@ XIN2 F@@ 28.01350E0 F* F-

XIN3 F@@ 44.010E0 F* F-

XIN4 F@@ 2.0159E0 F* F-

XIN5 F@@ 28.01E0 F* F- XIN1 F@@ F/

MW1 F!!

MW1 F@@ G1 F@@ F- G2 F@@ F/

HCH F!!

-0.425468E0

TGR F@@

0.286500E-2 -0.462073E-5

TGR F@@ F* F+ F* F+

0.877118E-3

TGR F@@

-0.556281E-5 0.881510E-8

TGR F@@ F* F+ F* F+

HCH F@@ F* F+

-0.824747E-6

TGR F@@

0.431436E-8 -0.608319E-11

TGR F@@ F* F+ F* F+

HCH F@@ 2E0 F** F* F+
BCH F!!

TGR F@@ TEM F!!
1 OPT !!
VIRGS

BMIX F@@ PGR F@@ F*
8.31451E-3 F/ TGR F@@ F/ 1E0 F+
ZONEW F!!

REPEAT

ERRNUM @@ 0=
NOT IF
1 FLAKC !!
THEN

FLAKC @@ 0= IF

-0.425468E0
0.877118E-3
-0.824747E-6
HCH F@@ F* F+
HCH F@@ F* F+
B011 F!!

0.286500E-2
-0.556281E-5
0.431436E-8
HCH F@@ F* F+
HCH F@@ F* F+
B111 F!!

-0.462073E-5
0.881510E-8
-0.608319E-11
HCH F@@ F* F+
HCH F@@ F* F+
B211 F!!

0.646422E-3
0.332805E-6 HCH F@@ F* F-
HCH F@@ F*
-0.302488E0 F+
C011 F!!

-0.422876E-5
0.223160E-8 HCH F@@ F* F+
HCH F@@ F*
0.195861E-2 F+
C111 F!!

0.688157E-8

0.367713E-11 HCH F@@ F* F-
HCH F@@ F*
-0.316302E-5 F+
C211 F!!

XIN1 F@@ XX1 F!!
XIN2 F@@ XX2 F!!
XIN3 F@@ XX3 F!!
XIN4 F@@ XX4 F!!
XIN5 F@@ XX5 F!!

263E0 TLOW F!!
3.38E0 THIGH F!!
0.5E-9 PLOW F!!
12E0 PHIGH F!!
8E0 DHIGH F!!

PGR F@@ PRE F!!
TGR F@@ TEM F!!
DGROS
DGROSS F@@ DB F!!

GR F@@ DOAIR F@@ F*
DB F@@ F/
MWX F!!

60E0 459.67E0 F+ 1.8E0 F/ TB F!!
14.73E0 6894.757E0 F* 1000000E0 F/ PB F!!

PB F@@ PRE F!!
TB F@@ TEM F!!
DGROS
DGROSS F@@ DB F!!

DB F@@ DEN F!!
TB F@@ TEM F!!
ZGROS
ZGROSS F@@ ZB F!!

THEN

;

PGROSS.TDS:

: PGROS

ZGROS

ZGROSS F@@ TEM F@@ F*
8.31451E-3 F*
DEN F@@ F*
PGROSS F!!
;

DGROSS.TDS:

```
EVAR FLAK

: DGROS

0 FLAK !! \ Bandera de RETURN

1E-6 TOL F!!

OE0 DEN F!!
PGROS
PGROSS F@@ X1 F!!

DHIGH F@@ DEN F!!
PGROS
PGROSS F@@ X2 F!!

X1 F@@ DEN F!!
PGROS
PGROSS F@@ PRE F@@ F-
F1 F!!

\ *****

F1 F@@ FABS TOL F@@ F<
IF
    X1 F@@ DGROSS F!!
    1 FLAK !!
THEN

    FLAK @@ 0 = IF \ ***I1

X2 F@@ DEN F!!
PGROS
PGROSS F@@ PRE F@@ F-
F2 F!!

F2 F@@ FABS TOL F@@ F<
IF
    X2 F@@ DGROSS F!!
    1 FLAK !!
THEN

    THEN \ ***01

    FLAK @@ 0 = IF \ ***I2

F1 F@@ F2 F@@ F* F0<
NOT IF
    ." DGROSS RAIZ NO ENCONTRADA "
    OE0 DGROSS F!!
    1 FLAK !!
THEN
    THEN \ ***02
```

```

FLAK @@ 0 = IF      \   ***I3

101 1 DO

    FLAK @@ 0 = IF      \   ***I4

0 DFLAG !!

F2 F@@ X1 F@@ F*
F1 F@@ X2 F@@ F* F-
F2 F@@ F1 F@@ F- F/
X3 F!!

X3 F@@ DEN F!!
PGROS
PGROSS F@@ PRE F@@ F-
F3 F!!

    THEN      \   ***O4

    FLAK @@ 0 = IF      \   ***I5

I 6 MOD 0=
IF
    X1 F@@ X2 F@@ F+ 2E0 F/
    DGROSS F!!
ELSE
    F1 F@@ F2 F@@ F-
    F1 F@@ F3 F@@ F- F*
    F2 F@@ F3 F@@ F- F* F0=
    IF
        1 FLAK !!
    THEN

        FLAK @@ 0 = IF      \   ***I6

        X1 F@@ F2 F@@ F* F3 F@@ F*
        F1 F@@ F2 F@@ F-
        F1 F@@ F3 F@@ F- F* F/

        X2 F@@ F1 F@@ F* F3 F@@ F*
        F2 F@@ F1 F@@ F-
        F2 F@@ F3 F@@ F- F* F/  F+

        X3 F@@ F1 F@@ F* F2 F@@ F*
        F3 F@@ F1 F@@ F-
        F3 F@@ F2 F@@ F- F* F/ F+
        DGROSS F!!

        DGROSS F@@ X1 F@@ F-
        DGROSS F@@ X2 F@@ F- F*
        F0< NOT IF
            X1 F@@ X2 F@@ F+ 2E0 F/
            DGROSS F!!
        THEN

```



```

      THEN          \ ***O6
THEN

      THEN \ ***O5

      FLAK @@ 0 = IF \ ***I7

DGROSS F@@ DEN F!!
PGROS
PGROSS F@@ PRE F@@ F-
FFF F!!

FFF F@@ FABS TOL F@@ F<
IF
  1 FLAK !!
THEN

      THEN \ ***O7
      FLAK @@ 0 = IF \ ***I8

FFF F@@ F3 F@@ F* F0<
IF
  1 DFLAG !!
  DGROSS F@@ X1 F!!
  FFF F@@ F1 F!!
  X3 F@@ X2 F!!
  F3 F@@ F2 F!!
THEN

F3 F@@ F1 F@@ F* 0E0 F>
IF
  1 DFLAG !!
  DGROSS F@@ X1 F!!
  FFF F@@ F1 F!!
THEN

DFLAG @@ 0=
IF
DGROSS F@@ X2 F!!
FFF F@@ F2 F!!
THEN
  THEN \ ***O8
LOOP

  THEN \ ***O3

FLAK @@ 0 = IF
0E0 DGROSS F!! ." DGROS NO CONVERGE"
THEN

;

```

VIRGS.TDS:

```

EVAR FLAKV

```

: VIRGS

0 FLAKV !!
0 ERRNUM !!

TOLD F@@ TEM F@@ F=
IF
BBMIX F@@ BMIX F!!
CCMIX F@@ CMIX F!!
1 FLAKV !!
THEN

FLAKV @@ 0 = IF

XIN1 F@@ XIN1 F@@ F* X11 F!!
XIN2 F@@ XIN2 F@@ F* X22 F!!
XIN3 F@@ XIN3 F@@ F* X33 F!!
XIN4 F@@ XIN4 F@@ F* X44 F!!
XIN5 F@@ XIN5 F@@ F* X55 F!!

OPT @@ 0=
IF
B111 F@@ B211 F@@
TEM F@@ F* F+ TEM F@@ F*
B011 F@@ F+
BCH F!!
THEN

0.740910E-3 -0.911950E-6
TEM F@@ F* F+ TEM F@@ F*
-0.144600E0 F+
BN2 F!!

0.403760E-2 -0.516570E-5
TEM F@@ F* F+ TEM F@@ F*
-0.868340E0 F+
BCO2 F!!

0.813385E-4 -0.987220E-7
TEM F@@ F* F+ TEM F@@ F*
-0.110596E-2 F+
BH2 F!!

0.602540E-3 -0.644300E-6
TEM F@@ F* F+ TEM F@@ F*
-0.130820E0 F+
BCO F!!

THEN

BCO2 F@@ BCH F@@ F* F0<
IF
." RAIZ NEGATIVA "
1 ERRNUM !!
1 FLAKV !!
THEN

```

FLAKV @@ 0 = IF

320E0 TEM F@@ F-
320E0 TEM F@@ F- F*
1.875E-5 F* 0.72E0 F+
BN2 F@@ BCH F@@ F+ F*
2E0 F/
B12 F!!

BCO2 F@@ BCH F@@ F* FSQRT
-0.865E0 F* B13 F!!

-0.250000E-6 TEM F@@ F*
0.271570E-3 F+ TEM F@@ F*
-0.521280E-1 F+ B14 F!!

0.518195E-6 TEM F@@ F*
-0.239381E-5 F+ TEM F@@ F*
-0.687290E-1 F+ B15 F!!

-0.204429E-5 TEM F@@ F*
0.161176E-2 F+ TEM F@@ F*
-0.339693E0 F+ B23 F!!

0.012E0 B24 F!!

X11 F@@ BCH F@@ F*
X22 F@@ BN2 F@@ F* F+
X33 F@@ BCO2 F@@ F* F+
X44 F@@ BH2 F@@ F* F+
X55 F@@ BCO F@@ F* F+

XIN2 F@@ XIN1 F@@ F*
B12 F@@ F* 2E0 F* F+

XIN3 F@@ XIN1 F@@ F*
B13 F@@ F* 2E0 F* F+

XIN4 F@@ XIN1 F@@ F*
B14 F@@ F* 2E0 F* F+

XIN5 F@@ XIN1 F@@ F*
B15 F@@ F* 2E0 F* F+

XIN3 F@@ XIN2 F@@ F*
B23 F@@ F* 2E0 F* F+

XIN4 F@@ XIN2 F@@ F*
B24 F@@ F* 2E0 F* F+

BMIX F!!

THEN

OPT @@ 1 =

```

```

IF
1 FLAKV !!
THEN

    FLAKV @@ 0 = IF

TEM F@@ TOLD F!!
BMIX F@@ BBMIX F!!

TEM F@@ 270E0 F-
0.0013E0 F*
0.92E0 F+ EEE F!!

1E0 3E0 F/ FFF F!!

TEM F@@ C211 F@@ F*
C111 F@@ F+ TEM F@@ F*
C011 F@@ F+
C11 F!!

TEM F@@ 0.611870E-7 F*
-0.398950E-4 F+ TEM F@@ F*
0.784980E-2 F+
C22 F!!

TEM F@@ -0.837030E-7 F*
0.348880E-4 F+ TEM F@@ F*
0.205130E-2 F+
C33 F!!

TEM F@@ 0.467095E-8 F*
-0.364887E-5 F+ TEM F@@ F*
0.104711E-2 F+
C44 F!!

    THEN

C11 F@@ F0<
IF
." TERMINO INVALIDO EN VIRGS "
1 ERRNUM !!
1 FLAKV !!
THEN

C33 F@@ F0<
IF
." TERMINO INVALIDO EN VIRGS "
1 ERRNUM !!
1 FLAKV !!
THEN

    FLAKV @@ 0 = IF

TEM F@@ 0.343051E-7 F*
-0.276578E-4 F+ TEM F@@ F*
0.736748E-2 F+ 3E0 F*
C15 F!!

```

```

TEM F@@ 0.157169E-7 F*
-0.168609E-4 F+ TEM F@@ F*
0.552066E-2 F+ 3E0 F*
C23 F!!

```

```

TEM F@@ -0.325798E-7 F*
0.806674E-5 F+ TEM F@@ F*
0.358783E-2 F+ 3E0 F*
C32 F!!

```

```

C11 F@@ X11 F@@ F* XIN1 F@@ F*
C22 F@@ X22 F@@ F* XIN2 F@@ F* F+
C33 F@@ X33 F@@ F* XIN3 F@@ F* F+
C44 F@@ X44 F@@ F* XIN4 F@@ F* F+
C23 F@@ X22 F@@ F* XIN3 F@@ F* F+
C32 F@@ X33 F@@ F* XIN2 F@@ F* F+
C15 F@@ X11 F@@ F* XIN5 F@@ F* F+

```

```

C11 F@@ C11 F@@ F* C22 F@@ F* FFF F@@ F**
XIN2 F@@ F* X11 F@@ F* EEE F@@ F* 3E0 F* F+

```

```

C11 F@@ C22 F@@ F* C22 F@@ F* FFF F@@ F**
XIN1 F@@ F* X22 F@@ F* EEE F@@ F* 3E0 F* F+

```

```

C11 F@@ C11 F@@ F* C33 F@@ F* FFF F@@ F**
XIN3 F@@ F* X11 F@@ F* 0.92E0 F* 3E0 F* F+

```

```

C11 F@@ C33 F@@ F* C33 F@@ F* FFF F@@ F**
XIN1 F@@ F* X33 F@@ F* 0.92E0 F* 3E0 F* F+

```

```

C11 F@@ C11 F@@ F* C44 F@@ F* FFF F@@ F**
XIN4 F@@ F* X11 F@@ F* 1.2E0 F* 3E0 F* F+

```

```

C11 F@@ C22 F@@ F* C33 F@@ F* FFF F@@ F**
XIN1 F@@ F* XIN2 F@@ F* XIN3 F@@ F* 1.1E0 F* 6E0 F* F+

```

```

CMIX F!!

```

```

CMIX F@@ CCMIX F!!

```

```

      THEN
;

```

ZGROSS.TDS:

```

EVAR FLAKZ

```

```

: ZGROS

```

```

0 FLAKZ !!
0E0 ZGROSS F!!
0 OPT !!
VIRGS

```

```
ERRNUM @@ 0<> IF
      1 FLAKZ !!
THEN

FLAKZ @@ 0= IF

DEN F@@ DEN F@@ F*
CMIX F@@ F*
DEN F@@
BMIX F@@ F* F+
1E0 F+
ZGROSS F!!

THEN
;
```

ANEXO C. SWITCHED-CAPACITOR VOLTAGE CONVERTERS

- _ **Output Current . . . 100 mA**
- _ **Low Loss . . . 1.1 V at 100 mA**
- _ **Operating Range . . . 3.5 V to 15 V**
- _ **Reference and Error Amplifier for Regulation**
- _ **External Shutdown**
- _ **External Oscillator Synchronization**
- _ **Devices Can Be Paralleled**
- _ **Pin-to-Pin Compatible With the LTC1044/7660**

Description

The LT1054 is a bipolar, switched-capacitor voltage converter with regulator. It provides higher output current and significantly lower voltage losses than previously available converters. An adaptive-switch drive scheme optimizes efficiency over a wide range of output currents. Total voltage drop at 100-mA output current is typically 1.1 V. This holds true over the full supply-voltage range of 3.5 V to 15 V. Quiescent current is typically 2.5 mA. The LT1054 also provides regulation, a feature not previously available in switched-capacitor voltage converters. By adding an external resistive divider, a regulated output can be obtained. This output is regulated against changes in both input voltage and output current. The LT1054 also can be shut down by grounding the feedback terminal. Supply current in shutdown is typically 100 mA. The internal oscillator of the LT1054 runs at a nominal frequency of 25 kHz. The oscillator terminal can be used to adjust the switching frequency or to externally synchronize the LT1054. The LT1054C is characterized for operation over a free-air temperature range of 0°C to 70°C. The LT1054I is characterized for operation over a free-air temperature range of -40°C to 85°C.

ANEXO D. PRECISION VOLTAGE REFERENCE

FEATURES

- ❖ +10V ± 0.0025 V OUTPUT
- ❖ VERY LOW DRIFT: 2.5ppm/°C max
- ❖ EXCELLENT STABILITY:
- ❖ 5ppm/1000hr typ
- ❖ EXCELLENT LINE REGULATION:
- ❖ 1ppm/V max
- ❖ EXCELLENT LOAD REGULATION:
- ❖ 10ppm/mA max
- ❖ LOW NOISE: 5mVp-p typ, 0.1Hz to 10Hz
- ❖ WIDE SUPPLY RANGE: 11.4VDC to 36VDC
- ❖ LOW QUIESCENT CURRENT: 1.4mA max
- ❖ PACKAGE OPTIONS: HERMETIC TO-99,
- ❖ PLASTIC DIP, SOIC

APPLICATIONS

- ❖ PRECISION-CALIBRATED VOLTAGE
- ❖ STANDARD
- ❖ D/A AND A/D CONVERTER REFERENCE
- ❖ PRECISION CURRENT REFERENCE
- ❖ ACCURATE COMPARATOR THRESHOLD
- ❖ REFERENCE
- ❖ DIGITAL VOLTMETERS
- ❖ TEST EQUIPMENT
- ❖ PC-BASED INSTRUMENTATION

DESCRIPTION

The REF102 is a precision 10V voltage reference. The drift is laser-trimmed to 2.5ppm/°C max (CM grade) over the industrial temperature range and 5ppm/°C max (SM grade) over the military temperature range. The REF102 achieves its precision without a heater. This results in low power, fast warm-up, excellent stability, and low noise. The output voltage is extremely insensitive to both line and load variations and can be externally adjusted with minimal effect on drift and stability. Single supply operation from 11.4V to 36V and excellent overall specifications make the REF102 an ideal choice for demanding instrumentation and system reference applications.

ANEXO E. SISTEMA EMBEBIDO TDS2020F

A continuación se detallan las características del TDS2020F y los principios de su lenguaje de programación llamado *Forth*, el cual siendo óptimo para sistemas embebidos basados en microcontrolador o microprocesador, solo es conocido por dedicados programadores de hardware y no es tan popular en el ámbito académico.

1. Sistema Embebido TDS2020F

El TDS2020F es un poderoso sistema embebido o microcomputador de control de 16 bits. A pesar de su pequeño tamaño y su bajo consumo de potencia, está equipado con importantes características que facilitan su uso para resolver problemas de control y recopilación de datos.

Gracias a su naturaleza interactiva de desarrollo y depuración (encontrar y remover errores del programa), se hace fácil la implementación de aplicaciones. Su librería de software provee soluciones instantáneas a muchos problemas de aplicación. El código fuente, escrito en el lenguaje de alto nivel *Forth*, se compila directamente en una memoria no volátil Flash-EEPROM.

2. Memoria “flash”

La compilación directa en una memoria Flash-EEPROM es conveniente y evita el costo y el tiempo de desarrollo en un programador de memoria PROM. Hay muchas otras ventajas, como que se puede borrar el programa remotamente por medio de un módem y recompilarlo. Por ejemplo, si el TDS2020F estuviera embebido en un vehículo o en tubo presurizado, se podría actualizar el programa sin necesidad de extraer la tarjeta.

3. Lenguaje *Forth*

La razón de programar en *Forth* es que para sistemas embebidos y microprocesadores, se obtienen resultados más rápidos y eficientes (menos líneas de código) con este lenguaje. Para la máquina es un tanto de más bajo nivel que 'C' pero es un lenguaje de más alto nivel que *assembler*. A diferencia de otros, es interactivo, ofreciendo un rápido desarrollo. El *Forth* multitarea, escrito especialmente para la tarjeta, da acceso fácil a todas sus características y permite que el *software* sea escrito rápidamente. El lenguaje es el mismo *Forth* Estándar Nacional Americano con muchas extensiones para explotar las características de *hardware* del TDS2020F. El *Forth* de 16 kbytes incluye controladores de teclado y

LCD, junto con muchas otras utilidades y un *assembler* totalmente simbólico. De esta forma, se pueden escribir los programas en un lenguaje de alto nivel, mezclándolo con *assembler* si se requiere.

4. Hardware

El TDS2020F está basado en el microprocesador Hitachi H8/532. Tiene 45k bytes de espacio para el programa compilado y más de 512k bytes de memoria "Flash" o RAM respaldada con batería para mantener los datos vitales mientras la tarjeta no está trabajando. Esto se puede expandir a más de un gigabyte con tarjetas PCMCIA o tarjetas "flash".

Un adaptador de red de área controlada (bus CAN) permite la interconexión rápida de los computadores TDS2020F, con un PC en la red si se requiere. También está provisto con los protocolos CAN de alto y bajo nivel ampliamente usados.

La tarjeta tiene entre 26 y 41 E/Ss paralelas dependiendo de la configuración y tiene dos puertos seriales RS232. Posee un conversor análogo/digital de 8 canales y 10 bits de resolución y contiene tres canales de conversión digital/análogo de 8 bits. Naturalmente se pueden conectar conversores externos de mayor resolución.

Las características adicionales incluyen cuatro contador-temporizadores en hardware (tres son de 16 bits), dos temporizadores separados de "watch-dog" (perro-guardián), reloj de tiempo del día, no volátil y multitarea. Necesita de una fuente de alimentación sencilla con 32mA de salida y además consume solo 155µA en un modo operacional de baja potencia, para aplicaciones como almacenamiento de datos.

El TDS2020F mide solo 10 x 8 cm, tiene pines para conectar con cable plano; o simplemente se puede usar como un componente insertado en una tarjeta más grande. Otra versión tiene un conector DIN41612 para usarlo en un "rack".

5. Usando Forth

Cuando se alcanza finalmente la etapa de manufactura de un producto basado en TDS2020F no es el final de *Forth*. Se puede usar para pruebas finales, reparación y mantenimiento, ya que el lenguaje está en la tarjeta.

A través de un conector que de acceso serial al computador *Forth* en el instrumento, se gana acceso al sistema de lenguaje interno con un PC o un terminal portátil. Se pueden tener disponibles todas las funciones que constituyen la aplicación.

Por ejemplo, en un termómetro electrónico, la rutina análogo/digital se puede ejercitar separadamente para ver si la falla está en la entrada del transductor o en el amplificador. Si no, se puede seguir con una prueba al LCD escribiendo unos pocos caracteres en el. Asimismo, se puede probar el teclado presionando una tecla para ver si genera el código esperado.

Si es necesario, la totalidad del código compilado se puede borrar de la Flash-EEPROM. Entonces se puede recompilar un nuevo código fuente, aún remotamente sobre un módem (o en una localización inaccesible).

Forth en la tarjeta es muy útil durante el diseño, pero la capacidad de acceder a un procedimiento de software individual en un producto terminado es única e invaluable.

6. Velocidad

La frecuencia de reloj del TDS2020F es 19.6608MHz y el tiempo de un ciclo es 102ns (nanosegundos). Las instrucciones se procesan en un promedio de alrededor de 3 millones por segundo (MIPS). La latencia de interrupción es solo de 2.3µs (microsegundos), más el tiempo para completar la instrucción actual.

7. Memoria

Dentro del Microprocesador H8/532: 32kbyte de PROM OTP (programable una vez), que lleva 16k de lenguaje de alto nivel *Forth* y 16k de espacio en blanco para programas adicionales de aplicación para el usuario. Además, contiene 1kbyte de RAM, de la cual la mitad se necesita para el sistema dejando 512 bytes libres para ser usados como variables para el programa de aplicación.

“Socket” de 28 pines: En este “socket” de la tarjeta se debe insertar una memoria Flash-EEPROM de 32kbyte no volátil para el programa final del aplicación del usuario. Debido a que alguna parte del mapa de memoria se usa para los registros en el chip y E/S (entrada-salida), solo 29kbytes están realmente disponibles. Una vez se ha finalizado el software, el “socket” puede soportar una EPROM 27C256 si se desea.

“Socket” de 32 pines: En este “socket” de la tarjeta se pueden insertar 512k, 128k o 32k de RAM, EEPROM o memoria Flash para variables adicionales, memoria auxiliar o almacenamiento de datos. Las memorias EEPROM y Flash conservan los datos en ausencia de potencia; mientras las RAM pueden hacerse no volátiles añadiendo una pequeña tarjeta de batería o una celda externa de litio. Las señales necesarias para expansión están disponibles en la tarjeta para extender indefinidamente la memoria fuera de la tarjeta.

Chip de reloj: El chip de reloj en la tarjeta tiene 239 bytes de RAM libre que son accesibles a través del bus I²C. Con batería externa, esta RAM (también como el reloj de tiempo real en cualquier RAM en el “socket” de 32 pines) se mantendrá en ausencia de potencia.

8. Entrada salida paralela

La capacidad del puerto paralelo del TDS2020F depende si usan o no el conversor AD y otras facilidades.

Los pines se pueden agrupar para diferentes funciones, así que el número de entradas y salidas paralelas libres depende de si se necesita conversión analógico/digital, reloj del tiempo del día no volátil, etc. A partir de esta tabla se puede verificar cuantos puertos de reserva están disponibles. En resumen, el número de bits de entrada o salida paralela está entre 26 y 41 dependiendo de las facilidades usadas.

9. Bus I²C

Este es un sistema de dos cables que se usa con periféricos de bajo costo tales como conversores A/D, relojes, E/S, RAMs, EEPROMs, etc comercializados por Philips, Signetics, Xicor, Microchip Technology y otros. La tarjeta tiene un chip de reloj PCF8583 y RAM conectados a este bus. Otros chips I²C se pueden añadir externamente.

10. Entrada salida serial

En la tarjeta hay dos controladores y receptores serial que usan formato y voltajes reales RS232. En el Puerto 1 se soportan todas las tasas de baudios de 75 a 38.4k y hay un MIDI adicional de 31.25k baudios. El Puerto 2 se puede usar a todas las tasas por encima de 4800 baudios. Aunque el sistema se alimenta con una fuente de alimentación sencilla de +6 a +16V, los puertos serie usan $\pm 8V$ generados internamente en la tarjeta para dar los niveles lógicos de salida que cumplan con las especificaciones RS232.

11. Microprocesador

El microprocesador H8/532, es un dispositivo de 16 bits con ocho registros de propósito general de 16 bits, hardware de multiplicación y división de 16 bits, e instrucciones de 8 y 16 bits. Las manipulaciones directas a bit operan con registros, toda la memoria y entrada-salida (E/S). La frecuencia del cristal es de 19.6608MHz y el dispositivo corre en el modo expandido máximo con un espacio de memoria de 1024k bytes.

El *hardware* que posee el chip del H8/532, incluye tres contadores o temporizadores de 16 bits, un contador/temporizador de 8 bits, puerto serial síncrono y asíncrono, y un sistema de interrupciones muy versátil (65 interrupciones y vectores DTC, 8 niveles de prioridad). El Controlador de Transferencia de Datos (DTC) habilita la transferencia entre E/S y memoria sin necesidad de software.

Todas las capacidades del procesador están disponibles al usuario del TDS2020F, pero a través del lenguaje *Forth* de alto nivel, que las hace más fáciles de utilizar. El acceso a algunas facilidades de *hardware* del microprocesador ya está construido en el sistema *Forth* pero el usuario es libre de usarlos también a través de *assembler*.

12. Soporte de teclado

El sistema de *software* y *hardware* permite que se conecten más de 64 teclas al sistema embebido, empleando solo ocho de las líneas de E/S paralelas. Se conecta un puerto particular de salida paralelo a un lado de una matriz de teclas 8 x 8 por medio de diodos. El otro lado regresa por medio de otros ocho diodos al bus de datos.

13. Pantalla de cristal líquido (LCD)

Cualquier LCD alfanumérico basado en el chip HD44780 se conecta directamente al TDS2020F; pues el *software* para controlarlo está construido dentro de la tarjeta. Un chip externo habilita más de 8 LCDs para ser conectados y el *software* los sirve a ellos simultáneamente.

14. Periféricos externos

Los buses de direcciones y datos están provistos en su totalidad, así como cinco direcciones decodificadas libres para su uso en la selección de periféricos del chip. Se pueden por ejemplo añadir LCDs, E/S extras paralelas, un chip de cuatro puertos seriales o dispositivos similares sin ningún chip de interfase.

Cada decodificador cubre un intervalo de memoria de 16 posiciones. Tres no están sincronizados con la señal de reloj y son capaces de manejar muchos periféricos como "*latches*" octales (sujetadores de ocho canales) o LCDs gráficos. Los otros dos están sincronizados con la señal de reloj. Uno es activo negativo, el otro es activo positivo. Este último es exactamente apropiado para entradas a displays LCDs alfanuméricos.

15. Conversor análogo a digital

El TDS2020F tiene un conversor A/D de ocho canales y 10 bits que da una parte en 1024 de resolución. El tiempo de conversión es de 21 μ s usando un programa abastecido en una librería. Un voltaje de entrada de 0V da 0 y uno de +5V da 1023. La referencia en la tarjeta es 5.00 \pm 0.05V y hay facilidad para conexión de una referencia externa. Si no se usa el A/D los ocho pines se convierten automáticamente en puertos de entrada paralelos.

16. Conversor digital a análogo

Hay tres canales de conversión D/A modulada con ancho de pulso (PWM) que actúan como tres bits extra de E/S paralela cuando no se usan. Se debe añadir un filtro externo en la salida (tal como un resistor de 10k y un capacitor de 1 μ F) para obtener el nivel análogo deseado.

17. Tiempo y fecha no volátiles

El TDS2020F tiene un chip de reloj PCF8583 que mantiene la fecha y el tiempo aún con el computador apagado si se alimenta con batería externa. La batería también se usa para mantener más de 512k bytes de RAM estática en la tarjeta. El consumo típico es de 3.5µA, dando a la batería un tiempo de vida limitado más por los 5 años de vida de fábrica, que por la capacidad de 950mA-hora propia de la batería.

La fecha y el tiempo son accesibles a una resolución de 813 nanosegundos debido a que otro reloj que corre bajo interrupciones refleja el reloj del chip. Se soportan los formatos de fecha Europeo y Americano.

El chip de reloj genera una interrupción no enmascarable (NMI) de un segundo. Por defecto no hace nada, pero se puede redireccionar para ejecutar una rutina de usuario en *Forth* de alto nivel o *Assembler*. El tiempo de repetición se puede cambiar en pasos muy pequeños para dar intervalos de 30ms (milisegundos) a un año.

18. Temporizador-contadores

Hay cuatro temporizadores de hardware (más el chip de reloj); uno es de 8 bits y los otros tres de 16 bits. Cada uno puede ser cronometrado internamente a cualquiera de tres tasas, o externamente para conteo de eventos. Todos ellos tienen dos registros de captura de salida y los tres temporizadores de 16 bits tienen un modo de captura de entrada. Múltiples interrupciones están asociadas con los temporizadores para procesar eventos asíncronos en *Forth* de alto nivel o *assembler*.

19. Temporizadores de “watchdog”

Si el microprocesador se cae por un pico o rizo de potencia o cualquier otra situación, uno de los temporizadores de “watchdog” (perro guardián) restaurará el sistema. El programa de aplicación se iniciará de nuevo. Hay un “watchdog” interno (por defecto 106ms) y un contador externo al microprocesador (426ms). Ambos se restauran por muchas palabras de *Forth* y normalmente no se notará su acción, pero cronometrarán y reiniciarán el sistema si algo va mal. Al reiniciar, se puede entrar una palabra de *Forth* definida por el usuario o aceptar el *reset* por defecto.

20. Modo de baja potencia

Se puede retirar la alimentación al controlador RS232 y poner el microprocesador en el modo de *software “standby”*. El consumo de corriente baja típicamente a 155µA más algunas cargas externas. Cuando una interrupción no enmascarada ocurra el computador despertará, encendiendo el puerto serie y continuando la ejecución del programa. Esto hace al TDS2020F ideal para aplicaciones portátiles y de recolección de datos.

21. Fuente de alimentación

Hace uso de una fuente sencilla de +6V a 16V y consume una corriente típica de 32mA. El modo operacional de baja potencia consume solo 155 μ A y no necesita soporte externo de *hardware*. La tarjeta tiene un generador de ± 8 V para ser usado por los puertos serie y éste se puede apagar por *software* para ahorrar potencia. La alimentación negativa también es útil para algunos periféricos externos. El regulador en la tarjeta puede dar más de 100mA, así que hay corriente extra disponible para alimentar cualquier circuitería externa particular a la aplicación.

22. Interfase de bus

El bus TDS2020F es compatible con casi cualquier chip periférico tales como conversores A/D y múltiples dispositivos de puerto serie. El bus de datos no está multiplexado. En la práctica, se limita la longitud de los buses de direcciones y datos fuera de la tarjeta a 300mm, a menos que se extienda con la circuitería sugerida.

23. Dimensiones

El tamaño de la tarjeta es 100mm x 80mm con huecos de montaje para tornillos de 2.5mm. La máxima altura, excluyendo los conectores para pines es de 15mm. Si se instala una tarjeta de aplicación sobre los pines (modo sándwich), la distancia entre el sistema embebido y la tarjeta de aplicación es de 11 ± 1 mm. Esta debería ser la longitud de los espaciadores si se usan. La temperatura de operación del sistema es de -10 a $+70^{\circ}\text{C}$. Todos los pines de conexiones coinciden con una matriz de 0.1 pulgadas, de modo que la tarjeta se puede montar fácilmente en una tarjeta prototipo con una matriz de huecos si se requiere. También hay un botón azul de *reset* en la esquina de la tarjeta.

24. Sistema Forth ANS

El lenguaje es una completa implementación del *Forth* Estándar Nacional Americano que incluye punto flotante. Hay muchas extensiones útiles para creadores de sistemas con única tarjeta. Por ejemplo, el número de microsegundos tomados por cualquier palabra *Forth* se pueden medir con exactitud en tiempo real. El lenguaje ha sido implementado específicamente para el procesador H8/532. Este usa las facilidades del chip donde sea posible; por ejemplo, las multiplicaciones y divisiones de *Forth* están construidas a partir de las instrucciones de hardware multiplicar/dividir de 16 bits. Sus instrucciones de manipulación de bit también se emplean, ellas son útiles especialmente para detectar y conmutar líneas sencillas de E/S.

Todos los registros *Forth* están implementados en registros de microprocesador y la pila de parámetros es la misma pila de hardware de las máquinas. Las dos pilas de *Forth* y las variables de usuario están dentro de la memoria del chip del microprocesador para una velocidad de acceso máxima. Dos registros de máquina

se dejan libres (usualmente), de modo que puedan ser usados por interrupciones rápidas sin necesidad de almacenarlos.

El código de aplicación se compila en una Flash-EEPROM no volátil para crear instantáneamente un sistema autónomo. Aparte de *Forth*, el sistema ROM tiene un *assembler* simbólico (lenguaje de máquina) que habilita al usuario para escribir código de máquina directamente en la tarjeta. No se requiere tanto trabajo con *assembler*, ya que una rutina en *assembler* se puede probar inmediatamente sin necesidad de descargar un archivo binario. La depuración interactiva del código de *assembler* es muy poderosa.

25. Extensiones ROM *Forth*.

Entre las extensiones incluidas en la ROM *Forth* están:

- Exploración de teclado.
- Displays de cristal líquido alfanuméricos.
- Soporte de periféricos de bus I²C que incluye RAM y reloj.
- Interrupciones escritas en código *assembler* o *Forth*.
- Multitarea.
- Reloj del tiempo del día.
- Servicio de temporizador "*watchdog*".
- Operación a baja potencia.
- Assembler* simbólico completo.
- E/S serial vectorizada y muchas otras palabras vectorizadas.
- Caracteres Hex, ASCII y de control.
- Aritmética de número doble (32-bit) y manipulación de pila.
- Medida de tiempo de ejecución exacta a un 1µs.

26. Multitarea.

El desarrollo de impresiones, programas de visualización y comunicaciones, como entidades separadas que usan el sistema multitarea, pueden correr todas al mismo tiempo en el TDS2020F.

La depuración de cada tarea se puede hacer independientemente de las otras y así se permite la corrida de un subprograma solo tecleando su nombre. La etapa final es combinar los programas individuales y esto es esencialmente un paso mecánico.

27. Software de desarrollo (*firmware*).

El ambiente de desarrollo *TDS-PC* para Windows, es el software de PC con facilidades para emulación del terminal y edición de múltiples ventanas. El código fuente se almacena en el disco del PC, aunque la compilación toma lugar dentro del TDS2020F. Una tecla inicia el proceso automático que envía el programa fuente del puerto serie del PC al TDS2020F para su compilación allí. Luego se depura el código

interactivamente en el TDS2020F. *TDS-PC* para Windows también corre en emuladores de PC Macintosh.

El código fuente puede ser escrito con muchos comentarios y estructuras correctamente separadas del margen (con sangría). Teclas especiales invocan la compilación y otras características. Hay una facilidad de incluir (INCLUDE), de modo que los archivos puedan ser anidados. Hay una interfase al editor usual de texto del código fuente, de modo que la escritura sea más fácil.

Una vez el programa está desarrollado y trabajando, no se necesita nada más para la configuración de desarrollo estándar. No se necesita un programador PROM porque el código se ha compilado en una Flash-EEPROM no volátil.

ANEXO F. CODIGO DEL FIRMWARE FINAL EN FORTH

LOG111.TDS: PROGRAMA PRINCIPAL

```
INCLUDE #7825P-A.TDS    \ Manejo del conversor ADS7825  (driver)
INCLUDE FPA.TDS         \ Rutina de Punto Flotante 1
INCLUDE FPAA2.TDS       \ Rutina de Punto Flotante 2
INCLUDE KEYF.TDS        \ Manejo de teclado 1

  HEX    \ Ubica en memoria las direcciones para los vectores de almacenamiento
B0BB8. 2CONSTANT PRES-W    \ Vector para Presiones promedio del minuto
9164A. 2CONSTANT TEMP-W    \ Vector para Temperaturas promedio del minuto
920DC. 2CONSTANT TEMP-HR   \ Vector para Temperatura promedio de la hora
B3CFC. 2CONSTANT PRES-HR   \ Vector para Presiones promedio de la hora.
8591C. 2CONSTANT PRES-DAY
85A48. 2CONSTANT TEMP-DAY
85B74. 2CONSTANT ACQFNC-HR
87794. 2CONSTANT ACQFC-HR
893B4. 2CONSTANT ACQFNC-DAY
894E0. 2CONSTANT ACQFC-DAY

DECIMAL

\ -----
\ VARIABLES

INCLUDE #EXTVAR2.TDS \ Define palabras para Manejo de memoria externa

EVAR FIRST 0 FIRST !!    \ Bandera para indicar primer vez que se enciende.

EVAR EDELAY    \ Variable para programar el periodo de la multitarea
61440 EDELAY !! \ En valor de 61440 equivale a 50ms

EVAR #INT      \ Usada para contar el numero de veces que pasa por INT
EVAR YAH       \ Bandera para indicar que ya hizo lo indicado en la hora
EVAR YAD

EVAR #LOG2 0 #LOG2 !!
EVAR #VEZ 0 #VEZ !!

2EVAR SLOGP 0. SLOGP 2!!
2EVAR SLOGT 0. SLOGT 2!!
2EVAR #IRQ1 0. #IRQ1 2!!
2EVAR #IRQ1-H 0. #IRQ1-H 2!!
2EVAR #IRQ1-D 0. #IRQ1-D 2!!

FEVAR PREPR 0E0 PREPR F!!
FEVAR TEMPR 0E0 TEMPR F!!
```

```

INCLUDE CAL12.TDS

\ -----
\ PALABAS PARA USO EN LA APLICACIÓN

DECIMAL

: START-LOG (  --  ) \ Habilita interrupción que permite adquirir señales
  $FFB0 6 ONE ;

: STOP-LOG (  --  ) \ Deshabilita interrupción que permite adquirir señales
  $FFB0 6 ZERO ;

: IRQ1 \ Esta interrupcion se activa con cada flanco de caida en el pin C27

@TIME TNEW 2!! \ Saca el # de TICKS que van en el dia y lo guarda
1TICK=53.33ms

STOP-LOG

#IRQ1 2@@ D0= IF @TIME Told 2!! AGA8-G2 THEN

#IRQ1 2@@ D0= NOT IF
  1E0
  TNEW 2@@ Told 2@@ D< IF
  1620000. Told 2@@ D- TNEW 2@@ D+ D>F F/ \ Halla la FREQ en hz del
  18.75E0 F* FREQ F!! \ transcurrido entre 2 interrupciones IRQ1
  ELSE TNEW 2@@ Told 2@@ D- D>F F/ \ Halla la FREQ en hz equivalente al
tiempo
  18.75E0 F* FREQ F!! \ transcurrido (resta de TICKS) entre dos
interrupciones
  THEN

  TNEW 2@@ Told 2!! \ Guarda el numero de TICKS actual en Told
  360E3 1PULSE F@@ F* RQMAX F@@ F/ FREQ F@@ F* %RQFI F!! \ % de rata
equivalente

  INTERP-C \ Con % interpolo y hallo el VOL. DEFLUJO CALIBRADO NO CORREG.(QFNC)

  QFNC-HR F@@ QFNC F@@ F+ QFNC-HR F!!
  CDAY-NC F@@ QFNC F@@ F+ CDAY-NC F!!

  FIRST @@ 1 = IF \ Si ya transcurrieron 2 seg donde se hallaron P T y FPV
(Z)
  AGA7 \ Se corrige el flujo con AGA7 y sale el vol. correg QFC
  QFC-HR F@@ QFC F@@ F+ QFC-HR F!! \ Y se acumula en el sumador
horario
  CDAY-C F@@ QFC F@@ F+ CDAY-C F!! \ a su vez en el CURR DAY corregido
  THEN

  #IRQ1 2@@ 3 UM/MOD DROP 0= IF AGA8-G2 THEN

THEN

#IRQ1 2@@ 1. D+ #IRQ1 2!! \ incrementa número de pulsos
#IRQ1-H 2@@ 1. D+ #IRQ1-H 2!!

```

```

#IRQ1-D 2@@ 1. D+ #IRQ1-D 2!!
START-LOG
RETURN;

-60 +ORIGIN ASSIGN IRQ1

: INIT-IRQ \ Inicializa o habilita el uso de IRQ1
$FFFC C@ $60 OR $FFFC C! \ Habilita IRQ1
$FFF0 C@ $88 AND \ Le pone prioridad máxima
$FE OR $FFF0 C! \
;
\ -----

: LOG2 \ Lee presión y temperatura
SLOGP 2@@ 1 A-D16 M+ SLOGP 2!! \ Lee PRESION CANAL 1
SLOGT 2@@ 0 A-D16 M+ SLOGT 2!! \ Lee TEMPERATURA CANAL 0
#LOG2 @@ 1+ #LOG2 !! ;

: PROM2 \ Promedia P y T cada 2 segundos y pasa a unidades °F y PSIG
SLOGP 2@@ D>F #LOG2 @@ S>F F/ MP1 F@@ F* BP1 F@@ F+ PREPR F!!
SLOGT 2@@ D>F #LOG2 @@ S>F F/ MT1 F@@ F* BT1 F@@ F+ TEMPR F!!
0 #LOG2 !! 0. SLOGP 2!! 0. SLOGT 2!! ;

: AVERAGE \ Promedio acumulativo de los resultados de PROM2
PREPR F@@ PREAV F@@ F+ 2E0 F/ PREAV F!!
TEMPR F@@ TEMAV F@@ F+ 2E0 F/ TEMAV F!! ;

\ -----

: INT \ Ésta es la multitarea que ocurre cada 50ms

LOG \ Recoge muestras de P y T

#INT @@ 1+ DUP #INT !! \ Incrementa el contador #INT

40 MOD 0= IF \ CADA 2 segs hace el PROMED de P y T
PROM2 \ ." =PT= "
FIRST @@ 1 = IF \ Si NO es la primer vez que se enciende el CF
AVERAGE THEN \ entonces hace el promedio acumulativo
FIRST @@ 0= IF \ Si se acaba de encender el CF
PREPR F@@ PREAV F!! \ Pone condiciones iniciales del promedio
TEMPR F@@ TEMAV F!! \ acumulativo igual al primer PROM2
1 FIRST !! \ quita bandera de primera vez
INIT-IRQ \ habilita interrupción de hardware, entrada de pulsos
THEN
THEN

#INT @@ 1200 MOD 0= IF \ Cada minuto guarda la PR y TEM en vector
PREAV F@@ PRES-W #MIN @@ FLOATSD D+ 2DUP POINT-P 2!! EF!
TEMAV F@@ TEMP-W #MIN @@ FLOATSD D+ 2DUP POINT-T 2!! EF!
#MIN @@ 1+ #MIN !!
0 #INT !! \ RESETEO DE INT
THEN

MINS @ 60 MOD \ Cada Hora (60 mins) se halla el promedio de la hora.
0= IF

```

```

YAH @@ 0= IF
  1 YAH !!    PROMED-HR
THEN
ELSE 0 YAH !!
  0. #IRQ1-H 2!! \ RESETEO DE IRQ-H
THEN

MINS @ 0= IF    \ Cada Dia (0 mins) se halla el promedio del dia
YAD @@ 0= IF
  1 YAD !!    PROMED-DY THEN
  ELSE 0 YAD !!
  0. #IRQ1-D 2!! \ RESETEO DE IRQ-D
THEN

START-LOG

EDELAY @@ LATER RETURN; \ fija tiempo de intervalo de multitarea

-27 +ORIGIN ASSIGN INT \ Asocia la palabra INT a la multitarea

: INITIALISE \ Inicializacion del sistema antes de entrar al lazo.
DIS          \ Deshabilita toda fuente de interrupción
$FFB0 6 ZERO \ Configura que se va a usar multitarea
ADS7825-INIT \ Inicializa el conversor AD
CEROS        \ Pone en ceros las variables
INIT-IRQ     \ Configura la interrupcion de hardware
EIS          \ Habilita interrupcioones
START-LOG ;  \ Comienza toma de señales

INCLUDE KEY3FF.TDS \ Manejo de teclado 2

: MAIN
  18.12.02 NOW    20.58.00 HRS W! W@ \ Pone alguna hora interna
  INITIALISE     ." START "
  ALL ; \ Despliega el menú en el display y atiende teclado.

: WORK \ Lazo externo que encierra el principal MAIN.
  BEGIN [' ] MAIN CATCH ERROR \
  AGAIN ;

```

CAL12.TDS: CALCULO Y ALMACENAMIENTO DE PROMEDIOS DE HORA Y DIA; E INICIALIZACIÓN DE VARIABLES

```

\ VARIABLES
\ Variables para conteo de pulsos y Acumulados de flujo

2EVAR Told    \ Cuenta de TICKS entre un pulso y otro
2EVAR TNEW    \ Cuenta de TICKS entre un pulso y otro (en el nuevo pulso)

FEVAR FREQ    \ Frecuencia equivalente Hz durante la vent
FEVAR QFNC-HR \ Acumulado de Volúmenes No Corregidos en 1 hora
FEVAR QFC-HR  \ Acumulado de Volúmenes Corregidos en 1 hora

\ Variables para promedios de PRE y TEM en una ventana.

```

```

2EVAR SUM1      \ Suma de lo leído en el periodo por el canal 1
2EVAR SUM2      \ Suma de lo leído en el periodo por el canal 2
FEVAR PREAV     \ Promedio acumulativo de las lecturas de Presion
FEVAR TEMAV     \ Promedio acumulativo de las lecturas de Temp.

2EVAR POINT-P \ Puntero al ultimo valor promedio de PRES-W
2EVAR POINT-T \ Puntero al ultimo valor promedio de TEMP-W
2EVAR  POINT-PH
2EVAR  POINT-TH

\ Variables para ecuaciones de Calibracion
FEVAR MX-P  300E0 MX-P F!!  \ Máxima presión
FEVAR MX-T  135E0  MX-T F!!  \ Máxima Temperatura

\ Valores iniciales de M y B de las ecuaciones de calibración Y=MX+B
FEVAR MP1   MX-P F@@ 32767E0 F/ MP1 F!!
FEVAR BP1   0E0 BP1 F!!
FEVAR MT1   MX-T F@@ 32E0 F- 32767E0 F/ MT1 F!!
FEVAR BT1   32E0 BT1 F!!
FEVAR MP2   MX-P F@@ 32767E0 F/ MP2 F!!
FEVAR BP2   0E0 BP2 F!!
FEVAR MT2   MX-T F@@ 32E0 F- 32767E0 F/ MT2 F!!
FEVAR BT2   32E0 BT2 F!!

\ Acumulados del dia
FEVAR CDAY-NC \ Acumulado no corregido del dia [ACF]
FEVAR CDAY-C  \ Acumulado corregido del dia [ACF]
FEVAR PDAY-NC \ Acum. no corr. del dia previo
FEVAR PDAY-C  \ Acum. correg. del dia previo

INCLUDE POLA-C.TDS \ Incluye Rutinas de Interpolacion Presion Constante.
INCLUDE AGA8G2.TDS \ Incluye Algoritmo AGA8-Gross2, sale Z y FPV
INCLUDE AGA7C.TDS  \ Incluye las palabras definidas en el Algoritmo AGA7

\ ** Variables para promedios de PRE y TEM en una hora.
EVAR  #MINH  \ Numero de minutos en la hora
EVAR  #MIN   \ Numero de minutos (temporal)
EVAR  #HORAS \ Numero de Horas transcurridas desde START
FEVAR SUM11  \ Sumatoria de promedios de ventanas Senal-1
FEVAR SUM22  \ Sumatoria de promedios de ventanas Senal-2
FEVAR PREAV-HR \ Promedio de Presion de la hora
FEVAR TEMAV-HR \ Promedio de Temper. de la hora

: PROMED-HR

\ Guarda los Vol No Corr y Corr. en los Vectores de Acumulados
correspondientes
QFNC-HR F@@ ACQFNC-HR #HORAS @@ FLOATSD D+ EF!
QFC-HR  F@@ ACQFC-HR  #HORAS @@ FLOATSD D+ EF!

0 #MIN !! \ Resetea el numero de veces que pasa por MIN
0E0 QFNC-HR F!! \ Resetea los acumulados de flujo para la siguiente hora
0E0 QFC-HR  F!! \ tanto no corregido como corregido

0E0 SUM11 F!! \ Alista sumadores de presion en cero
0E0 SUM22 F!! \ Alista sumadores de temperat en cero

```

```

\ Halla el numero de minutos transcurridos en la hora con la resta de punteros
\ Dividida en 6 bytes que es lo que ocupa un dato flotante
\ No todas las horas son de 60 min, especialmente la primera, cuando se
enciende

```

```

POINT-P 2@@ PRES-W D- D>S 6 / 1+ #MINH !!

```

```

#MINH @@ 0 DO \ Halla la suma de todas los promedios de cada minuto

```

```

PRES-W I 6 * M+ EF@ SUM11 F@@ F+ SUM11 F!!

```

```

TEMP-W I 6 * M+ EF@ SUM22 F@@ F+ SUM22 F!!

```

```

LOOP

```

```

\ Divide en numero de ventanas hallando el promedio de la hora para cada senal

```

```

SUM11 F@@ #MINH @@ S>F F/ PREAV-HR F!!

```

```

SUM22 F@@ #MINH @@ S>F F/ TEMAV-HR F!!

```

```

\ Almacena los promedios en los Vectores correspondientes.

```

```

PREAV-HR F@@ PRES-HR #HORAS @@ FLOATSD D+ 2DUP POINT-PH 2!! EF!

```

```

TEMAV-HR F@@ TEMP-HR #HORAS @@ FLOATSD D+ 2DUP POINT-TH 2!! EF!

```

```

#HORAS @@ 1+ #HORAS !! \ Aumenta en 1 el numero de promedios hora hechos

```

```

;

```

```

\ Variables para los promedios y acumulados del dia.

```

```

EVAR #HOUR \ Numero de horas del dia, en general 24, exepto el primer dia

```

```

EVAR #DIAS \ Numero de dias transcurridos desde START

```

```

2EVAR POINT-DP

```

```

2EVAR POINT-DT

```

```

2EVAR POINT-Hold \ Puntero antiguo

```

```

FEVAR PREAV-DAY

```

```

FEVAR TEMAV-DAY

```

```

FEVAR SUMD1

```

```

FEVAR SUMD2

```

```

: PROMED-DY

```

```

\ Guarda los Vol No Corr y Corr. en los Vectores de Acum DEL DIA

```

```

CDAY-NC F@@ ACQFNC-DAY #DIAS @@ FLOATSD D+ EF!

```

```

CDAY-C F@@ ACQFC-DAY #DIAS @@ FLOATSD D+ EF!

```

```

\ Hace Dia Previo = Dia Actual e inicia en cero el Conteo del nuevo dia

```

```

CDAY-NC F@@ PDAY-NC F!! 0E0 CDAY-NC F!!

```

```

CDAY-C F@@ PDAY-C F!! 0E0 CDAY-C F!!

```

```

0E0 SUMD1 F!! \ Alista sumadores de pres y temp de todo el dia

```

```

0E0 SUMD2 F!!

```

```

\ Halla el numero de horas del dia, con la resta de punteros

```

```

POINT-PH 2@@ POINT-Hold 2@@ D- D>S 6 / 1+ #HOUR !!

```

```

POINT-PH 2@@ POINT-Hold 2!! \ Graba puntero nuevo en antiguo

```

```

\ Suma todos lo promedios horarios de PRES y TEMP

```

```

#HOUR @@ 0 DO

```

```

PRES-HR I FLOATSD D+ EF@ SUMD1 F@@ F+ SUMD1 F!!

```

```

TEMP-HR I FLOATSD D+ EF@ SUMD2 F@@ F+ SUMD2 F!!

```

```

LOOP

```

```

\ Los divide en el numero de horas resultando el promedio del dia

```

```
SUMD1 F@@ #HOUR @@ S>F F/ PREAV-DAY F!!
SUMD2 F@@ #HOUR @@ S>F F/ TEMAV-DAY F!!
```

```
\ Almacena los promedios en los vectores DAY correspondientes y deja un
puntero.
```

```
PREAV-DAY F@@ PRES-DAY #DIAS @@ FLOATSD D+ 2DUP POINT-DP 2!! EF!
TEMAV-DAY F@@ TEMP-DAY #DIAS @@ FLOATSD D+ 2DUP POINT-DT 2!! EF!
```

```
#DIAS @@ 1+ #DIAS !! \ Incrementa en 1 la cuenta de días.
;
```

```
\ Palabra que inicializa en Ceros una serie de variables, contadores y
sumadores
```

```
\ y inicializa algunos punteros a la primera casilla de los vectores de
alamacen
```

```
: CEROS
0E0 FREQ F!! 0E0 QFNC-HR F!! 0E0 QFC-HR F!!
0 #HORAS !! 0 #DIAS !! 0E0 SUM11 F!! 0E0 SUM22 F!!
0E0 CDAY-NC F!! 0E0 CDAY-C F!! 0E0 PDAY-NC F!! 0E0 PDAY-C F!!
PRES-W POINT-P 2!! TEMP-W POINT-T 2!! PRES-HR POINT-Hold 2!!
0 FIRST !! 0 YAH !! 0 YAD !! 0 #INT !! 0 #LOG2 !!
0 #MIN !! 0. SLOGP 2!! 0. SLOGT 2!! 0. #IRQ1 2!!
;
```

```
\ PALABRAS PARA MOSTRAR LOS DATOS GUARDADOS EN LA MEMORIA
```

```
: LGPH CR \ Muestra todos los promedios horarios de presión
#HORAS @@ 0 DO
2 SPACES PRES-HR I 6 * M+ EF@ F. 1 SPACES
LOOP ;
```

```
: LGTH CR \ Muestra todos los promedios horarios de temperatura
#HORAS @@ 0 DO
2 SPACES TEMP-HR I 6 * M+ EF@ F. 1 SPACES
LOOP ;
```

```
: LGPD CR \ Muestra todos los promedios diarios de presión
#DIAS @@ 0 DO
2 SPACES PRES-DAY I 6 * M+ EF@ F. 1 SPACES
LOOP ;
```

```
: LGTD CR \ Muestra todos los promedios diarios de temperatura
#DIAS @@ 0 DO
2 SPACES TEMP-DAY I 6 * M+ EF@ F. 1 SPACES
LOOP ;
```

```
: LGQNCH CR \ Muestra los volúmenes no corregidos acumulados por hora
#HORAS @@ 0 DO
2 SPACES ACQFNC-HR I 6 * M+ EF@ F. 1 SPACES
LOOP ;
```

```
: LGQCH CR \ Muestra los volúmenes corregidos acumulados por hora
#HORAS @@ 0 DO
2 SPACES ACQFC-HR I 6 * M+ EF@ F. 1 SPACES
LOOP ;
```



```
: LGQNCB CR \ Muestra los volúmenes no corregidos acumulados por Dia
#HORAS @@ 0 DO
2 SPACES ACQFNC-DAY I 6 * M+ EF@ F. 1 SPACES
LOOP ;
```

```
: LGQCD CR \ Muestra los volúmenes corregidos acumulados por Dia
#HORAS @@ 0 DO
2 SPACES ACQFC-DAY I 6 * M+ EF@ F. 1 SPACES
LOOP ;
```

KEYF.TDS: DEFINE PALABRAS BASICAS DE MANEJO DE DISPLAY Y TECLADO

```
HEX CREATE CODES
```

```
"U C, "3 C, "2 C, "1 C, BA C, 43 C, 4F C, 7E C,
"D C, "6 C, "5 C, "4 C, 11 C, 0F C, EE C, 23 C,
". C, "9 C, "8 C, "7 C, 33 C, 89 C, 2A C, 12 C,
"E C, "- C, "0 C, "C C, FC C, 23 C, 75 C, 82 C,
DECIMAL
```

```
VARIABLE MANY 0 MANY !
VARIABLE MODO
VARIABLE #LIST
VARIABLE #LISTold
```

```
20 CONSTANT WIDE \ Numero de caracteres del reglón del display de 4 líneas
```

```
\ -----
\ PALABRAS PARA USO EN LA APLICACIÓN DE TECLADO Y DISPLAY
```

```
: {PUT} ( c n -- ) \ Character c is written to position n on display.
  DUP CASE DUP 0 WIDE WITHIN
    IFCASE ENDOF
      \ line 1 has positions 0 TO WIDE-1
      DUP WIDE [ WIDE 2* ] LITERAL WITHIN
      IFCASE [ 64 WIDE - ] LITERAL + ENDOF
      \ line 2 has positions 64 to 63+WIDE
      DUP [ WIDE 2* ] LITERAL [ WIDE 3 * ] LITERAL WITHIN
      IFCASE [ WIDE NEGATE ] LITERAL + ENDOF
      \ line 3 has positions WIDE to 2*WIDE
      DUP [ WIDE 3 * ] LITERAL [ WIDE 4 * ] LITERAL WITHIN
      IFCASE [ 64 WIDE 2* - ] LITERAL + ENDOF
      \ line 4 has positions 64+WIDE to 63+2*WIDE
      DROP 0 0AT \ put to home position if not 0 to 4*WIDE-1
      SWAP \ bring CASE selector back to top of stack
      ENDCASE <PUT> ; \ place character in corrected position on LCD
```

```
: <LCD ( -- ) \ Revector EMIT to use LCD
  'EMIT @ R> 2>R ['] LCDEMIT 'EMIT ! ;
: LCD> ( -- ) \ Restore vector
  2R> >R 'EMIT ! ;
```

```
' {PUT} 'PUT ! \ revector LCD placement primitive to new version
\ at compile time
```

```
\ -----
\ DEMONSTRATION Exclude from final application code
```

```

: INILCD ( -- ) \ Test on LM041 display
  0          \ no blink
  0          \ cursor on
  1          \ display on
  0          \ no display shift on write
  1          \ auto incr of char position on write
  $38        \ 2 line LCD (electrically), 5 x 7 matrix
  PRIME      \ initialise LCD
  ['] {PUT} 'PUT ! \ revector LCD primitive to new version {PUT}
;

: TRANSLATE ( n1 - n2 ) \ Traduce el código de una tecla oprimida
CODES + 1- C@ ;

: KKEY ( - n ) \ Espera hasta que se oprima una tecla
BEGIN NEWKEY ?DUP UNTIL ;

: REQU
KKEY TRANSLATE
;

\ Solicita un numero (varias teclas) hasta que se oprima Enter (Tecla E)
\ a la vez va visualizando lo que se oprime y finalmente se traduce.

: REQUEST ( - )
0 MANY !
PAD 15 BLANK PAD
BEGIN KKEY TRANSLATE DUP "E <>
WHILE DUP LCDEMIT OVER C! 1+
MANY @ 1+ MANY !
REPEAT 2DROP ;

\ Misma anterior pero si oprime Tecla C, se borra el ultimo carácter escrito
: REQUESTC ( - )
0 MANY !
PAD 15 BLANK PAD
BEGIN KKEY TRANSLATE DUP "C = IF 0 MANY ! EXIT THEN DUP "E <>
WHILE DUP LCDEMIT OVER C! 1+
MANY @ 1+ MANY !
REPEAT 2DROP ;

\ Pasa la cadena traducida a un número entero
: TRADN 0. PAD MANY @ >NUMBER 2DROP DROP ;

\ Pasa la cadena traducida a un número flotante
: TRADF PAD MANY @ >FLOAT DROP ;

```

KEY3FF.TDS: MENUS Y INTERFASE DE USUARIO EN DISPLAY Y TECLADO

```

FEVAR VOLT      0E0 VOLT F!!
FEVAR CASET     0E0 CASET F!!
FEVAR REFCAL    0E0 REFCAL F!!
EVAR IDECO

```

```

: L1- 0 AT ! ;
: L2- 20 AT ! ;

```

```

: L3- 40 AT ! ;
: L4- 60 AT ! ;
: MEN1 <LCD ." ==== CIG-ECOGAS ==== LCD> ;
: MEN2 <LCD ." === ECOFLOW2020T === LCD> ;
: MEN3 <LCD ." 1-DISPLAY 2-ALARMAS LCD> ;
: MEN4 <LCD ." 3-CALIBRA 4-CONFIGU LCD> ;

: MEN WIPE MEN1 MEN2 MEN3 MEN4 ;
: MEN-A WIPE MEN 1 MEN2 ;

: MENDI0 WIPE
L1- <LCD ." MODO DISPLAY LCD>
L3- <LCD ." Escoja con flechas LCD>
L4- <LCD ." [CLEAR para volver] LCD> 1000 MS ;

: MENDI1
CDAY-C F@@ L1- <LCD ." CuDy CF: " #F. LCD>
RQFC      F@@ L2- <LCD ." Rat CFH: " #F. LCD>
PREPR     F@@ L3- <LCD ." Pre PSI: " #F. LCD>
TEMPR     F@@ L4- <LCD ." Temp °F: " #F. LCD> ;

: MENDI2
PDAY-C F@@ L1- <LCD ." PvDy CF: " #F. LCD>
RQFNC     F@@ L2- <LCD ." URatCFH: " #F. LCD>
FLOWK     F@@ L3- <LCD ." Flw Con: " #F. LCD>
FPV       F@@ L4- <LCD ." FPV : " #F. LCD> ;

: MENDI3
#IRQ1 2@@ L1- <LCD ." #PULSOS: " D. LCD>
PREAV  F@@ L2- <LCD ." PREAV : " #F. LCD>
TEMAV  F@@ L3- <LCD ." TEMAV : " #F. LCD>
#HORAS @@ L4- <LCD ." #HORAS: " . LCD> ;

: MENAL1 WIPE
L1- <LCD ." ** MODO ALARMAS ** LCD>
L3- <LCD ." Escoja alarma (1-4) LCD>
L4- <LCD ." [CLEAR para volver] LCD> 2000 MS ;

: MENAL2
L1- <LCD ." 1-MAX. PRESION " LCD>
L2- <LCD ." 2-MAX. TEMPERATURA " LCD>
L3- <LCD ." 3-MIN. VOLTAJE " LCD>
L4- <LCD ." 4-MAX. VOLTAJE " LCD> ;

: MENCA1 WIPE
L1- <LCD ." *MODO CALIBRACION* LCD>
L3- <LCD ." Escoja sensor (1-4) LCD>
L4- <LCD ." [CLEAR para volver] LCD> 2000 MS ;

: MENCA2
L1- <LCD ." 1-PRESION - Brazo1 LCD>
L2- <LCD ." 2-TEMPERA - Brazo1 LCD>
L3- <LCD ." 3-PRESION - Brazo2 LCD>
L4- <LCD ." 4-TEMPERA - Brazo2 LCD> ;

: MENCA3
L1- <LCD ." ** CALIBRACION ** LCD>

```

```

L2- <LCD ." Valores a congelar:" LCD>
L3- <LCD ." Presion [@F]:" LCD>
L4- <LCD ." Tempera[psi]:" LCD> ;

: Menco WIPE
L1- <LCD ." *MODO CONFIGURACION*" LCD>
L2- <LCD ." Escoja dato (1-4)" LCD>
L3- <LCD ." Mas datos con flechas" LCD>
L4- <LCD ." [CLEAR para volver]" LCD> 2000 MS ;

: Menco1
L1- <LCD ." 1-MP1 " LCD>
L2- <LCD ." 2-BP1 " LCD>
L3- <LCD ." 3-MT1 " LCD>
L4- <LCD ." 4-BT1 " LCD> ;

: Menco2
L1- <LCD ." 1-KCAL " LCD>
L2- <LCD ." 2-KFL " LCD>
L3- <LCD ." 3-CURRDAY-NC " LCD>
L4- <LCD ." 4-CURRDAY-C " LCD> ;

: Menco3
L1- <LCD ." 1-RQMAX DE TURBINA " LCD>
L2- <LCD ." 2-Fraccion Molar N" LCD>
L3- <LCD ." 3-Fraccion Molar CO2" LCD>
L4- <LCD ." 4-Fraccion Molar H" LCD> ;

: MENP <LCD ." PRUEBA " LCD> ;

: PASSW WIPE DECIMAL
L2- <LCD ." *Entre su password*" LCD>
L3- <LCD ." y oprima ENTER " LCD>
L4- <LCD ." :: " LCD>
REQUEST TRADN ;

\ ***** DISPLAY *****

: LIST-1 MENDI1 ;
: LIST-2 MENDI2 ;
: LIST-3 MENDI3 ;

: MOD01
WIPE 1 #LIST ! 1 #LISTold !
BEGIN
#LISTold @ #LIST @ <> IF WIPE THEN
#LIST @ #LISTold !

#LIST @ 1 = IF LIST-1 THEN
#LIST @ 2 = IF LIST-2 THEN
#LIST @ 3 = IF LIST-3 THEN

INKEY ?DUP 0> IF
TRANSLATE CASE
"U OF #LIST @ 2 = IF 1 #LIST ! THEN
#LIST @ 3 = IF 2 #LIST ! THEN END OF
"D OF #LIST @ 2 = IF 3 #LIST ! THEN

```

```

        #LIST @ 1 = IF 2 #LIST ! THEN END OF
ENDCASE
THEN

INKEY TRANSLATE "C = UNTIL
;

\ ***** ALARMAS *****

: PI-ALL
L2- <LCD ." OLD= " LCD> L3- <LCD ." NEW= " LCD>
L4- <LCD ." ENTER" LCD> ;

: PIDE-MP 26 AT ! MENP 3000 MS WIPE ;
: PIDE-MT 46 AT ! MENP 3000 MS WIPE ;
: PIDE-mV 66 AT ! MENP 3000 MS WIPE ;
: PIDE-MV 26 AT ! MENP 3000 MS WIPE ;

: MODO2
WIPE 1 #LIST ! 1 #LISTold !
BEGIN
#LISTold @ #LIST @ <> IF WIPE THEN
MENAL2
INKEY ?DUP 0> IF
TRANSLATE CASE
    "1 OF WIPE L1- <LCD ." ** MAX. PRESION **" LCD> PI-ALL PIDE-MP END OF
    "2 OF WIPE L1- <LCD ." ** MAX. TEMPERA **" LCD> PI-ALL PIDE-MT END OF
    "3 OF WIPE L1- <LCD ." ** MIN. VOLTAJE **" LCD> PI-ALL PIDE-mV END OF
    "4 OF WIPE L1- <LCD ." ** MAX. VOLTAJE **" LCD> PI-ALL PIDE-MV END OF
ENDCASE
THEN
100 MS
INKEY TRANSLATE "C = UNTIL ;

\ ***** CALIBRACION *****

: CA-ALL WIPE L1- <LCD ." LEE= " LCD>
L2- <LCD ." PRE:0-300 TEM:32-135" LCD>
L3- <LCD ." Cer[0]-Full[9]-OK[1]" LCD>
L4- <LCD ." Opcion: " LCD>
500 MS ;

: CALOK 71 AT ! <LCD ." ..CAL.." LCD> 1500 MS ;

: CALI-P1
CA-ALL
5 AT ! PREPR F@@ <LCD #F. LCD>
\ 25 AT ! REQUESTC TRADF REFCAL F!!
67 AT ! <LCD ." > " LCD> REQU DUP LCDEMIT
CASE
    "0 OF PREPR F@@ -1E0 #F* BP1 F!! CALOK END OF
    "9 OF MX-P F@@ MP1 F@@ #F* PREPR F@@ #F/ MP1 F!! CALOK END OF
    "1 OF 69 AT ! <LCD ." OK " LCD> 1500 MS END OF
ENDCASE WIPE ;

: CALI-T1
CA-ALL

```

```

5  AT ! TEMPR F@@ <LCD  #F. LCD>
\ 25 AT ! REQUESTC TRADF REFCAL F!!
55 AT ! REQU DUP LCDEMIT
CASE
    "0 OF BT1 F@@ 2E0 #F*  TEMPR F@@ #F-  BT1 F!! CALOK END OF
    "9 OF MX-T F@@ MT1 F@@ #F* TEMPR F@@ #F/  MT1 F!! CALOK END OF
    "1 OF 69 AT ! <LCD ." OK" LCD> 1500 MS END OF
ENDCASE  WIPE ;

: CALI-P2
CA-ALL
5  AT ! PREPR F@@ <LCD  #F. LCD>
25 AT ! REQUESTC TRADF REFCAL F!!
55 AT ! REQU DUP LCDEMIT
CASE
    "0 OF BP2 F@@ REFCAL F@@ #F-  BP2 F!! CALOK END OF
    "9 OF REFCAL F@@ 32767E0 #F/  MP2 F!! CALOK END OF
    "1 OF <LCD ." OK " LCD> 1500 MS END OF
ENDCASE  WIPE ;

: CALI-T2
CA-ALL
5  AT ! TEMPR F@@ <LCD  #F. LCD>
25 AT ! REQUESTC TRADF REFCAL F!!
55 AT ! REQU DUP LCDEMIT
CASE
    "0 OF BT2 F@@ REFCAL F@@ #F-  BT2 F!! CALOK END OF
    "9 OF REFCAL F@@ 32767E0 #F/  MT2 F!! CALOK END OF
    "1 OF <LCD ." OK " LCD> 1500 MS END OF
ENDCASE  WIPE ;

: MODO3
WIPE 1 #LIST ! 1 #LISTold !
MENCA1 1000 MS  WIPE
BEGIN
#LISTold @ #LIST @ <> IF WIPE THEN
MENCA2
INKEY ?DUP 0> IF
TRANSLATE CASE
    "1 OF WIPE L1- <LCD ." * PRESIDON Braz1 *" LCD> 500 MS CALI-P1 END OF
    "2 OF WIPE L1- <LCD ." * TEMPERA Braz1 *" LCD> 500 MS CALI-T1 END OF
    "3 OF WIPE L1- <LCD ." * PRESIDON Braz2 *" LCD> 500 MS CALI-P2 END OF
    "4 OF WIPE L1- <LCD ." * TEMPERA Braz2 *" LCD> 500 MS CALI-T2 END OF
ENDCASE
THEN
50 MS
INKEY TRANSLATE "C = UNTIL
;

\ ***** CONFIGURACION *****

: CONF1 WIPE
L2- <LCD ." OLD= " LCD>  \ PONE
L3- <LCD ." NEW= " LCD>  \ REQUESTC
L4- <LCD ." [ ENTER o CLEAR ]" LCD> ;

```

```

: ASIG    MANY @ 0 <> ;
: INTR-N  @@ 26 AT ! <LCD . LCD> 100 MS 46 AT ! REQUESTC TRADF F>S ASIG WIPE ;
: INTR-F  F@@ 26 AT ! <LCD #F. LCD> 100 MS 46 AT ! REQUESTC TRADF ASIG WIPE ;

: MODO4
WIPE    1 #LIST ! 1 #LISTold ! DECIMAL
PASSW DUP ." NUMERO=" . 999 <> IF
    WIPE
    L2- <LCD ." !Password Invalido!" LCD> 2000 MS EXIT
    THEN
MENCO 2000 MS WIPE
BEGIN
    #LISTold @ #LIST @ <> IF WIPE THEN
    #LIST @ #LISTold !
    #LIST @ 1 = IF MENCO1 THEN
    #LIST @ 2 = IF MENCO2 THEN
    #LIST @ 3 = IF MENCO3 THEN

    INKEY ?DUP 0> IF
    TRANSLATE CASE
    "U OF #LIST @ 2 = IF 1 #LIST ! THEN
        #LIST @ 3 = IF 2 #LIST ! THEN ENDOF
    "D OF #LIST @ 2 = IF 3 #LIST ! THEN
        #LIST @ 1 = IF 2 #LIST ! THEN ENDOF
    "1 OF CONF1
        #LIST @ 1 = IF MP1 INTR-F IF MP1 F!! THEN THEN
        #LIST @ 2 = IF KCAL INTR-F IF KCAL F!! THEN THEN
        #LIST @ 3 = IF RQMAX INTR-F IF KCAL F!! THEN THEN
        WIPE ENDOF
    "2 OF CONF1
        #LIST @ 1 = IF BP1 INTR-F IF BP1 F!! THEN THEN
        #LIST @ 2 = IF KFL INTR-F IF KFL F!! THEN THEN
        #LIST @ 3 = IF %NI INTR-F IF %NI F!! THEN THEN
        WIPE ENDOF
    "3 OF CONF1
        #LIST @ 1 = IF MT1 INTR-F IF MT1 F!! THEN THEN
        #LIST @ 2 = IF CDAY-NC INTR-F IF CDAY-NC F!! THEN THEN
        #LIST @ 3 = IF %CO2 INTR-F IF %CO2 F!! THEN THEN
        WIPE ENDOF
    "4 OF CONF1
        #LIST @ 1 = IF BT1 INTR-F IF BT1 F!! THEN THEN
        #LIST @ 2 = IF CDAY-C INTR-F IF CDAY-C F!! THEN THEN
        #LIST @ 3 = IF %HI INTR-F IF %HI F!! THEN THEN
        WIPE ENDOF
    ENDCASE
    THEN
    100 MS
    INKEY TRANSLATE "C = 50 MS UNTIL
;

\ ***** LCD-KEY *****
\ Palabra Principal que involucra todo, y es la tarea de fondo o background

: ALL
BEGIN
    INILCD

```

```

MEN REQU
CASE
    "1 OF WIPE L1- MENDIO  1 MODO ! 1000 MS MODO1 ENDOF
    "2 OF WIPE L2- MENAL1  2 MODO !  MODO2 ENDOF
    "3 OF WIPE 3 MODO ! MODO3 ENDOF
    "4 OF WIPE 4 MODO ! MODO4 ENDOF
    "0 OF WIPE START ENDOF
ENDCASE
WIPE
KEY? UNTIL
;

```

AGA7C.TDS: NORMA AGA7

```

FEVAR RQFC \ OUTPUT: Rata de Flujo Corregida a Cond. Base [CFH]
0E0 RQFC F!!
FEVAR QFC \ OUTPUT: Volumen de Flujo Corregido a Cond. Base [CF] en la
ventana
0E0 QFC F!!

FEVAR FLOWK 0E0 FLOWK F!!
FEVAR KFL 1.019E0 KFL F!!

: AGA7

TBASE F@@ 459.67E0 F+ FPV F@@ F* FPV F@@ F*
TEMPR F@@ 459.67E0 F+ F/
PREPR F@@ PATMO F@@ F+ F* PBASE F@@ F/
KFL F@@ F* FLOWK F!!
FLOWK F@@ QFNC F@@ F*
QFC F!!

\ CR ." ESTE ES RQFC " RQFC F@@ F. CR

QFC F@@ 3600E0 F* FREQ F@@ F* RQFC F!!

;

```

POLA-C.TDS: RUTINA DE INTERPOLACIÓN – CURVA DE TURBINA -

```

\ Dos palabras para fácil acceso a un elemento de un vector dado
: 6F@@ ( VECTOR INDICE-- ELEMENTO) \ Trae el elemento enesimo
1- 6 * + F@@ ;
: 6F!! ( VECTOR INDICE-- ELEMENTO) \ Guarda en el elemento enesimo
1- 6 * + F!! ;

FEVAR KCAL 1.0E0 KCAL F!! \ CONSTANTE DE CALIBRACION
FEVAR RQMAX \ INPUT: Rata de Flujo Maximo (ACFH)
FEVAR 1PULSE \ INPUT: Equivalencia en ACFH de 1 Pulso de la Turbina
35000E0 RQMAX F!!
100E0 1PULSE F!!

\ INPUT: % de Rata de Flujo de Curva de Calibración
FEVAR %QF RAM 15 6 * ALLOT ROM
5E0 %QF 1 6F!!
7E0 %QF 2 6F!!
10E0 %QF 3 6F!!

```



```

15E0 %QF 4 6F!!
20E0 %QF 5 6F!!
30E0 %QF 6 6F!!
50E0 %QF 7 6F!!
75E0 %QF 8 6F!!
100E0 %QF 9 6F!!

```

\ INPUT: Factores de calibracon de la Curva de Calibración del Fabricante

```
FEVAR 100/K RAM 15 6 * ALLOT ROM
```

```

1.001001E0 100/K 1 6F!!
1.003009E0 100/K 2 6F!!
1.003009E0 100/K 3 6F!!
1.003009E0 100/K 4 6F!!
1.003009E0 100/K 5 6F!!
1.003009E0 100/K 6 6F!!
1.001001E0 100/K 7 6F!!
0.990001E0 100/K 8 6F!!
0.998004E0 100/K 9 6F!!

```

```
FEVAR %RQFI \ INPUT: Porcentaje Leido de Rata de Flujo Bruta
```

```
FEVAR RQFI \ Rata de Flujo Bruta
```

```
FEVAR RQFNC
```

```
0E0 RQFNC F!! \ OUTPUT: Rata de Flujo Calibrada NO CORREGIDA (ACFH)
```

```
FEVAR QFNC
```

```
0E0 QFNC F!! \ OUTPUT: Flujo Calibrado NO CORREGIDO (ACF)
```

```
FEVAR 100/KO \ Valor de Constante de Calibracion
```

```
FEVAR QF1 \ Limites de intervalo usado en interpolacion
```

```
FEVAR QF2 \ " "
```

```
FEVAR 100/K1 \ " "
```

```
FEVAR 100/K2 \ " "
```

```
FEVAR MTURC \ Pendiente de la recta de interpolacion Turbina Constante
```

```
EVAR NE 9 NE !! \ Transitorias
```

```
EVAR CON 1 CON !! \ Transitorias
```

```
: INTERP-C
```

```
1 CON !!
```

```
BEGIN
```

```
%QF CON @@ 6F@@ %RQFI F@@ F< WHILE
```

```
%QF CON @@ 6F@@ QF1 F!!
```

```
%QF CON @@ 1 + 6F@@ QF2 F!!
```

```
100/K CON @@ 6F@@ 100/K1 F!!
```

```
100/K CON @@ 1 + 6F@@ 100/K2 F!!
```

```
CON @@ 1 + CON !!
```

```
REPEAT
```

```
100/K2 F@@ 100/K1 F@@ F-
```

```
QF2 F@@ QF1 F@@ F- F/
```

```
MTURC F!!
```

```
%RQFI F@@ RQMAX F@@ F* 100E0 F/
```

```
RQFI F!!
```

```
MTURC F@@ %RQFI F@@ QF1 F@@ F- F*
```

```
100/K1 F@@ F+
```

```
100/KO F!!
```

```
RQFI F@@ 100/KO F@@ F*
```

RQFNC F!!

\ Saca la Rata de Flujo Calibrado
\ No Corregido (RQFNC) en [ACFH] y luego lo convierte a unidades de
\ volumen de flujo QFNC No Corregido [ACF].

RQFNC F@@ 3600E0 F/ FREQ F@@ F/ KCAL F@@ F*
QFNC F!!

;

ANEXO G. MANUAL DE INSTALACIÓN Y OPERACIÓN ECUFLOW2020T

INTRODUCCIÓN

El Computador de Flujo de Gas Electrónico (EFC) ECOFLOW-2020T/P es un computador controlado que incluye transductores de presión y temperatura. El computador está diseñado para obtener precisión, confiabilidad y facilidad de mantenimiento. El ECOFLOW-2020T esta en capacidad de operar junto a un medidor primario tipo turbina y el ECOFLOW-2020P esta diseñado para operar junto a un medidor primario tipo platina de orificio.

El bajo consumo de potencia de los circuitos integrados y las tarjetas electrónicas implementadas le permiten al EFC operar al menos 1 año únicamente con alimentación de la batería.

El EFC incorpora un teclado (16 teclas) y un *display* de dos líneas alfanuméricas montados en el interior de la puerta del encapsulado, lo que facilita la operación en campo al permitir el acceso a la información de la estación. Los parámetros de la estación, las alarmas, la configuración y la calibración pueden ser manejados usando el teclado y el *display* sin utilizar ningún dispositivo externo.

El operador puede también usar un computador portátil y ejecutar el *software* incluido en la unidad para configurar los parámetros del EFC. Este *software* le permite al operador descargar la información almacenada por el EFC y configurar algunos valores de los parámetros. El ECOFLOW-2020T realiza el calculo del volumen corregido usando el reporte AGA-7 y el ECOFLOW-2020P calcula el volumen corregido usando el reporte AGA-3.

El diseño intrínsecamente seguro del EFC permite la operación en un ambiente con la siguiente clasificación: Clase 1, Grupo D.

INSTALACIÓN

CHEQUEO INICIAL

1. Identifique todas las partes exteriores de la unidad. En la Figura 1 se muestra el diagrama para el ecoflow -2020T con un solo brazo de medición.
2. Suelte el cerrojo ubicado en el lado derecho del encapsulado y abra la puerta del encapsulado.

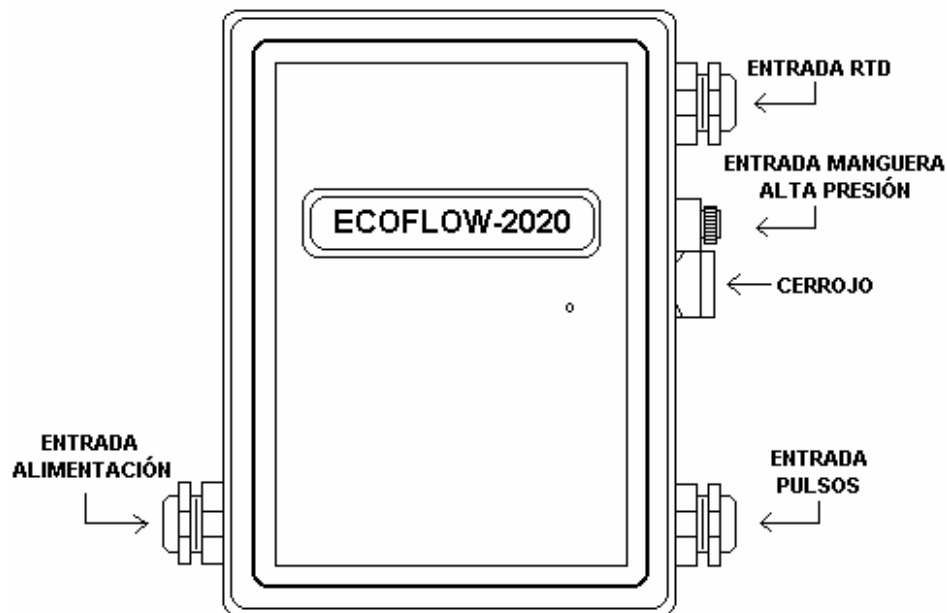


Figura 1. Parte exterior del ECUFLOW 2020T.

3. Verifique que el EFC contiene todas las partes listadas a continuación.

- (1) Encapsulado Nema 4X
- (1) *Display*
- (1) Teclado matricial
- (1) Tarjeta de acondicionamiento de señal
- (1) Tarjeta CPU
- (1) Una batería de respaldo

Sólo para el ECUFLOW-2020T

- (2) Sensor de presión estática
- (2) RTD

Sólo para el ECUFLOW-2020P

- (1) Sensor de presión diferencial (DP)
- (1) Sensor de presión estática
- (1) RTD

4. Si la unidad va a ser alimentada a través de la batería de 12V conéctela a los terminales de la tarjeta de acondicionamiento etiquetados como $\pm 12V_{bat}$. Si la unidad utilizará la alimentación de la red, conecte este suministro DC a los terminales de la tarjeta de acondicionamiento etiquetados como $\pm 12VDC$, como se muestra en la Figura 2.

ADVERTENCIA

No exceda la cantidad de voltaje de entrada recomendado: 12VDC.

5. Llevando el ECUFLOW-2020T/P a modo de Display, se pueden observar los valores de los parámetros sobre el *display* a través del uso de las teclas de desplazamiento ↑ ↓. Este modo de operación se explica detalladamente en la sección 2.1.

6. Empleando el teclado y *display* se pueden cambiar los valores de los parámetros, visualizar o verificar alarmas activas, o calibrar la unidad.

ALIMENTACIÓN PARA EL ECUFLOW-2020T/P

La unidad puede ser alimentada de dos maneras:

1. Conectando la alimentación de red a la unidad.
2. A través de una batería recargable de litio que suministra la alimentación para el EFC para aproximadamente 1 año. Si el suministro principal falla, una batería de respaldo ubicada en el interior de la unidad alimentará las unidades de memoria y el reloj de tiempo real. La alimentación de respaldo puede mantener datos históricos por un período de hasta siete años. Cuando la alimentación de respaldo entra en funcionamiento, el EFC volverá a una operación normal solo hasta que la batería principal sea reemplazada. El rango para el suministro de alimentación es de 11,5 – 15 VDC.

SEÑAL DE PULSO DE ENTRADA

Sólo para el ECUFLOW-2020T

El EFC puede ser configurado para recibir una señal de contacto seco o una señal de pulsos desde el medidor de turbina, la cual representa el volumen no corregido.

TRANSDUCTORES EN EL ECUFLOW-2020T/P

Para sensar la temperatura del gas, el EFC emplea un detector resistivo de temperatura (RTD) de platino altamente lineal y estable. Para la medición de la temperatura del encapsulado se emplea un circuito integrado de precisión ubicado dentro de la tarjeta de acondicionamiento.

Sólo para el ECUFLOW-2020T

El ECUFLOW-2020T usa dos transductores de presión de precisión tipo *strain gauge* montados dentro la unidad, que combina máxima precisión con bajo consumo de potencia. Estos sensores son empleados para la medición de la presión estática. Cada sensor hace parte de un brazo de medición.

Sólo para el ECUFLOW-2020P

El conjunto de sensores de presión del ECUFLOW-2020P esta conformado por un transductor de presión diferencial (DP) y uno de presión estática dispuestos para realizar tomas de presión tipo brida.

ADVERTENCIA!

DESPRESURICE el medidor y la tubería asociada antes que realice las conexiones de presión en la tubería. Una falla puede generar una explosión e incendio, causando serias lesiones al personal y daños en los predios que se realice la medición. NO INTENTE conectar cualquier tubería o accesorios al medidor o tubo cuando se está bajo presión. NO FUME mientras está cerca del gas o se esta probando la presión al medidor.

1.4.1. CONEXIÓN DE LA MANGUERA DE ALTA PRESIÓN

Sólo para el ECUFLOW-2020T

Conecte la manguera de alta presión al sensor de presión del brazo de medición que se va a emplear. Recuerde usar cinta PTFE (Teflón®, etc.) o un sello de tubería enrollándolo completamente sobre las conexiones. Si la manguera de alta presión es de mayor longitud que la requerida para la instalación, entonces se debe enrollar evitando dejar círculos menores a un radio de $\frac{3}{4}$ " y con curvas puntiagudas. Se debe estar seguro de haber sellado toda la conexión herméticamente. Posteriormente se deben realizar pruebas de fuga a la manguera.

NOTA

Se recomienda que emplee una válvula de corte entre el instrumento y la fuente de presión.

1.4.2. INSTALACIÓN DEL SENSOR DE PRESIÓN DIFERENCIAL

Sólo para el ECUFLOW-2020P

Conecte el cable de instrumentación del transmisor de presión diferencial en las entradas respectivas del ECUFLOW-2020P. El sensor posee una brida que le permite realizar la conexión al proceso. Conecte los *tubing* de alta presión y baja presión a las brida de entrada que están etiquetadas con las letra H y L, respectivamente. Es necesario que el lado de Alta del Transmisor DP se conecte aguas arriba de la platina por que si no las lecturas de DP serán incorrectas. No olvide realizar estas conexiones herméticamente.

1.4.3. INSTALACIÓN DEL SENSOR DE TEMPERATURA

El EFC incluye un sensor de temperatura tipo RTD. El cable de la RTD se puede enrollar la longitud que sea necesaria para prevenir posibles daños al instrumento o poner en peligro el ambiente que rodea el equipo. El sensor está diseñado para ser ajustado dentro de un termopozo estándar. No olvide sellar herméticamente la conexión.

Puede usar aceite o glicol etileno para mejorar la transferencia de calor desde el termopozo al sensor. Sin embargo, deje suficiente espacio para el fluido a utilizar entre el termopozo y la punta del sensor, con el fin de evitar una compresión hidráulica sobre el sensor, y evitar daños en el mismo.

Se recomienda que el sensor se instale 1 ó 2 diámetros de tubería aguas abajo de la ubicación del medidor. El diámetro de tubería mínimo a través del cual se puede insertar el termopozo es 4". Para tuberías más grandes, inserte el termopozo al menos el 50% del diámetro de la tubería interna, pero no exceda más del 75%. También, instale el termopozo de tal forma que la punta del mismo apenas llegue a la mitad del flujo en la tubería. No posicione el termopozo donde pueda ser expuesto a la luz directa del sol; si es necesario instale una capa protectora que cubra el sensor del sol.

La Tabla 1 contrasta el diámetro de la tubería con relación a la longitud de inserción del sensor en la misma.

Tabla 1. Referencia de inserción del termopozo

Diámetro tubería		Longitud de inserción	
pulg.	mm	pulg.	mm
4	100	2.5	64
8	200	4.5	115
12	300	6	150

COMUNICACIÓN

El ECUFLOW-2020T/P esta equipado con un puerto serial RS-232C. Este puerto le permite a un operador configurar y recolectar datos con un computador portátil estándar (IBM®, Compaq®, etc.). Para esta función se requiere el *software* que se incluye con la unidad. El EFC se comunica a una rata de 9600 baudios con computadores portátiles o de control conectados directamente al puerto serial.

PUESTA A TIERRA DEL ECUFLOW-2020T/P

Debido a que la potencia aplicada a los equipos puede causar daños y lesiones graves al personal de campo, se deben tener en cuenta ciertas consideraciones cuando se instala el EFC sobre sitios de medición de gas natural. La mas importante es el sistema de puesta a tierra. La siguiente información esta diseñada para indicar las reglas y prácticas generales, y no para sustituir las definidas en el Código Eléctrico Nacional, ni en la recomendación de Instalaciones Eléctricas para áreas clasificadas de gas publicada por la Asociación Americana de Gas (AGA). El

buen entendimiento de las normas locales es fundamental para un trabajo de instalación apropiado y legal.

Todo el EFC y el equipo de medición de gas debe ser conectado al mismo sistema de tierras, de tal manera que cualquier voltaje inesperado sobre el sistema sea rápidamente derivado a tierra y no a través de la línea de potencia o del cable de los pulsos del medidor de turbina. Esto se realiza empleando una barra de tierra común a la cual se le conectan de forma segura todos los chasis de los equipos, los encapsulados metálicos, la tubería de gas aislada, las interfaces de utilidad, las líneas de teléfono, la potencia AC, la fuente de protección secundaria y los buses de tierra de la seguridad intrínseca. Se debe usar un cable de tierra de cobre sólido de tamaño y tipo apropiados para unir el EFC a la barra de tierra. Si se utiliza más de una barra, como sucede cuando los equipos están separados por cierta distancia, todas las barras deben ser unidas de manera adecuada. Se recomienda usar cobre de calibre 6 o 8 para conexión a tierra.

Cuando se pone a tierra un equipo electrónico que esta siendo montado sobre la tubería, se debe tener cuidado de no desconectar algún sistema de protección catódica si ha sido instalado para proteger los sistemas de la corrosión. La conexión al sistema de puesta tierra de un tubo de gas protege al EFC pero resulta perjudicial al sistema de protección catódica. Aunque la tubería de gas esta enterrada en la tierra, normalmente se aísla del sistema de puesta a tierra a través de algún tipo de capa protectora. Los sobrevoltajes provenientes de relámpagos intempestivos cercanos al sistema, pueden viajar a través de la tubería empotrada en la tierra o por las líneas y conexiones eléctricas.

Las líneas de tierra y cualquier equipo montado sobre ellas pueden ser aislados de la protección catódica utilizando bridas de aislamiento echas para este propósito. Si un soporte del medidor u otra estructura es aislada con estas bridas, puede ser necesario conectarlas entre sí mediante caminos subterráneos con el objetivo de mantener uniforme el voltaje catódico sobre ellas. Estas bridas de aislamiento no son garantizan la prevención de altos potenciales de descargas eléctricas. Por esta razón, es recomendable que se conecte a tierra la parte aislada del medidor.

MODOS DE OPERACIÓN

A través del *display* del EFC se puede acceder a cuatro opciones de trabajo a saber:

- Display.
- Alarmas.
- Calibración.
- Configuración.

MODO DISPLAY

En este modo se puede conocer el valor actual de las principales variables que maneja y calcula la unidad. Aparecen en grupos de cuatro variables y en su orden se enumeran a continuación:

Current Day
Rata Corregida
Presión línea
Temperatura
Previous day
Rata no corregida
Factor de flujo
Zeta, (factor de supercompresibilidad)
Frecuencia
Volumen no corregido
Fecha
Número de días de trabajo (desde la ultima descarga)

Para moverse sobre las doce variables se emplean las teclas desplazamiento $\uparrow\downarrow$. Cuando desee visualizar una variable específica, ubique el cursor sobre ella y oprima la tecla enter. Este menu no se puede editar.

MODO ALARMA

En este modo se puede configurar los diferentes tipos de alarmas que trabaja el EFC:

Max Presión: Se configura la presión máxima de operación.
Max Temperatura: Se configura la temperatura máxima de operación.
Max Voltaje: Se configura el voltaje máximo de alimentación.
Min Voltaje: Se configura el voltaje mínimo de alimentación.

MODO CALIBRACIÓN

En este modo se pueden configurar o introducir las constantes de calibración de cada uno de los sensores. Para el caso del ECUFLOW-2020T, el menu es el siguiente:

P1
T1
P2
T2

MODO CONFIGURACIÓN

A través de este modo se configuran los parámetros del EFC. El menu de este modo es el siguiente:

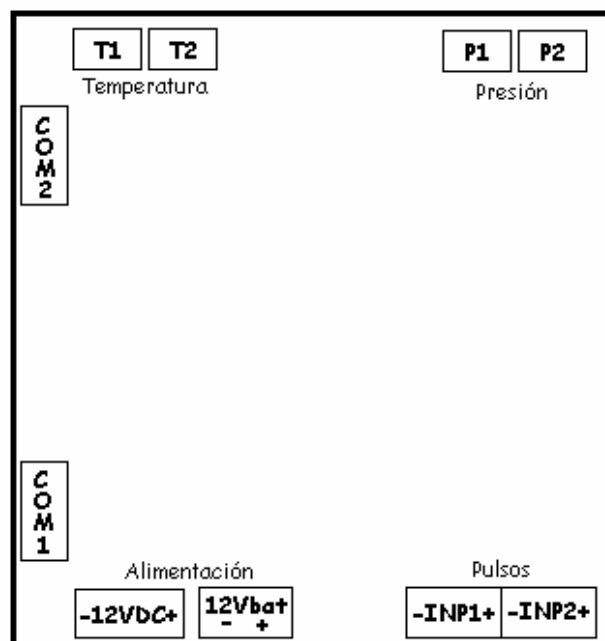
Numero de ID
Presion Base
Tempera Base
Presion Atmosf

Gravedad Especifica
Full Scala Presion
Full Scala Tempera
Fecha y Hora
Modo de Muestreo
Fraccion Molar N
Fraccion Molar CO2
Fraccion Molar H

Para moverse sobre las doce variables se emplean las teclas desplazamiento $\uparrow\downarrow$. Cuando desee acceder a una variable específica, ubique el cursor sobre ella y oprima la tecla enter. En la pantalla siguiente aparecerá el valor actual del parámetro y un espacio disponible para introducir el nuevo valor. Para confirmar cualquiera de los dos valores oprima la tecla enter.

APÉNDICE A. DIAGRAMA DE CONECTORES

TARJETA DE ACONDICIONAMIENTO ECUFLOW-2020T



TARJETA DE ACONDICIONAMIENTO ECUFLOW-2020P

