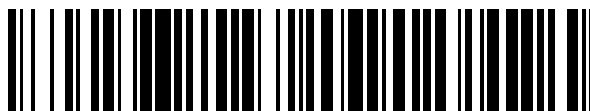


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 368 121**

51 Int. Cl.:

G01M 3/02 (2006.01)

G01M 3/26 (2006.01)

G01M 3/32 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **04777658 .8**

96 Fecha de presentación: **07.07.2004**

97 Número de publicación de la solicitud: **1651939**

97 Fecha de publicación de la solicitud: **03.05.2006**

54 Título: **MÉTODO Y APARATO DE DETECCIÓN DE FUGAS EN UN TANQUE DE ALMACENAMIENTO.**

30 Prioridad:
07.07.2003 US 614748

45 Fecha de publicación de la mención BOPI:
14.11.2011

45 Fecha de la publicación del folleto de la patente:
14.11.2011

73 Titular/es:
MASS TECHNOLOGY, INC.
4302 HIGHWAY 42
KILGORE, TX 75662, US

72 Inventor/es:
WOLFORD, Jimmy;
WOLFORD, Bernie;
LOCKERD, Clark y
SLAUGHTER, Ricky

74 Agente: **Tomas Gil, Tesifonte Enrique**

ES 2 368 121 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Método y aparato de detección de fugas en un tanque de almacenamiento

MENCIÓN A LA SOLICITUD PREVIA

- 5 [0001] Este documento es una solicitud de continuación respecto a la solicitud de patente estadounidense, nº de serie 10/614.748, presentada el 07 de julio de 2003 en la cual se reivindica la prioridad según el título 35 del Código de Estados Unidos, §120 y según las disposiciones del Tratado de Cooperación en materia de patentes.

ANTECEDENTES DE LA INVENCION

1. Campo de la invención.

- 10 [0002] La presente invención tiene que ver con un método y dispositivo para suministrar un sistema de detección de fugas seguro, preciso y rentable en un tanque de almacenamiento; y más particularmente, con un método y dispositivo donde la integridad del contenido de un tanque de almacenamiento se determina mediante mediciones de masa del producto almacenado. 2. Información de antecedentes.

- 15 [0003] Los tanques de almacenamiento tienen un papel fundamental en la economía actual. La economía, a escala mundial, depende del funcionamiento apropiado de estos tanques, ya que su uso está extendido en diferentes industrias y prácticamente en todas las regiones del mundo. Como consecuencia del papel fundamental que estos tanques de almacenamiento desempeñan, la integridad de los tanques es algo muy valioso. Es decir, los propietarios de tanques de almacenamiento están dispuestos a invertir grandes cantidades de dinero en el mantenimiento e inspección de dichos tanques.

- 20 [0004] Estos tanques son de todas las formas y tamaños, se pueden encontrar tanto bajo tierra como en la superficie, y se utilizan para almacenar una amplia gama de materiales. La capacidad de un tanque de almacenamiento oscila de cientos a millones de galones y se utilizan para almacenar una asombrosa variedad de productos; estos tanques de almacenamiento se usan comúnmente para almacenar materiales peligrosos.

- 25 [0005] Como se puede imaginar, existe una amplia gama de problemas asociados al mantenimiento de la integridad del tanque de almacenamiento, particularmente con los tanques de almacenamiento en superficie. Dadas las enormes dimensiones de los tanques en superficie, los productos corrosivos contenidos en los tanques, la increíble masa del producto almacenado, y las condiciones meteorológicas extremas a las que los tanques son sometidos, es obvio que las fugas en tanques de almacenamiento en superficie son un problema común. Usando los criterios para la determinación de umbrales de detección de fugas de la Agencia de Protección Ambiental de Estados Unidos (EPA) de ,05 galones por hora en un tanque subterráneo de 10.000 galones, este umbral equivaldría a un nivel de
30 detección de 15 galones por hora en un tanque en superficie de 80.000. Dado el limitado número de sistemas capaces de cumplir con el umbral de detección de fugas de la EPA en un tanque de almacenamiento subterráneo y las dificultades añadidas asociadas a los tanques en superficie, se ve fácilmente la dificultad de proteger y detectar las fugas.

- 35 [0006] No obstante, la dificultad conocida en la prevención de fugas en tanques de almacenamiento no mitiga los deberes o responsabilidades de las partes responsables. Las tremendas consecuencias económicas y medioambientales y la amenaza de litigios y costes de limpieza asociados a fugas en tanques de almacenamiento obligan a las partes responsables a invertir grandes sumas de dinero en el mantenimiento e inspección de los tanques. Las inspecciones de tanques son costosas con respecto a la cantidad de dinero invertido, el peligro que representan para los inspectores y el medio ambiente y el tiempo de inactividad de la producción. De hecho, estas
40 inspecciones a menudo dejan fuera de servicio un tanque durante más de un mes. La amenaza de responsabilidades legales también obliga a las partes responsables a invertir dinero de forma innecesaria para el mantenimiento de estos tanques. Además, las responsabilidades legales no terminan con los litigios y los costes de limpieza.

- 45 [0007] En la actualidad, las partes responsables son, en algunos países, encarceladas como un resultado directo de fugas en tanques de almacenamiento. Estas fugas han contaminado agua subterránea circundante, parte de la cual sirve como agua potable para los residentes locales. Como tal, las instalaciones vinculadas con tales incidentes se cierran hasta que se puede establecer el cumplimiento con los reglamentos de emisiones más allá de toda duda razonable. A su vez, tal comprobación depende de la comprobación de sistemas y métodos de detección seguros y suficientemente precisos para comprobar dicho cumplimiento. Cada día que las instalaciones cerradas permanecen
50 inoperativas se suma a una ya enorme pérdida de dinero.

[0008] Antes de la presente invención (que se describirá en detalle más adelante), simplemente no existen sistemas o métodos conocidos mediante los cuales se puedan satisfacer los requisitos de detección de fugas. Los sistemas de detección de fugas disponibles actualmente carecen de umbrales de detección lo suficientemente bajos como para detectar fugas hasta los límites superiores permisibles de fuga para tanques de almacenamiento en superficie.

- 55 [0009] Claramente, por las razones expuestas anteriormente, hay una necesidad seria e inmediata de la capacidad de determinar con bastante más precisión que lo actualmente posible a través del uso de sistemas y métodos disponibles actualmente, la presencia y grado de fuga en tanques de almacenamiento en superficie, al menos hasta

el punto de comprobar el cumplimiento de los reglamentos o leyes sobre fugas en tanques de almacenamiento aplicables .

Comparación con tecnologías conocidas en el campo

- 5 [0010] Sistemas de detección de fugas en tanques de almacenamiento se conocen en la técnica; como se divulga en los documentos US4862733 y US5201212A1, aunque estos productos presentan muchos problemas. Los sistemas actuales son imprecisos, o proporcionan datos erróneos para todas o cualquiera de las cuestiones incluyendo: la consistencia del terreno que actúa como la base del tanque, la estratificación de temperatura del producto del tanque, fuentes de ruido externas, dilatación térmica del contenido del tanque, nivel de la capa freática, contaminación previa del terreno y/o la dinámica de la pared del tanque.
- 10 [0011] Además, algunos dispositivos de detección se pueden usar sólo cuando el tanque de almacenamiento está vacío, y ningún sistema o método conocido garantiza una inspección integral del tanque. La forma más común de dicho sistema es una prueba con caja de vacío; @ sin embargo, este sistema está destinado sólo para soldaduras y no se suele aplicar al tanque al completo. El escaneo de suelo con flujo magnético también se usa, pero no es eficaz al examinar el área de la superficie del suelo cercana a las paredes de superficie o donde hay obstrucciones físicas.
- 15 La detección ultrasónica se usa, pero sólo es eficaz para áreas pequeñas de la superficie. La detección de gas también se usa, pero los tipos de materiales almacenados en el tanque pueden obstaculizar este método.
- [0012] Otros sistemas comunes de detección de fugas emplean un sensor de nivel. No obstante, incluso grandes cambios de volumen producen pequeños cambios de nivel, ya que el área de corte transversal de la superficie líquida en estos tanques es muy grande. Esto, combinado con la expansión diferencial y el cambio de temperatura del líquido almacenado y su vapor, hace que este tipo de sistema de detección sea inconsistente y casi inútil.
- 20 [0013] Finalmente, los sistemas de detección de medición de masa se conocen en la técnica. No obstante, los sistemas y métodos relacionados disponibles actualmente no tienen la capacidad de precisión que se indica arriba como crucial en la actualidad (y que, como se describe más abajo, es proporcionada por los sistemas y métodos de la presente invención). Los sistemas actuales de detección de fugas de medición de masa en la técnica están limitados por variaciones de la cubierta del tanque que resultan de los efectos de la temperatura en el chapado de la cubierta del tanque. Como tal, los sistemas de detección de medición de masa conocidos son sólo lo suficientemente sensibles como para usarse en tanques más pequeños, tanques de almacenamiento típicamente subterráneos. No obstante, como se verá en la especificación a seguir, la presente invención supera las variaciones de cubierta de tanque y otros defectos de la tecnología actual conocida en este campo mediante un equipo, técnicas e interpretación para la recogida y corrección de datos.
- 30 [0014] Teniendo en cuenta las graves consecuencias de un fallo en la detección de fugas significantes en un tanque de almacenamiento, actualmente no detectables mediante el uso de sistemas o métodos conocidos, existe una necesidad imperiosa de un sistema y un método con los que poder detectar fugas muy pequeñas incluso en tanques muy grandes, de una manera idealmente segura y rentable.
- 35 [0015] Bien serviría a los que son responsables de mantener la integridad del tanque de almacenamiento proporcionar un sistema de detección seguro, preciso y rentable que no dependa de variables independientes tales como la temperatura fluida, la estratificación fluida, o la estabilización del tanque, y que se pueda utilizar de una manera eficaz conservando así los recursos medioambientales e industriales.

RESUMEN DE LA INVENCION

- 40 [0016] En vista de lo anterior, es un objetivo de la presente invención proporcionar un aparato de detección de fugas en un tanque de almacenamiento con un umbral de detección muy bajo que se pueda utilizar de una manera eficaz conservando así los recursos medioambientales e industriales.
- [0017] Otro objetivo de la presente invención es proporcionar un aparato para la detección segura de fugas en un tanque de almacenamiento
- 45 [0018] Otro objetivo de la presente invención es proporcionar un aparato para la detección precisa de fugas en un tanque de almacenamiento
- [0019] Otro objetivo de la presente invención es proporcionar un aparato para la detección rentable de fugas en un tanque de almacenamiento
- 50 [0020] Otro objetivo de la presente invención es proporcionar un aparato para la detección no intrusiva de fugas en un tanque de almacenamiento
- [0021] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento donde el contenido del tanque de almacenamiento no tiene que ser extraído
- [0022] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que no implique aditivos químicos
- 55 [0023] Otro objetivo de la presente invención es proporcionar un aparato para la detección inmediata de fugas en un tanque de almacenamiento

- [0024] Otro objetivo de la presente invención es proporcionar un aparato para la detección concluyente de fugas en un tanque de almacenamiento
- [0025] Otro objetivo de la presente invención es proporcionar un aparato para la detección cuantitativa de fugas en un tanque de almacenamiento
- 5 [0026] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que no dependa de cambios de temperatura del fluido
- [0027] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que no dependa de la estratificación del fluido
- 10 [0028] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que no requiera tiempo de estabilización del tanque
- [0029] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que requiera sólo una mínima preparación del tanque
- [0030] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que haya sido evaluado por un tercer laboratorio independiente reconocido por la EPA
- 15 [0031] Otro objetivo de la presente invención es proporcionar un método con un umbral de detección muy bajo que se pueda utilizar de una manera eficaz conservando así los recursos medioambientales e industriales.
- [0032] Otro objetivo de la presente invención es proporcionar un método para la detección segura de fugas en un tanque de almacenamiento
- 20 [0033] Otro objetivo de la presente invención es proporcionar un método para la detección precisa de fugas en un tanque de almacenamiento
- [0034] Otro objetivo de la presente invención es proporcionar un método para la detección rentable de fugas en un tanque de almacenamiento
- [0035] Otro objetivo de la presente invención es proporcionar un método para la detección no intrusiva de fugas en un tanque de almacenamiento
- 25 [0036] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento donde el contenido del tanque de almacenamiento no tiene que extraerse
- [0037] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que no implique aditivos químicos
- 30 [0038] Otro objetivo de la presente invención es proporcionar un aparato para la detección inmediata de fugas en un tanque de almacenamiento
- [0039] Otro objetivo de la presente invención es proporcionar un aparato para la detección concluyente de fugas en un tanque de almacenamiento
- [0040] Otro objetivo de la presente invención es proporcionar un aparato para la detección cuantitativa de fugas en un tanque de almacenamiento
- 35 [0041] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que no dependa de cambios de temperatura del fluido
- [0042] Otro objetivo de la presente invención es proporcionar un aparato para la detección de fugas en un tanque de almacenamiento que no dependa de la estratificación del fluido
- 40 [0043] Otro objeto de la presente invención es proporcionar un método para la detección de fugas en un tanque de almacenamiento que no requiera tiempo de estabilización de tanque
- [0044] Otro objetivo de la presente invención es proporcionar un método para la detección de fugas en un tanque de almacenamiento que requiera sólo una mínima preparación del tanque
- [0045] Otro objetivo más de la presente invención es proporcionar un método para la detección de fugas en un tanque de almacenamiento que haya sido evaluado por un tercer laboratorio independiente reconocido por la EPA.
- 45 [0046] La presente invención proporciona una solución segura, extremadamente precisa y rentable para los problemas mencionados anteriormente. Los resultados de las pruebas relacionadas con la presente invención proporcionan una determinación precisa de la integridad del contenido, y en caso de fuga, un índice de fuga volumétrico preciso. La presente invención no está restringida por el tipo de fluido, la temperatura del fluido, el nivel del fluido, o el tamaño del tanque.
- 50 [0047] A diferencia de los productos conocidos en la técnica, la presente invención proporciona un sistema de detección intrínsecamente seguro. El sistema de detección de fugas de la presente invención usa un vataje lo suficientemente bajo (como se establece en el National Electric Code) de manera que los componentes del sistema

se pueden colocar en el área clase I del tanque. De hecho, la presente invención dispone que los componentes del sistema de detección de fugas se coloquen dentro del tanque de almacenamiento. Como se describirá en la especificación a seguir, la colocación de componentes de detección de filtración en el tanque usados en combinación con técnicas de control de sistema y software de corrección de datos proporcionan una precisión no es posible con los productos conocidos en la técnica.

[0048] Además, no se requiere ninguna inspección física de los tanques para la práctica del presente sistema. Por lo tanto, no hay necesidad de drenar, limpiar o entrar en el tanque. Al no necesitar inspección física, ni los inspectores ni el entorno se ven expuestos al contenido del tanque. Al no existir la necesidad de drenar el contenido del tanque, la práctica de la presente invención no produce subproductos peligrosos asociados al proceso de drenaje/limpieza, y se evita el peligro en el transporte y almacenamiento del producto drenado. Finalmente, los sistemas y métodos de la presente invención no requieren la mezcla de aditivos químicos con el contenido del tanque. Por lo tanto, los derrames y fugas casuales se evitan por completo.

[0049] La práctica de la presente invención es rentable. La estructura del tanque o la base y tierra circundante no se alteran, como tal; el tiempo de preparación y los costes de inversión de capital se minimizan. La presente invención no es invasiva y no requiere inspección manual del tanque. Por lo tanto, el funcionamiento del tanque no se obtaculiza, de manera que no se produce una interrupción en la producción. No hay ningún coste relacionado con la manipulación, transporte, eliminación, o almacenamiento del material peligroso extraído. Finalmente, las pruebas se pueden realizar simultáneamente para reducir más el tiempo total invertido e indentificar rápidamente las áreas problemáticas.

[0050] La característica determinante de los sistemas de detección de fugas de medición de masa es la sensibilidad del aparato. Esto es, cuanto más bajo sea el nivel de umbral de detección de fugas de un dispositivo, más eficaz será al detectar las fugas. La presente invención, al utilizar una combinación de técnicas y componentes no conocidos en la técnica, proporciona un umbral de detección de fugas que es mucho inferior al de cualquier dispositivo conocido. De forma más importante, el sistema de la presente invención proporciona un lugar para los componentes de medición de masa en el propio tanque de almacenamiento, eliminando así el ruido exterior asociado a unidades de burbujeo requeridas por otros productos en la técnica. El sistema asegura los componentes de medición de masa dentro de un vacío y mantiene constante la temperatura de los componentes de medición de masa durante todo el proceso de medición. Además, el sistema corrige errores en los datos atribuidos a la dinámica de la cubierta del tanque de almacenamiento y a la imprecisión inherente de los dispositivos de medición de masa. Este proceso de corrección de datos se analizará en detalle en la especificación a seguir.

[0051] Tal y como se menciona, las variaciones en la cubierta del tanque limitan la eficacia de los sistemas de detección de medición de masa actualmente conocidos. Los sistemas y métodos de la presente invención superan las variaciones en la cubierta del tanque mediante la recogida de datos y técnicas de corrección de datos. Primero, los datos se recogen mediante el uso de un transductor de presión de cristal de cuarzo (las especificaciones y uso de este transductor se explicarán con más detalle en la Descripción Detallada de las Formas de Realización Preferentes). Una unidad terminal remota intrínsecamente segura, conectada al transductor de presión, registra los datos de presión durante un período de tiempo (preferiblemente de una a cinco noches). La temperatura atmosférica y la presión barométrica se registran y analizan con precisión para calcular cualquier cambio en la masa del fluido en el tanque. Estos datos experimentan una regresión para dar una línea de pendiente que se convierte en una tasa de fuga, normalmente en galones por hora.

[0052] Los datos generados por el transductor se recogen en una base de 24 horas. Sólo se analizan los datos con una cantidad suficientemente baja de ruido externo. Tales datos se obtienen normalmente por la noche y con unas condiciones meteorológicas buenas. También, el software de corrección de datos recoge el coeficiente de dilatación para cualquier tanque de almacenamiento dado. Esta corrección elimina el efecto de la energía radiante solar en el área del tanque de superficie, que puede afectar adversamente a la medición de la masa del producto almacenado. Los datos nocturnos se corrigen para las condiciones atmosféricas y las variaciones en la cubierta del tanque. Estas mediciones y correcciones permiten al sistema conseguir reiteradamente la exactitud establecida en condiciones reales de forma rutinaria.

[0053] Para conseguir una precisión incluso mayor, el sistema de detección de fugas de la presente invención proporciona unos medios de medición barométricos independientes que registran constantemente la presión barométrica durante el proceso de recogida de datos. Estos medios de medición de presión barométrica independientes usados en combinación con el software de corrección de datos, corrigen cualquier deriva del cero asociada al transductor de presión individual. Es decir, este sistema corrige el error inherente presente en cualquier transductor cuando el transductor se desvía de su calibración inicial.

[0054] El uso del aparato implica asegurar una combinación de componentes de medición de masa precisos, incluyendo un transductor de presión de cristal de cuarzo altamente preciso, en un recipiente sellado al vacío. Este recipiente se baja luego a la superficie inferior de un tanque de almacenamiento. Una referencia diferencial se coloca justo sobre una superficie líquida. La presión, medida en el suelo del tanque ("presión del fondo del tanque") y la presión atmosférica, medida justo sobre la superficie líquida, son registradas por el transductor de presión diferencial microsensible mencionado anteriormente, registradas en tiempo real y procesadas posteriormente usando una rutina de análisis de datos para calcular con precisión cualquier cambio en la masa de fluido contenida en el tanque para

determinar si hay una pérdida. El presente sistema, usando el transductor específico y cuando se usa de la manera y con la interpretación de datos aquí descritas, es capaz de detectar fugas en un tanque de almacenamiento en la superficie en un umbral inferior a ,9 galones por hora con una probabilidad de detección del 95 % en un tanque B de 100.000 mucho más preciso que lo que es posible con cualquier sistema de detección de fugas cuantitativo actualmente disponible. Esto, de forma cuantitativa, equivale a la detección de diferenciales de presión equivalentes a menos de 1/10.000 pulgadas de presión de columna de agua, un nivel de tolerancia necesario para conseguir tales umbrales de detección.

[0055] El método y aparato de la presente invención proporciona un modo eficaz y seguro para detectar fugas muy pequeñas en tanques muy grandes. Particularmente, la presente invención proporciona una mejora formidable en la exactitud y el umbral de detección de fugas, que permite a sus usuarios conseguir resultados superiores a los actualmente posibles.

[0056] Así, satisfaciendo los objetivos anteriores, una forma de realización de la presente invención proporciona sistemas y métodos para resolver cada uno de los problemas planteados con los sistemas de detección de fugas en un tanque de almacenamiento actualmente disponibles.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0057] El anexo A es un ejemplar impreso del código fuente del programa de ordenador denominado en este caso como el programa RTU.

La Fig. 1 es un diagrama de bloques que describe el trazado general del presente sistema de detección de fugas.

La Fig. 2 es una vista sagital y en sección transversal del receptáculo ("medios de cubierta de protección") del sistema de detección de fugas.

DESCRIPCIÓN DETALLADA DE LA FORMA DE REALIZACIÓN PREFERENTE

[0058] En los dibujos y la descripción que sigue, en referencia a la Figura 1, una forma de realización preferente de un sistema de detección de fugas en un tanque de almacenamiento según la presente invención se designa generalmente como sistema 10. Una forma de realización de la presente invención se muestra para incluir un recipiente sellado al vacío 12, que aloja y protege una pluralidad de componentes de medición de masa y componentes de control de sistema. En la forma de realización preferente, el recipiente sellado al vacío 12 está hecho de un metal sustancialmente no corrosivo (aluminio, por ejemplo), sin embargo, cualquier material que sea resistente a la corrosión y ofrezca suficiente protección a los componentes incluidos es adecuado para su uso en la presente invención. El recipiente 12 está directamente sumergido en un tanque de almacenamiento 60 y reposa sobre la superficie inferior 62 del tanque de almacenamiento. El recipiente 12 contiene además una boquilla de sellado al vacío 14 y una apertura para el transductor en el lateral superior 20. La boquilla de sellado al vacío 14 permite que los medios de comunicación pasen del interior del recipiente al exterior del recipiente manteniendo la integridad del interior del recipiente al vacío. La boquilla de vacío 14 contiene además una apertura para los medios de medición de presión barométrica 16 y una apertura para el transductor en el lateral inferior 18.

[0059] En su extremo próximo un tubo flexible del recipiente 15 forma una sello hermético líquido con la boquilla de sellado al vacío 14. Extendiéndose desde la boquilla de sellado al vacío 14, el tubo flexible del recipiente 15 pasa a través de un hueco de la superficie superior del tanque de almacenamiento 64 a un área exterior de la región de clase I del tanque de almacenamiento 60 (la región de clase I se refiere a las zonas peligrosas designadas por el National Electric Code donde sólo se pueden introducir niveles de vataje de potencia inferior a determinados niveles prescritos). El tubo flexible del recipiente 15 sirve como conducto para la extensión de los medios de comunicación a través de la boquilla de vacío 14 y como referencia atmosférica cuando se usa como tubo flexible de referencia de los medios de medición de presión barométrica. El tubo flexible del recipiente 15 permite a la apertura del lateral inferior para el transductor 18 y a la apertura de los medios de medición de presión barométrica 16 estar directamente expuestas a presión atmosférica mientras se mantiene un sello hermético fluido con la boquilla de sellado al vacío 14, conservando así la integridad del recipiente al vacío 12.

[0060] Dentro del recipiente sellado al vacío 12 se encuentra el transmisor de presión diferencial 22. En la forma de realización preferente, el transmisor de presión diferencial 22 está compuesto por un transductor de presión de cristal de cuarzo de alta precisión 24. El transductor 24 contiene un cristal de cuarzo de oscilación y tiene una resolución de 1x10⁻⁸, como se conoce en la industria. Tal unidad está disponible en Paroscientific, Inc. como el modelo n°. 6015-G. La resolución definitiva alcanzable con un transductor está limitada por su nivel de ruido. El sistema 10 reduce inmensamente el ruido aumentando así la resolución del transmisor 24. En el sistema 10, el transmisor 22 se ha modificando con respecto a su configuración original de modo que se puede sumergir directamente en el tanque de almacenamiento 60. Esta modificación ha eliminado la dependencia de cualquier unidad de borboteo (eliminando así el ruido asociado a tales unidades) como es necesario para otros productos. Como se describirá con más detalle en esta sección, el transductor 24 se mantiene a una temperatura constante y se asegura al vacío para reducir más el ruido.

[0061] El transductor de presión de cristal de cuarzo 24 está además compuesto por el lateral inferior del transductor 26. El lateral inferior del transductor 26 es una referencia diferencial que recibe el valor de presión barométrica en la

superficie líquida. El tubo del lateral inferior del transductor 28 forma un sello hermético de aire en su extremo próximo con el lado inferior del transductor 26 y se extiende a través del recipiente al vacío 12 donde forma un sello hermético de aire en su extremo distal en la apertura lateral inferior del transductor 18 de la boquilla de sellado al vacío 14. El tubo del lateral inferior del transductor 28 permite al lateral inferior del transductor 26 recibir la presión barométrica del punto de referencia en la superficie líquida mientras permite al recipiente 12 permanecer al vacío.

[0062] El transductor de presión de cristal de cuarzo 24 está además compuesto por el lateral superior del transductor 30. El lateral superior del transductor de presión de cristal de cuarzo 30 es un punto de referencia de presión, que mide la suma de la presión barométrica e hidrostática en la superficie inferior del tanque 30. El lateral superior del transductor 30 contiene un tubo saliente del lateral superior del transductor 32. En la forma de realización preferente, el tubo del lateral superior del transductor 32 se llena de un líquido de detección de presión y se extiende a través de la apertura del lado superior del transductor 20 donde se transfiere al producto contenido en el tanque 60. El tubo del lateral superior del transductor 32 está rodeado por un tubo de conexión 34. En la forma de realización preferente, el tubo de conexión 34 se desliza a lo largo del tubo del lateral superior 32 y forma un sello hermético fluido en la apertura del lateral superior 20. El tubo de conexión 34 permite al tubo del lateral superior 32 extenderse a través de la apertura del lateral superior 20 mientras se mantiene la integridad del recipiente al vacío 12.

[0063] El transductor 24 resta el valor recibido en el lateral inferior del transductor 26 al valor recibido en el lateral inferior del transductor 30 para obtener la presión ejercida por la masa del producto almacenado. El transmisor 22, por comunicación digital, manda luego esta información procesada al registrador de datos 30. Estos datos se transmiten por los medios de transferencia de datos 23. En la forma de realización preferente, el medio de transferencia de datos 23 es un cable estándar de comunicaciones en bus. No obstante, se podría prever fácilmente unos medios de transferencia de datos tales como comunicación inalámbrica que funcionarían también por igual. El medio de transferencia de datos 23 se extiende desde la salida del transmisor de presión diferencial 22 a través de la boquilla de sellado al vacío 14 y continúa, separado del contenido del tanque de almacenamiento 60 por el tubo flexible del recipiente 15, al registrador de datos 30.

[0064] También dentro del recipiente 12 se encuentra el transmisor de corriente 34. El transmisor de corriente 34 sirve como parte de un esquema de regulación de temperatura usado para mantener el contenido del recipiente 12 a una temperatura constante durante el proceso de recogida de datos. El transmisor de corriente 34, en la forma de realización preferente, acciona un calefactor resistente 36 mediante un simple sistema de control on/off. Un disipador de calor 38 actúa en combinación con el transmisor de corriente 34 y el calefactor resistente 36 actúa para regular la temperatura del recipiente 12. Mientras el esquema de regulación de temperatura anterior se ha descrito haciendo referencia a una forma de realización, se podrían imaginar fácilmente otros esquemas de regulación de temperatura que funcionarían también por igual. El medio de transferencia de datos 39 se extiende desde la salida del transmisor de corriente 34 a través de la boquilla de sellado al vacío 14 y continúa, separado del contenido del tanque de almacenamiento 60 por el tubo flexible del recipiente 15, al registrador de datos 30. En la forma de realización preferente, el medio de transferencia de datos 39 es un cable estándar de comunicaciones en bus. No obstante, se podrían concebir fácilmente unos medios de transferencia de datos tales como una comunicación inalámbrica que funcionarían también por igual.

[0065] El uso de este esquema de regulación de temperatura para mantener el transmisor 22 a una temperatura constante aumenta aún más la precisión del aparato actual. La temperatura absoluta a la que el transmisor 22 se mantiene no es crítica, ya que la constancia de la temperatura afecta a la integridad de las mediciones del sujeto. Como una cuestión de sentido práctico y económico, la temperatura del transmisor 22 se mantiene, según el modo actualmente preferente de la presente invención, a una temperatura de aproximadamente 1°F sobre la temperatura ambiente del producto (petróleo o gasolina, por ejemplo) en el tanque 60. Si, por ejemplo, el producto está a 50°F, el transmisor 22 se mantiene a 51°F, si el producto está a 90° F, el transmisor 22 se mantiene a 91°F y así sucesivamente.

[0066] También se encuentra dentro del recipiente 12 el medio de medición de presión barométrica 40. El medio de medición de presión barométrica 40 sirve como referencia independiente para la presión atmosférica real. En la forma de realización preferente, el medio de medición de presión barométrica 40 puede ser cualquier barómetro estándar que mande señales que sean procesadas por el registrador de datos 30. El medio de medición barométrica 40 es muy útil para aumentar la precisión del sistema 10. Todos los transductores disminuyen en exactitud con el tiempo ya que pierden la calibración con respecto a la presión atmosférica real. Esto se conoce como deriva del cero. No obstante, la presente invención emplea un medio de medición barométrica 40 que sirve como medida independiente de presión atmosférica real permitiendo así la corrección de datos durante cualquier período de tiempo prolongado. Como se comentará en esta sección, la corrección de los datos usando valores tomados del medio de medición de presión barométrica 40 se basa en el software y aumenta enormemente la precisión de la invención actual.

[0067] El tubo del medio de medición barométrica 42 forma un sello de aire hermético en su extremo próximo al medio de medición barométrica 40 y se extiende a través del recipiente al vacío 12 donde éste forma un sello hermético en su extremo distal a la apertura del medio de medición barométrica 16 de la boquilla de sellado al vacío 14. El tubo del medio de medición barométrica 42 permite al medio de medición barométrica 40 recibir la presión barométrica del punto de referencia en la superficie del líquido dentro del tanque de almacenamiento 60, mientras

permite que el interior del recipiente 12 (con el transmisor 22 instalado en su interior) permanezca al vacío para eliminar sustancialmente cualquier variación causada por el medio ambiente en el rendimiento del instrumento). El medio de transferencia de datos 43 se extiende desde la salida del medio de medición de presión barométrica 40 a través de la boquilla de sellado al vacío 14 y continúa, separado del contenido del tanque de almacenamiento 60 por el tubo flexible del recipiente 15, al registrador de datos 30. En la forma de realización prefente, el medio de transferencia de datos 43 es un cable estándar de comunicaciones en bus. No obstante, se podrían concebir fácilmente unos medios de transferencia de datos tales como la comunicación inalámbrica que funcionaría también por igual.

[0068] Aunque no es necesario, el ordenador remoto 30 se aloja por lo general en un apartado diferente, tal como la unidad de campo 50. Conforme a las rutinas descritas más adelante y el ejemplar del código del ordenador representado en el Anexo 1 adjunto al presente documento e incorporado aquí por referencia, el registrador de datos procesa los datos recibidos desde el transmisor 22, el transmisor de corriente 34, el calefactor resistente 36, el disipador de calor 38, y el medio de medición de presión barométrica 40. El registrador de datos 30 se comunica con el ordenador remoto 70 por el medio de transferencia de datos 72.

[0069] El software comienza a funcionar con el inicio de la recogida de datos en el fondo del tanque, junto con las condiciones medioambientales y atmosféricas. Los datos se recogen automáticamente mediante la programación controlada del ordenador en un determinado período de tiempo, preferiblemente de 36 a 60 horas. El tiempo de la prueba depende del tamaño del tanque y de las condiciones atmosféricas del lugar. En la forma de realización prefente, el medio de transferencia de datos 72 es un cable de comunicaciones de bus estándar. No obstante, uno podría fácilmente prever unos medios de transferencia de datos tales como la comunicación inalámbrica que funcionaría también por igual.

[0070] Como se comentará e ilustrará de aquí en adelante, el ordenador remoto 70 contiene un software que ejecuta regresiones lineales de datos recibidos desde el registrador de datos 30. Esta regresión detecta cambios minúsculos en la masa del producto almacenado, indicando así la presencia de la más pequeña de las fugas. Conforme aumenta la compilación de datos, la regresión se vuelve más precisa. El módulo de post procesamiento y el software del ordenador remoto 70 es independiente del registrador de datos 30.

[0071] Hay dos programas o módulos de software implicados en el sistema de detección de fugas en un tanque de almacenamiento de la presente invención: el programa RTU y el programa de regresión lineal.

[0072] El programa RTU es ejecutado por el registrador de datos 30 y es el responsable de obtener (rutina 100) y corregir (rutina 200) las lecturas de presión del transmisor 22, controlar la temperatura del transmisor 22 (rutina 300), calcular los ajustes para las expansiones de la cubierta de tanque (rutina 400), obtener la temperatura del transmisor 22 (rutina 500), y almacenar los datos. Los datos adquiridos por el programa RTU se almacenan dentro del registrador de datos 30 en memoria no volátil 31.

[0073] El propósito del programa RTU es interrogar a un transmisor de presión diferencial inteligente (transmisor 22) mediante una conexión en serie. La lectura de presión del transmisor 22 es la diferencia de la lectura de presión del lateral inferior del transductor 26 y el lateral superior del transductor 30. Este valor de presión se modifica por dos variables adicionales para mejorar la exactitud de la lectura. El programa ejecuta la corrección de la presión barométrica; un barómetro análogo (tal como el medio de medición de presión barométrica 40) proporciona la señal que se envía para corregir la presión de transmisión por errores debidos a cambios en la presión barométrica, medidos en la superficie superior del contenido del tanque de almacenamiento 60. También, el programa supervisa la temperatura ambiente para compensar los cambios en el diámetro del tanque que de no ser así desajustarían la interpretación de datos, destinado solamente a detectar variaciones en el contenido del tanque de almacenamiento 60 causado por fugas. Cualquier cambio en el diámetro del tanque se registra en los cálculos del transmisor 22, atribuyendo adecuadamente todas las variaciones en presión diferencial (ya corregidas por las variaciones en presión atmosférica, como se ha mencionado anteriormente) a las variaciones en el contenido del tanque de almacenamiento 60, tales como fugas.

[0074] La rutina 100, obteniendo las lecturas de presión del transmisor 22, se realiza cada minuto de la siguiente manera: en el paso 101 se envía una orden al transmisor 22 para obtener una nueva muestra de presión, en el paso 102 se envía una orden para esperar la nueva muestra, en el paso 103 se envía una orden para pedir la nueva muestra. Los datos vuelven como un número psi del código ASCII.

[0075] La rutina 200, ajustando la lectura de presión del transductor 24 según el medio de medición barométrica 40, se realiza de la siguiente manera: en el paso 201 el usuario introduce parámetros de instalación reflejando: (1) un factor de corrección barométrico predeterminado que está determinado en laboratorio para establecer cada transmisor 22, y para más tarde permitir la calibración del transmisor del Azero point@ 22; (2) el coeficiente de expansión para el tanque de almacenamiento 60 (un factor fácilmente calculado por personas bastante expertas en el campo pertinente, aplicando principios de ingeniería de materiales comunes a estándares pertenecientes al diseño y materiales de tanques de almacenamiento); (3) la temperatura ambiente en el lugar de instalación en el tiempo de la instalación (para usarse al calcular las variaciones dimensionales en el tanque 60 según el coeficiente de dilatación mencionado; y (4) la densidad relativa del producto contenido en el tanque de almacenamiento 60. El usuario también introduce el ajuste de temperatura deseada para el calefactor resistente 36.

[0076] En el paso 202 se realizan los siguientes cálculos:

$$\text{Hadj} = \text{H} - \text{Bcf} * (\text{B} - 14.5)$$

$$\text{Hadj} / \text{sg}$$

5

donde (para el presente estudio, aunque no se refleje precisamente en los mismos términos en el código de fuente anexo) Hadj = presión de descarga derivada o ajustada; H = presión de descarga en pies en agua (Hpsi x 2,037); Hpsi = presión de descarga en libras por pulgada cuadrada (medida en el lateral superior 30 del transmisor 22); B = lectura de barómetro en libras por pulgada cuadrada; sg = densidad relativa del contenido del tanque 60; y Bcf = factor de corrección barométrica. El número 14,5 es un número un tanto arbitrario que se acerca bastante a un intervalo previsto de presión barométrica real medida. Este factor es sustraído de la presión barométrica medida para reducir determinadas cifras calculadas a un nivel menor y más factible para el tratamiento posterior (regresión lineal, etc.) en el rastreo de diferencias diminutas en el contenido del tanque de almacenamiento. En el paso 203 estos datos de presión derivados o ajustados se utilizan para calcular la masa del producto en el tanque basándose en el diámetro del tanque.

[0077] La rutina 300, controlando la temperatura del transductor 24, se realiza de la siguiente manera: en el paso 301 la salida digital al transmisor de corriente 34 se activa cuando la lectura de temperatura de la entrada análoga del disipador 38 está 1 grado por debajo del valor de ajuste de temperatura, en el paso 302 la salida digital al transmisor de corriente 34 se desactiva cuando la temperatura es 0,1 grado sobre el valor de ajuste.

[0078] La rutina 400, ajustando la masa de contenido previamente derivada para la expansión de la cubierta del tanque, se realiza de la siguiente manera: en el paso 401 la temperatura ambiental se calcula según el promedio para obtener una temperatura para usar al calcular el cambio en el diámetro del tanque, este cálculo requiere que el coeficiente de dilatación y el diámetro del tanque sean introducidos por el usuario o bien en la puesta en marcha (como se ha mencionado previamente) o bien en cualquier momento, el resultado obtenido se utiliza para ajustar la masa total en el tanque para las indicaciones falsas erróneas efectuadas por cambios en el contenido del tanque 60, para producir puramente variaciones relacionadas con fugas (sin asumir ninguna adición de intensidad o eliminación de contenidos por otros medios).

[0079] La rutina 500, obteniendo la temperatura del transductor 24, se realiza de la siguiente manera: en el paso 501 se envía una orden al transmisor 22 para obtener una nueva temperatura, en el paso 502 se envía una orden para esperar la lectura de la temperatura, en el paso 503 se envía una orden para pedir que el transmisor 22 envíe la nueva lectura.

[0080] Finalmente, el programa RTU es el responsable del almacenamiento de los datos. La cantidad de almacenamiento de datos disponible determina cuántos días de datos se almacenan para la recuperación. Se almacena un registro por minuto. La organización de la fecha es por días. El registro por cada minuto incluye: (1) el contenido del tanque en libras (como un número de punto flotante, formato IEEE de 32 bits), (2) la presión barométrica (como un número entero de x100-16 bits), (3) la temperatura ambiente (como un número entero de x100-16 bits). Otros datos, como temperaturas de transductor precedentes, el diámetro del tanque y los coeficientes de dilatación del tanque, también se pueden almacenar como datos corrientes.

[0081] El segundo programa de software del sistema de detección de fugas en un tanque de almacenamiento de la invención reivindicada es el programa de regresión lineal. El ordenador remoto 70 ejecuta este programa. La rutina 700, la regresión lineal de los datos recibidos, se realiza de la siguiente manera: en el paso 701 el fichero de datos creado por el programa RTU se crea y se realiza el análisis de fugas, en el paso 702 las secciones de datos se seleccionan para la calidad del tiempo atmosférico durante esa sección particular de datos -sólo los datos nocturnos se usan típicamente para minimizar los ruidos externos en el análisis, en el paso 703 un ajuste lineal mejor se usa para los puntos de datos en cada sección de datos- cuando las secciones de datos que representan duraciones de nivel de ruido apropiadamente bajas se incluyen en la mejor regresión de datos de ajuste, la pendiente de la línea de ajuste indica mejor el índice de fuga. El cálculo de la regresión lineal y un mejor ajuste son sencillos y se podrían realizar con un software común tal como Microsoft Excel.

[0082] Se cree que, mientras sea eficaz y seguro, el presente dispositivo evitará inconvenientes significantivos y proporcionará una utilidad sustancial a quienes deseen detectar fugas en tanques de almacenamiento. En concreto, el presente dispositivo permitirá detectar fugas muy pequeñas en tanques de almacenamiento muy grandes de una manera rentable y consistente.

[0083] Aunque la invención ha sido descrita haciendo referencia a formas de realización específicas, estas no se debe interpretar en un sentido limitado. Varias modificaciones de las formas de realización descritas, al igual que formas de realización alternativas de la invención que estén dentro del campo de la invención serán evidentes para los expertos en la técnica en referencia a las reivindicaciones.

5 ANEXO A

[0084]

```

/*****
*
* Mass Technologies - Data logger
*
*
*****/

// use 2700H for root memory reserve

#include eziobl17.lib
#include aasc.lib
#include aascz0.lib
#include aascz1.lib
#include aascsc.lib
#include vdriver.lib
#include msz.lib
#include flashstore.lib

#define pi 3.1415927

////////////////////////////////////
//
// Configuration Data - stored in Flash
//
////////////////////////////////////

float calib[6][2] = {
//      {-4979, 16.68}, // min,max for Baro sensor psi for 0-10V converter
//      {0.0, 36.0},
//      {00.0, 100.0}, // for xducer temp degF (0-100)
//      {00.0, 150.0}, // for tanktemp 1 (0-150)
//      {00.0, 150.0}, // for tanktemp 2 (0-150)
//      {00.0, 150.0}, // for tanktemp 3 (0-150)
//      {00.0, 150.0} // for tanktemp 4 (0-150)
};

////////////////////////////////////
//
// Installation Data - stored in Flash
//
////////////////////////////////////

```

```

float desired_xducer_temp = 60.0;
float Bcf                = 0.004;
float coef_of_expn       = 1.000;
float tank_dia           = 60.0;
float sg                 = 1.00;
float initial_temp       = 60.0;
float day_start          = 0000;

////////////////////////////////////
//
// Logging Data - Current data stored in ram, previous days
//                stored in flash
//
////////////////////////////////////

struct {                                // holds 6 hours of logging data
    int initial1, initial2; // set to 1234, (rec#) when memory is initialized
    int year;
    int month;
    int day;
    int hour;
    int min;
    float mass[240];
    unsigned int baro[240];
    int avg_temp[240];
    int xducer_temp[240];
} datastorage;

union { datastorage f; char flash[2560]; } data;

union { float f; unsigned int i[2]; } modbusd;

////////////////////////////////////
//
// Runtime data storage -- it is not necessary to save
//                this between runs
//
////////////////////////////////////

#define BUFFSIZE 64
struct _Channel *chan1;                // channel for serial baro sensor
char readBuf1[BUFFSIZE];              // storage buffer for data from sensor
char writeBuf1[BUFFSIZE];             // storage for data to sensor
struct _Channel *chan4;               // channel for serial to terminal
char readBuf4[BUFFSIZE];              // storage buffer for data from terminal

```

```

char writeBuf4[BUFSIZE];           // storage for data to terminal
char eol[3] = {13, 10, 0};        // end of line (carriage return, line feed) string
char baro_cmnd[24];                // command to baro sensor
char baro_resp[64];                // response from baro sensor
char serial_temp[24];              // latest temp from baro sensor
char serial_pressure[24];          // latest pressure from baro sensor
char term_in[64];                  // line received from terminal
char term_out[64];                 // data to send to terminal
char date_time[24];
shared float avg_tank_temp;        // average of the four tank temp sensors
shared float xducer_temp_min;     //
shared float xducer_temp_max;     //
shared float xducer_temp_avg;     //
shared float xducer_ytemp_min;    //
shared float xducer_ytemp_max;    //
shared float xducer_ytemp_avg;    //
int heater;
int start_hour, start_min;        //
int start_year;
char *flashptr;
int current_record;
int modbusaddr;

union { datastorage f; char flash[2560]; } old;

////////////////////////////////////
//
// Current sensor values -- updated as new data arrives
//
//
////////////////////////////////////

shared float cur_xducer_sensor;
shared float cur_xducer_temp;
shared float cur_local_baro;
shared float cur_pod_temp;
shared float cur_tank_temp_1;
shared float cur_tank_temp_2;
shared float cur_tank_temp_3;
shared float cur_tank_temp_4;

shared float mass;
shared float baro;

int year, month, day, hour, minute;

```

```

int monthdayhour;

////////////////////////////////////
//
// read date/time from RTC
//
//
////////////////////////////////////

void read_datetime(){
    struct tm time;

    tm_rd(&time);
    year = time.tm_year + 1900;
    month = time.tm_mon;
    day = time.tm_mday;
    hour = time.tm_hour;
    minute = time.tm_min;
    monthdayhour = month*1000 + day*100 + hour;
    return;
}

void write_datetime(){
    struct tm time;

    time.tm_year = year-1900;
    time.tm_mon = month;
    time.tm_mday = day;
    time.tm_hour = hour;
    time.tm_min = minute;
    tm_wr(&time);
}
////////////////////////////////////
//
// Read record# into root memory
//
//
////////////////////////////////////

void flash_read(int record){

    long a;

    a=flashstorage + (record*sizeof(data.flash));

```

```

xmem2root(
    a,
    &old.flash,
    sizeof(data.flash)
);
return;
}

////////////////////////////////////
//
// Write current logging data into
// flash memory at record#
//
////////////////////////////////////

void flash_write(int record){
    int r;

    r=WriteFlash(
        flashstorage+ ( record*sizeof(data.flash) ),
        data.flash,
        sizeof(data.flash)
    );
    // if (r!=0)
    // printf("\nFlash error %d, writing record %d",r,record);
    return;
}

////////////////////////////////////
//
// Ready storage system for use
//
//
////////////////////////////////////

void flash_restart(){
    // go down flash storage, if any record is not yet intialized,
    // initialize it
    int i;

    for (i=0; i<36; i++) {
        flash_read(i);

        if (old.f.initial1!=1234 || old.f.initial2!=i) {
            old.f.year = 0;

```

```

    old.f.day = 0;
    old.f.hour = 0;
    old.f.initial1=1234;
    old.f.initial2=i;
    WriteFlash(
        flashstorage+ ( i*sizeof(old.flash) ) ,
        old.flash,
        sizeof(old.flash)
    );
}
}
return;
}

```

```

////////////////////////////////////
//
// return index number for record
//
//
////////////////////////////////////

```

```

getindex(int h, int m){
    return ( ((h*60+m)%240) );
}

```

```

////////////////////////////////////
//
// Return index # for current data point
//
//
////////////////////////////////////

```

```

getcurrentindex(){
    read_datetime();
    return(getindex(hour,minute));
}

```

```

////////////////////////////////////
//
// Format a date/time string from

```



```

// record # and index in a flash record
// return: year-mm-dd hh:mm
// if record # is < 0, use current (RAM) record
// if record # is not <0, use flash record that should
// have been copied to RAM as 'old'
////////////////////////////////////

int formatdt(int recnum, int index) {
    int lyear, lmonth, lday, lhour, lminute;

    if (recnum < 0) {
        lyear = data.f.year;
        lmonth = data.f.month;
        lday = data.f.day;
        lhour = data.f.hour;
    } else {
        lyear = old.f.year;
        lmonth = old.f.month;
        lday = old.f.day;
        lhour = old.f.hour;
    }
    lminute = index;
    lhour = lhour + ((int)lminute/60);
    lminute = lminute % 60;
    sprintf(date_time, "%4d-%02d-%02d %02d:%02d", lyear, lmonth, lday, lhour,
        lminute);
    if (lyear < 2001 || lyear > 2100) lyear = 0;
    return(lyear);
}

////////////////////////////////////
//
// Scale input values into engineering units
//
//
////////////////////////////////////

float scale(float x, int sensor_number){
    float m, b;
    float result;

    if (sensor_number > 0) x = (x-2.0)*1.25; // base inputs > 1 on 4-20mA sigs.

    m = (calib[sensor_number][1]-calib[sensor_number][0])/10;
    b = calib[sensor_number][0];

```

```

result = m*x + b;

//printf("scaled: %f\n", result);
return result;
}

////////////////////////////////////
//
// Head correction for barometric pressure changes
// returns corrected head in feet
//
////////////////////////////////////

float HcBaro
( float Hpsi,      // Head in psi
  float B,        // barometer in psi
  float sg         // specific gravity
) {

    float H;      // head in feet

    H = Hpsi * 2.307; // result in feet of water
    H = H - Bcf * ( B - 14.5 ); // result is in

    return H / sg;
}

////////////////////////////////////
//
// Head correction for temperature change
// returns Head correction amount
//
////////////////////////////////////

float HcTemp
( float Di,      // initial diameter in feet
  float Ti,      // initial temp in deg F
  float T,       // current temp in deg F
  float sg,      // specific gravity
  float H,       // current Head in feet
  float fac      // coef of expansion
)
{
    float Ci;    // initial circ in feet

```

```

float Cf;           // final circ in feet
float Df;           // final dia in feet
float Ai;           // initial area ft^2
float Af;           // final area ft^2
float dV;           // change in tank volume

Ci = Di * pi;
Cf = Ci + ( Ci * 5.6e-6 * fac * (T - Ti));
Df = Cf / pi;
Ai = Di*Di / 4. * pi;
Af = Df*Df / 4. * pi;
dV = (Ai - Af) * H;
return H - (dV / Af);
}

clear_local_store(){
  int i, h;

  read_datetime();
  h = hour - (hour % 4);

  data.f.initial1 = 1234;
  data.f.initial2 = -1;

  for (i=0; i<240; i++) {
    data.f.mass[i]=-1.0;
    data.f.baro[i]=0;
    data.f.avg_temp[i]=0;
  }
  data.f.year = year;
  data.f.month = month;
  data.f.day = day;
  data.f.hour = h;
  data.f.min = 0;
}

check_local_store(){
  int i, h;

  // check data store in ram -- if it is not correct for the current
  // data, write to flash, and initial RAM

  read_datetime();
  h = hour - (hour % 4);

```

```

if ( data.f.year != year || data.f.month != month
    || data.f.day != day || data.f.hour != h
    || data.f.initial1 != 1234 || data.f.initial2 != -1)
{
    flash_write(current_record);
    current_record = (current_record+1)%36;
    clear_local_store();
}
}

main() {
    int pos;                                // used for baro serial input processing
    int i, index;
    int rr, ii;
    float f;
    union { float f; char c[4]; } u;
    float Hc;

    int lastminute;                        // used to track time changes

    VdInit();
    eioBrdInit(0);

    if (current_record < 0) current_record = 0;
    current_record %= 36;

    // open channel for baro serial transmissions
    chan1 = aascOpen(DEV_Z1, 0, ASCI_PARAM_8N1+ASCI_PARAM_1200*8,NULL);
    aascSetReadBuf (chan1, readBuf1, sizeof(readBuf1));
    aascSetWriteBuf(chan1, writeBuf1, sizeof(writeBuf1));
    aascTxSwitch(chan1,1);
    aascRxSwitch(chan1,1);

    // open channel for serial to terminal
    chan4 = aascOpen(DEV_SCC, 0, SCC_B+SCC_8N1+SCC_1200*8,NULL);
    aascSetReadBuf (chan4, readBuf4, sizeof(readBuf4));
    aascSetWriteBuf(chan4, writeBuf4, sizeof(writeBuf4));
    aascTxSwitch(chan4,1);
    aascRxSwitch(chan4,1);

    // open up modbus comms
    msaZ0(modbusaddr,9600,0);              // modbus ascii Rs232, 9600,8,n,1

```

```

strcpy(term_out,"MTC data logger. v1.30 \r\n\r\n*");

read_datetime();
lastminute = minute;
xducer_temp_min = 9999.9;
xducer_temp_max = 0.0;
xducer_temp_avg = 0.0;
xducer_ytemp_min = 9999.9;
xducer_ytemp_max = 0.0;
xducer_ytemp_avg = 0.0;
cur_local_baro = 14.5;

start_hour = (int)(day_start/100);
start_min = (int)(day_start - (start_hour*100));

while(1==1) {

    // do analog input processing
    costate analog_inputs always_on {
        f=eioBrdAI(0);
        cur_local_baro = ((cur_local_baro*10)+scale(f, 0))/11.0;
        f=eioBrdAI(1);
        cur_pod_temp = scale(f,1);
        f=eioBrdAI(2);
        cur_tank_temp_1 = scale(f,2);
        f=eioBrdAI(3);
        cur_tank_temp_2 = scale(f,3);
        f=eioBrdAI(4);
        cur_tank_temp_3 = scale(f,4);
        f=eioBrdAI(5);
        cur_tank_temp_4 = scale(f,5);
        waitfor (DelayMs(1000));
    }

    costate data_processing always_on {
        read_datetime();
        if (lastminute != minute) {                // a new minute -- do processing

            // watch for new day -- reset avg/min/max values
            if (hour == 0 && minute == 0) {
                xducer_ytemp_min = xducer_temp_min;
                xducer_ytemp_max = xducer_temp_max;
                xducer_ytemp_avg = xducer_temp_avg;
                xducer_temp_min = 9999.9;
            }
        }
    }
}

```

```

    xducer_temp_max = 0.0;
    xducer_temp_avg = cur_xducer_temp;
}

// record min/max/avg baro temp
if (xducer_temp_min > cur_xducer_temp) xducer_temp_min = cur_xducer_temp;
if (xducer_temp_avg < 1.0) xducer_temp_avg = cur_xducer_temp;
if (xducer_temp_max < cur_xducer_temp) xducer_temp_max = cur_xducer_temp;
xducer_temp_avg = ((xducer_temp_avg*9.0)+cur_xducer_temp)/10.0;

// create avg tank temp
//avg_tank_temp =
(cur_tank_temp_1+cur_tank_temp_2+cur_tank_temp_3+cur_tank_temp_4)/4.0;    //removed
9/10/02
avg_tank_temp = cur_tank_temp_1;

// calc mass
Hc = HcBaro ( cur_xducer_sensor , cur_local_baro , sg );
Hc = HcTemp ( tank_dia, initial_temp, avg_tank_temp, sg, Hc, coef_of_expn);
mass = Hc;
lastminute = minute;

// check for room in RAM, write out record in flash if full
check_local_store();           // reset local data store

// save results
index = getcurrentindex();           // get index for data
data.f.mass[index] = mass;
data.f.baro[index] = (int)(cur_local_baro*1000); //cur_xducer_sensor*2307); changed
2/18/02
data.f.avg_temp[index] = (int)(avg_tank_temp*100);
data.f.xducer_temp[index] = (int)(cur_xducer_temp*100);
}
waitfor(DelayMs(250));
}

// run heater relay
costate heater_relay always_on {
    if (cur_pod_temp < desired_xducer_temp-0.1) {
        eioBrdDO(BankB(0),1);    // turn on output
        heater = 1;
    }
    if (cur_pod_temp > desired_xducer_temp+0.1) {
        eioBrdDO(BankB(0),0);    // turn off output
        heater = 0;
    }
}

```

```

    }
    waitfor (DelayMs(1000));
}

// display heater status by blinking on board LED when heater should be on
costate heater_status always_on {
    if (heater) {
        switchLED(0);
        waitfor (DelayMs(100));
        switchLED(1);
        waitfor (DelayMs(100));
    } else {
        switchLED(1);
        waitfor (DelayMs(100));
        switchLED(0);
        waitfor (DelayMs(2000));
    }
}

// do serial output to baro sensor
costate baro_output always_on {
    // if output string is non-zero in length, output it to the sensor
    if (strlen(baro_cmnd)>0) {
        //printf("S:%s:\n", baro_cmnd);
        aascFlushRdBuf(chan1); // flush out input buffer
        baro_resp[0]=0; // zero out response string
        aascWriteBlk(chan1,baro_cmnd,strlen(baro_cmnd),1);
        aascWriteBlk(chan1,eol,2,1); // send end of line
        baro_cmnd[0]=0; // zero out command
    }
    yield;
}

// do serial input from baro sensor
costate baro_input always_on {
    if (aascScanTerm(chan1, 0x0a)>0){
        pos = aascReadBlk(chan1, baro_resp,sizeof(baro_resp),0);
        baro_resp[pos] = 0; // mark end of received string
        // go down string, end string at first <cr>
        for (i=0; i<pos; i++) if (baro_resp[i]==13) baro_resp[i]=0;
        //printf("R:%s:\n", baro_resp);
        aascFlushRdBuf(chan1);
    }
    yield;
}

```

```

// do serial output to terminal
costate term_output always_on {
    // if output string is non-zero in length, output it to the sensor
    if (strlen(term_out)>0 && (aascWriteBufFree(chan4) > strlen(term_out)+3) ) {
        aascWriteBlk(chan4,term_out,strlen(term_out),1);
    }
    if (term_out[strlen(term_out)-1]!='\n')
        aascWriteBlk(chan4,eol,2,1);           // send end of line
    term_out[0]=0;                           // zero out command
}
yield;
}

// do serial input from terminal
costate term_input always_on {
    if (aascScanTerm(chan4, 0x0d)>0){
        pos = aascReadBlk(chan4, term_in,sizeof(baro_resp),0);
        term_in[pos] = 0;                     // mark end of received string
        // go down string, end string at first <cr>
        for (i=0; i<pos; i++) if (term_in[i]==13) term_in[i]=0;
        //printf("Rt%d:%s:\n", strlen(term_in), term_in);
        aascFlushRdBuf(chan4);
    }
    yield;
}

// do modbus processing
costate modbus_io always_on{
    msRun();
    yield;
}

// do terminal processing
costate terminal_io always_on {
    if (strlen(term_in)>0) {                  // process command
        for (i=0; i<strlen(term_in); i++) term_in[i]=toupper(term_in[i]);
        if (strcmp(term_in,"CI")==0) {       // current inputs command?
            sprintf(term_out,"\r\nTrans: %e\r\nTrans Temp :
%f",cur_xducer_sensor,cur_xducer_temp);
            waitfor(strlen(term_out)==0);
            sprintf(term_out,"Baro: %f", cur_local_baro);
            waitfor(strlen(term_out)==0);
            sprintf(term_out,"Heatsink Temp: %f", cur_pod_temp);
            waitfor(strlen(term_out)==0);
            sprintf(term_out,"Ambient Temp: %f\r\nProduct 1 Temp:

```



```

%f",cur_tank_temp_1,cur_tank_temp_2);
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Product 2 Temp: %f\r\nProduct 3 Temp:
%f",cur_tank_temp_3,cur_tank_temp_4);
    waitfor(strlen(term_out)==0);
}
if(strcmp(term_in,"CV")==0) { // current values
    sprintf(term_out,"\r\nCor. Head: %f",mass);
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Baro: %f", cur_local_baro);
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Ambient Temp: %f",avg_tank_temp);
    //sprintf(term_out,"\r\nCor. Head: %f\r\nBaro: %f\r\nAmbient Temp: %f", mass,
cur_local_baro, avg_tank_temp);
    waitfor(strlen(term_out)==0);
}
if(strcmp(term_in,"XT")==0) { // xducer temp info
    sprintf(term_out, "\r\nToday");
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Min: %f\r\nMax: %f\r\nAvg:
%f",xducer_temp_min,xducer_temp_max,xducer_temp_avg);
    waitfor(strlen(term_out)==0);
    sprintf(term_out, "\r\nYesterday");
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Min: %f\r\nMax: %f\r\nAvg:
%f",xducer_ytemp_min,xducer_ytemp_max,xducer_ytemp_avg);
    waitfor(strlen(term_out)==0);
}
if(strcmp(term_in,"DT",2)==0) { // current date?
    if(term_in[2]!='=') { // set date/time
        year = (term_in[3]-48)*10+term_in[4]-48+2000;
        month = (term_in[5]-48)*10+term_in[6]-48;
        day = (term_in[7]-48)*10+term_in[8]-48;
        hour = (term_in[9]-48)*10+term_in[10]-48;
        minute =(term_in[11]-48)*10+term_in[12]-48;
        write_datetime();
    }
    read_datetime();
    sprintf(term_out,"\r\nYear: %d",year);
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Month: %d\r\nDay: %d\r\nHour: %d",month,day,hour);
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Minute:%d", minute);
    waitfor(strlen(term_out)==0);
}
}

```

```

if (strcmp(term_in,"TT",2)==0) { // transducer temp
    if (term_in[2]!='\0') { // set transducer temp
        u.f = atof(&(term_in[3]));
        i=WriteFlash(phy_adr(&desired_xducer_temp),u.c,sizeof(u.c));
    }
    sprintf(term_out,"\r\nHeatsink Temp: %f",cur_pod_temp);
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Setpoint: %f",desired_xducer_temp);
    waitfor(strlen(term_out)==0);
    sprintf(term_out,"Heater: %s",heater?"ON":"OFF");
    waitfor(strlen(term_out)==0);
}
if (strcmp(term_in,"SD",2)==0) { // transducer temp
    if (term_in[2]!='\0') { // set transducer temp
        u.f = atof(&(term_in[3]));
        i=WriteFlash(phy_adr(&day_start),u.c,sizeof(u.c));
        start_hour = (int)(day_start/100);
        start_min = (int)(day_start-start_hour*100);
    }
    sprintf(term_out,"\r\nStart of Day: %2d:%2d",start_hour,start_min);
    waitfor(strlen(term_out)==0);
}
if (strcmp(term_in,"MA",2)==0) { // t
    if (term_in[2]!='\0') {
        modbusaddr = atoi(&(term_in[3]));
    }
    sprintf(term_out,"\r\nModbus Addr: %d",modbusaddr);
    waitfor(strlen(term_out)==0);
}
if (strcmp(term_in,"TD",2)==0) { // tank dia
    if (term_in[2]!='\0') {
        u.f = atof(&(term_in[3]));
        i=WriteFlash(phy_adr(&tank_dia),u.c,sizeof(u.c));
    }
    sprintf(term_out,"\r\nTank Dia: %f",tank_dia);
    waitfor(strlen(term_out)==0);
}
if (strcmp(term_in,"SG",2)==0) { // specific gravity
    if (term_in[2]!='\0') {
        u.f = atof(&(term_in[3]));
        i=WriteFlash(phy_adr(&sg),u.c,sizeof(u.c));
    }
    sprintf(term_out,"\r\nSpecific G: %f",sg);
    waitfor(strlen(term_out)==0);
}

```

```

if(strncmp(term_in,"BC",2)==0) { // baro correction factor
    if (term_in[2]!='=') {
        u.f = atof(&(term_in[3]));
        i=WriteFlash(phy_adr(&Bcf),u.c,sizeof(u.c));
    }
    sprintf(term_out,"\r\nBaro Correct: %f",Bcf);
    waitfor(strlen(term_out)==0);
}
if(strncmp(term_in,"CE",2)==0) { // coef of expansion
    if (term_in[2]!='=') {
        u.f = atof(&(term_in[3]));
        i=WriteFlash(phy_adr(&coef_of_expn),u.c,sizeof(u.c));
    }
    sprintf(term_out,"\r\nCoef of Expn: %f",coef_of_expn);
    waitfor(strlen(term_out)==0);
}
if(strncmp(term_in,"TT",2)==0) { // initial temp
    if (term_in[2]!='=') {
        u.f = atof(&(term_in[3]));
        i=WriteFlash(phy_adr(&initial_temp),u.c,sizeof(u.c));
    }
    sprintf(term_out,"\r\nInitial Temp: %f",initial_temp);
    waitfor(strlen(term_out)==0);
}
if(strncmp(term_in,"CS",2)==0) { // clear storage
    for (rr=0; rr<36; rr++) {
        data.f.year = 0;
        data.f.day = 0;
        data.f.hour = 0;
        data.f.initial1 = 1234;
        data.f.initial2 = rr;
        flash_write(rr);
    }
    sprintf(term_out,"\r\nAll data storage has been cleared.");
    waitfor(strlen(term_out)==0);
}
if(strncmp(term_in,"DD",2)==0) { // dump data in comma delimited format
    strcpy(term_out, "\r\n");
    waitfor(strlen(term_out)==0);
    // start with ram record, and go back up until we've reached zero
    for (ii=index; ii>=0; ii--) {
        if (data.f.baro[ii]!=0 || data.f.avg_temp[ii]!=0) { // print this record
            if (formatdt(-1,ii)>0) {
                sprintf(term_out,"%s, %f, %d, %d, %d",date_time,data.f.mass[ii],
                    data.f.baro[ii],data.f.avg_temp[ii],data.f.xducer_temp[ii]);
            }
        }
    }
}

```

```

        waitfor(strlen(term_out)==0);
    }
}
}
// pickup the previous record and print it also,
// repeat until we've come back to the current record.
rr = current_record - 1;
do {
    yield;
    flash_read(rr);
    for (ii=240-1; ii>=0; ii--) {
        if ((old.f.baro[ii]!=0) || (old.f.avg_temp[ii]!=0)) { // print this record
            if (formatdt(rr,ii)>0){
                sprintf(term_out,"%s, %f, %d, %d, %d",date_time,old.f.mass[ii],
                    old.f.baro[ii],old.f.avg_temp[ii], old.f.xducer_temp[ii]);
                waitfor(strlen(term_out)==0);
            }
        }
    }
    rr--;
    if (rr<0) rr=36-1;
} while (rr != current_record);
}
sprintf(term_out,"*");
waitfor(strlen(term_out)==0);
term_in[0]=0;
}
yield;
}

// run serial baro sensor
costate baro_io always_on {
    strcpy(baro_cmnd,"*0100P3");
    waitfor(DelaySec(45));
    if (strlen(baro_resp)>6) strcpy(serial_pressure,&(baro_resp[5]));
    else strcpy(serial_pressure,"0.0");
    cur_xducer_sensor = atof(serial_pressure);
    strcpy(baro_cmnd,"*0100Q3");
    waitfor(DelaySec(10));
    if (strlen(baro_resp)>6) strcpy(serial_temp,&(baro_resp[5]));
    else strcpy(serial_temp,"0.0");
    cur_xducer_temp = atof(serial_temp)*9.0/5.0+32.0;
    waitfor(DelaySec(4));
}
}

```

```

}

// get *E*ven word for *F*loating point modbus register
unsigned int EF(float f){
    modbusd.f = f;
    return modbusd.i[0];
}

// *O*dd *W*ord
unsigned int OF(float f){
    modbusd.f = f;
    return modbusd.i[1];
}

// write *E*ven word of modbus bus floating point write
void EW(float *f, int i){
    modbusd.i[0] = i;
    *f = modbusd.f;
}

void OW(float *f, int i){
    modbusd.i[1] = i;
    *f = modbusd.f;
}

/*=====
=====*\
        Read Output Coil (Registers 00001-00020)
\*=====
=====*/

int
msOutRd ( unsigned   wCoil,
          int        *pnState
          )

{
    return MS_BADADDR;
}

/*=====
=====*\
        Write Output Coil (Registers 00001-00020)
\*=====
=====*/

```

```
int
msOutWr (unsigned wCoil,
         int      bState
        )
```

```
{
    return MS_BADADDR;
}
```

```
/*=====*\
    Read Input Coil (Registers 10001-10020)
\*=====*/
```

```
int
msIn (unsigned wCoil,
      int      *pnState
     )
```

```
{
    return MS_BADADDR;
}
```

```
/*=====*\
    Read Input Register (Registers 30001-3000A)
\*=====*/
```

```
int
msInput (unsigned wReg,
         unsigned *pwValue
        )
```

```
{
    return MS_BADADDR;
}
```

```
/*=====*\
    Read Register 40000
\*=====*/
```

```
/** BeginHeader msRead */
```

```
int msRead ( unsigned wReg,unsigned *pwValue );
/** EndHeader */
```

```
int
msRead ( unsigned wReg,
          unsigned *pwValue
        )
{
    int rvalue;
    int recnum, item;
    int currec;

    rvalue = 0;
    switch (wReg) {
        case 0:
            *pwValue = year;
            break;
        case 1:
            *pwValue = month;
            break;
        case 2:
            *pwValue = day;
            break;
        case 3:
            *pwValue = hour;
            break;
        case 4:
            *pwValue = minute;
            break;
        case 10:
            *pwValue = EF(desired_xducer_temp);
            break;
        case 11:
            *pwValue = OF(desired_xducer_temp);
            break;
        case 12:
            *pwValue = EF(tank_dia);
            break;
        case 13:
            *pwValue = OF(tank_dia);
            break;
        case 14:
            *pwValue = EF(sg);
            break;
        case 15:
```

```

    *pwValue = OF(sg);
    break;
case 16:
    *pwValue = EF(coef_of_expn);
    break;
case 17:
    *pwValue = OF(coef_of_expn);
    break;
case 18:
    *pwValue = EF(initial_temp);
    break;
case 19:
    *pwValue = OF(initial_temp);
    break;
case 20:
    *pwValue = start_hour;
    break;
case 21:
    *pwValue = start_min;
    break;
case 22:
    *pwValue = EF(Bcf);
    break;
case 23:
    *pwValue = OF(Bcf);
    break;
case 100:
    *pwValue = EF(mass); // ??
    break;
case 101:
    *pwValue = OF(mass); // ??
    break;
case 102:
    *pwValue = EF(cur_xducer_temp);
    break;
case 103:
    *pwValue = OF(cur_xducer_temp);
    break;
case 104:
    *pwValue = EF(cur_tank_temp_1);
    break;
case 105:
    *pwValue = OF(cur_tank_temp_1);
    break;
case 106:

```



```

    *pwValue = EF(cur_tank_temp_2);
    break;
case 107:
    *pwValue = OF(cur_tank_temp_2);
    break;
case 108:
    *pwValue = EF(cur_tank_temp_3);
    break;
case 109:
    *pwValue = OF(cur_tank_temp_3);
    break;
case 110:
    *pwValue = EF(cur_tank_temp_4);
    break;
case 111:
    *pwValue = OF(cur_tank_temp_4);
    break;
case 112:
    *pwValue = EF(xducer_temp_min);
    break;
case 113:
    *pwValue = OF(xducer_temp_min);
    break;
case 114:
    *pwValue = EF(xducer_temp_max);
    break;
case 115:
    *pwValue = OF(xducer_temp_max);
    break;
case 116:
    *pwValue = EF(xducer_temp_avg);
    break;
case 117:
    *pwValue = OF(xducer_temp_avg);
    break;
case 118:
    *pwValue = EF(xducer_ytemp_min);
    break;
case 119:
    *pwValue = OF(xducer_ytemp_min);
    break;
case 120:
    *pwValue = EF(xducer_ytemp_max);
    break;
case 121:

```

```

    *pwValue = OF(xducer_ytemp_max);
    break;
case 122:
    *pwValue = EF(xducer_ytemp_avg);
    break;
case 123:
    *pwValue = OF(xducer_ytemp_avg);
    break;
case 124:
    *pwValue = EF(cur_local_baro);
    break;
case 125:
    *pwValue = OF(cur_local_baro);
    break;
case 126:
    *pwValue = EF(cur_xducer_sensor);
    break;
case 127:
    *pwValue = OF(cur_xducer_sensor);
    break;
default:
    rvalue = MS_BADADDR;
    break;
}

if (wReg >= 1000) {

    recnum = wReg/1000;
    item = wReg%1000;
    currec = getcurrentindex();
    if (recnum < currec) { // use current ram data
        switch (item) {
            case 0:
                *pwValue = data.f.year;
                break;
            case 1:
                *pwValue = data.f.month;
                break;
            case 2:
                *pwValue = data.f.day;
                break;
            case 3:
                if (data.f.hour < 12)
                    *pwValue = (int)(recnum*2)/60;
                else

```

```

    *pwValue = 12 + (int)(recnum*2)/60;
    break;
case 4:
    *pwValue = (recnum * 2)%60;
    break;
case 6:
    *pwValue = EF(data.f.mass[currec-recnum-1]);
    break;
case 7:
    *pwValue = OF(data.f.mass[currec-recnum-1]);
    break;
case 8:
    *pwValue = data.f.baro[currec-recnum-1];
    break;
case 9:
    *pwValue = data.f.avg_temp[currec-recnum-1];
    break;
}
rvalue = 0;
}
else { // use flash data
    flash_read(getcurrentindex());
    switch (item) {
    case 0:
        *pwValue = old.f.year;
        break;
    case 1:
        *pwValue = old.f.month;
        break;
    case 2:
        *pwValue = old.f.day;
        break;
    case 3:
        if (old.f.hour<12)
            *pwValue = (int)((currec-recnum-1)*2)/60;
        else
            *pwValue = 12 + (int)((currec-recnum-1)*2)/60;
        break;
    case 4:
        *pwValue = ((currec-recnum-1) * 2)%60;
        break;
    case 6:
        *pwValue = EF(old.f.mass[currec-recnum-1]);
        break;
    case 7:

```

```

        *pwValue = OF(old.f.mass[currec-recnum-1]);
        break;
    case 8:
        *pwValue = old.f.baro[currec-recnum-1];
        break;
    case 9:
        *pwValue = old.f.avg_temp[currec-recnum-1];
        break;
    }
    rvalue = 0;
}
}

// return rvalue;
return 0;
}

/*=====
   *\
   Write Register
   *\
   =====*/

/** BeginHeader msWrite */
int msWrite ( unsigned wReg,unsigned wValue );
/** EndHeader */

int
msWrite ( unsigned wReg,
          unsigned wValue
          )
{
    int rvalue, i;
    union { float f; char c[4]; } u;

    rvalue = 0;
    switch (wReg) {
    case 0:
        year = wValue;
        write_datetime();
        break;
    case 1:
        month = wValue;
        write_datetime();
        break;

```

```

case 2:
    day = wValue;
    write_datetime();
    break;
case 3:
    hour = wValue;
    write_datetime();
    break;
case 4:
    minute = wValue;
    write_datetime();
    break;
case 10:
    u.f = desired_xducer_temp;
    EW(&u.f, wValue);
    i=WriteFlash(phy_adr(&desired_xducer_temp),u.c,sizeof(u.c));
    break;
case 11:
    u.f = desired_xducer_temp;
    OW(&u.f, wValue);
    i=WriteFlash(phy_adr(&desired_xducer_temp),u.c,sizeof(u.c));
    break;
case 12:
    u.f = tank_dia;
    EW(&u.f, wValue);
    i=WriteFlash(phy_adr(&tank_dia),u.c,sizeof(u.c));
    break;
case 13:
    u.f = tank_dia;
    OW(&u.f, wValue);
    i=WriteFlash(phy_adr(&tank_dia),u.c,sizeof(u.c));
    break;
case 14:
    u.f = sg;
    EW(&u.f, wValue);
    i=WriteFlash(phy_adr(&sg),u.c,sizeof(u.c));
    break;
case 15:
    u.f = sg;
    OW(&u.f, wValue);
    i=WriteFlash(phy_adr(&sg),u.c,sizeof(u.c));
    break;
case 16:
    u.f = coef_of_expn;
    EW(&u.f, wValue);

```

```

    i=WriteFlash(phy_adr(&coef_of_expn),u.c,sizeof(u.c));
    break;
case 17:
    u.f = coef_of_expn;
    OW(&u.f, wValue);
    i=WriteFlash(phy_adr(&coef_of_expn),u.c,sizeof(u.c));
    break;
case 18:
    u.f = initial_temp;
    EW(&u.f, wValue);
    i=WriteFlash(phy_adr(&initial_temp),u.c,sizeof(u.c));
    break;
case 19:
    u.f = initial_temp;
    EW(&u.f, wValue);
    i=WriteFlash(phy_adr(&initial_temp),u.c,sizeof(u.c));
    break;
case 20:
    start_hour = wValue;
    i = day_start - (int)(day_start / 100)*100;
    u.f = start_hour + i;
    i=WriteFlash(phy_adr(&day_start),u.c,sizeof(u.c));
    break;
case 21:
    start_min = wValue;
    u.f = ((int)(day_start/100))*100 + start_min;
    i=WriteFlash(phy_adr(&day_start),u.c,sizeof(u.c));
    break;
case 22:
    u.f = Bcf;
    EW(&u.f, wValue);
    i=WriteFlash(phy_adr(&Bcf),u.c,sizeof(u.c));
    break;
case 23:
    u.f = Bcf;
    OW(&u.f, wValue);
    i=WriteFlash(phy_adr(&Bcf),u.c,sizeof(u.c));
    break;
default:
    rvalue = MS_BADADDR;
    break;
}
return rvalue;
}

```

REIVINDICACIONES

1. Aparato para detectar fugas en un tanque de almacenamiento de líquido comprendiendo:

medios sensores de presión diferencial con un componente de medición de baja presión y un componente de medición de alta presión;

5 medios de cubierta de protección para la protección de dichos medios sensores de presión diferencial de materiales y condiciones cuando se sumerge en el contenido líquido de dicho tanque de almacenamiento;

estando dicho componente de medición de alta presión configurado para medir la presión del fondo del tanque dentro de un líquido en el que dichos medios de cubierta se sumergen;

10 estando dicho componente de medición de baja presión configurado para recibir datos indicativos de presión atmosférica sustancialmente en la superficie de dicho líquido;

medios de medición de presión barométrica configurados para medir la presión atmosférica sustancialmente en la superficie de dicho líquido cuando dichos medios de cubierta de protección se sumergen en él;

15 presión barométrica y medios de cálculo de presión diferencial están configurados para recibir unos primeros y segundos datos reflejando, respectivamente, dicha presión del fondo del tanque y dicha presión barométrica sustancialmente en dicha superficie de dicho líquido, y para ajustar dichos primeros datos para eliminar sustancialmente las variaciones sobre dichas medidas de dicha presión del fondo del tanque provocada solamente por variaciones de presión atmosférica para producir una presión atmosférica ajustada a la presión del fondo del tanque;

20 medios de medición de la temperatura ambiente para medir la temperatura ambiente cerca de dicho tanque de almacenamiento;

25 presión barométrica y dinámica del tanque ajustada a medios de cálculo de presión del fondo del tanque configurados para recibir unos terceros datos indicativos de dicha presión atmosférica ajustada a la presión del fondo del tanque, para recibir unos cuartos datos indicativos de mediciones de la temperatura ambiente mediante dichos medios de medición de temperatura ambiente, para recibir unos quintos datos indicativos de las características de expansión de dicho tanque de almacenamiento, y para ajustar dichos terceros datos con referencia a dichos cuartos y quintos datos para eliminar sustancialmente las variaciones sobre mediciones y cálculos de dicha presión barométrica ajustada a la presión del fondo del tanque, provocadas solamente por cambios dimensionales en dicho tanque de almacenamiento que resultan de las variaciones de temperatura atmosféricas, para producir una presión del fondo del tanque ajustada al tanque dinámico;

30 medios de cálculo de masa del contenido del tanque para calcular el contenido de masa de dicho tanque de almacenamiento sustancialmente basado en dicha presión del fondo del tanque ajustado dinámico y datos de entrada del usuario reflejando las características físicas de dicho contenido en dicho tanque de almacenamiento.

35 2. Aparato según la reivindicación 1, donde dichos medios de cubierta de protección se configuran manteniendo al menos algunos componentes de dicho aparato dentro de dichos medios de cubierta de protección, incluyendo dicho sensor de presión diferencial, pero exceptuando dicho componente de medición de baja presión y dicho componente de medición de alta presión, en vacío sustancial.

40 3. Aparato según la reivindicación 1 ó 2, comprendiendo además medios de gestión de temperatura de la cubierta de protección configurados para mantener la temperatura de dicho cubierta de protección sustancialmente a una temperatura específica de usuario.

4. Aparato según la reivindicación 1, 2 ó 3, donde los medios de detección de presión barométrica comprenden

45 unos primeros medios de medición de presión barométrica configurados para medir la presión atmosférica sustancialmente en la superficie de dicho líquido cuando dichos medios de cubierta de protección se sumergen en éste y para comunicar tal medición con dicho componente de medición de baja presión de dicho sensor de presión diferencial; y

unos segundos medios de medición de presión barométrica configurados para medir la presión atmosférica sustancialmente en la superficie de dicho líquido cuando dichos medios de cubierta de protección se sumergen en él;

50 y donde dicha presión barométrica y medios de cálculo de presión diferencial se configuran para recibir primeros y segundos datos reflejando, respectivamente, dicha presión del fondo del tanque y dicha presión barométrica medida por dichos primeros medios de medición de presión barométrica y para calcular la presión atribuible al contenido líquido de dicho tanque de almacenamiento, y recibiendo datos de dichos segundos medios de medición barométricos y, basado en el mismo, ajustando dicha presión atribuible a dicho contenido de dicho tanque de almacenamiento para eliminar sustancialmente las variaciones sobre dichas mediciones de las mismas provocadas solamente por variaciones de presión atmosférica para producir una presión atmosférica ajustada a la presión del fondo del tanque.

5. Aparato según la reivindicación 4, donde dichos medios de cubierta de protección se configuran manteniendo al menos algún componente de dicho aparato en ellos, incluyendo dicho sensor de presión diferencial, pero exceptuando dicho componente de medición de baja presión y dicho componente de medición de alta presión, sustancialmente al vacío.
- 5 6. Método para la detección de fugas en un tanque de almacenamiento; que incluye los pasos de:
 - selección del aparato según cualquiera de las reivindicaciones precedentes,
 - selección de los medios de almacenamiento de datos para recoger una pluralidad de datos de la masa del contenido generados por dichos medios de cálculo de masa del contenido del tanque en una pluralidad de puntos en el tiempo;
 - 10 selección de medios informáticos configurados para generar un indicio perceptiblemente humano de cambios en dichos datos de la masa del contenido entre una pluralidad de dichos puntos en el tiempo;
 - colocación de dichos medios de cubierta de protección sustancialmente en una superficie interior del fondo de dicho tanque de almacenamiento;
 - accionamiento de dicho aparato; y
 - 15 observación de indicios de cambios de los datos en dichos datos de masa del contenido atribuibles a fugas de dicho tanque de almacenamiento para detectar tal fuga.
7. Método según la reivindicación 6, comprendiendo además los pasos de:
 - fijación de los orificios de entrada y salida de flujo de dicho tanque de almacenamiento antes del dicho accionamiento de dicho aparato; y
 - 20 procesamiento sustancialmente selectivo de dichos datos de masa del contenido que se generaron aproximadamente entre la puesta de sol y el amanecer en un lugar de dicho sistema.

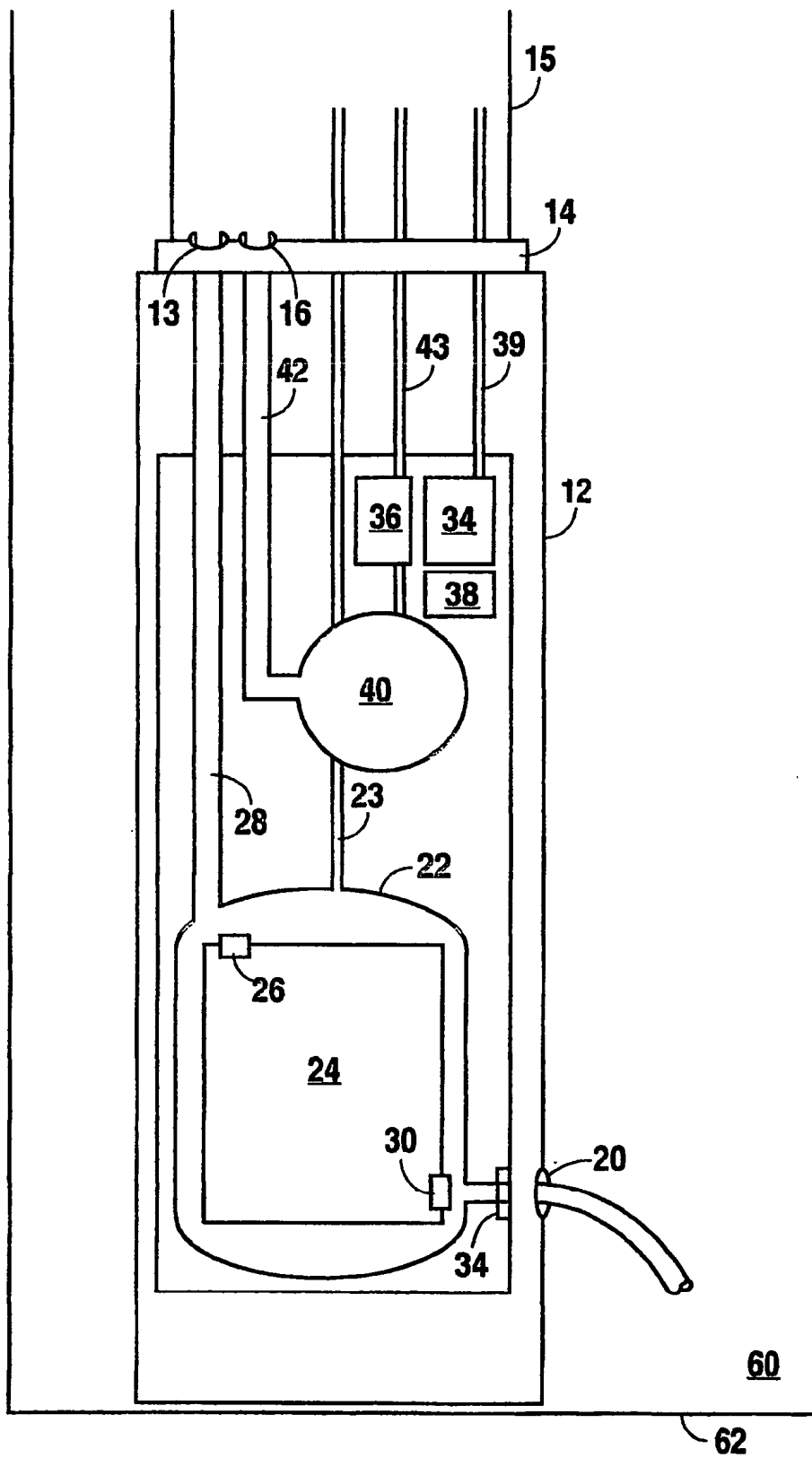


Fig. 1

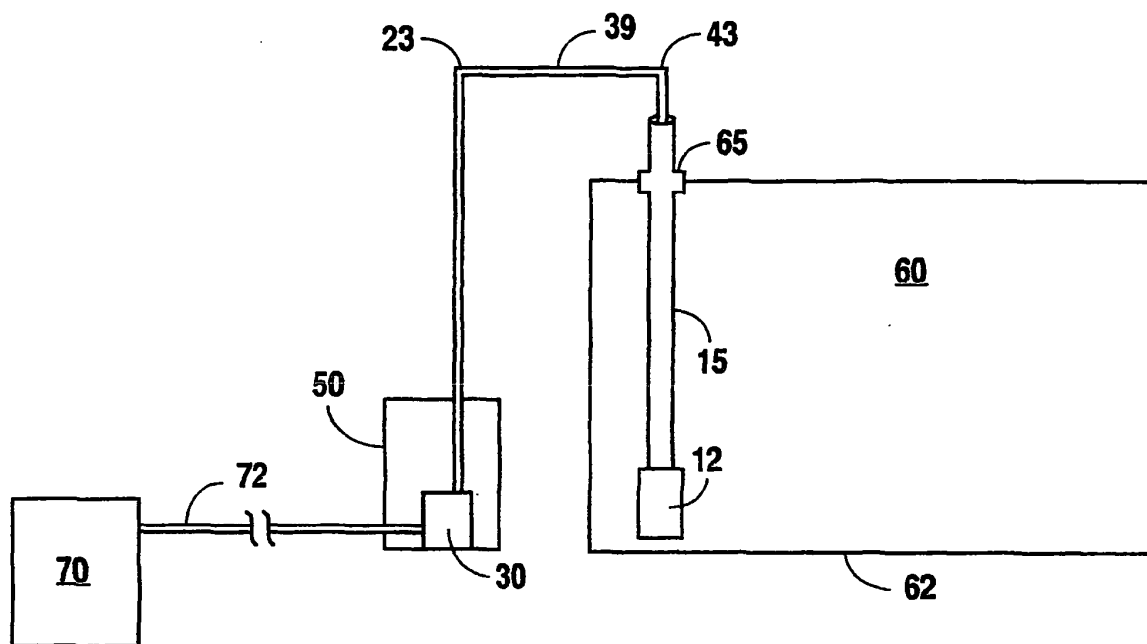


Fig. 2