

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**CONTROL AUTOMATIZADO Y MONITOREO DE MEDIDOR
VOLUMÉTRICO DE PETRÓLEO DIESEL 2 INSTALADO A
1000 METROS MAR ADENTRO DESDE UNA PLANTA
PESQUERA**

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

WILINGTON YUNIOR CORDERO VENTURA

PROMOCIÓN

2006-I

**LIMA – PERÚ
2011**

**CONTROL, AUTOMATIZADO Y MONITOREO DE MEDIDOR VOLUMETRICO DE
PETROLEO DIESEL 2 INSTALADO A 1000 METROS MAR ADENTRO DESDE UNA
PLANTA PESQUERA**

Gracias a Dios por la maravillosa familia con la que siempre he contado y en especial a mi madre que lucha todos los días de su vida por seguir a mi lado y que es, a quien dedico mi esfuerzo siempre en todo lo que hago

SUMARIO

En el presente informe se propone un sistema que permite la automatización, el control y el monitoreo del suministro de petróleo diesel 2 desde las plantas pesqueras a través de una tubería a embarcaciones que se encuentran a una distancia considerable mar adentro.

A diferencia del sistema de despacho que existe en la actualidad, en donde los medidores volumétricos de petróleo diesel 2 instalados se encuentran en planta, en este informe se considera la instalación de estos medidores con registrador electrónico en una plataforma flotante a 1 kilómetro mar adentro aproximadamente, que es el lugar en donde se hace el despacho a las embarcaciones. Mediante este sistema los encargados de operar estos medidores seguirán laborando desde planta y podrán monitorear y controlar remotamente estos equipos a través de una interfaz.

ÍNDICE

INTRODUCCION	1
CAPÍTULO I	
PLANTEAMIENTO DE INGENIERÍA DEL PROBLEMA	
1.1 Deficiencia de despachos de petróleo diesel 2 en la industria pesquera.....	2
1.2 Falta de control de despachos de combustible.....	3
1.3 Deficiencia de horas efectivas de trabajo de operarios.....	3
1.4 Objetivos del trabajo.....	3
CAPÍTULO II	
MARCO TEÓRICO CONCEPTUAL	
2.1 Actividad pesquera.....	4
2.2 Medidores de desplazamiento positivo de fluido.....	4
2.3 Registrador electrónico.....	6
2.4 Válvula solenoide para despachos predeterminados.....	7
2.5 JAVA y las comunicaciones cliente / servidor.....	8
2.6 La comunicación a través de Radiofrecuencia.....	9
2.7 Modulación.....	9
2.8 Espectro ensanchado por salto de frecuencias.....	9
2.9 Radio modem.....	11
CAPÍTULO III	
METODOLOGÍA PARA LA SOLUCIÓN DEL PROBLEMA	
3.1 Control remoto de medidor volumétrico de petróleo diesel 2.....	14
3.2 Formato de mensajes comando y mensajes respuesta.....	15
3.3 Conexiones para control de bomba y válvula solenoide.....	16
3.4 Código de programación para plataforma de control e interfaz gráfica.....	18
CAPÍTULO IV	
ANÁLISIS DE RESULTADOS	
4.1 Centralización de control y monitoreo de medidor volumétrico.....	20
4.2 Precisión en el despacho de combustible a las embarcaciones.....	21
4.3 Detección de fugas de combustible por averías.....	22
4.4 Lenguaje de programación JAVA como software libre.....	23
CONCLUSIONES Y RECOMENDACIONES	24
ANEXO A	

ESPECIFICACIONES DEL LCR II.....	25
ANEXO B	
DIAGRAMA DEL CABLEADO – LECTROCOUNT LCR II.....	27
ANEXO C	
ESPECIFICACIONES FUNCIONALES Y FORMATO DE MENSAJES DEL PROTOCOLO LCP.....	29
ANEXO D	
TABLAS DE CABLEADO DE TERMINALES DEL REGISTRADOR ELECTRÓNICO...38	
ANEXO E	
MODELO DE COMUNICACIÓN CLIENTE SERVIDOR EN JAVA.....	41
ANEXO F	
CANALIZACIONES Y TECNICAS FHSS PARA IEEE 802.11b.....	43
ANEXO G	
HOJA TÉCNICA DE RADIO MODEM, MODELO FGR2-PE, PARA ETHERNET Y RS-485.....	45
ANEXO H	
HOJA TÉCNICA DE RADIO MODEM, MODELO FGRIQ, PARA TRANSMISION DE VOLTAJES CONTINUOS.....	48
ANEXO I	
HOJA TÉCNICA DE VALVULAS DE SOLENOIDES DE DOS ESTADOS.....	51
ANEXO J	
CÓDIGOS EN LENGUAJE DE PROGRAMACION JAVA CORRESPONDIENTES A LA SIMULACION DE LA INTERFAZ GRAFICA PARA EL CONTROL DE UN MEDIDOR VOLUMÉTRICO DE PETRÓLEO.....	53
ANEXO K	
CÓDIGOS EN LENGUAJE DE PROGRAMACION JAVA CORRESPONDIENTES A LA SIMULACION DE UN EQUIPO MEDIDOR VOLUMÉTRICO DE PETRÓLEO.....	59
ANEXO L	
IMAGEN REAL DE PLATAFORMA FLOTANTE EN PLENO DESPACHO A EMBARCACIONES.....	63
BIBLIOGRAFÍA.....	65

INTRODUCCION

En la industria pesquera peruana, todas las plantas suministran petróleo diesel 2 a embarcaciones que se encuentran a una distancia considerable de las mismas mar adentro. Los equipos de bombeo y los medidores volumétricos de petróleo involucrados en la cuantificación de estos despachos se encuentran siempre en planta. No existe ningún medio adicional que permita determinar con certeza si el volumen de petróleo que se despachó desde planta es el mismo que se entregó a la embarcación respectiva, ya que hay una trayectoria de tubería bastante extensa a considerar por donde se envía el petróleo.

En el presente informe de suficiencia, se considera un sistema con el que una planta pesquera pueda ser capaz de suministrar petróleo diesel 2 a embarcaciones que se encuentran a una distancia considerable de las plantas mar adentro de una manera más eficiente a la que se da en la actualidad.

También se considera un entorno gráfico con el que, el operario encargado de poner en marcha el sistema pueda visualizar en tiempo real el normal funcionamiento del sistema en plena actividad y tomar algunas acciones de control desde su entorno de trabajo hacia el sistema cuando se quiera interrumpir el proceso de manera oportuna por sucesos inesperados.

CAPÍTULO I PLANTEAMIENTO DE INGENIERÍA DEL PROBLEMA

1.1 Deficiencia de despachos de petróleo diesel 2 en la industria pesquera

La industria pesquera peruana hace despachos de petróleo diesel 2 a sus embarcaciones destinadas a la pesca marítima, estos despachos son realizados desde planta por medio de equipos de bombeo y tuberías, estas tuberías pueden alcanzar dimensiones de varios cientos de metros que por lo general llegan a extenderse hasta 1 Km como se aprecia en la figura 1.1 desde la planta pesquera hasta una plataforma flotante mar adentro en donde dichas embarcaciones reciben el petróleo.

Como parte de la cuantificación de estos despachos, se emplean medidores de volumen de petróleo que siempre se encuentran en planta y que miden el volumen despachado desde planta, sin embargo no existe ningún medio que pueda determinar con certeza si el volumen de petróleo que se recibió en la plataforma es la misma que se despachó desde planta.

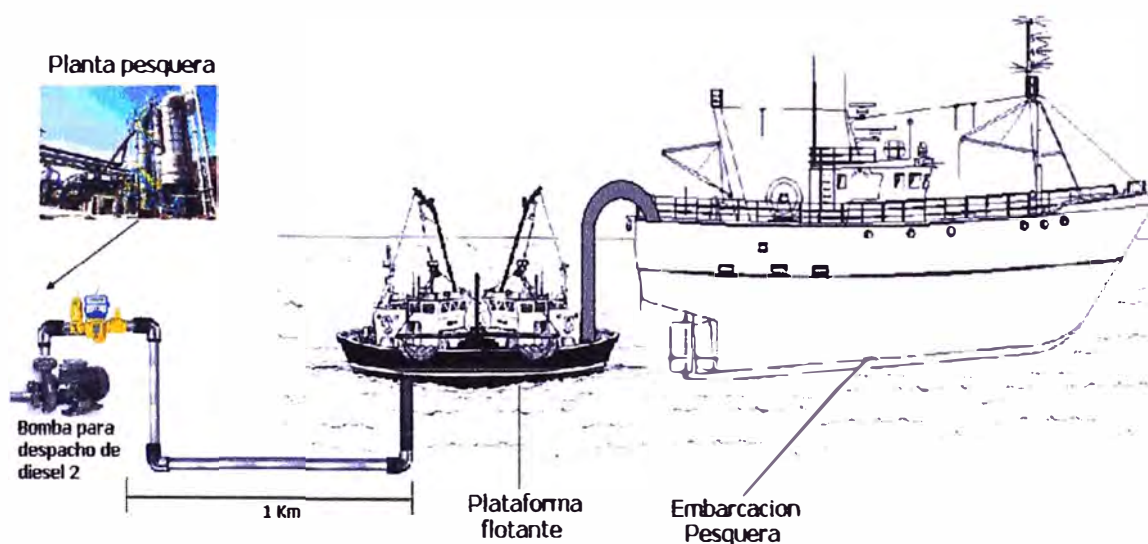


Fig. 1.1 Sistema de despacho típico de petróleo diesel 2

Considerando que generalmente el diámetro de la tubería para despacho de petróleo es 4 pulgadas y para una trayectoria de 1 Km., habría un volumen de 8.167 m³ o su equivalente de 1,854 galones involucrados que no se podría determinar con certeza si se logró recibir en su totalidad en la plataforma, por lo que se presenta esta incertidumbre en cada despacho que se hace.

1.2 Falta de control de despacho de combustible

Todas las plantas pesqueras, cuentan con un área de logística y almacén de materiales que se encarga de llevar el control del movimiento interno, de ingresos y salidas de mercancías y materiales. El petróleo diesel 2 que es el combustible con el que cuentan es también parte de una mercancía que no debería escapar de su control.

Sin embargo las oficinas y almacenes de esta área siempre se encuentran distanciadas de los medidores volumétricos y no hay un medio de control de despachos más que la correcta observación por parte del operario encargado de la supervisión del normal funcionamiento del medidor. Esta supervisión consiste en visualizar el registrador del medidor y hacer la respectiva parada del despacho cuando se haya completado el volumen solicitado, luego este operario informa a su supervisor o ingresa a la base de 50 y 100 minutos aproximadamente observando el medidor.

1.3 Deficiencia de horas efectivas de trabajo de operarios

Los operarios encargados de supervisar los despachos, siempre deben estar observando el normal funcionamiento de todo el sistema de despacho todo el tiempo que tome este. Generalmente los volúmenes solicitados por las embarcaciones son de 2000 a 3000 galones, y considerando que el caudal entregado por la bomba de despacho es por lo general de 100 galones por minuto, entonces el operario deberá permanecer entre 20 y 30 minutos aproximadamente observando el medidor por cada despacho.

1.4 Objetivos del Informe operarios

El objetivo principal del informe es presentar un sistema que permita realizar la cuantificación exacta del petróleo diesel 2 que se despacha a las embarcaciones.

También se logrará la automatización de dicho sistema de despacho para que se pueda tener el control total centralizado desde las oficinas de logística de planta con la participación de la menor cantidad de operarios.

CAPÍTULO II MARCO TEORICO CONCEPTUAL

2.1 Actividad pesquera

Es la actividad comercial de capturar peces y otras especies marinas con el objeto de obtener recursos alimenticios para su comercialización. De estos recursos alimenticios se buscan subproductos como aceites y harinas que no van destinados al consumo humano directo. La mayor parte de la actividad pesquera se realizada cerca de las costas, esto debido a que en general las aguas que se extienden sobre la plataforma continental desde las costas, son más ricas en fauna gracias a una mayor disponibilidad de nutrientes, provenientes del continente o de fenómenos de surgencia.

Cada industria pesquera cuenta con embarcaciones marítimas en las inmediaciones del mar, diseñadas y equipadas especialmente para ser utilizadas en las labores de pesca. Dichas embarcaciones funcionan con motores diesel de combustión interna de gran capacidad, por lo que las plantas pesqueras necesitan constantemente abastecerles de petróleo diesel 2. Este despacho se realiza desde una plataforma instalada a una distancia considerable mar adentro, cada planta cuenta con una plataforma flotante para dicho propósito.

2.2 Medidores de desplazamiento positivo de fluidos

Los medidores de desplazamiento positivo tienen como objetivo medir la cantidad de fluido que circula por un conducto. Consiste en una cámara en donde se encuentran alojadas unas piezas móviles que generan divisiones de volúmenes que son contabilizados progresivamente de acuerdo a las vueltas que generan dichas piezas.

Dependiendo del fabricante, la estructura interna de la cámara de medición puede variar en su diseño. Para fines específicos, se considerará un diseño típico de medidor que es usado comúnmente en la industria pesquera para la medición volumétrica de petróleo diesel 2.

El grupo LIQUID CONTROLS de IDEX Corporation de U.S.A. maneja una línea de medidores para este propósito como el que se muestra en la figura 2.1.

En la figura 2.2 se puede apreciar que el cuerpo del medidor (1) está diseñado con tres cilindros (2). Tres rotores, el rotor de bloqueo (3) y dos rotores de desplazamiento (4, 5), giran de manera sincronizada dentro de los cilindros. Los tres rotores están apoyados por placas de asiento (6, 7). Los extremos de los rotores salen fuera de las placas de asiento.

El engrane del rotor de bloqueo (8) está colocado sobre el extremo del rotor de bloqueo (3). Los engranajes de los rotores de desplazamiento (9, 10) están colocados sobre los extremos de los rotores de desplazamiento. Estos engranajes establecen la relación sincronizada entre los tres rotores.



Fig.2.1 Medidor volumétrico de desplazamiento positivo

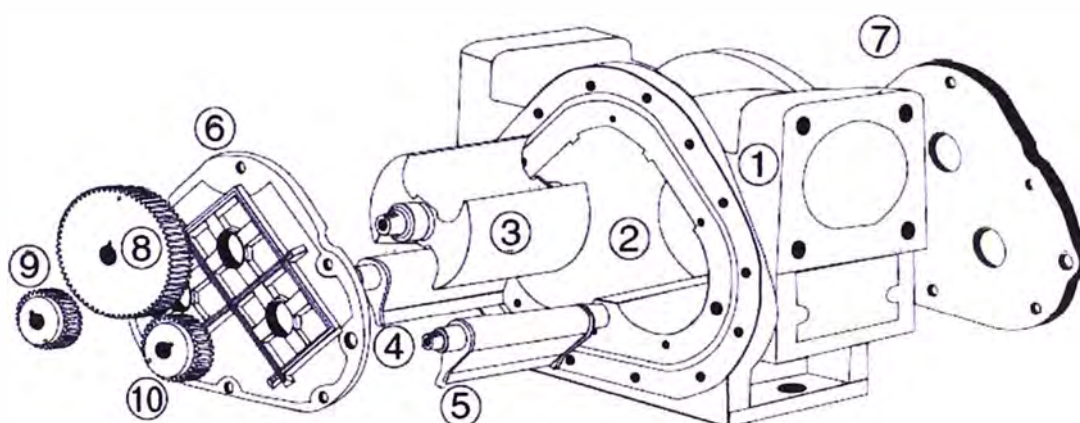


Fig.2.2 Despiece de la cámara de medición

En la figura 2.3 se puede ver que cuando el fluido pasa a través del cuerpo del medidor, el ensamblaje de rotores gira. El líquido está separado en secciones uniformes por la rotación de los rotores. El desplazamiento del fluido ocurre de manera simultánea.

Cuando entra el fluido, otra porción de fluido está siendo separada y medida. Al mismo tiempo, el fluido corriente arriba está siendo desplazado fuera del medidor hacia la línea de descarga. Ya que se conoce el volumen de los cilindros, y la misma cantidad de fluido

pasa a través del medidor con cada revolución del rotor de bloqueo, el volumen preciso de líquido que ha pasado a través del medidor puede ser determinado



Fig.2.3 Diagrama de flujo de medición

2.2 Registrador Electrónico

LIQUID CONTROLS maneja una serie de registradores electrónicos para el monitoreo del volumen de producto despachado. Uno de ellos es el modelo LectroCount LCR-II como se aprecia en la figura 2.4, que consta de un microprocesador basado en un registrador electrónico que es usualmente instalado en los medidores de desplazamiento positivo. El LCR-II registra el volumen despachado en una pantalla LCD de 6 dígitos.

En el anexo A se puede apreciar las especificaciones técnicas del modelo LCR II de registrador electrónico.



Fig. 2.4 Registrador Electrónico

Actualmente el modelo de placa de CPU utilizado por este registrador es el 84040 que se muestra en la figura 2.5, en donde se puede apreciar los diferentes terminales de entrada y salida en el LCR II.

En el anexo B se tiene información mas detallada de todos los terminales (funciones y cableado) de la placa 84040.

Como se aprecia en la fig. 2.5, a través del conector J2 se tiene el puerto RS 485 con 2 líneas compatibles con el estándar SAE J1708. Es por medio de este puerto que se puede lograr la comunicación con un dispositivo o plataforma externa mediante el

protocolo de propósito general llamado LCP desarrollado por LIQUID CONTROLS, del cual su formato de mensajes se aprecia con detalle en el anexo C.

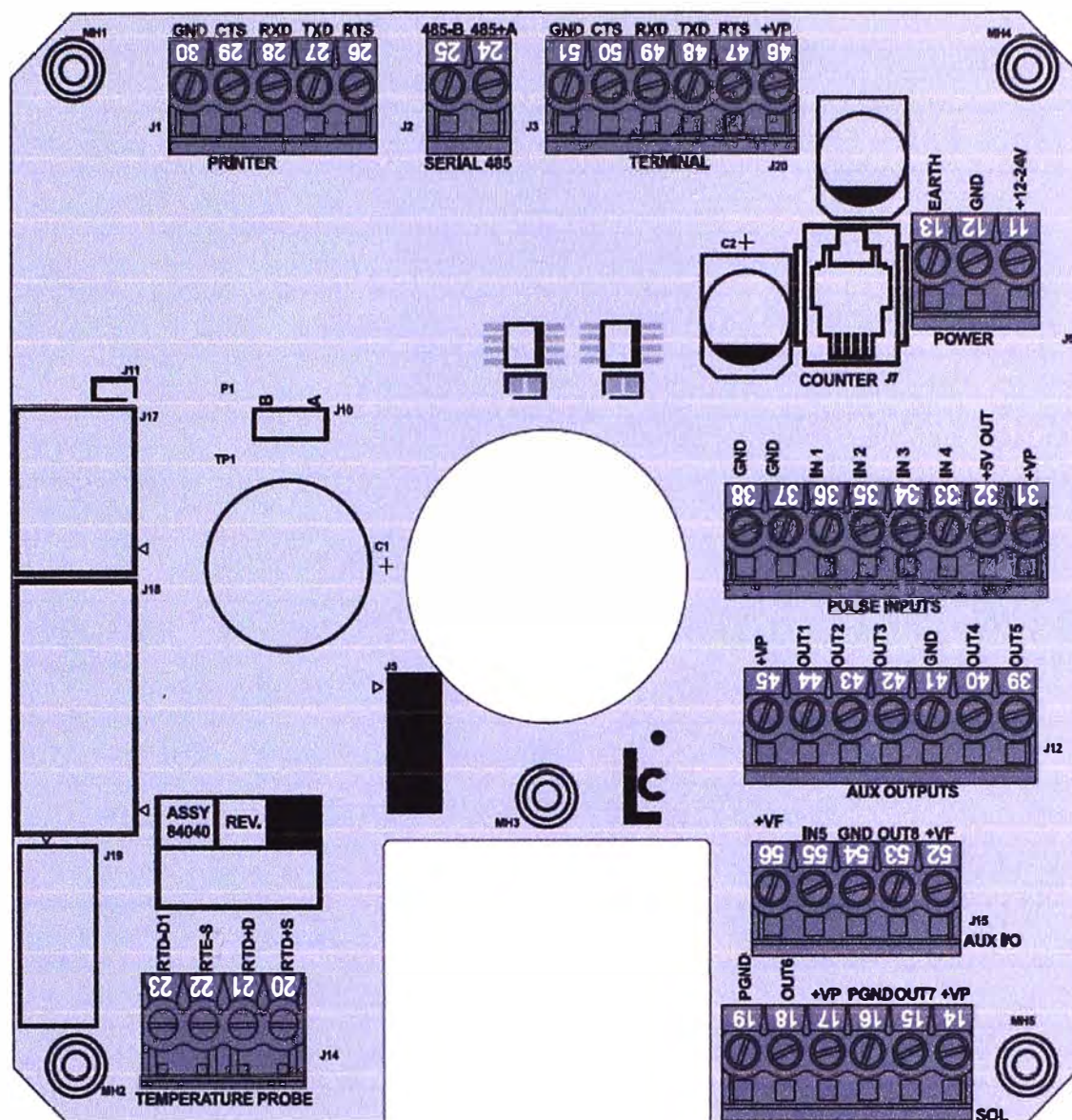


Fig. 2.5 Placa CPU, modelo 84040

2.3 Válvula Solenoide para despachos predeterminados

Este tipo de válvula es un dispositivo operado eléctricamente con la finalidad de controlar el flujo del petróleo diesel 2. Generalmente esta válvula se cierra por presión y consta de 2 partes fundamentales: el solenoide y la válvula. El solenoide es un electroimán que consiste de una bobina de alambre de cobre aislado, el cual está enrollado alrededor de una superficie de cuerpo cilíndrico, este convierte energía eléctrica en energía mecánica para actuar la válvula, cuando se envía corriente eléctrica a través de estos devanados, actúa como un electroimán, tal y como se describe en la figura 2.6. El campo magnético que se crea, es la fuerza motriz para abrir la válvula. Un resorte se

utiliza para devolver el actuador y la válvula de nuevo a su estado de reposo cuando el flujo de corriente se retira.

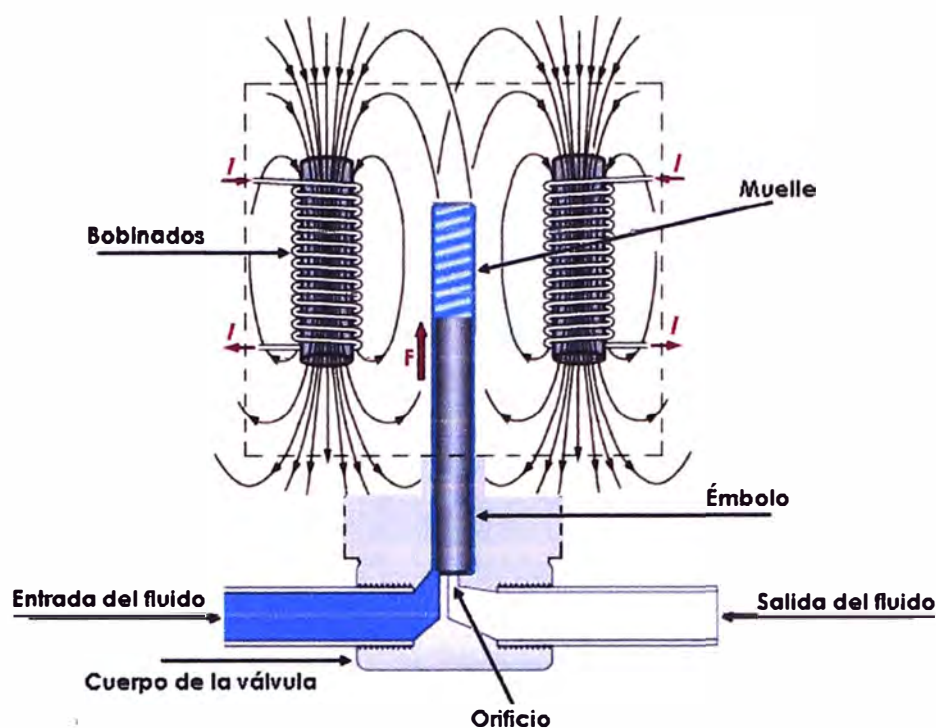


Fig. 2.6 Válvula solenoide

2.4 JAVA y las comunicaciones Cliente / Servidor

Como lenguaje de programación, JAVA se convirtió en la primera plataforma informática para computadoras desarrollada por Sun Microsystems en 1995, en un intento por resolver simultáneamente los problemas que se le plantean a los desarrolladores de software por la proliferación de arquitecturas incompatibles.

Debido a la existencia de distintos tipos de ordenadores y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de ordenador utilizado, por lo que desarrollaron un código neutro, el cual se ejecutaba sobre una maquina virtual de JAVA o JVM por sus siglas en ingles.

JAVA es un lenguaje orientado a objetos, por lo que está conformado de interfaces de programación de aplicaciones (más conocido por sus siglas en ingles API), las cuales se encuentran provistas de un conjunto de clases utilitarias para efectuar toda clase de tareas necesarias dentro de un programa.

Cualquier aplicación que se desarrolle, se apoya en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero todas forman parte del propio lenguaje.

En JAVA, la API o paquete `java.net` permite realizar conexiones y transacciones a través de una red. Un ejemplo muy común es el modelo Cliente/Servidor, en donde el

servidor establece un puerto y espera durante un cierto tiempo a que el cliente establezca la conexión. Cuando el cliente solicite una conexión, el servidor abrirá la conexión con un método específico como se puede observar en el anexo E.

2.5 La comunicación a través de radiofrecuencia

Este tipo de comunicación actúa mediante ondas electromagnéticas que se propagan por el espacio. Estas ondas corresponden al espectro de radiofrecuencia o RF en la porción menos energética del espectro electromagnético ubicada entre aproximadamente 3 KHz y 300 Ghz, y pueden ser transmitidas con la generación de corriente alterna a una antena. La comunicación mediante ondas electromagnéticas lleva información desde un punto (emisor) hasta otro (receptor) sin la presencia de cables como se representa en la figura 2.7.



Fig. 2.7 Sistema de comunicaciones de radiofrecuencia

2.6 Modulación

Es una operación mediante la cual ciertas características de una onda denominada portadora o información, se modifica en función de otra denominada moduladora para que pueda ser transmitida como se aprecia en la figura 2.8. El proceso inverso consiste en separar de la señal modulada, la onda que contiene solamente la información, este proceso es denominado desmodulación.

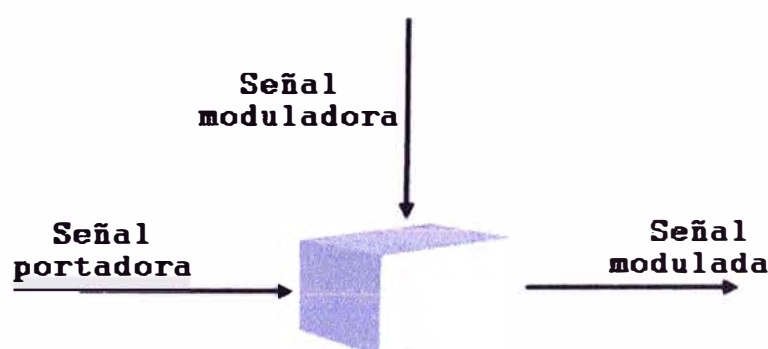


Fig. 2.8 Esquema de modulación de una señal

2.7 Espectro ensanchado por salto de frecuencias

Esta tecnología también denominada por sus siglas en inglés FHSS consiste en transmitir una parte de la información en una determinada frecuencia durante un intervalo de tiempo que es menor de 400ms, luego de transcurrido este tiempo se cambia la frecuencia de emisión y se continúa con la transmisión a otra frecuencia, por lo que cada

tramo de la información se va transmitiendo en una frecuencia distinta durante un intervalo muy corto de tiempo. El orden de los saltos en frecuencia se determina según una secuencia seudo aleatoria almacenada en tablas, y que tanto el emisor y el receptor deben conocer. Para el rango de frecuencias de 2,4 GHz a 2,4835 GHz la técnica FHSS se organiza en 79 canales con un ancho de banda de 1 MHz cada uno como se puede apreciar en el anexo F.

Para fines específicos en este informe y debido a que la información a transmitir es digital, consideraremos una modulación en particular. La modulación por desplazamiento de frecuencia gaussiana o por sus siglas en ingles GFSK que es una versión mejorada de la modulación por desplazamiento de frecuencia o FSK. La modulación por desplazamiento de frecuencia consiste en la transmisión digital de información binaria de ceros y unos empleando 2 frecuencias diferentes como se puede apreciar en la figura 2.9.

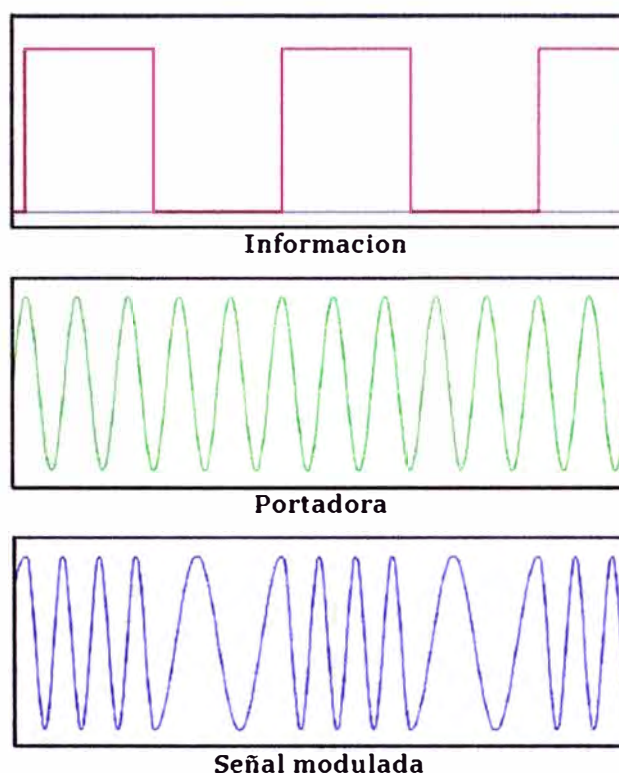


Fig. 2.9 Modulación binaria FSK

En la modulación digital, a la relación de cambio a la entrada del modulador se le llama bit-rate y tiene como unidad el bit por segundo o bps. A la relación de cambio a la salida del modulador se le llama baud-rate y que es en esencia la velocidad o cantidad de símbolos por segundo.

En FSK el bit rate es igual al baud rate, así por ejemplo un 0 binario se puede representar con una frecuencia f_1 , y el 1 binario con una frecuencia distinta f_2 .

Al modulador de FSK clásico con un filtrado gaussiano previo de la serie de bits se le conoce como GFSK. Con el filtrado previo, se obtiene un espectro de energía más estrecho de la señal modulada, por lo que se puede obtener mayores velocidades de transferencia sobre un mismo canal.

La modulación GFSK de 2 niveles o 2GFSK es un tipo de modulación donde un 1 lógico es representado mediante una desviación positiva de la frecuencia de la onda portadora, y un 0 mediante una desviación negativa de la misma proporcionando una velocidad de 1Mbps como se muestra en la figura 2.10.

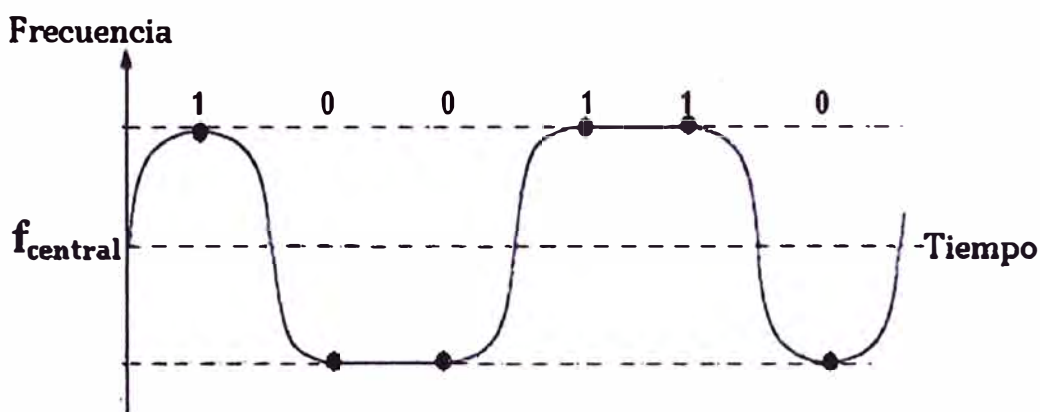


Fig. 2.10 Modulación 2GFSK

2.9 Radio módem

Es un sistema que permite la conexión de dos o más puntos de comunicación a través de la radiofrecuencia, los enlaces vía radio están basados en sustituir el medio físico convencional de transmisión con el fin de brindar comunicación a largo alcance y ahorrando la instalación de cables. Los radio módems pueden ser transmisores o receptores. Los radio módems transmisores permiten la recepción de información a través de una interface de datos de entrada como ETHERNET para luego enviar la información a través de una antena vía radiofrecuencia, los radio módems receptores lo hacen de manera inversa.

Para fines específicos de este informe, se considerará radio módems típicos para este propósito. FreeWave de USA maneja la serie de radio módems FGR2, la cual cuenta con puertos ETHERNET y seriales. También maneja una serie de radio módems especial denominada FGRIO que, aparte de manejar puertos seriales cuenta también con salidas y entradas digitales para la transmisión de voltajes continuos.

Estos radio módems llegan a tener alcances de hasta 96 kilómetros en línea de vista limpia en el caso del FGR2 y de 3 kilómetros en el caso del FGR2IO, por lo que pueden considerarse en distancias que no superan los 2 kilómetros en los sistemas de despacho de petróleo diesel 2 de la industria pesquera. Las hojas técnicas de los radio módems de

las series FGR2 y FGR10 se encuentran especificadas al detalle en los anexos G y H respectivamente.

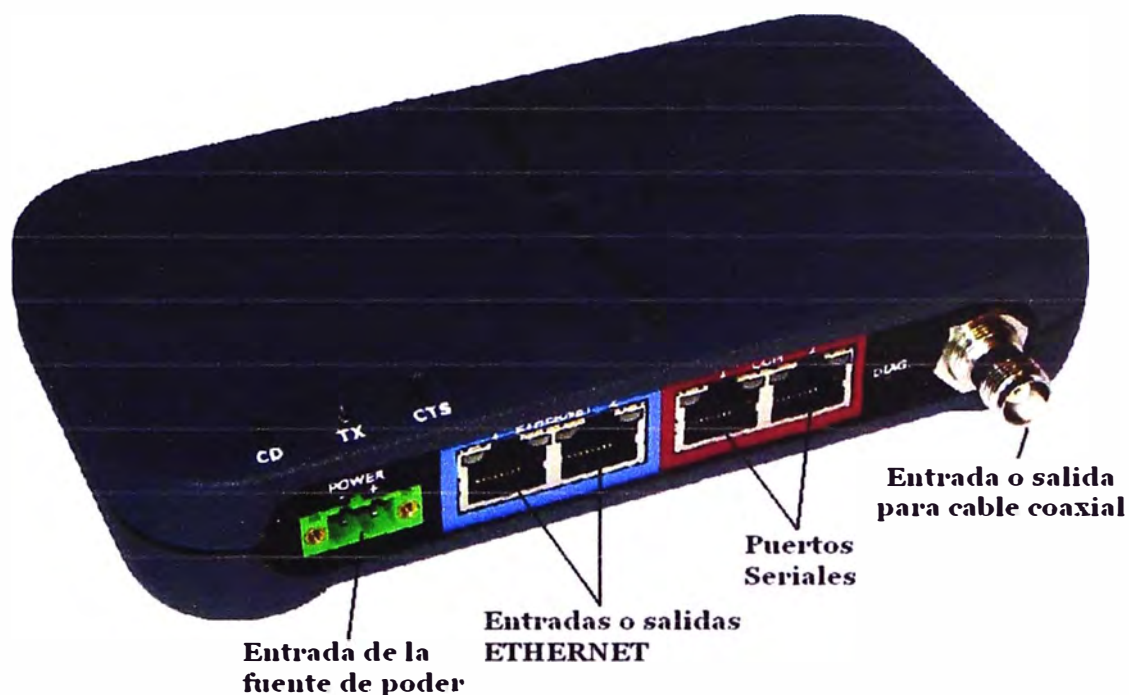


Fig. 2.11 Radio módem típico para envío y/o recepción de información vía Radiofrecuencia

En la figura 2.11 se puede observar un equipo radio módem típico para establecer una comunicación vía radiofrecuencia, en este caso es similar a la serie FGR2

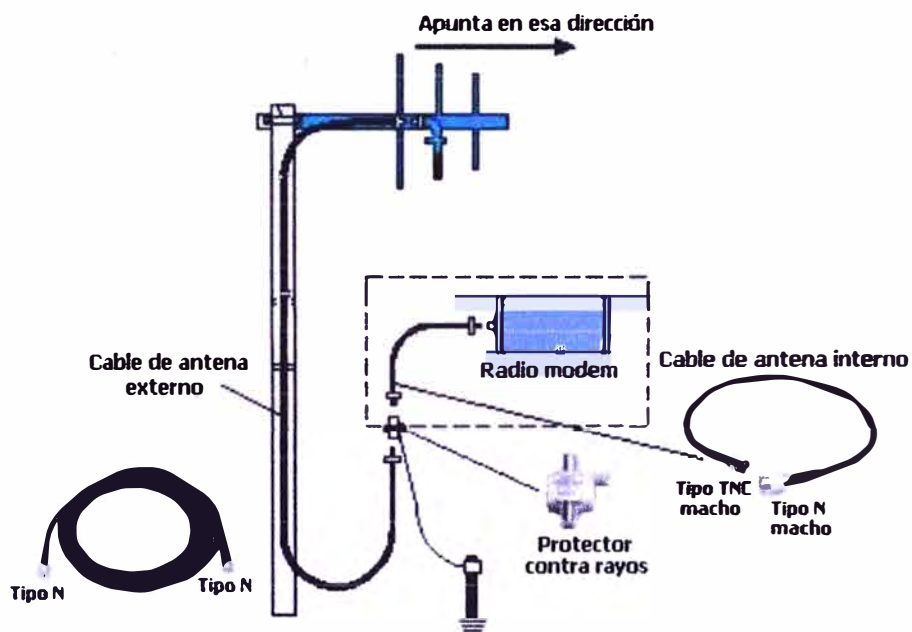


Fig. 2.12 Instalación típica de una antena direccional transmisora

Las antenas usadas pueden ser de dos tipos: direccionales que son antenas capaces de concentrar la mayor parte de la energía radiada de manera localizada aumentando así

la potencia emitida hacia el receptor, mientras que las antenas omnidireccionales radian o captan por igual en todas las direcciones.

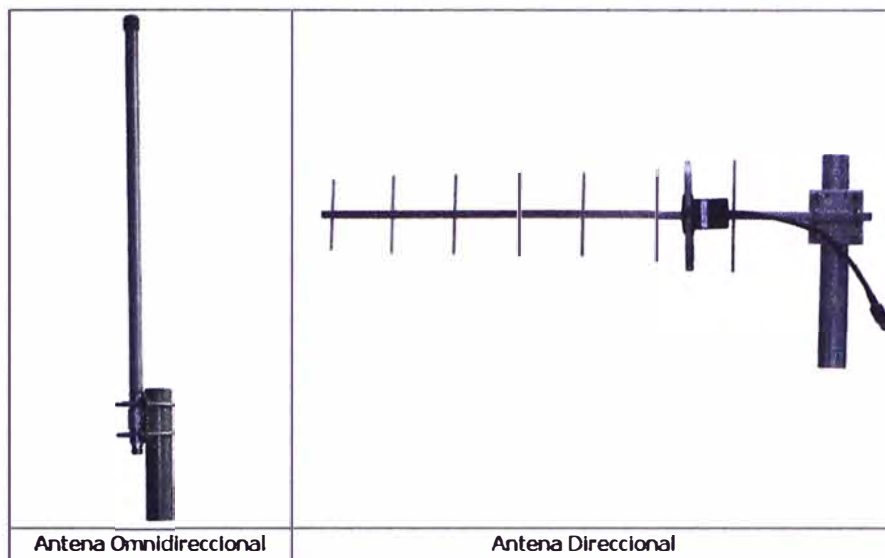


Fig. 2.13 Tipos de antenas

En la figura 2.12 se puede apreciar la instalación típica de una antena del tipo direccional, mientras que en la figura 2.13 se observa los 2 tipos de antenas que pueden ser usados para este propósito.

CAPÍTULO III METODOLOGÍA PARA LA SOLUCIÓN DEL PROBLEMA

3.1 Control Remoto de medidor volumétrico de petróleo diesel 2

La solución alternativa que se plantea en este informe consiste en la instalación del medidor volumétrico en la plataforma flotante con la intención de contabilizar el despacho que se realizó en dicha plataforma como se muestra en la figura 3.1, quedando así determinado con certeza la cantidad exacta que se despachó a la embarcación.

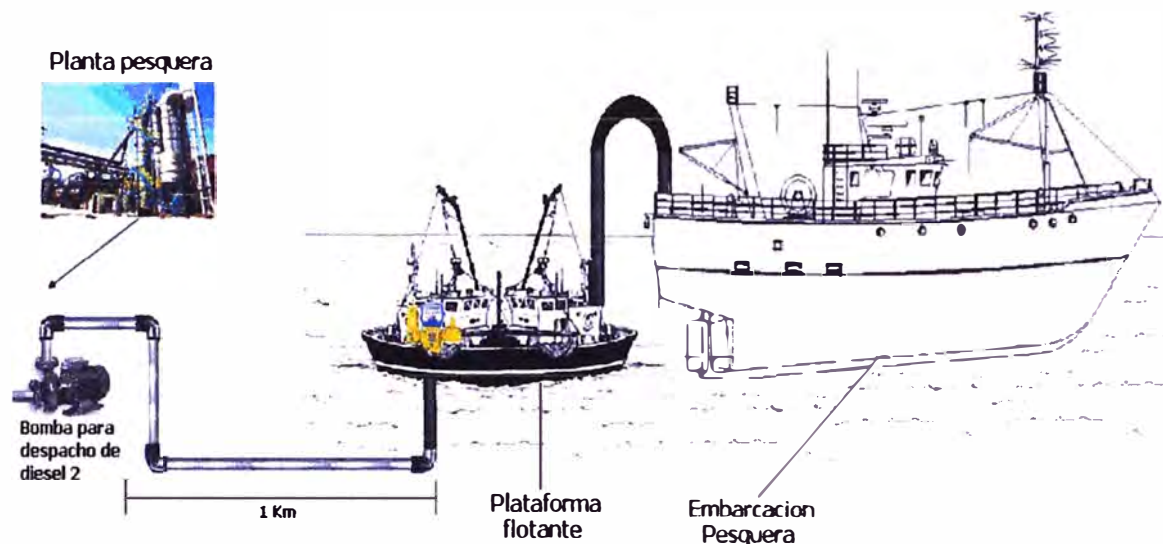


Fig. 3.1 Esquema de sistema de despacho alternativo

Sin embargo, a diferencia de tener el equipo en planta, no es práctico tener un operario de almacén supervisando los despachos en plataforma, ya que tendría que movilizarse desde planta un kilómetro mar adentro aproximadamente de ida y vuelta por cada despacho lo que significaría cerca de 40 minutos solo de viaje.

Se plantea poder operar el medidor remotamente desde la planta por medio de un sistema inalámbrico basado en radiofrecuencia y una interfaz gráfica realizada en lenguaje de programación JAVA como plataforma de control y operación.

De acuerdo al ANEXO C en donde se especifica que los equipos medidores de la marca LIQUID CONTROLS cuentan con puertos seriales RS 232 y RS 485 para comunicación de propósito general, podemos considerar al puerto RS 485 para establecer una comunicación bidireccional, ya que se enviarán mensajes comandos

desde la interfaz gráfica y se recibirán mensajes respuesta desde el medidor volumétrico de acuerdo al protocolo LCP y los formatos de mensajes especificados en dicho anexo.

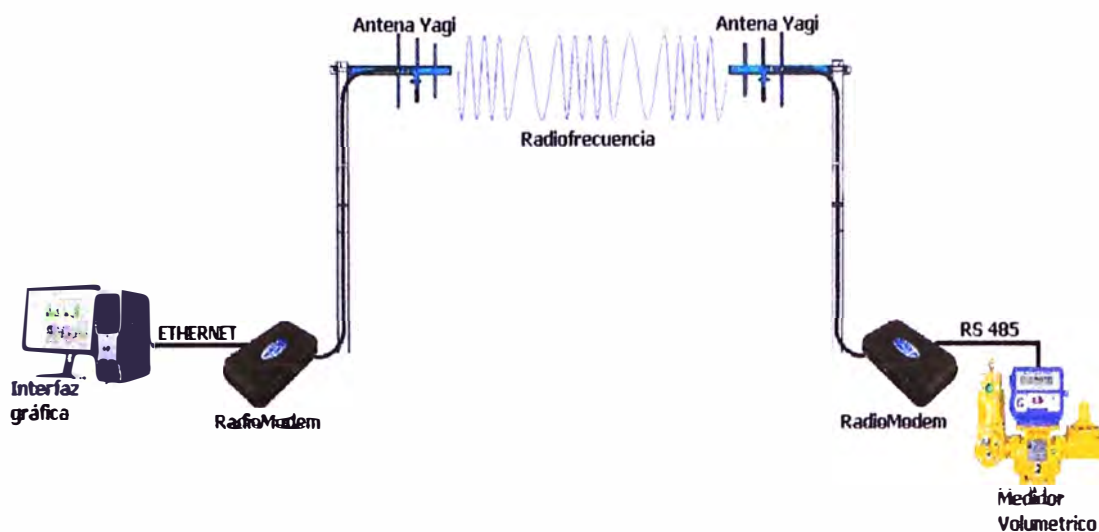


Fig. 3.2 Sistema de control y monitoreo remoto de medidor volumétrico

Como se puede apreciar en la figura 3.2, se considera una interfaz gráfica en planta desde donde un operario podrá controlar remotamente el medidor volumétrico en la plataforma flotante que se encuentra cientos de metros mar adentro.

3.2 Formato de mensajes comando y mensajes respuesta

De acuerdo a las especificaciones funcionales y formato de mensajes descrito en el anexo C, se considera el protocolo LCP descrito para las tramas de mensajes comando y mensajes respuesta.

El formato de mensaje consta de una secuencia de caracteres de 8 bit (formato hexadecimal) cada uno como se describe en la figura 3.3.

<Hacia><Desde><Estado><Longitud><Dato 0><Dato 1>.....<Dato n>

Fig. 3.3 Formato de mensaje

En donde:

- <Hacia>: viene a ser representado por la dirección de destino hacia donde se desea enviar.
- <Desde>: viene a ser representado por la dirección desde donde está siendo enviado.
- <Estado>: representa el estado del mensaje. La representación de cada bit se puede apreciar en el anexo C.
- <Longitud>: representa la longitud de la porción del mensaje.
- <Dato i>: Byte de la información numero i.

Como ejemplo, podemos observar que si deseamos saber la identificación del equipo medidor que se encuentra en el nodo de dirección 255 al que se desea remotar,

entonces se debe ejecutar un mensaje comando desde el punto de control con dirección asignada de 250 para obtener dicha identificación de la siguiente manera:

- <FAh> <FFh> <02h> <01h> <00h>

El mensaje de respuesta por parte del medidor sería de la siguiente manera:

- <FFh> <FAh> <80h> <0Dh> <00h> <02h> <53h> <52h> <32h> <30h> <30h> <62h> <32h> <2Eh> <30h> <35h> <00h>

Se puede apreciar que el mensaje de respuesta está siendo enviado desde el equipo medidor con nodo de dirección 250 (<FAh>) hacia la dirección 255 (<FFh>) desde donde se envió el mensaje comando. En el estado del mensaje (<80h>) se puede apreciar que el bit 7 se encuentra asignado con el valor de 1, lo que indica que el mensaje es de respuesta. También se observa que la longitud del mensaje es de 13 (<0Dh>), lo que indica que la información de mensaje contiene en código ASCII el nombre del equipo y la revisión "SR200b2.05".

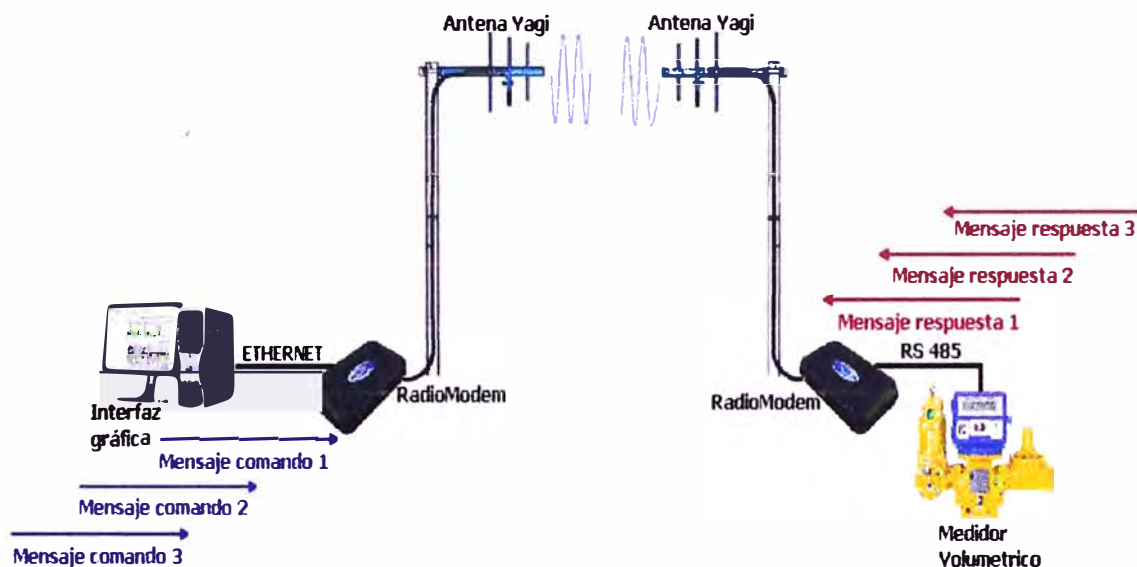


Fig. 3.4 Esquema del envío y recepción de mensajes en el equipo de control remoto

3.3 Conexiones para control de bomba y válvula solenoide

Del Anexo D podemos observar que una de las salidas auxiliares del registrador electrónico es el J12, en donde el terminal 44 corresponde al control de la bomba de despacho de petróleo. Este terminal conjuntamente con el terminal 41 que corresponde a la tierra (GND) entrega un voltaje de tensión continua dependiendo del estado del equipo, cuando el equipo no se encuentra haciendo ningún despacho de combustible este voltaje es de 0 VDC, y cuando el equipo se encuentra en plena actividad de despacho este voltaje es de 5 VDC.

Con esta propiedad podemos encender automáticamente la bomba que hace el despacho y apagarla automáticamente cuando haya concluido dicho despacho, esto se

puede conseguir mediante un relé que se comporte como un interruptor dentro del circuito del motor de la bomba con la fuente de energía eléctrica como se puede apreciar en la figura 3.5.

Debido a que la salida auxiliar J12 se encuentra en el equipo medidor volumétrico y la bomba en planta, se debe tener en cuenta nuevamente una conexión inalámbrica como se muestra en la figura 3.6.

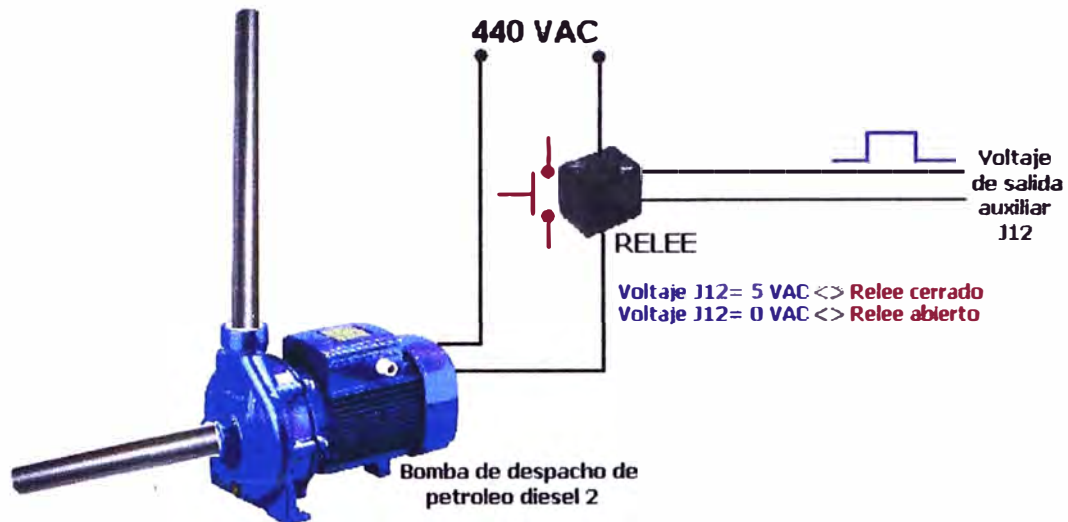


Fig. 3.5 Esquema del control automático de bomba

Se puede observar en la figura 3.6 que los radio módems para este propósito tienen que tener entradas y salidas digitales, y que el relé por lo general debe estar instalado en un tablero eléctrico.

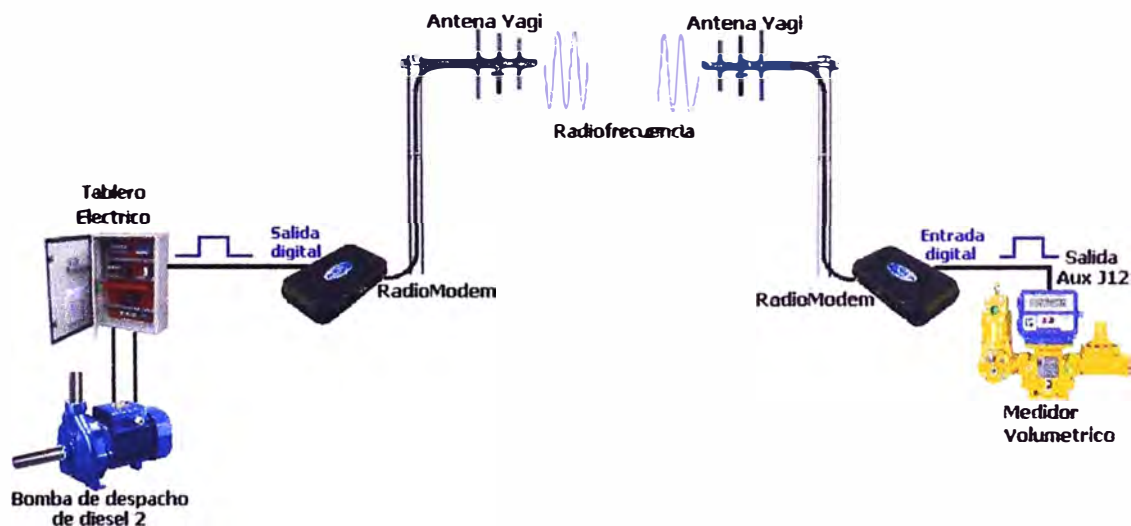


Fig. 3.6 Sistema de control remoto de bomba

La válvula solenoide para predeterminar despachos es un accesorio adicional con el que cuenta LIQUID CONTROLS para sus líneas de medidores volumétricos,

generalmente el modelo E-7 de válvula de solenoides de dos estados es empleado para este propósito, el cual va instalado por el lado de salida del fluido del medidor volumétrico como se aprecia en la figura 3.7 y su hoja técnica es descrita en el anexo I.

Se puede observar del Anexo B y D que las conexiones de la válvula solenoide con el registrador electrónico se realizan en la conexión J13, en donde se observa el cableado respectivo de los 2 solenoides.

Al momento de realizar un despacho predeterminado, la válvula se abrirá automáticamente para dejar pasar el fluido y progresivamente ir cerrándose conforme vaya llegando al valor predeterminado y lograr hacer un despacho exacto.

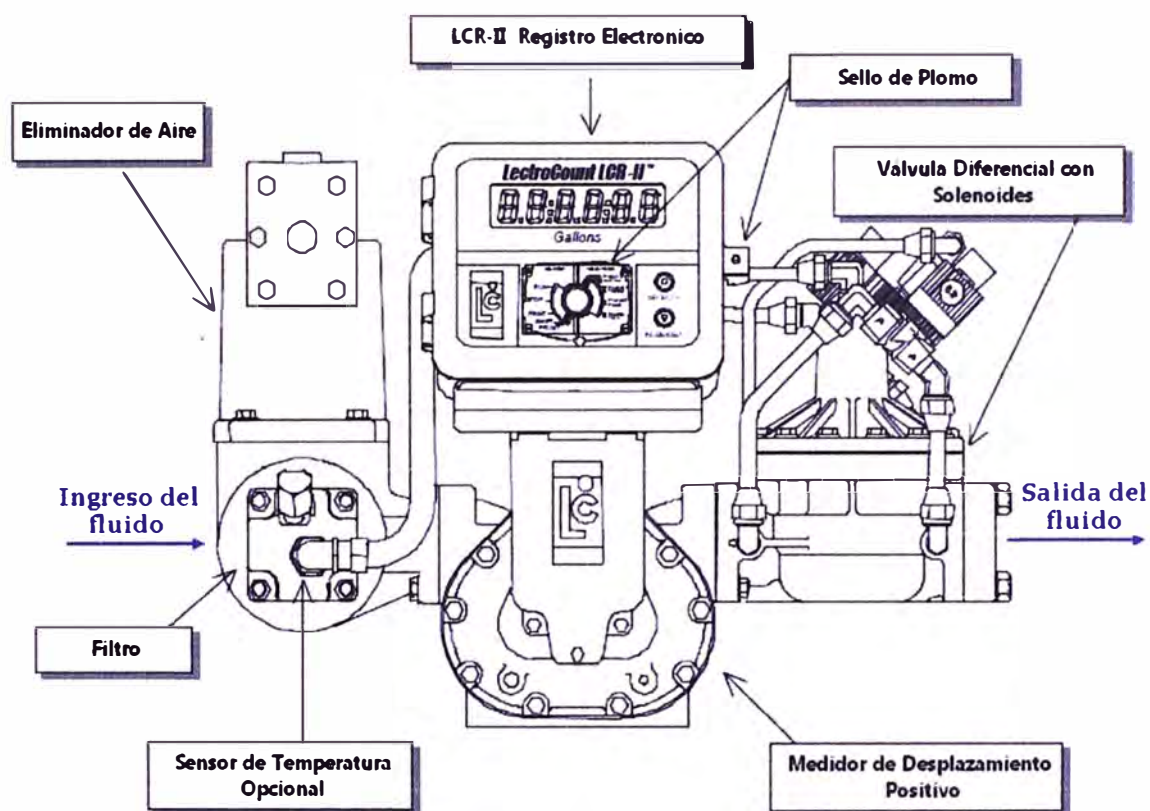


Fig. 3.7 Equipo medidor volumétrico y sus respectivos accesorios

3.4 Código de programación para plataforma de control e interfaz grafica

Se ha elaborado un código en lenguaje de programación JAVA con el cual se logra simular un medidor volumétrico en una computadora, que es remotamente controlado vía ETHERNET desde otra computadora en donde se simula una interfaz. Este control permite realizar lo siguiente:

- Predeterminar un volumen para ser despachado
- Detección de interrupción de un despacho

Se consigue predeterminar un despacho desde la computadora en donde se simula una interfaz enviando un comando mensaje vía ETHERNET a la computadora que simula

al medidor volumétrico. La detección de una interrupción se puede interpretar como fuga de combustible, ya que una parte de la tubería que llega al medidor es de un material flexible conocido como HDPE como se aprecia en la figura 3.8, esto con la finalidad de poder darle cierto grado de movimiento a la plataforma flotante que no se encuentra totalmente fija sobre el mar.

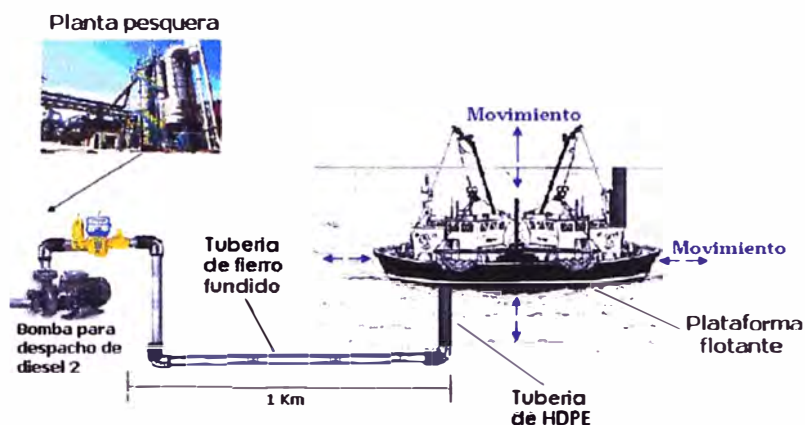


Fig. 3.8 Esquema de la tubería para despacho de petróleo diesel 2

Los códigos desarrollados tanto para la interfaz gráfica como para la simulación del equipo medidor volumétrico se encuentran especificados en los anexos J y K respectivamente.

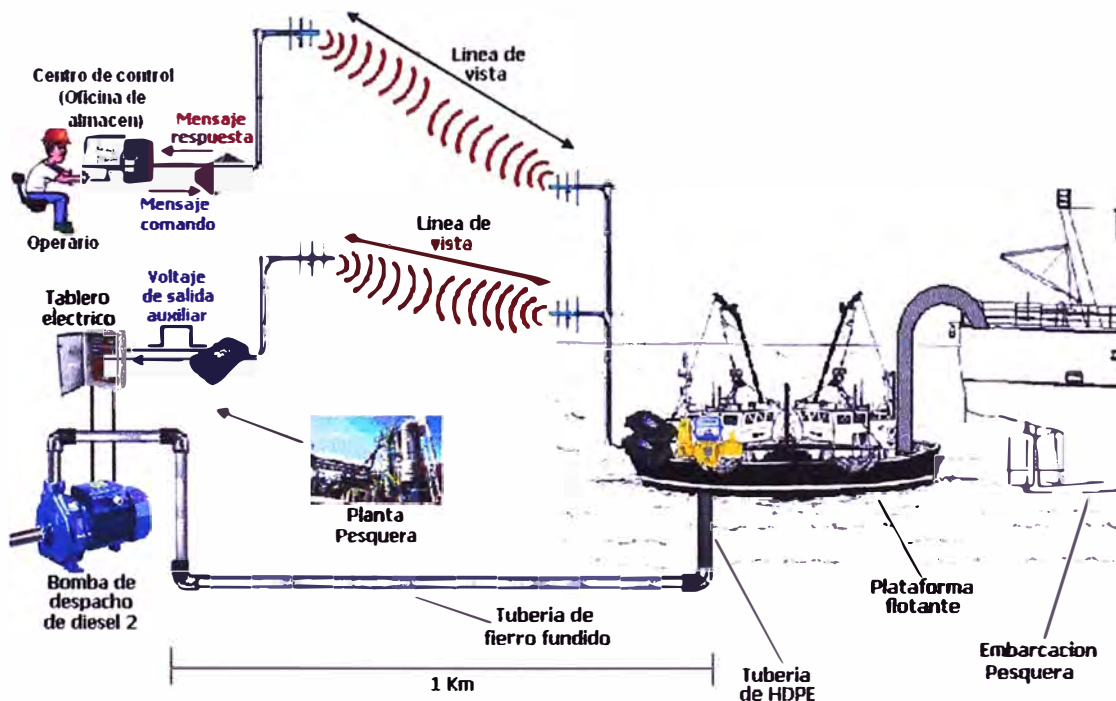


Fig. 3.9 Sistema de monitoreo y control de despacho de petróleo diesel 2

En la figura 3.9 se tiene un esquema del sistema de monitoreo y control del despacho del petróleo que se propone en el presente informe.

CAPÍTULO IV ANÁLISIS DE RESULTADOS

Teniendo como referencia el actual sistema de despacho de combustible a las embarcaciones pesqueras, y presentando como alternativa la propuesta descrita para este propósito, se describen las siguientes ventajas

4.1 Centralización del control y monitoreo del medidor volumétrico

La interfaz gráfica desarrollada en una computadora en las oficinas de almacén permite el control remoto del medidor y de la bomba que hace el despacho del combustible desde planta, con lo que se consigue reducir el número de operarios encargados de supervisar el despacho.

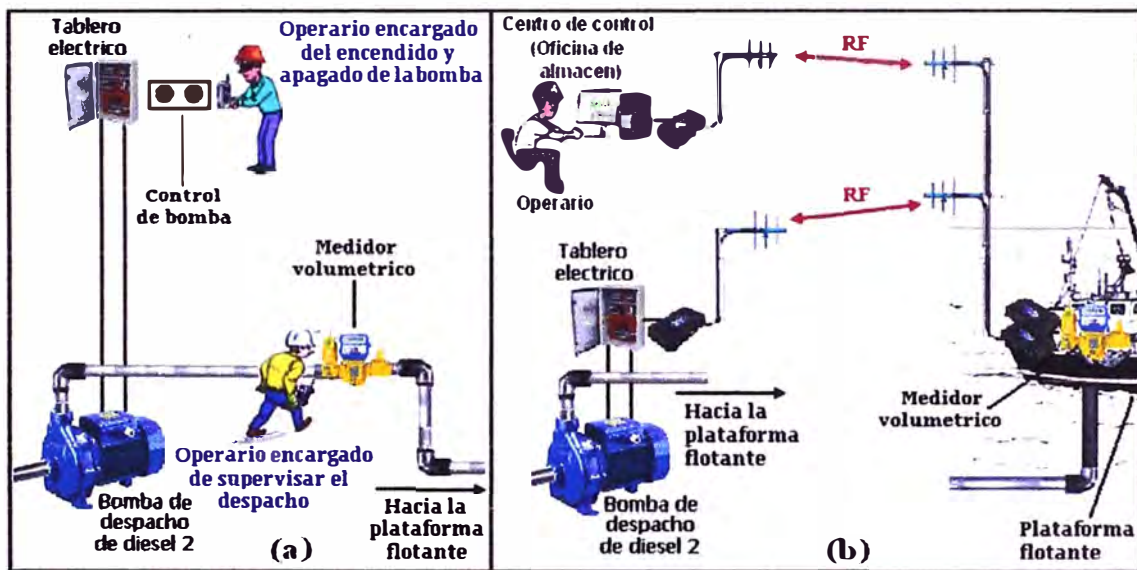


Fig. 4.1 Esquema comparativo entre el sistema de despacho actual y el sistema Propuesto

Como se observa en la figura 4.1, en el lado izquierdo se tiene un esquema del sistema de despacho convencional con el que actualmente cuentan las plantas pesqueras, en donde se puede apreciar que se necesitan de 2 operarios para la supervisión del medidor volumétrico y del control de la bomba respectivamente, esto es necesario ya que, generalmente el control de la bomba no se encuentra cerca del medidor. Además de esto y debido a que el medidor cuenta con una válvula de solenoides que se cierra automáticamente haciendo un despacho exacto que fue predeterminado desde planta, es necesario apagar la bomba en ese instante, ya que con

la válvula totalmente cerrada y la bomba aun encendida, estará ejerciendo una presión considerable con el combustible a las partes internas del equipo medidor con lo cual lo desgastará prematuramente.

A diferencia del sistema de despacho convencional, en el lado derecho de la figura 4.1 se tiene el esquema del sistema de despacho propuesto, en donde se puede apreciar que solo es necesario de un operador para el control total de un despacho, ya que se tiene el control de la bomba por parte del medidor vía radiofrecuencia y el control y monitoreo del medidor también vía radiofrecuencia desde las oficinas del almacén.

4.2 Precisión en el despacho de combustible a las embarcaciones

Debido a que en el sistema de despacho propuesto, el medidor volumétrico se encuentra en la plataforma flotante, este podrá contabilizar exactamente lo que se este entregando a las embarcaciones.

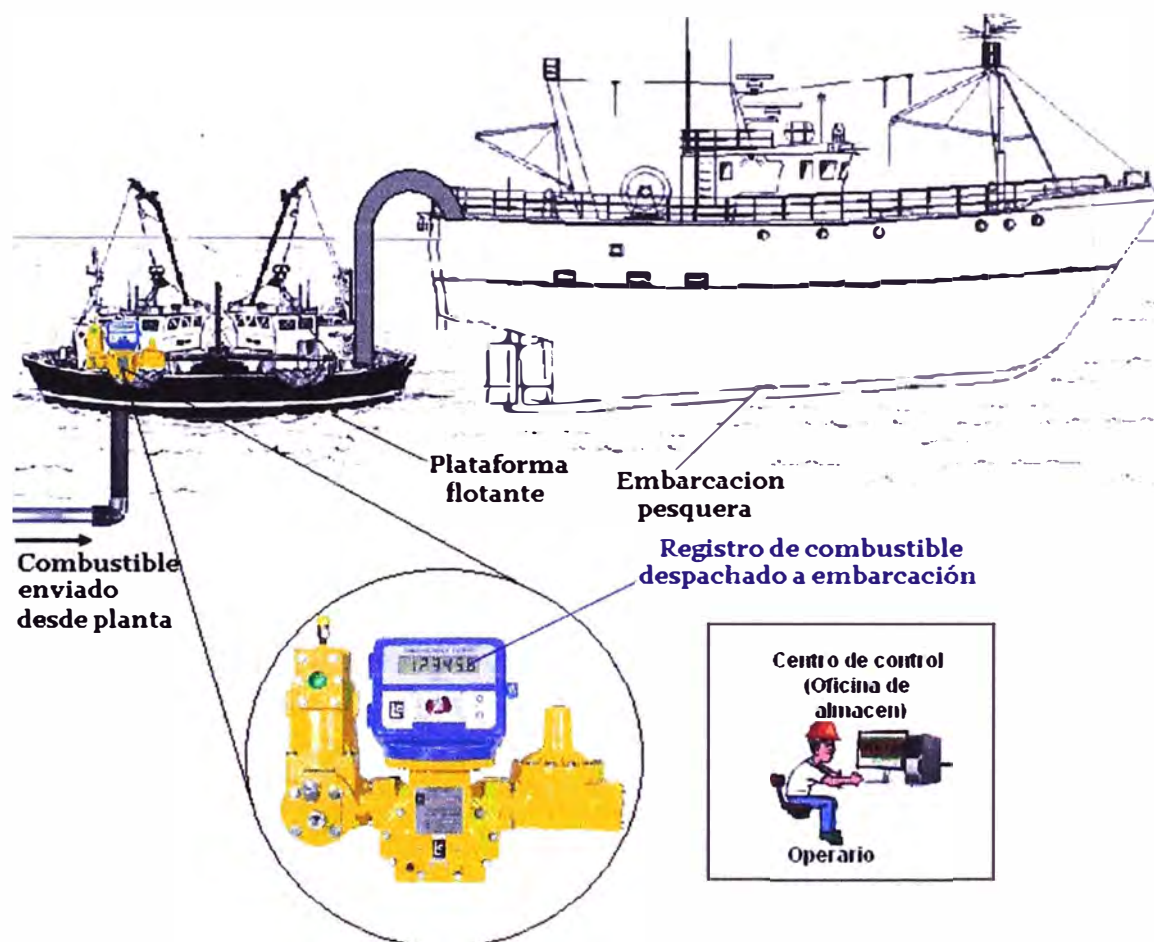


Fig. 4.2 Sistema de despacho de combustible propuesto en plataforma flotante

Como se observa en la figura 4.2, el registrador electrónico del medidor contabilizará el volumen de lo que realmente se esté entregando a las embarcaciones pesqueras, y esta cuantificación podrá ser vista desde las oficinas de almacén.

4.3 Detección de fugas de combustible por averías

Las fugas de combustible por averías en las tuberías ocurren con frecuencia debido a que, parte de esta tubería es de un material flexible conocido como HDPE, el cual con el movimiento de la plataforma en el mar tiende a averiarse, y como se observa en la parte superior de la figura 4.3, no hay ningún medio que pueda detectar la fuga de combustible.

Como consecuencia se pierde combustible que se va al mar y termina siendo un daño ambiental considerable.

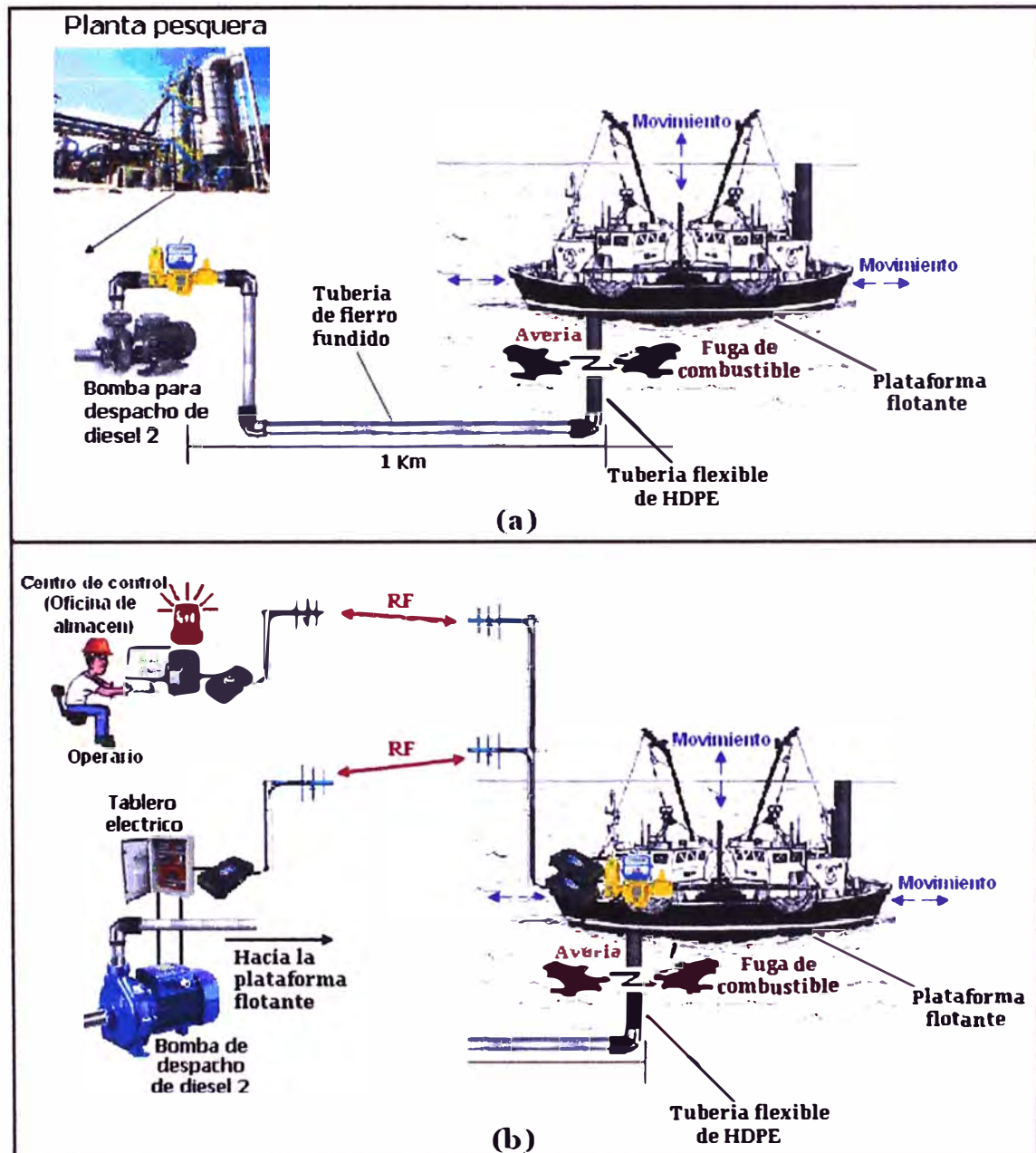


Fig. 4.3 Avería de tubería flexible en sistema de despacho de combustible

Como se observa en el esquema del sistema de despacho propuesto graficado en la parte inferior de la figura 4.3, en donde la interfaz puede detectar dicha avería

considerando el cambio brusco en la velocidad del fluido antes de llegar al valor predeterminado, esto se encuentra especificado en el código desarrollado en el lenguaje de programación JAVA en el anexo I.

4.4 Lenguaje de programación JAVA como software libre

Una de las ventajas de realizar una interfaz grafica en JAVA es la del costo cero al acceso y a la aplicación. Esto permite que el código generado pueda ser modificado tantas veces como se quiera haciendo que el sistema de despacho propuesto sea un sistema que pueda ampliarse para mejoras o para el control de más equipos como se muestra en la figura 4.4.

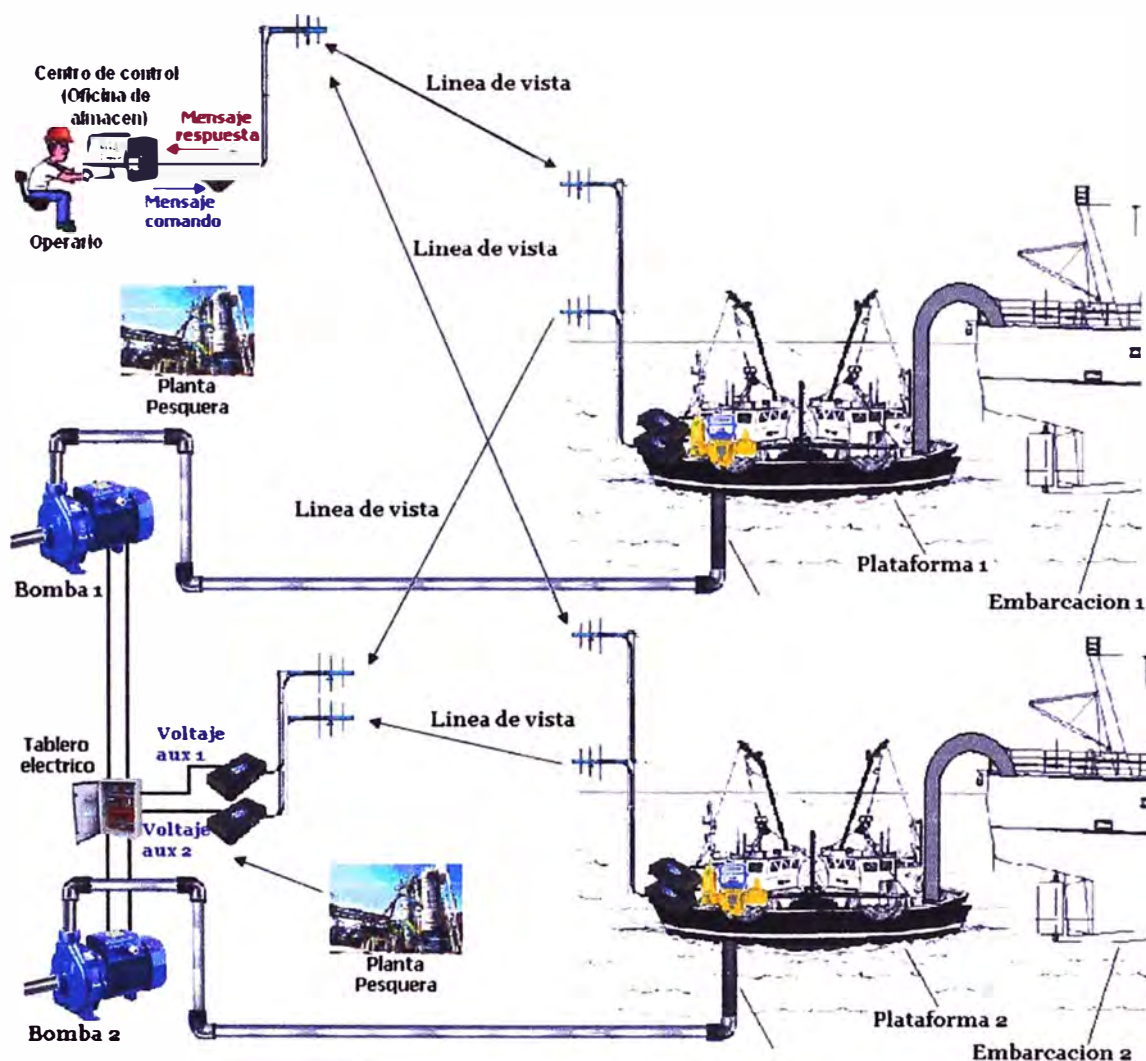


Fig. 4.4 Esquema de control y monitoreo de despacho de combustible de 2 plataformas flotantes

CONCLUSIONES Y RECOMENDACIONES

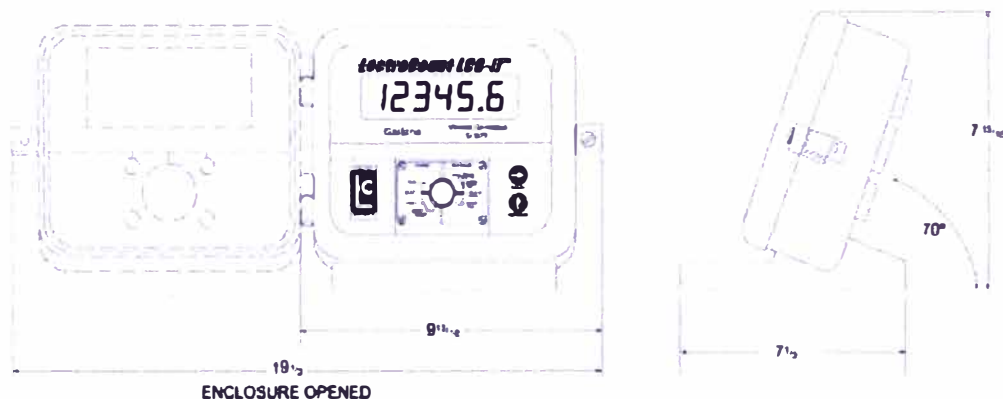
1. La automatización que se presenta en este informe para el sistema de despacho a embarcaciones tiene ventajas significativas cuando se hace comparación con el sistema de despacho actual, tanto en lo efectivo por la precisión en el despacho a las embarcaciones como en lo óptimo y centralizado ya que el monitoreo de todo el sistema sería realizado desde una interfaz gráfica por un solo operador.
2. Debido a que el sistema de despacho presentado en este informe se encuentra desarrollado íntegramente en lenguaje de programación JAVA el cual permite la modificación de los códigos desarrollados, se puede generar algunos comandos alternativos como una alerta al cambio brusco del caudal del fluido que se puede interpretar como la detección de fugas de combustible por averías en las tuberías previniendo así un desastre ambiental mayor, también este sistema puede permitir ampliaciones futuras como la adición de más equipos a ser remoteados y controlados.
3. Es importante tener en cuenta la correcta operación del sistema propuesto ya que, de no encontrarse listo para realizar un despacho podría dañar algunos equipos involucrados como el equipo medidor o la bomba de despacho, también podría verse afectada la cuantificación del fluido. Algunos ejemplos de estos inconvenientes son las válvulas cerradas que interrumpen el despacho o el sobre dimensionamiento de la bomba cuando ésta entrega un caudal de fluido por encima del permitido por el equipo medidor.
4. Debido a las condiciones ambientales de alta humedad y alto grado de corrosión provocado por las brisas marinas, se recomienda que los radio módems y antenas empleadas para este propósito que se encuentran expuestas a estas condiciones sean a prueba de agua que, de acuerdo a las especificaciones sobre los grados de protección IPX, deberían de ser IPX-1 o mejor.

ANEXO A
ESPECIFICACIONES DEL LCR II

LCR-II Especificaciones

El *LectroCount LCR-II™* es un registro electrónico de microprocesador que le ofrece las mismas funciones de operación avanzadas de previos modelos pero con beneficios adicionales. El *LectroCount LCR-II™* le ofrece las siguientes funciones:

- Completamente Aprobado por "Weights & Measures" de E.U.A.
- Aprobado como Clase I, División 2, Grupo C y D
- Encapsulado NEMA 4X con bisagra en la tapa de desplegado y un solo punto de atornillado sellable con sello de plomo
- Puertos para conexión de Conduit. 2 de 3/4" NPT y 5 de 1/2" NPT
- Operación con la mayoría de medidores. El *LCR-II™* acepta entradas de medidores de Desplazamiento Positivo, de Turbina, Electromagnético y Másico.
- Calibración Lineal de Multi-punto para mayor precisión a cualquier razón de flujo
- Productos Múltiples
- Compensación Electrónica de Volumen por Temperatura
- Desplegado LCD de número grande y luz de fondo
- Totalizador de 10 dígitos
- Dos botones de control en el panel frontal con funciones de:
 - "SELECT" para acceso a menús internos
 - "INCREASE"
- Predeterminado, visualización de la razón de flujo y temperatura, selección de productos y más.



Especificaciones

Ambientales:

Encapsulado: Clase I, División 2, Grupo C y D
NEMA 4X

Temp. de Operación: -40° to 70°C (-40° to 158°F)

Aprobaciones:

EUA Weights & Measures NTEP COC 86-022
Canadá Weights and Measures Pendiente

Comunicaciones

RS232/485. Teclado Lap Pad
Computadora Hand Held, PC

Impresor:

Epson 200, 300, 290, 295,
Okidata ML 184T. Axiohm Blaster

Eléctricas

Voltaje: +9 to 28 VDC, 3A Máximo (incluyendo válvulas solenoide)

Entradas:

Sensor de Temp.: Sensor Resistivo RTD de Platino de 4 hilos.
Entradas de Pulso: +5 to 28-voltios pico a pico, de colector abierto de un solo canal o de Cuadratura 0-1500 Hz por canal

Salidas:

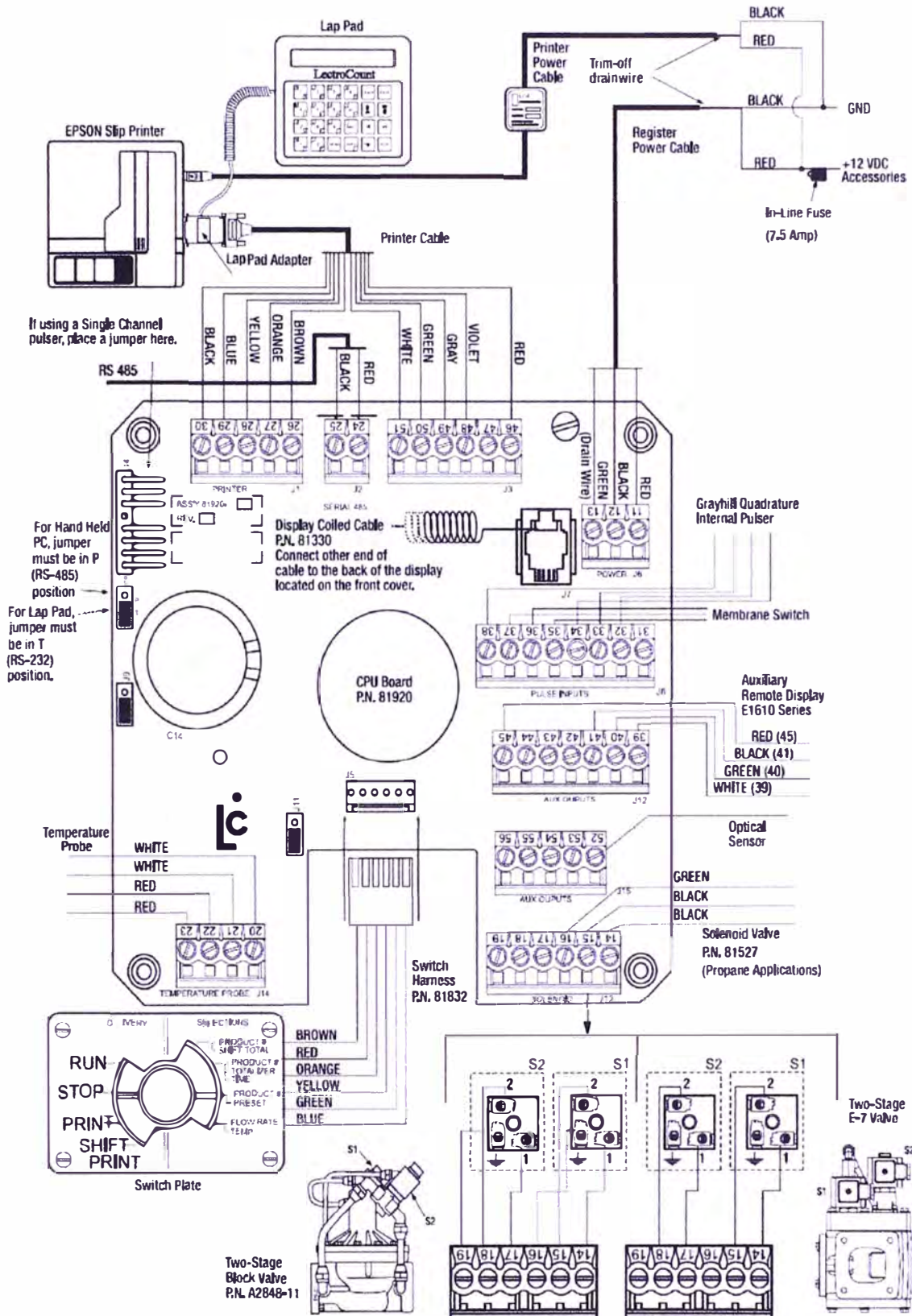
Válvula de Control(2): Salida de dren abierto para válvulas de control de dos etapas, 1 A máximo por canal.

Auxiliares (2): Salida de dren abierto para control auxiliar de propósito general. Aux (1), 1 A Máximo.
Aux. (2), 150 mA Máximo.

Salida de Pulsos: Salida de dren abierto, 1 A Máximo.

ANEXO B
DIAGRAMA DE CABLEADO – LECTROCOUNT LCR II

LectroCount® LCR-II® Wiring Schematic



ANEXO C
ESPECIFICACIONES FUNCIONALES Y FORMATO DE MENSAJES DEL
PROTOCOLO LCP

1 Functional Specifications

The LCP is a general purpose protocol that is to be used in products designed and developed by Liquid Controls, LLC. It provides the general means of communications between a variety of devices over an RS-232, RS-422, or RS-485 hardware link.

1.1 General Requirements

The following list shows the requirements of LCP.

- Any device using LCP can be the master, a slave, or both.
- Message collision control is performed via timing.
- Supported baud rates will be 2400, 4800, 9600, 19200, 57600, or 115200. Note that the LCR and LCR-II boards do not support 115200 baud.
- Characters will be transmitted using one start bit, 8 data bits, and one stop bit. No parity bit is transmitted. It is assumed that the CRC following each message will ensure data integrity.

1.2 LCP Message Format

In the message format described below, each sequence of characters enclosed in angle brackets represents one 8-bit byte which can contain binary data in the range of 0 to 255 decimal.

LCP messages use the following basic format:

```
~~<to><from><status><len><data0>...<datalen-1><crcl><crch>
```

<to> Destination node address of the message being sent. This number can range from 0 to 255. Node address 0 is used to broadcast messages to all nodes in the network.

<from> Source node address of the message being sent. This number can range from 1 to 255.

<status> Status of message. The status byte looks as follows:

Bit	Description	
0	Message Identifier. This bit will toggle between 0 and 1 so that the slave device can distinguish between new messages and repeat messages.	
1	Synchronization. If this bit is set, the message received will be processed as a new message regardless of the state of the message identifier bit.	
	Command Messages (Bit 7 = 0)	Response Messages (Bit 7 = 1)
2	Check Request. Set if the host wants to check on the status of the previous request that was delayed due to the busy bit.	Busy. Set by the slave device to indicate to the host that it can't respond to the request but will do so as soon as possible.
3	Abort Request. Set if the host wants to abort a delayed request.	Request Aborted. Set if the slave aborted an active delayed request.
4	Reserved.	No Request Active. Set if the slave receives a check request and there is no delayed request active.
5	Reserved.	Buffer Overrun. Set if the receive buffer in the slave device was not large enough to contain the entire message.
6	Reserved.	Not Supported. Set if the command in the message received by the slave is not supported by the slave device.
7	Message Type. Set to 0 indicating the message is a command message from a host.	Message Type. Set to 1 indicating the message is a response message from a slave.

<len> Length of the data portion of the message.

<data_i> Data byte number i. The contents of the data portion of the message is device specific. See Section 2 for the details of the specific messages for the LCR and LCR-II.

<crc_l> Low order byte of CRC.

<crc_h> High order byte of CRC.



2 Data Portion of LCR Specific Messages

The field numbers, device status byte, printer status byte, delivery status word, delivery code word, command bytes, and security level are described in detail in the **LCP Device #02: LCR manual, Publication #500219** or can be determined via the defines in LCP02.h.

It is important to note that all integers, longs, and floats are stored in the LCR with the MSB at the lowest address and the LSB at the highest address. This is opposite of how the Intel 80x86 processor stores its numeric data. For example, a no-flow timer of 180 will be transmitted to the host as 00h followed by B4h. In an Intel machine, these bytes need to be swapped to B4h followed by 00h before being used by the application.

LCR volume fields are stored as long integers with an implied decimal point as defined by Field #39. Floating point numbers are stored using the IEEE-754 format.

Data Type Definitions

Type	Name	Description
A	ASCII Character	Contains a one byte ASCII character. The number of characters stored in the field is <i>size</i> .
AZ	ASCIIZ String	Contains a NUL terminated string of ASCII characters. The maximum length of the string is <i>size-1</i> .
D	Double Floating Point	Contains an eight byte floating point number in IEEE-754 format.
F	Floating Point	Contains a four byte floating point number in IEEE-754 format.
SB	Signed Byte	Contains a one byte integer in the range -128 to 127.
SI	Signed Integer	Contains a signed integer where the range is operating system dependant. On 16-bit machines, it is a two byte signed integer in the range -32768 to 32767. On 32-bit machines, it is a four byte signed integer in the range -2147483648 to 2147483647.
SL	Signed Long	Contains a four byte integer in the range -2147483648 to 2147483647.
SS	Signed Short	Contains a two byte integer in the range -32768 to 32767.
ST	Structure	Data element is a structure where the contents vary depending on the structure definition.
UB	Unsigned Byte	Contains a one byte integer in the range of 0 to 255.
UI	Unsigned Integer	Contains an unsigned integer where the range is operating system dependant. On 16-bit machines, it is a two byte unsigned integer in the range 0 to 65535. On 32-bit machines, it is a four byte unsigned integer in the range 0 to 4294967295.
UL	Unsigned Long	Contains a four byte integer in the range 0 to 4294967295.
UN	Union	Data element is a union where the contents vary depending on the union definition.
US	Unsigned Short	Contains a two byte integer in the range 0 to 65535.
V	LCR Volume	Contains a signed four byte integer in the range -2147483648 to 2147483647 with an implied decimal point defined by the decimals field in the record.

Get Product ID (Generic LCP Message)

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to get the product ID. Set to 00h.
Total	01			

Return from Get Product ID

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	productID	Product identification byte. The LCR will return 02h.
02	n	AZ	name	ASCIIZ string containing the product name and revision string. The string length of name should be no larger than 15 implying that n should be no larger than 16.
Total	n+02			

Get Field Data

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to get field data. Set to 20h.
01	01	UB	fieldNum	Field number to get.
Total	02			

Return from Get Field Data

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
02	n	?	fieldData	Field data. Size and type dependant on field number.
Total	n+02			

Set Field Data

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to set field data. Set to 21h.
01	01	UB	fieldNum	Field number to set.
02	n	?	fieldData	Field data. Size and type dependant on field number.
Total	n+02			

Return from Set Field Data

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
Total	02			



Print Text on LCR Printer

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to print text. Set to 22h.
01	n	UB	prnText	Text to print.
Total	n+01			

Return from Print Text on LCR Printer

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
Total	02			

Note that the following "Get Machine Status" message is identical to the "Get Delivery Status" message defined below with the exception that the printer status is returned. If the printer is offline, a two second delay will occur during the "Get Machine Status" request whereas no such delay will occur during a "Get Delivery Status" request. If the printer status is not needed, use the "Get Delivery Status" message below.

Get Machine Status

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to get machine status. Set to 23h.
Total	01			

Return from Get Machine Status

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
02	01	UB	prnStatus	Printer status byte.
03	02	US	delStatus	Delivery status word.
05	02	US	delCode	Delivery code word.
Total	07			

Issue Command

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to issue command. Set to 24h.
01	01	UB	command	Command byte.
Total	02			

Return from Issue Command

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
Total	02			



Set Device Address

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to set device address. Set to 25h.
01	01	UB	newAddr	New device address.
Total	02			

Return from Set Device Address

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
Total	02			

Get Version Number

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to get version number. Set to 26h.
Total	01			

Return from Get Version Number

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
02	02	UB	version	Major and minor version number of LCR RS-485 software.
Total	04			

Get Security Level

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to get security level. Set to 27h.
Total	01			

Return from Get Security Level

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
02	01	UB	security	Security level in the LCR.
Total	03			



Note that the following "Get Delivery Status" message is identical to the "Get Machine Status" message defined above with the exception that the printer status is not returned. If the printer is off-line, a two second delay will occur during the "Get Machine Status" request whereas no such delay will occur during a "Get Delivery Status" request. If the printer status is not needed, this is the message to use.

Get Delivery Status

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to get machine status. Set to 28h.
Total	01			

Return from Get Delivery Status

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	01	UB	devStatus	Device status byte.
02	02	US	delStatus	Delivery status word.
04	02	US	delCode	Delivery code word.
Total	06			

Activate Pump & Print

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to activate Pump & Print mode. Set to 29h.
Total	01			

Return from Activate Pump & Print

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
Total	01			



Get Transaction Record (SR216)

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to get the oldest transaction record. Set to 2Ah.
Total	01			

Return from Get Transaction Record

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
01	04	F	temperature	Temperature of product at the end of the delivery.
05	04	UL	customerID	Customer ID number.
09	04	SL	saleNumber	Sale number.
0D	04	V	gross	Gross volume delivered.
11	04	V	net	Net volume delivered.
15	02	US	status	Delivery status at the end of the delivery.
17	01	UB	compType	Active compensation type.
18	01	UB	dateFormat	Date format used in the date and time field.
19	01	UB	decimals	Decimal setting for the volume fields.
1A	01	UB	product	Product ID delivered.
1B	01	UB	qtyUnits	Unit of measure for the volume fields.
1C	01	UB	tempScale	Temperature scale.
1D	12	AZ	dateTime	Date and time at the end of the delivery.
Total	2F			

Delete Transaction Record (SR216)

Offset	Size	Type	Name	Description
00	01	UB	msgID	Message ID to delete the oldest transaction record. Set to 2Bh.
Total	01			

Return from Delete Transaction Record

Offset	Size	Type	Name	Description
00	01	UB	rc	Return code.
Total	01			



ANEXO D
TABLA DE CABLEADO DE LOS TERMINALES DEL REGISTRADOR ELECTRÓNICO

Tablas de Cableado

Este apéndice provee una descripción tabular del cableado de las terminales del LCR-II. Consulte estas tablas cuando haya cableado incorrectamente una terminal o para diagnosticar un problema que se presente en campo.

CONEXION DE VOLTAJE (J6)

Conector/ Terminal	Señal	Color de Cable	Conectese a:
J6-13	Earth	Verde/Blindaje	Sin Conexion*
12	GND	Negro	Tierra
11	+12V IN	Rojo	+12VDC (circuito de accesorios del vehiculo)

*En el extremo que se conecta a la caja de fusible, corte el blindaje y ponga cinta eléctrica.

NOTA: El cable de voltaje del LCR-II incluye un fusible y porta-fusible para protección del sistema contra un corto circuito en el cable. Liquid Controls recomienda que el fusible siempre sea instalado cuando no existe otro tipo de protección en el vehiculo para este circuito.

CONEXION PUERTO RS-485 (J2)

Conector/ Terminal	Señal	Color de Cable		Conectese a:
		User Supplied	Cable #81513	
J2-25	485-B	Negro	Violeta	Terminal
24	485-A	Rojo	Rojo	Terminal

NOTA: El Jumper J10 de la tarjeta del LCR-II se debe de mover a posición 485.

CONEXION DEL SENSOR DE TEMP. (J14)

Conector/ Terminal	Señal	Color de Cable	Conectese a:
J14-23	RTD-D	Rojo	Sensor de Temp.
22	RTD-S	Rojo	Sensor de Temp.
21	RTD-D	Blanco	Sensor de Temp.
20	RTD-S	Blanco	Sensor de Temp.

CONEXION IMPRESOR (J1)

Conector/ Terminal	Señal	Color de Cable		Conectese a:
		Cable #81464	Cable #81513	
J1-30	GND	Verde	Negro	Impresor, Pin 7
29	CTS	Negro	Azul	Impresor, Pin 20
28	RXD	Rojo	Amarillo	Impresor, Pin 2
27	TXD	Café	Naranja	Impresor, Pin 3
26	RTS	Blanco	Café	Impresor, Pin 6

CONEXION VALVULA SOLENOIDE (J13)

Conector/ Terminal	Señal	Color de Cable	Conectese a:
J13-19	Earth	Verde	Tierra Valvula Solenoide 2
18	Out 6	Negro o Rojo	Solenoide 2
17	+12V	Negro o Rojo	Solenoide 2
16	Earth	Verde	Tierra Valvula Solenoide 1
15	Out 7	Negro o Rojo	Solenoide 1
14	+12V	Negro o Rojo	Solenoide 1

CONEXION LAP PAD R2-232 (J3)

Conector/ Terminal	Señal	Color de Cable		Conectese a:
		Cable #81467	Cable #81513	
J3-51	GND	Verde	Blanco	Tierra, Pin 5
50	CTS	Blanco	Verde	RTS Terminal, Pin 4
49	RXD	Café	Gris	TXD Terminal, Pin 3
48	TXD	Rojo	Violeta	RXD Terminal, Pin 2
47	RTS	*	*	CTS Terminal, Pin 8
46	+12V	Negro	Rojo	Lap Pad +12, Pin 8

NOTA: El Jumper J10 de la tarjeta del LCR-II se debe de mover a la posición de 232.

* Para una terminal estandar RS-232 (que no sea del Lap Pad), el cable negro se conecta a J3-47.

CONEXION DE BOTONES EXTERNOS Y DE PULSADOR INTERNO (J8)

Conector/ Terminal	Señal	Color de Cable	Conectese a:
J8-38	GND	Negro	Tierra Pulsador Interno o de POD
37	GND	Negro	Tierra del Boton de Control
36	In1	Violeta	Boton de Control INCREASE
35	In2	Gris	Boton de Control SELECT
34	In3	Verde	Pulsador Interno o POD "B"
33	In4	Blanco	Pulsador Interno o POD "A"
32	+5V Out	Rojo	Pulsador Int "+"
31	+12 V Out	Rojo	POD "+"

Tablas de Cableado

SALIDAS AUXILIARES (J12)

Conector/ Terminal	Señal	Color de Cable	Conectese a:
J12-45	+12V Out	N/A	
44	Out 1 (Aux. 1)	N/A	Bomba o Control Aux.
43	Out 2 (Aux. 2)	N/A	Direccion de Flujo
42	Out 3	N/A	Entrada de Pulso Para Sistema Aux.
41	GND	N/A	
40	Out 4	N/A	Entrada Pantalla
39	Out 5	N/A	Entrada Pantalla

GENERADOR DE PULSOS DE QUAD. VEEDER ROOT (J8)

Conector/ Terminal	Señal	Color de Cable	Conectese a:
J8-38	GND	Blanco	Tierra Pulsador
37	-	-	-
36	-	-	-
35	-	-	-
34	In 3	Naranja	Salida Canal B
33	In 4	Negro	Salida Canal A
32	-	-	-
31	+12V Out	Rojo	Entrada +12V

NOTA: Deberán conectarse resistencias de 300 ohms entre terminales de CANAL B y Tierra, y entre CANAL A y Tierra.

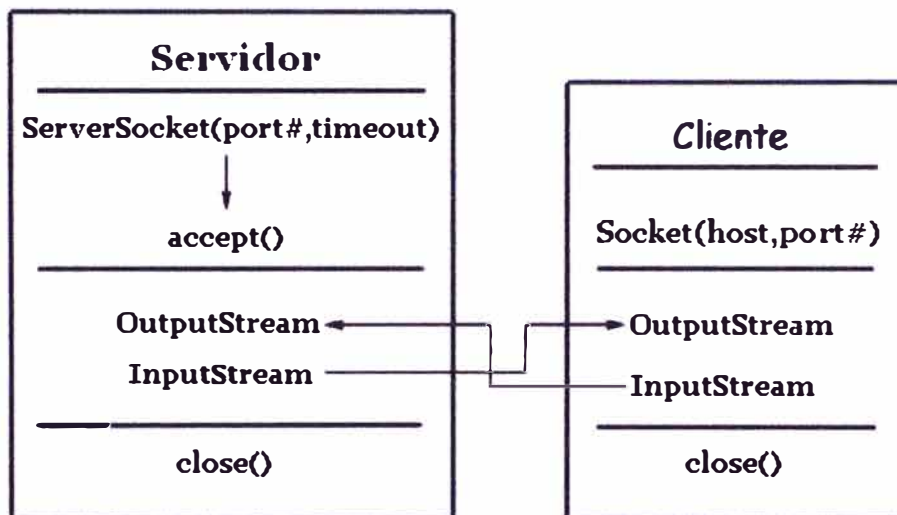
POD LIQUID CONTROLS (J8)

Conector/ Terminal	Señal	Color de Cable	Conectese a:
J8-38	GND	Negro	Tierra Pulsador
37	-	-	-
36	-	-	-
35	-	-	-
34	In 3	Verde	Salida Canal B
33	In 4	Blanco	Salida Canal A
32	-	-	-
31	+12V Out	Rojo	Entrada +12V

PULSADOR DE UN SOLO CANAL (J8)

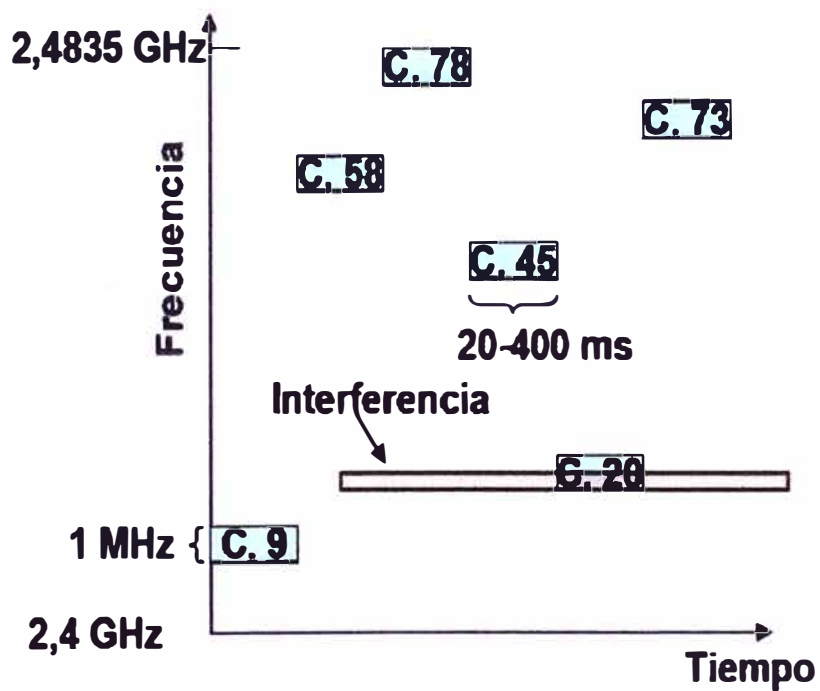
Conector/ Terminal	Señal	Color de Cable	Conectese a:
J8-38	GND	N/A	Tierra Pulsador
37	-	-	-
36	-	-	-
35	-	-	-
34	-	-	-
33	In 4	N/A	Salida de Pulso
32	-	-	-
31	+12V Out	Rojo	Entrada +12V

ANEXO E
MODELO DE COMUNICACIÓN CLIENTE SERVIDOR EN JAVA



ANEXO F
CANALIZACIONES Y TECNICAS FHSS PARA IEEE 802.11b

canal	frecuencia	canal	frecuencia	canal	frecuencia	canal	frecuencia
		21	2 421	41	2 441	61	2 461
2	2 402	22	2 422	42	2 442	62	2 462
3	2 403	23	2 423	43	2 443	63	2 463
4	2 404	24	2 424	44	2 444	64	2 464
5	2 405	25	2 425	45	2 445	65	2 465
6	2 406	26	2 426	46	2 446	66	2 466
7	2 407	27	2 427	47	2 447	67	2 467
8	2 408	28	2 428	48	2 448	68	2 468
9	2 409	29	2 429	49	2 449	69	2 469
10	2 410	30	2 430	50	2 450	70	2 470
11	2 411	31	2 431	51	2 451	71	2 471
12	2 412	32	2 432	52	2 452	72	2 472
13	2 413	33	2 433	53	2 453	73	2 473
14	2 414	34	2 434	54	2 454	74	2 474
15	2 415	35	2 435	55	2 455	75	2 475
16	2 416	36	2 436	56	2 456	76	2 476
17	2 417	37	2 437	57	2 457	77	2 477
18	2 418	38	2 438	58	2 458	78	2 478
19	2 419	39	2 439	59	2 459	79	2 479
20	2 420	40	2 440	60	2 460	80	2 480



ANEXO G
HOJA TÉCNICA DE RADIO MODEM, MODELO FGR2-PE, PARA CONEXIONES
ETHERNET Y RS-485



1880 S. Flatiron Court, Suite F
Boulder, CO 80301

tf 866.923.6168

p 303.381.9200

f 303.786.9948

www.freewave.com
sales@freewave.com

FGR2-PE

Industrial Grade 2.4GHz WiFi and Range Radio

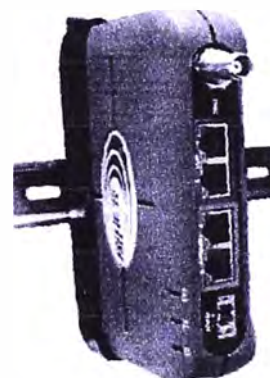
Overview:

The FGR2-PE offers industrial Serial and Ethernet wireless connectivity using license-free spread spectrum for data communication over long distances. The FGR2-PE is compatible with other FreeWave FGR plus family radios and now offers users a second, switched Ethernet port as well as two Serial ports to meet the demands of customers' wireless automation requirements. The future-proof combination of two Serial ports and two switched Ethernet ports give customers the ability to transition from Serial to Ethernet data communications without having to replace their wireless communications infrastructure. This cost-effective flexibility of the FGR2-PE product also reduces the need for additional hardware.

The FGR2-PE offers industrial grade wireless security and encrypted communications, all in one package. All radios are designed, manufactured and 100% tested in Boulder, Colorado.

Features:

- Wide Input Voltage range - 6 to 30 Volt DC.
- Low current draw, at 12 Volts:
 - 140 mA in full time receive
 - 550 mA transmit current
- Strong Signal Performance - Maintains high sensitivity even in harsh RF conditions.
- Versatile - A single radio can operate simultaneously as a slave and as a repeater.
- High Noise Immunity - Superior performance in noise congested environments.
- Secure - Proprietary spread spectrum technology helps prevent Denial of Service (DoS) attacks and unauthorized access.
- High Speed - Up to 154 kbps over-the-air throughput.
- Long Range:
 - Point-to-Point - Up to 60 miles with clear line of sight (LOS).
 - Point-to-Multipoint - Up to 60 miles with clear line of sight (LOS).
- Error Free - 32 bit CRC with automatic retransmission.
- Industrial Grade - 100% tested for RF performance from -40° C to +75° C.
- UL Class 1, Division 2 (pending)



Shown with optional DIN rail

FGR2-PE

Transmitter

Frequency Range	902-928 MHz (FHSS)
Output Power	5 mW to 1 watt
Range - Line of Sight	Point to Point: 60 Miles, Point to Multipoint: 60 miles
Modulation	2 level GFSK
Occupied Bandwidth	230.4 kHz
Hopping Patterns	15 per Band, 105 total user selectable
Hopping Channels	112
Frequency Zones	16 Zones, 7-8 Channels per zone
RF Connector	TNC

Receiver

Sensitivity	-110 dBm for BER 1×10^{-4} at 115 Kbs, -106 dBm for BER 1×10^{-4} at 153.6 Kbs
Selectivity	20 dB at f.c +/- 230 kHz
System Gain	140 db

Data Transmission

Error Detection	32 bit CRC, Retransmit on Error
Data Encryption	AES 128 bit encryption and proprietary spread spectrum technology
Data Interface	Ethernet, EIA-232, 485, 422
Data Connector	Ethernet, 2x 10/100 Base T, Auto Crossover and 2x Serial, RJ45
Protocol	Ethernet: IEEE 802.3 TCP/IP, DHCP, ICMP, UDP, ARP, Multicast, TFTP

Data Interface

Connector	2x Serial RJ 45, 2x 10/100 Ethernet RJ 45
-----------	---

Power Requirements

Operating Voltage	6 to 30 VDC			
Current (mA)	Mode	6 VDC	12 VDC	30 VDC
	Transmit	1.1 A	550 mA	220 mA
	Receive	252 mA	150 mA	63 mA
	Idle	140 mA	71 mA	32 mA

General Information

Operating Temperature Range	-40° C to +75° C. (-40° F to +167° F)
Dimensions	6.8 in (L) x 3.8 in (W) x 1.4 in (H) / 17.3 cm (L) x 9.6 cm (W) x 3.5 cm (H)
Weight	1.3 lbs. / 0.6 kg
Humidity	0 to 95% non-condensing

*Contact FreeWave for implementation details

4.26.10



1880 S. Flatiron Court, Suite F
Boulder, CO 80301

tf 866.923.6168
p 303.381.9200
f 303.786.9948

www.freewave.com
sales@freewave.com

ANEXO H
HOJA TÉCNICA DE RADIO MODEM, MODELO FGRI0, PARA TRANSMISION DE
VOLTAJES CONTINUOS



1880 S. Flatiron Court, Suite F
Boulder, CO 80301

t 866.923.6168
p 303.381.9200
f 303.786.9948

www.freewave.com
sales@freewave.com

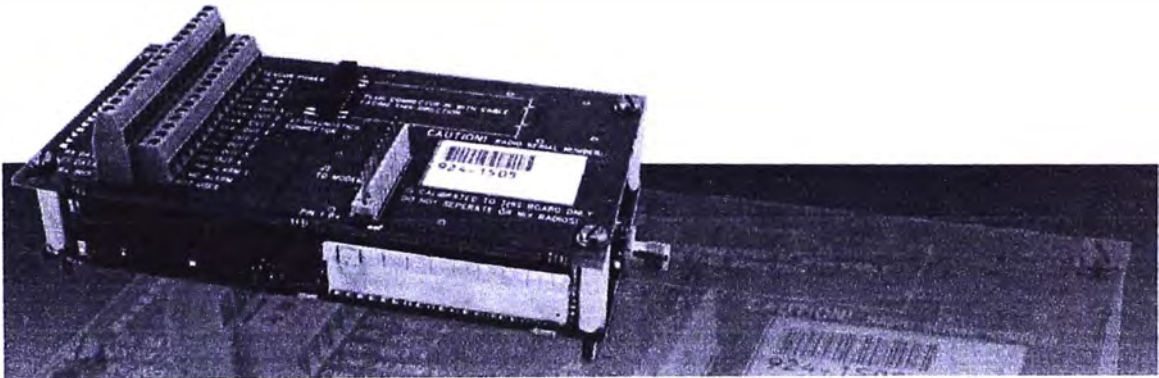
IO - SERIES

Overview:

The FreeWave® Technologies FGRIIO System provides outstanding performance and versatility in wireless transmission of process-control signals. FGRIIO offers "transparent" acquisition, transport and reconstruction of analog, digital and power signals, eliminating the need for associated buried wiring. The RTU requires no altered programming. The FGRIIO is Class 1 Division 2 Approved and is lower-cost and provides better signal integrity than vulnerable wiring.

Features:

- Frequency Hopping - Communication and diagnostics between the IO Master and the IO Slaves.
- Affordable - Low installation and long-term maintenance costs.
- Low latency - Less than one second signal delay.
- High Accuracy - FGRIIO System analog signal fidelity is factory calibrated and drift with time and temperature is much less than that of transducers.
- Short Range/Low power - Suitable for solar powered installations.
- Error Free Communications - 32 bit CRC with automatic retransmission.
- Industrial Grade Specifications - 100% tested for RF performance from -40° C to +75° C.
- Master Input voltage range - Input voltage range is 6-30 VDC at full RF output power.
- Noise Immunity - Superior performance in noise congested environments.
- Secure - proprietary spread spectrum technology prevents unauthorized access.
- Slave Radio - Accepts 2 Digital Inputs (DI), 2 Analog Inputs (AI) and switches 2 Digital Outputs (DO).
- Master Radio - Mirrors signals for up to 4 Slaves and provides Link and Command Alarm signals.
- Wire Replacement - FGRIIO System accuracy is not diminished by distance as it may be in wired systems.



IO-SERIES

Transmitter				
Frequency Range	902-928 MHz (FHSS)	Hopping Patterns	15 per Band, 105 total, user selectable	
Output Power	5 mW to 1 Watt (+30 dBm)	Hopping Channels	50 to 112, user selectable	
Range, Line of Sight	2 Miles with clear LOS to IO Slave, 60 miles to network Master	Hopping Bands	7, user selectable	
Modulation	2 level GFSK	RF Connector	Type SMA	
Occupied Bandwidth	230 kHz			
Master Receiver		Master Analog Outputs		
Sensitivity	For 10 ⁻⁶ BER -108 dBm, 10 ⁻⁴ BER -110 dBm	Number of Outputs	4, can be mapped to up to 4 slaves	
Selectivity	20 dB at fc ± 115 kHz, 60 dB at fc ± 145 kHz	Accuracy, Resolution	+/- .1%, 16 bit	
System Gain	140 dB	Output Range	.2-5.62V, >10 Kohm Load Resistance	
Master Digital Outputs		Master Digital Inputs		
Number of Outputs	4 per Master, 1 Link, 1 Command Alarm	Number of Inputs	4	
Output Connector	Mini Phoenix (3.55mm)	Master Input to Slave Output Delay	1 sec. Max	
Slave Input to Master Output Delay	1 sec. Max	Voltage Range	0 - 30 V	
Signal Output Voltage Range	0 - 4.6 V			
Data Transmission				
Error Detection	32 bit CRC, Retransmit on Error			
Data Encryption	Dynamic Key Substitution			
Link Throughput	115.2 Kbps			
Data Interface	Serial			
Protocol	RS232 / 485 / 422, 1200 Baud to 115.2 KBaud			
Data Connector	10-pin header with locking ramp, 0.1 inch spacing, power/data connector.			
Diagnostics Interface				
Connector	Separate 20-pin PCB header			
Power Requirement				
Operating Voltage	6 to 30 VDC			
Current (mA)	Mode	6VDC	12 VDC	30 VDC
	Transmit	1A	500 mA	200 mA
	Receive	140 mA	86 mA	43 mA
	idle	120 mA	70 mA	38 mA
General Information				
Operating Temperature Range	-40° C to +75° C			
Dimension	140 L x 70 W x 34 H (mm)			
Weight	137 g			
Humidity	0 to 95% non-condensing			

FreeWave® Radios Require Professional Installation.

Specifications may change at any time without notice. © 2019 FreeWave Technologies, Inc.



1880 S. Flatiron Court, Suite F
Boulder, CO 80301

tf 866.923.6168
p 303.381.9200
f 303.786.9948

www.freewave.com
sales@freewave.com

ANEXO I
HOJA TÉCNICA DE VÁLVULA DE SOLENOIDES DE DOS ESTADOS



Specification Sheet

E-7 Electronic Two-Stage Preset Valve



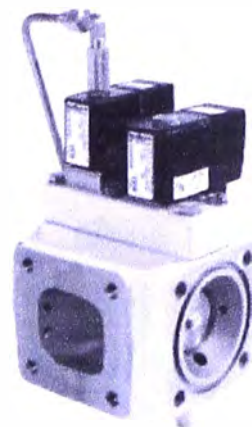
The Liqui Controls E-7 Electronic, Two-Stage, Preset Valve is designed for refined fuel applications including gasoline, gasohol, diesel fuel, and fuel oil. The valve is available as part of a complete, new metering system with electronic registration or as a retrofit when upgrading from mechanical to electronic registration. The E-7 valve may be used with M5, M7, and M10 Class 1 meters.

Features of the E-7 Valve include:

- Indexible design for 90° rotational adjustment of valve outlet.
- Interchangeability (same mounting dimensions) with the V-7 and K-7 mechanical valves
- Two-stage operation for preset deliveries
- Fast response for quick valve shut-off

The E-7 Series electronically-actuated piston valve is mounted on the meter outlet and provides a compact installation. The valve is operated by two solenoids. These solenoids control two-stage closure with a dwell period to eliminate hydraulic shock.

The valve solenoids are wired to a Liqui Controls LectroCount Electronic Register and are controlled by the register software. During a delivery, the valve opens to allow full flow through the meter. For preset deliveries, the full flow solenoid de-energizes at the point where dwell flow is activated. This reduces the flow rate through the meter. Once the delivery preset volume is reached, the valve fully closes without hydraulic shock to the system.



Technical Specifications

Environmental Rating
NEMA 4X

Safety
Designed to meet Class I, Division 2 requirements

Materials of Construction
Valve Body: Aluminum
Elastomers: Viton O-Rings, Teflon Seal, Acetal Guide Rings

Pilot Control System
Solenoids, Tubing and Fittings: Brass solenoids and fittings with flared copper tubing

Temperature Rating
-40° to 160°F (-40° to 71°C)

Pressure Rating
Maximum non-shock working pressure
• 150 PSI (10.3 BAR)
Maximum differential pressure
• 100 PSI (6.9 BAR)

Maximum Nominal Flow
150 GPM (568 L/min) for 30 SSU products

Viscosity Range
30 to 100 SSU

Products
Class 1: refined fuels (gasoline, gasohol, diesel fuel, and fuel oil)

Solenoids
Voltage: +10.2 to 13.6 VDC
(Optional Voltage): +20.4 to 27.2 VDC
Current: S1: 1 Amp, S2: 1 Amps maximum

Specifications

www.lcmeter.com

ANEXO J
CÓDIGOS EN LENGUAJE DE PROGRAMACION JAVA CORRESPONDIENTES A LA
SIMULACION DE LA INTERFAZ GRAFICA PARA EL CONTROL DE UN MEDIDOR
VOLUMETRICO DE PETROLEO

CONTROLADOR**ControladorServer.java**

```

package com.controla.controlador;

import java.io.IOException;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Willy
 */
public class ControladorServer implements Runnable{
    ServerSocket server;
    Socket socket;
    boolean active=false;
    public boolean isActive(){
        return active;
    }
    public void init(int port) {
        try {
            server=new ServerSocket(port);
        } catch (IOException ex) {
            Logger.getLogger(ControladorServer.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public void action(){
        active=true;
        new VentanaControlador(socket);
        active=false;
    }
    public void run(){
        try {
            init(1321);
            InetAddress host = InetAddress.getLocalHost();
            System.out.println("[Controlador] Servidor iniciado en el puerto 1321");
            System.out.println("[Controlador] Hostname:"+host.getHostName());
            System.out.println("[Controlador] Host Address:"+host.getHostAddress());
            while(true){
                socket=server.accept();
                if(isActive()){
                    System.out.println("Existe un surtidor activo");
                }else{
                    System.out.println("Activando el surtidor.");
                    this.action();
                }
            }
        } catch (IOException ex) {
            Logger.getLogger(ControladorServer.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```
}
}
```

IniciarControlador.java (Clase principal)

```
package com.controla.controlador;
/**
 *
 * @author Willy
 */
public class IniciarControlador {
    public static void main(String[] args) {
        // TODO code application logic here
        ControladorServer controlador=new ControladorServer();
        Thread thread=new Thread(controlador,"Controlador");
        thread.start();
    }
}
```

VentanaControlador.java

```
package com.controla.controlador;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.text.DecimalFormat;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.SwingWorker;
/**
 *
 * @author Willy
 */
public class VentanaControlador extends JFrame implements Runnable {
    Socket socket=null;
```

```

JTextField txtIndicador;
JPanel panel=new JPanel();
JTextArea taskOutput;
private Task tarea;
JTextField txtCantidad;
int cantidadMaxima=0;
double cantidadDespachada=0.0;
JButton cmdParar;
//boolean detenido=false;
Random random = new Random();
double factor=0.0;
public VentanaControlador(Socket s){
socket=s;
Thread thread=new Thread(this);
thread.start();
}

public void initComponents(){
//Panel para los componentes graficos
panel.setLayout(new BorderLayout());
txtIndicador=new JTextField();
Font font=new Font(Font.SANS_SERIF,Font.PLAIN,64);
txtIndicador.setFont(font);
txtIndicador.setBackground(Color.BLACK);
txtIndicador.setForeground(Color.GREEN);
// Area para informacion de avance
taskOutput = new JTextArea(5, 20);
taskOutput.setMargin(new Insets(5,5,5,5));
taskOutput.setEditable(false);
// Agrega el panel al formulario
txtCantidad=new JTextField();
txtCantidad.setEditable(false);
cmdParar=new JButton("Parar");
cmdParar.addActionListener(new ActionListener() {

@Override
public void actionPerformed(ActionEvent e) {
tarea.cancel(true);
}
});
panel.add(cmdParar,BorderLayout.EAST);
panel.add(txtCantidad,BorderLayout.SOUTH );
panel.add(txtIndicador);
add(panel, BorderLayout.PAGE_START);
add(new JScrollPane(taskOutput), BorderLayout.CENTER);
panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
//Propiedades basicas del formulario
this.setSize(600, 400);
this.setTitle("Controlador");
this.setVisible(true);
// Inicia la tarea
try {
Thread.sleep(5000);
} catch (InterruptedException ex) {

```

```

Logger.getLogger(VentanaControlador.class.getName()).log(Level.SEVERE, null, ex);
}
tarea = new Task();
tarea.addPropertyChangeListener(new PropertyChangeListener() {
@Override
public void propertyChange(PropertyChangeEvent evt) {
if ("progress" == evt.getPropertyName()) {
int progress = (Integer) evt.getNewValue();
if(progress==100){
txtIndicador.setText(""+(double) cantidadMaxima);
cantidadDespachada=(double) cantidadMaxima;
}else{
cantidadDespachada=(double)progress*factor;
txtIndicador.setText(""+ new DecimalFormat("#.0#").format(cantidadDespachada));
}
}
});
tarea.execute();
}

```

```

public void run(){
BufferedReader in = null;
try {
//Imprime lo que pasa el cliente
this.initComponents();
in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
String textoRemoto=in.readLine();
cantidadMaxima=Integer.parseInt(textoRemoto);
factor=(double) cantidadMaxima/100.0;
//System.out.println("factor:"+factor);
String mensaje="[Controlador] Cantidad a despachar:"+textoRemoto+" galones";
txtCantidad.setText(mensaje);
System.out.println(mensaje);
//accion();
} catch (IOException ex) {
Logger.getLogger(VentanaControlador.class.getName()).log(Level.SEVERE, null, ex);
} finally {
try {
in.close();
} catch (IOException ex) {
Logger.getLogger(VentanaControlador.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

```

class Task extends SwingWorker<Void, Void> {

```

```

boolean detenido=false;

```

```

@Override
public Void doInBackground() {
Random random = new Random();
int progress = 0;

```

```

setProgress(0);

while (progress < 100) {
//Sleep for up to one second.
if(this.isCancelled()){
detenido=true;
break;
}
try {
Thread.sleep(random.nextInt(500));
} catch (InterruptedException ignore) {}
progress++;
setProgress(Math.min(progress, 100));
}
return null;
}

@Override
public void done() {
setCursor(null);
if(!detenido){
String mensaje=" El despacho s"
+ ""
+ "Se completó con éxito."+
" Se despacharon "+new DecimalFormat("#.0#").format(cantidadDespachada)+" galones";
taskOutput.append(mensaje);
llamarDespachador(mensaje);
}else{
String mensaje=" [Atencio] El despacho se ha detenido. "+
" Se despacharon "+new DecimalFormat("#.0#").format(cantidadDespachada)+" galones";
taskOutput.append(mensaje);
llamarDespachador(mensaje);
}
}
}

public void llamarDespachador(String mensaje) {
try {
String hostName="Willy";
Socket socket = new Socket(hostName, 1322);
/** Escritura en el socket*/
PrintWriter out=new PrintWriter(socket.getOutputStream(),true);
out.println(mensaje);
} catch (IOException ex) {
Logger.getLogger(VentanaControlador.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
}

```


ANEXO K
CÓDIGOS EN LENGUAJE DE PROGRAMACION JAVA CORRESPONDIENTES A LA
SIMULACION DE UN EQUIPO MEDIDOR VOLUMETRICO DE PETROLEO

DespachadorServer

```

package com.controla.despachador;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
/**
 *
 * @author Willy
 */
public class DespachadorServer implements Runnable{
    ServerSocket server;
    Socket socket;
    public void init(int port){
        try {
            server=new ServerSocket(port);
        } catch (IOException ex) {
            Logger.getLogger(DespachadorServer.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public void run() {
        try {
            init(1322);
            InetAddress host = InetAddress.getLocalHost();
            System.out.println("[Despachador] Servidor iniciado en el puerto 1322");
            System.out.println("[Despachador] Hostname:"+host.getHostName());
            System.out.println("[Despachador] Host Address:"+host.getHostAddress());
            while(true){
                socket=server.accept();
                accion(socket);
            }
        } catch (IOException ex) {
            Logger.getLogger(DespachadorServer.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public void accion(Socket socket){
        try {
            // aqui implementar la accion que deseas ejecutar
            BufferedReader in = new BufferedReader(new
            InputStreamReader(socket.getInputStream()));
            String textoRemoto=in.readLine();
            String titulo="Mensaje desde el Controlador";
            JOptionPane.showMessageDialog(new JFrame(),textoRemoto, titulo,
            JOptionPane.INFORMATION_MESSAGE);
        } catch (IOException ex) {
            Logger.getLogger(DespachadorServer.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```
}
}
```

IniciarDespachador (Clase principal)

```
package com.controla.despachador;
/**
 *
 * @author Willy
 */
public class IniciarDespachador {
    public static void main(String[] args) {
        VentanaDespachador despachar= new VentanaDespachador();
        DespachadorServer despachador=new DespachadorServer();
        Thread thread=new Thread(despachador,"Despachador");
        thread.start();
    }
}
```

VentanaDespachador

```
package com.controla.despachador;
import java.awt.BorderLayout;
import java.awt.Cursor;
import java.awt.FlowLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.Inet4Address;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
/**
 *
 * @author Willy
 */
public class VentanaDespachador extends JFrame {
    JPanel panel=new JPanel();
    JTextField txtCantidad;
    JLabel labCantidad;
    JButton startButton;
    JTextArea areaMensajes;
    public VentanaDespachador(){
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```

this.setSize(600, 400);
this.setTitle("Despachador");
this.initComponents();
this.setVisible(true);
}
public void initComponents(){
//Panel para los componentes graficos
panel.setLayout(new FlowLayout());
startButton = new JButton("Iniciar");
startButton.setActionCommand("iniciar");
startButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
// realiza la llamada al controlador remoto.
llamarControlador();
//startButton.setEnabled(false);
//setCursor(Cursor.getPredefinedCursor(Cursor.WAIT_CURSOR));
}
});
txtCantidad=new JTextField(12);
labCantidad=new JLabel("Cantidad (galones): ");
panel.add(labCantidad);
panel.add(txtCantidad);
panel.add(startButton);
// Para recibir mensajes
// Area para informacion de avance
areaMensajes = new JTextArea(5, 20);
areaMensajes.setMargin(new Insets(5,5,5,5));
areaMensajes.setEditable(false);
add(new JScrollPane(areaMensajes), BorderLayout.CENTER);
panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
add(panel, BorderLayout.PAGE_START);
}
public void llamarControlador() {
try {
String hostName="Willy";
Socket socket = new Socket(hostName, 1321);
/** Escritura en el socket*/
PrintWriter out=new PrintWriter(socket.getOutputStream(),true);
out.println(txtCantidad.getText());
areaMensajes.append(" [Despachador] "+new java.util.Date()+" : Enviando a despachar,
"+txtCantidad.getText()+" galones.\n");
} catch (IOException ex) {
Logger.getLogger(VentanaDespachador.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
}

```

ANEXO L
IMAGEN REAL DE PLATAFORMA FLOTANTE EN PLENO DESPACHO A
EMBARCACIONES



BIBLIOGRAFÍA

- [1] "Internal Messages for LCP" IDEX Corporation, revision F, 2005, LIQUID CONTROLS
- [2] <http://home.java.net>
- [3] Jonathank Knudsen and Sing Li, "Beginning J2ME from Novice to Professional", third Edition
- [4] <http://es.wikipedia.org/wiki/Electroválvula>
- [5] <http://www.freewave.com>
- [6] Frenzel, A.M.; Carrasco, A.; Monachesi,E.; Chaile, M.G, "Física de las Ondas Radioeléctricas dentro del Estandar IEEE 802.11b", 2010
- [7] <http://www.monografias.com/trabajos/java/java.shtml>
- [8] <http://www.sapiens.itgo.com/documents/doc68.htm>