

ULL

Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología



Sistema para la monitorización y procesamiento de caudales de combustible en buques rápidos.

Trabajo Final de Grado

Grado en Ingeniería Electrónica Industrial y Automática

Tutores

Ing. Imanuel Berdi (Fred Olsen S.A.)

Dr. Alberto Hamilton Castro (Universidad de La Laguna)

Alumno

Jorge Sebastián Visser Ortiz

Septiembre, 2014

D. **Immanuel Berdi** con DNI 541178611 Ingeniero en Electrónica
miembro del Departamento TÉCNICO de la compañía Fred
Olsen S.A.

y

D. **Alberto Hamilton Castro**, con DNI 43773884P profesor Titular de
Universidad adscrito al Departamento de Ingeniería Informática de la Universidad
de La Laguna

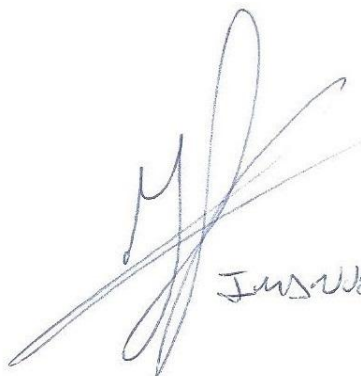
CERTIFICAN

Que la presente memoria titulada:

*“Sistema para la monitorización y
procesamiento de caudales de
combustible en buques rápidos”*

ha sido realizada bajo su dirección por D. Jorge Sebastián Visser Ortiz, con DNI
78518564-Z.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos
oportunos firman la presente en La Laguna a 4 de Septiembre de 2014



IMMANUEL BERDI

Agradecimientos:

A mi familia y a Laura.

A Imanuel Berdi y Alberto Hamilton Castro por tutorizarme en este proyecto.

A la Compañía Fred Olsen Express S.A., por permitirme el desarrollo de este proyecto en sus instalaciones, por su asesoramiento y su inestimable colaboración.

Resumen

Teniendo en cuenta las prioridades estratégicas que actualmente se están fijando en la marina mercante para desarrollar mejores factores de eficiencia y reducir el consumo de combustible, se hace esencial adoptar procedimientos e inversiones que permitan alcanzar este objetivo con el fin de mantener la competitividad en el sector.

En este sentido, en el presente trabajo se propone el diseño de un sistema integral para el control del consumo de combustible en buques rápidos.

Este sistema está diseñado para ser instalado en buques de la naviera Fred Olsen Express S.A. El diseño comprende un primer subsistema formado por caudalímetros másicos de coriolis, instalados en los circuitos de alimentación y retorno de combustible de los motores Cat3618, y un controlador integrado que calcula los parámetros de consumo neto y los envía a través de una red de área local (LAN) mediante el protocolo MODBUS/TCP . El segundo subsistema consiste en una interfaz SCADA, diseñada específicamente para proporcionar al operador un acceso fiable a estos parámetros, e instalado en una estación de trabajo conectada a la misma red.

Abstract

Considering current strategic priorities to develop improved efficiency factors and reducing fuel consumption in the merchant marine, it is essential to invest and adopt procedures to achieve this objective, in order to maintain competitiveness in this sector. Thus, in the present work, the design of a system to fast-ferries fuel consumption monitoring is proposed.

This system is designed to install in Fred Olsen Express S.A. fast ferries. The design comprises a first subsystem formed by coriolis mass flowmeters, installed in the supply and return circuits of Cat3618 engines, and an embedded controller that calculates the net consumption parameters which are sent via a local area network (LAN) using the MODBUS / TCP protocol. The second subsystem consists on a SCADA interface, specifically designed to provide the operator a reliable access to these parameters, and installed on a workstation connected to the same network.

ÍNDICE

Capítulo 1.-

- 1.- Introducción, 15
- 1.1.- Alternativas, 16
- 1.2.- Qué solución se propone, 17

Capítulo 2.-

- 2.- Antecedentes y motivaciones, 19
- 2.1.- El consumo de combustible en el transporte marítimo, 20
 - 2.1.1.- El ciclo operativo del buque, 21
 - 2.1.2.- Perfil operativo, 22
 - 2.1.3.- La resistencia al avance, 23
 - 2.1.4 Estimación del consumo de combustible, 23
 - 2.1.5.- Factores de incertidumbre y el reto de la precisión en la estimación del consumo, 25
- 2.2.- Nueva legislación sobre combustibles en el transporte marítimo., 25
- 2.3.- Impacto de la legislación vigente en el precio del combustible marítimo, 26
- 2.4.- Importancia del control de consumo en el transporte marítimo en la actualidad, 27
- 2.5.- Algunas medidas importantes adoptadas por las principales compañías de transportes a escala global., 27

Capítulo 3.-

- 3.- Ámbito y Alcance del Proyecto, 29
- 3.1.- Ámbito general, 29
- 3.2.- Ámbito Específico, 29

Capítulo 4.-

- 4.- Materiales y Métodos, 30
- 4.1.- Especificaciones técnicas del proyecto, 30
- 4.2.- Características De La Nave, 32
- 4.3.- El motor Caterpillar 3618, 32
- 4.3.- Cálculo de consumo de combustible, 34
- 4.4.- Caudal Másico Vs. Caudal Volumétrico, 36
 - 4.4.1.- Arquitectura de un sistema con medidores volumetricos, 38
 - 4.4.2.- Arquitectura de un sistema con medidores de caudal másico, 40
- 4.5.- Caudalímetro másico kröne modelo optimass 1000, 42
 - 4.5.1.- Precisión Optimass 1000, 43
 - 4.5.2.- Arquitectura Optimass 1000, 43
 - 4.5.3.- Pérdidas de carga, 46
 - 4.5.4.- Dimensiones y peso, 47
 - 4.5.5.- Condiciones de operación, 48

Capítulo 5.-

- 5.- Preprocesado y Acondicionamiento de señales (Optimass-1000), 49
- 5.1.- Acondicionamiento De La Salida Digital Para El Caudalímetro, 50
- 5.2.- Simulaciones Circuito Con LT-Spice, 52

Capítulo 6.-

- 6.- Sistema de adquisición y procesado de datos, 54
- 6.1.- El microcontrolador AVR mega 2560, arquitectura y características, 54

- 6.2.- Arduino AVR mega 2560, 57
- 6.2.1.- Entorno de programación
 - Arduino 1.5.5-R2, 59
- 6.3.- La ethernet shield de Arduino, 60
- 6.4.- Implementación, 62
- 6.4.1.- Entradas, 62
- 6.4.10.- Descripción del código fuente:, 85
- 6.4.2.- Salidas, 63
- 6.4.3.- Cálculos y procedimientos, 63
- 6.4.4.- Configuración de los
 - temporizadores del AVR mega 2560, 64
- 6.4.5.- Limitaciones del sistema
 - Arduino, 69
- 6.4.6.- Programación del controlador en
 - Arduino, 72
- 6.4.7.- Desarrollo de un servidor de
 - comunicación vía Modbus/TCP, 73
 - 6.4.7.1.- El protocolo Modbus, 74
 - 6.4.7.2.- El Protocolo MODBUS
 - TCP/IP, 80
 - 6.4.7.3.-Ventajas del Protocolo
 - MODBUS/TCP, 82
- 6.4.8.- Funciones Básicas de la librería
 - Ethernet, 84
- 6.4.9.- Funciones básicas de la librería
 - 'mudbus.h', 84
- 6.5 Topología física de la red (LAN), 95

Capítulo 7.-

- 7.- Sistema SCADA, 96
- 7.1.- El sistema SCADA IGSS Free 50, 96
- 7.2.- Modos de direccionamiento, 98
- 7.4.- Diseño de un GUI(Graphical User
 - Interface) para la representación de
 - datos en pantalla con
 - SCADA de IGSS., 99

Capítulo 8.-

- 8.- Pruebas y Test, 102
- 8.1.-Monitorización de las comunicaciones
 - mediante WireShark, 102

Capítulo 9.-

- 9.- Discusión, 106
- 9.1.- Resultados, 106
- 9.2.- Líneas abiertas, 106
- 9.3.- Alternativas comerciales, 107

Capítulo 10.-

- 10.- Conclusiones, 109

Chapter 10.-

- 10.- Conclusions, 111

Capítulo 11.-

- 11.- Bibliografía, 113

.....

Anexo 1.-

- Presupuesto, 119

Anexo 2.-

- Código Fuente, 125

I.- Memoria

Capítulo 1.-

INTRODUCCIÓN

Vivimos en una sociedad cambiante y acelerada, en donde muchos de los sistemas adoptados en el pasado se están mostrando manifiestamente caducos. La inversión de las tendencias afecta a todos los elementos de la sociedad, obligándonos a replantearnos los antiguos modelos y adaptándolos a esta nueva época.

El sector de la energía en el transporte marítimo no es una excepción. Los modelos que se han venido adoptado en la industria del transporte por mar se basa en premisas que se ajustan a una realidad que poco a poco se va quedando sin efecto. Y sus consecuencias se dejan entrever en cifras, tal como se puede interpretar del último *Review of maritime Transport, 2013* de la "United Nations Conference on Trade and Development".

A pesar de que alrededor del *80% del comercio mundial, en volumen, y más del 70% en valor, se transporta por mar y transita por puertos de todo el mundo*¹ y del espectacular incremento del comercio marítimo internacional que está tocando máximos históricos, cabe destacar que este auge no ha acrecentado la rentabilidad del sector. También hay que destacar que cada vez hay menos naves operativas y los pequeños armadores tienden a desaparecer. Siendo los grandes armadores los que compensan esa tendencia con un incremento del volumen de carga de sus naves, los barcos son cada vez menos, pero su capacidad es mayor. Esto hace persistir un exceso de oferta que ejerce una fuerte presión a la baja sobre los fletes.

Ante esta situación los transportistas están poniendo en marcha diferentes estrategias, entre ellas, medidas para reducir el consumo de combustible.

Por otro lado, particularmente en Europa se están dando pasos de carácter político para controlar las emisiones de gases contaminantes procedente del transporte marítimo lo que

⁽¹⁾ Datos obtenidos de *Review of maritime Transport, 2013*

obliga a las actuales flotas a aumentar la demanda de combustibles más refinados, lo que junto con la limitada capacidad de producción y refino de estos combustibles, hace prever una escasez de oferta a medio plazo lo incrementará drásticamente los costes de estos combustibles. Esta afectación será particularmente grave en el caso de buques rápidos y ultraligeros que no tienen posibilidad de usar otros tipos de combustibles menos refinados.

Uniéndolo a una demanda cada vez mayor podemos sacar dos conclusiones sencillas:

- ✚ La tendencia de los precios permanecerá al alza.
- ✚ Es evidente que la configuración actual se hace insostenible, y hay que cambiar las pautas de consumo.

1.1.- Alternativas

Al margen de que la realidad nos obliga a explorar nuevas tecnologías como las naves propulsadas con GNL⁽²⁾, combustible más barato y menos contaminante, con el que ya operan algunas navieras en el norte de Europa y Latinoamérica. La evolución natural nos permite vislumbrar que la tendencia actual pasa por una solución de compromiso como aumentar la eficiencia de los recursos disponibles, y por tanto un estricto control del consumo.

Esto no es un secreto, las principales navieras a nivel mundial ya están adoptando medidas más o menos rápidas para adaptarse gradualmente a esa tendencia.

Está claro que, quienes adopten una posición pionera en este sentido conseguirán una enorme ventaja estratégica.

² *Gas Natural Licuado*

1.2.- Qué solución se propone

Se propone desarrollar un sistema que proporciones datos del consumo y rendimiento de los buques, teniendo en cuenta las siguientes especificaciones:

- ✚ Debe ser Versátil: Para poder adaptarse a cualquier nave.
- ✚ Polivalente: Debe adaptarse a todo tipo de combustibles.
- ✚ Debe poder comunicarse con el resto de la instrumentación del buque para lo cual se debe adoptar un protocolo de comunicación.
- ✚ Portátil: fácilmente instalable y transportable.
- ✚ Extremadamente Preciso: Un 3% supone una desviación inaceptable en los caudales de trabajo.
- ✚ Amigable para el operador
- ✚ Accesible: Se ha de poder acceder a los datos con facilidad, para poder procesarlos, copiarlos, graficarlos, etc.
- ✚ Fiable: debe ser un sistema robusto y tolerante a fallos.
- ✚ Económico: debe ser más económico que los sistemas disponibles en el mercado.

Si bien todos los factores anteriormente expuestos son importantes se ha de hacer especial mención a la precisión, dado que una pequeña desviación en el valor instantáneo introduciría un error de toneladas en un cómputo global, lo cual sería inaceptable. Se ha de prestar especial cuidado en la selección de los sensores apropiados así como en la propagación del error.

La principal ventaja competitiva de este proyecto radica en su finalidad, si bien este punto se desarrollara en adelante con más detalle, a grandes rasgos un estudio detallado del

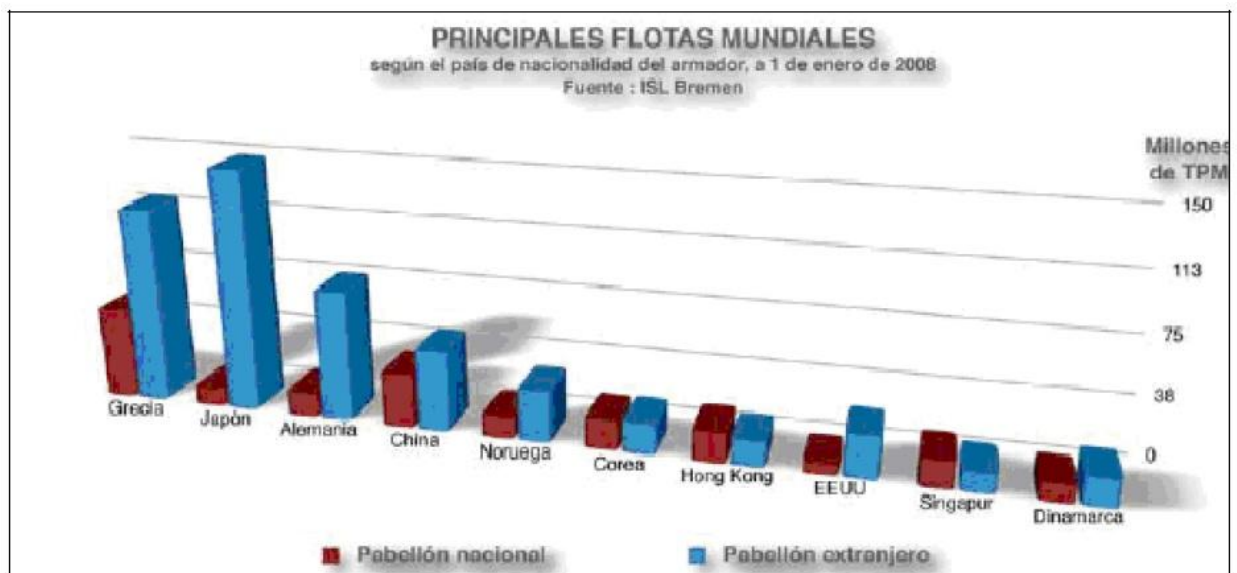
consumo de los buques permitirá estudiar el rendimiento global de estos, las pautas de carga y descarga, revisión de maniobras, etc. Para mejorar la eficiencia y en última instancia, la rentabilidad del transporte.

Capítulo 2.-

ANTECEDENTES Y MOTIVACIONES

Según el *Review of maritime Transport, 2013* elaborado por la UNCTAD³ el comercio marítimo se incrementó un 4,3% en 2012, el informe también refleja que el ritmo de entrega de nuevos buques ha disminuido, sin embargo, la constante incorporación de nuevos buques a un mercado con un exceso de oferta junto con la debilidad de la economía mundial, ejercieron una fuerte presión a la baja sobre los fletes. Los bajos precios de flete registrados en 2012 disminuyeron los ingresos de los transportistas hasta niveles iguales, o incluso inferiores, a los costes operacionales. Los transportistas emplearon varias estrategias para remediar la situación, en particular, adoptando medidas para la reducción del consumo de combustible.

Figura 1. Principales Flotas Mundiales



Fuente: ANAVE

³ "United Nations Conference on Trade and Development".

Los retos más importantes que afronta el sector en la actualidad además de la seguridad, son los costos del aprovisionamiento energético, y los problemas conexos como el cambio climático y la sostenibilidad ambiental.

2.1.- El consumo de combustible en el transporte marítimo

El Consumo de combustible a lo largo del ciclo de operaciones de un buque depende de muchos factores, aunque existe una relación directa entre distancia recorrida y consumo, y entre tonelaje de carga y consumo, no es realista estimar el consumo de combustible sólo mediante estos factores. Existen otros factores muy fundamentales a tener en cuenta como son la velocidad, la climatología, el lastre, los tiempos de espera, etc. Todos estos factores se calculan en general por separado, y se suman a un computo global obteniendo una cifra media sobre Km recorridos y Toneladas de carga.

En general la ruta de navegación a seguir por un buque para ir de un puerto a otro depende de las personas que dirigen el mismo, estando estos obligados a respetar las directrices y acuerdos internacionales, así como las normas de los países cuyas aguas territoriales están atravesando.

Es evidente que, salvo problemas que afecten a la seguridad del buque y la carga, la ruta elegida es aquella que exige un menor consumo y/o tiempo de navegación.

La influencia en el estado de la mar y las condiciones de viento en la navegación del buque, son tan importantes que hacen que compense el navegar por una ruta más larga, a fin de evitar malas condiciones meteorológicas, incrementando los consumos por unidad de distancia en porcentajes muy significativos, además de someter al buque a unos esfuerzos que aumentan los gastos de mantenimiento.

2.1.1.- El ciclo operativo del buque

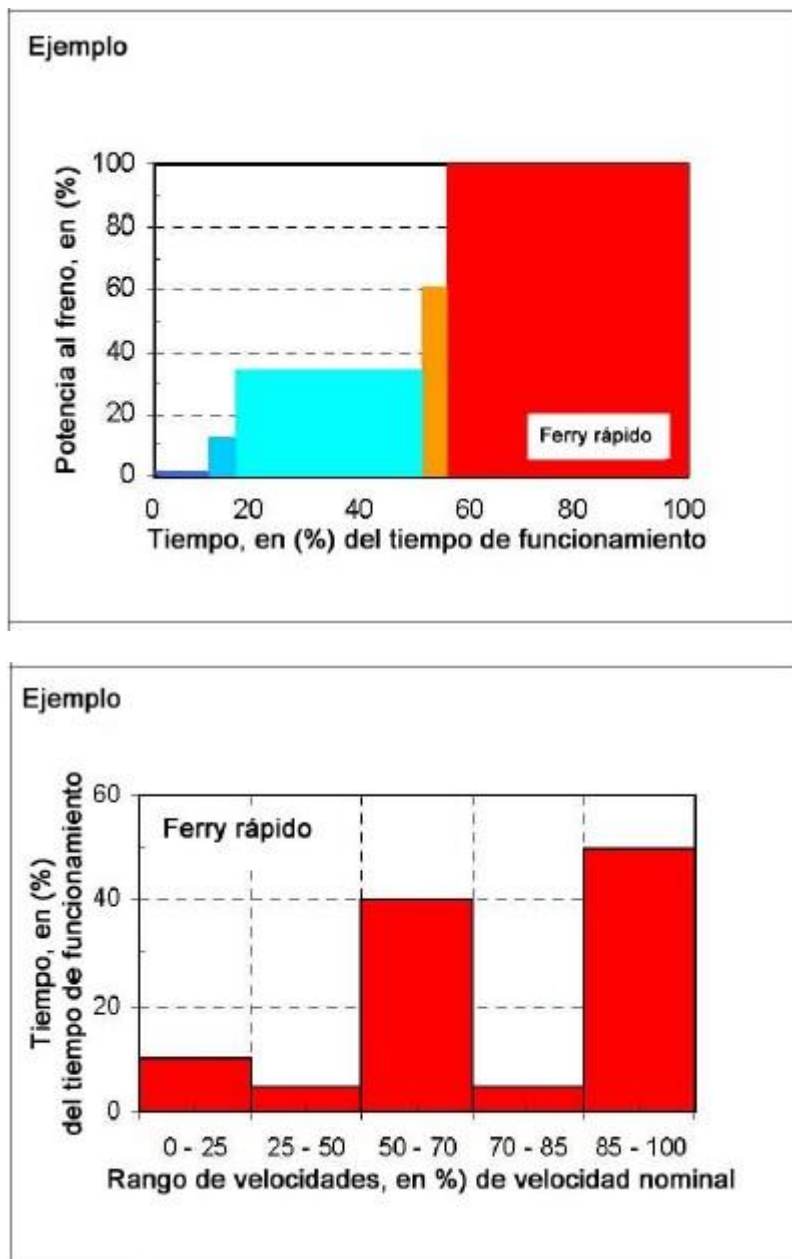
En el servicio operativo del buque se suelen distinguir 4 etapas principales y una complementaria:

- ✚ **Etapas de cruce:** el buque navega, en mar abierto, a la velocidad de servicio.
- ✚ **Etapas de precaución:** El buque está llegando a puerto y reduce su velocidad. también se puede dar este tipo de navegación por estar pasando por zonas de riesgo, o con problemas de navegabilidad.
- ✚ **Etapas de maniobra:** El buque está en bocana de puerto, generalmente con el práctico a bordo y los remolcadores preparados para realizar la maniobra de atraque.
- ✚ **Etapas de Muelle:** El buque está atracado en el muelle pudiendo distinguirse dos sub-etapas:
 - ✚ **Etapas de Carga/Descarga:** El buque está atracado y realizando operaciones de carga y/o descarga, con lo que su consumo es mayor que en la siguiente sub-etapa.
 - ✚ **Etapas de Hottelling:** El buque está atracado sin realizar maniobras de carga y descarga, sin embargo existe un consumo permanente debido a generadores de energía eléctrica, sistemas de refrigeración, etc.
- ✚ **Etapas de Paro y/o Varada:** Además de estas etapas, durante el año el buque está inactivo en ciertos periodos para operaciones de tráfico, bien por falta de fletes, o bien porque se están realizando operaciones de revisión y/o reparaciones.

2.1.2.- Perfil operativo

El tiempo entre la partida de un buque, y su llegada a puerto puede dividirse en varios tiempos parciales en los que se navega con rangos de velocidad constantes. A esta lista de tiempos parciales y rangos de velocidad se la llama perfil operativo.

figura 2. perfil Operativo (fast ferry)



fte: E-navales

2.1.3 .- La resistencia al avance:

La resistencia al avance de un buque a una velocidad determinada se define como el valor de la fuerza que es necesario vencer para desplazar dicho buque en aguas tranquilas, a esa velocidad.

Existen tres tipos de Resistencia al avance:

- ✚ Resistencia hidrodinámica (debida al agua)
- ✚ Resistencia aerodinámica (debida al aire)
- ✚ Resistencia accidental

La potencia necesaria para vencer esta resistencia recibe el nombre de "Potencia de remolque" o "Potencia efectiva", y su expresión es:

$$P_E = R \cdot V$$

siendo **R** y **V** la resistencia al avance y la velocidad de avance respectivamente.

2.1.4 Estimación del consumo de combustible

Con la información disponible puede calcularse el consumo de combustible para una potencia al freno del motor Diesel (P_B) y una velocidad del mismo (n).

Si no se dan las tolerancias en el diagrama de consumo de combustible, se debe añadir a los resultados un margen de un 5%.

$$B = \frac{(P_B \cdot b_e)}{\rho_{fuel}} \left(m^3/h \right)$$

Siendo:

- ✚ **b_e** = Consumo específico de combustible en (kg/kWh)
- ✚ **B** = Consumo de Combustible en (m³/h)
- ✚ **P_b** = Potencia al Freno del motor diesel en (Kw)
- ✚ **ρ_{fuel}** = Densidad del combustible en (Kg/m³)

Los consumidores adicionales, por ejemplo, los grupos generadores deben añadirse para calcular el consumo de combustible total. Si solo se conoce la potencia eléctrica generada en Kw, se puede usar como estimación para el rendimiento del alternador, un valor de, por ejemplo el 95%.

$$B = B_{propulsión} + B_{gensets} + B_{auxiliary} \left(\frac{m^3}{h} \right)$$

Siendo:

- ✚ **B** = Consumo de Combustible en (m³/h)


La ecuación puede ser utilizada para cualquier potencia al freno (PB) y velocidad (n) en el diagrama de funcionamiento. Si el consumo debe ser calculado para períodos de tiempo de un perfil operativo, debe usarse la siguiente ecuación.


$$B = \frac{(P_{B1} \cdot B_{e1} \cdot t_1 + \dots + P_{Bn} \cdot B_{en} \cdot t_n)}{100 \cdot \rho_{fuel}} \left(\frac{m^3}{h} \right)$$

Siendo:

- ✚ **B_e** = Consumo específico de combustible en (kg/kWh)
- ✚ **t₁** = Primer Período de tiempo en un perfil operativo, en (%)
- ✚ **t_n** = Último Período de tiempo en un perfil operativo, en (%)

 **B** = Consumo de Combustible en (m³/h)

 **P_B** = Potencia al Freno del motor diesel en (Kw)

 **ρ_{fuel}** = Densidad del combustible en (Kg/m³)

2.1.5. - Factores de incertidumbre y el reto de la precisión en la estimación del consumo

Todos los factores anteriormente citados son absolutamente necesarios para realizar una buena estimación de cara a las características de diseño de naves, cálculo de rutas, etc. No obstante los elementos de incertidumbre son demasiado importantes para despreciarlos. Por ejemplo, un casco sucio provoca una pérdida hidrodinámica con respecto a un casco limpio; el estado de la mar, las condiciones climáticas, son efectos que afectan notablemente al consumo. Las desviaciones respecto de las estimaciones son críticas. Lo que hace imprescindible obtener datos de consumo en tiempo real y de manera particularizada para cada nave. Esto agiliza la toma de decisiones y el estudio del rendimiento de distintas maniobras lo que se traduce en un ahorro considerable de los costos de combustible.

2.2.- Nueva legislación sobre combustibles en el transporte marítimo.

La Directiva 2012/33/UE del Parlamento Europeo y del Consejo, de 21 de noviembre de 2012 , por la que se modifica la Directiva 1999/32/CE del Consejo en lo relativo al contenido de azufre de los combustibles para uso marítimo se rige por las normas elaboradas en la Organización Marítima Internacional (OMI), y reduce progresivamente el contenido máximo de azufre de los combustibles para uso marítimo del 3,5 % actual al 0,5 % en enero de 2020. En algunos ecosistemas frágiles, tales como el Mar Báltico y el Mar del Norte, incluido el Canal de la Mancha, el contenido máximo de azufre se reducirá al 0,1 % ya en 2015.

2.3.- Impacto de la legislación vigente en el precio del combustible marítimo.

Derivado directamente de las directivas anteriormente descritas, se prevé un aumento drástico de la demanda de combustible diesel más refinado. Solventar tal demanda requeriría grandes inversiones en las viejas instalaciones de refinado que actualmente debido a su escasa capacidad de producción están quedando obsoletas en vez de eso las empresas están optando por la construcción de nuevas mega-refinerías en países emergentes y cerrar sus refinerías en Europa, lo cual agrava el problema. Mientras esta desestabilización entre la capacidad de oferta y la demanda se prolongue en el tiempo se estima que el precio de gasoil refinado aumente significativamente, poniendo en cuestión la rentabilidad de muchas empresas navieras.

Este hecho ha llevado a muchas navieras, sobre todo las que operan en el mar Báltico, a replantearse en sus planes estratégicos este tipo de combustible, adquiriendo flotas de buques impulsados por GNL.

Sin embargo, el uso de gas natural licuado como combustible marino aumentará de forma gradual, no radical. Según prevé la sociedad de clasificación Lloyd's Register en su informe "Global Marine Fuel Trends 2030", se prevé que el GNL supondrá en 2030 un 11% como máximo del conjunto de combustibles para buques. El estudio añade que mientras un 31% de los quimiqueros y petroleros usarán GNL en 2030, sólo lo hará el 5% de los portacontenedores.

Lloyd's Register también considera que, en los próximos 16 años, la cuota de fuel oil pesado en el mercado de bunkering no se reducirá. *"El fuel oil pesado todavía estará muy presente en 2030, representando entre un 47 y un 66% del total de combustibles marinos usados. En 2030, la demanda total de fuel oil pesado será al menos la misma y en algunos escenarios podría aumentar hasta un 23%, respecto de los niveles de 2010. Pero, con la demanda global de combustible marino duplicándose en 2030, otros combustibles experimentarán una mayor tasa de crecimiento para responder a esa demanda."*

2.4.- Importancia del control de consumo en el transporte marítimo en la actualidad

Un término intermedio es redoblar los esfuerzos para mejorar la eficiencia. Una mejora de un 1% en el rendimiento de los grandes buques podría suponer un ahorro de cientos de miles de euros en las cuentas de una empresa, teniendo en cuenta además las tasas por cuotas de emisión de gases de efecto invernadero.

2.5.- Algunas medidas importantes adoptadas por las principales compañías de transportes a escala global.

Los beneficios potenciales de la implementación de este sistema se pueden extrapolar haciendo un ejercicio de Benchmarking, con casos de éxito en la marina mercante. En este sentido, se han investigado algunas medidas importantes adoptadas por las principales compañías de transportes a escala global.

✚ Maersk Ahorra 90 Millones en poco menos de tres años gracias a medidas de control de consumo y eficiencia:

Según el **Maersk Sustainability Report 2013**, la naviera mercante más importante del mundo Maersk que fue pionera en implementar medidas de control de la eficiencia en su flota mercante ha podido implementar medidas de ahorro equivalentes a unos 90 millones de dólares en costos de energía, gracias a la transformación en la tecnología de sus buques que le permiten hacer más eficiente el consumo de combustibles.

Los Indicadores de Rendimiento (KPI, por sus siglas en inglés), han servido como un sistema de medición muy preciso desde 2009, permitiendo establecer los parámetros para alcanzar estándares de rendimiento cada vez mayores. El Indicador de Rendimiento de los buques ha sido ideado con el fin de desarrollar diferentes alternativas que permitan el ahorro en combustible y minimizar el impacto ambiental de éstos.

Durante los primeros tres años de la implementación de las medidas, se economizaron 160 mil toneladas de combustible gracias a la utilización de propulsión eficiente en sus motores.

Luego de observar estos resultados positivos en el proceso de KPI en la flota de Maersk, más de 150 naves han mostrado mejoras desde la segunda mitad del año pasado.

Hapag-Lloyd redujo la velocidad de sus barcos:

Según la plataforma Oceana, la tendencia global de los armadores es reducir la velocidad de sus buques para aumentar la eficiencia y disminuir el consumo. En este sentido la corporación Hapag-Lloyd redujo la velocidad de algunos de sus barcos en tan sólo 5 nudos, un 20%, lo que le ahorró un 50% de gastos de combustible.

Además, asegura que si se limitase la velocidad de los barcos se reducirían las emisiones de gases contaminantes.

Capítulo 3.-

ÁMBITO Y ALCANCE DEL PROYECTO

3.1.- Ámbito general

En un ámbito general la finalidad de este proyecto es el desarrollo de un sistema de control y mejora continua en la eficiencia de los buques rápidos. Implementando procedimientos y agilizando la toma de decisiones en base a datos objetivos de consumo para cada nave. Como se ha podido documentar en casos de éxito con las principales navieras mercantes a escala planetaria, el control del consumo de combustible en la industria naval resulta crítico de cara a mantener la competitividad y viabilidad económica de las navieras. En este sentido un importante esfuerzo de cara a obtener datos a tiempo real acerca del consumo objetivo del buque están más que justificados.

3.2.- Ámbito Específico

En un ámbito más específico, los objetivos de este proyecto se circunscriben al diseño y prototipado de un sistema empotrado para la medición y cotejo en tiempo real de diferentes parámetros de consumo y eficiencia del buque, estos datos se ciernen al caudal másico de combustible de refrigeración, caudal másico de combustible que entra en la cámara de combustión, y el cálculo de consumos acumulado e instantáneo. Estos datos deben ser parametrizados para poder contrastarse frente a otros datos como valores de potencia, aceleración y velocidad del buque; así como otros referentes a la densidad del combustible o el estado operativo de la nave.

Estos datos además deben ser almacenados para posteriores estudios pormenorizados, y deben ser presentados en una interfaz amigable y sencilla de operar.

Capítulo 4.- MATERIALES Y MÉTODOS

4.1.- Especificaciones técnicas del proyecto:

De cara a la implementación de este proyecto debemos tener en cuenta las siguientes especificaciones de diseño, ya que son críticas para un correcto desempeño.

Especificaciones Técnicas Del Proyecto

ERROR:	máx. 2 %
RÉGIMEN DE FUNCIONAMIENTO:	2 ton/hora
FONDO DE ESCALA:	10 ton/hora
PERDIDAS DE CARGA:	máx. - 0.39 bar
MEDIO:	Líquido
TEMPERATURA:	máx. 70.0 c
PRESIÓN:	1 bar
DENSIDAD:	0.89 kg/l operation
VISCOSIDAD:	11.0 mm²/s

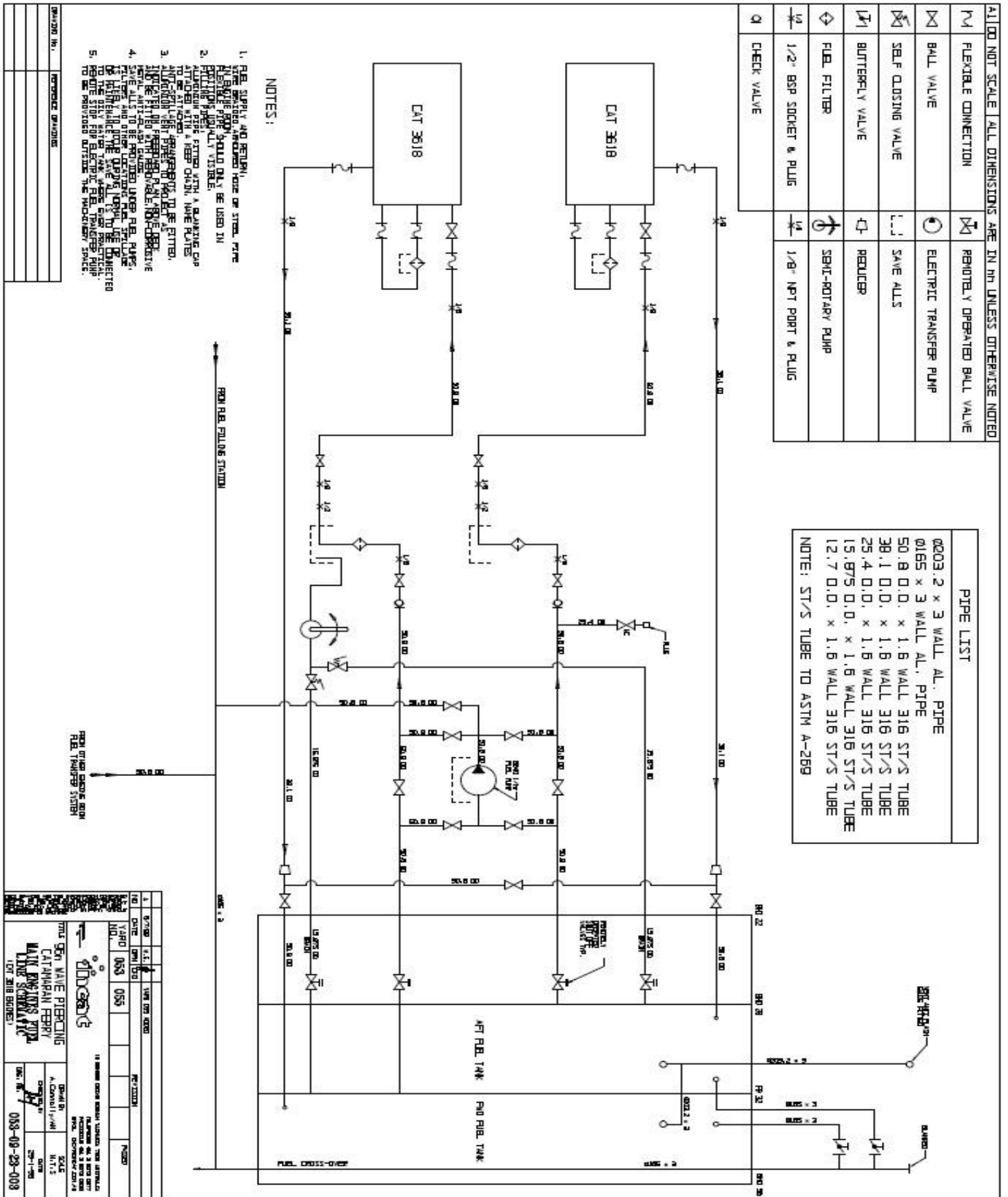


figura 3. Circuito De Alimentación Y Retorno De Combustible En Buques Rápidos

Fte: Fred Olsen S.A.

4.2.- Características de la nave

Las naves donde se instalarán los dispositivos de medición son buques tipo catamarán, dotados de dos cascos de aluminio paralelos unidos por sus cubiertas, certificados como naves de gran velocidad, dedicados al transporte de carga rodada, pasajeros y vehículos en régimen de equipaje. Son propulsados por cuatro motores principales Caterpillar **CAT 3618**, que impulsan cuatro toberas de chorro de agua **KaMeWa 125 S11** y que le proporcionan una velocidad máxima de 37 nudos.

4.3.- El motor Caterpillar 3618

Los Motores CAT3618 son Motores Diesel con una potencia unitaria de 7.200 Kw a 1.050 rpm.



CATERPILLAR® ENGINE SPECIFICATIONS

V-18, 4-Stroke-Cycle-Diesel

Emissions	IMO compliant
Bore — mm (in)	280 (11.0)
Stroke — mm (in)	300 (11.8)
Displacement — L (cu in)	333 (20,320)
Rotation (from flywheel end)	CCW or CW
Compression Ratio	12.6:1
Aspiration	Turbocharged-Aftercooled
Low Idle Speed — rpm	350
Rated Speed — rpm	1050
Average Piston Speed — m/s (ft/s)	10.5 (34.4)
Engine Firing Pressure — bar (psi)	185 (2,680)
BMEP — bar (psi)	24.7 (358)
BSFC — g/bkW-h (lb/hp-h)	201 (.330)

figura 4. Tabla de especificaciones del motor CAT3618

Fte: Spec. Sheet Caterpillar

PERFORMANCE DATA

3618 @ 1050 rpm

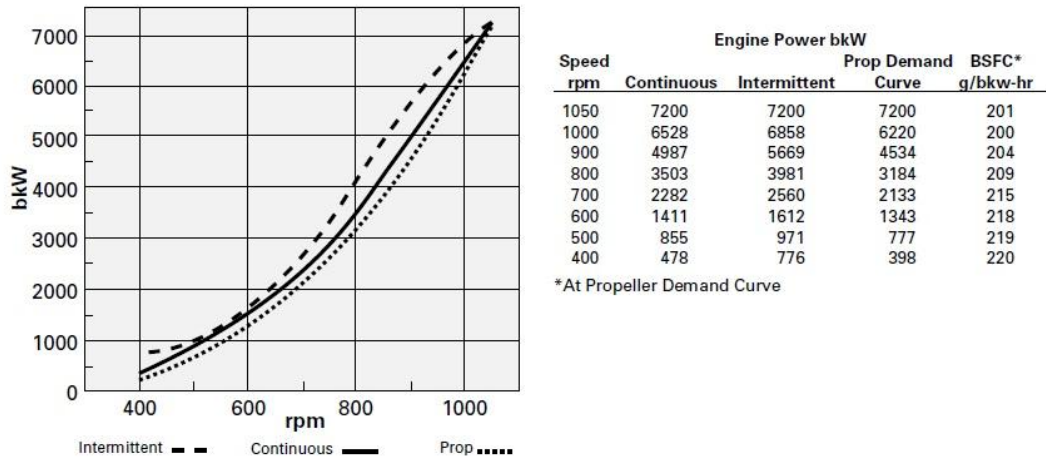
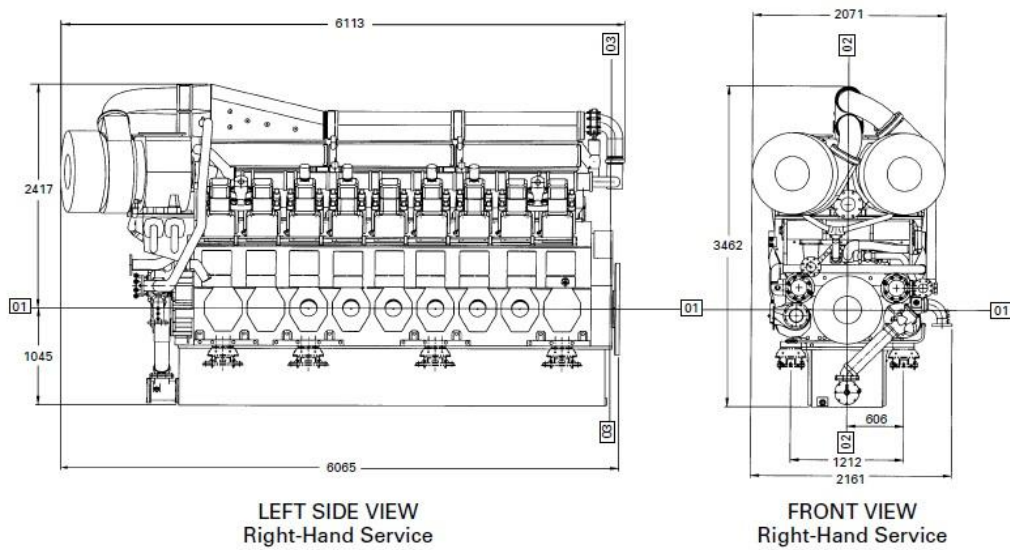


figura 5. Gráfica y tabla de rendimientos
Fte: Spec. Sheet Caterpillar

DIMENSIONS



Engine	Overall Length mm (in)	Overall Width mm (in)	Overall Height mm (in)
3618	6113 (241)	2161 (85)	3462 (136)

figura 6. Dimensiones
Fte: Spec. Sheet Caterpillar

4.3.- Cálculo de consumo de combustible

Como puede verse en el circuito de alimentación y retorno de combustible en buques rápidos (figura 3), No existe un unico flujo de combustible desde los depósitos hacia los motores, el sistema es un poco más complejo. La razón de esto, es que en este tipo de buques existen dos flujos de gasoil que entran hacia el motor, uno es el gasoil que entrara en la camara de combustión, y el otro es el flujo de refrigeración, ya que estos motores se refrigeran con el mismo combustible. El combustible que entra a refrigerar el motor es devuelto a través de un flujo de salida hacia unos intercambiadores de calor y luego nuevamente hacia los depósitos. También hemos de tener en cuenta un par de detalles de caracter constructivo. Para empezar cada buque posee cuatro motores los cuales son alimentados por dos bombas de combustible. la configuración es por tanto dos circuitos independientes formados por una bomba y dos motores, además cada motor posee un filtro de partículas para el combustible.

Otra cosa que hemos de tener en cuenta es que al existir un caudal de refrigeración, este estará a diferente temperatura en el flujo de retorno que en el de alimentación. Para ser más exactos mientras que el flujo de alimentación estará a una temperatura de 20-28 grados celcius, la temperatura del caudal de retorno estará entre 60-68 grados celsius. Esto es de vital importancia, sobre todo si el método de medición de los flujos es volumétrico, ya que la densidad del combustible varía con la temperatura y de no considerarlo introduciría un error inaceptable en el proceso de medición.

Entendemos por tanto que un método correcto de medición del consumo de combustible, y el más objetivo, consiste en hallar la diferencia entre el caudal de alimentación y el caudal de retorno para cada uno de los motores, obteniendo el consumo por cada motor, y sumar los valores resultantes en los 4 motores para obtener un consumo global. Además para aumentar la precisión y disminuir la complejidad del proceso se realizará este proceso mediante 8 caudalímetros, dos por motor, ubicados en cada uno, uno en el circuito de entrada y uno en el circuito de retorno.

Obteniendo de esta forma una dimensión objetiva del consumo neto de combustible se plantean algunos retos:

La precisión:

Dado que vamos a realizar ocho mediciones diferentes y además debemos operar con esos valores debemos asegurarnos que a pesar de la propagación del error, el resultado sigue otorgando un altísimo grado de precisión. Esto no es fácil con la mayoría de caudalímetros disponibles en el mercado.

Pérdidas de carga:

El caudalímetro, como instrumento de medición, debe formar parte del circuito de combustible, y por tanto produce unas pérdidas de carga. Estas pérdidas de carga deben ser tenidas en cuenta, puesto que deben estar dentro de los rangos admitidos por el circuito de alimentación del buque.

Temperatura del fluido:

Como se explicó anteriormente se han de tener en cuenta los cambios físicos producidos en el fluido debido al cambio de temperatura, en especial los cambios de volumen y viscosidad.

4.4.- Caudal Másico Vs. Caudal Volumétrico

Teniendo en cuenta la oferta de caudalímetros disponible en el mercado podemos dividirlos en dos grandes tipos:

- ✚ Caudalímetros de Caudal Másico

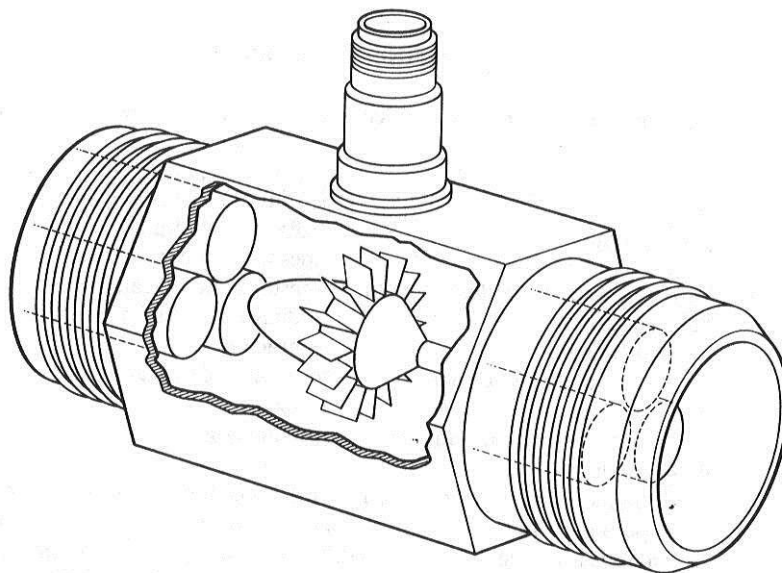
- ✚ Caudalímetros de Caudal Volumétrico (de rodete)

A continuación se elaborará una breve comparativa funcional de las dos alternativas. En dicha comparativa se establece un ratio equivalente a 5/10 al valor umbral referente a los parámetros más importantes de cada una, como son precisión Máxima y Mínima (% de error relativo), pérdidas de carga (bar), coste (€), etc. Los ratios representan la distancia marginal con respecto a dicho umbral.

Figura 7. Comparativa Funcional Caudalímetros

Tipo Caudalímetro	Presión Máx (1-10)	Presión min (1-10)	Pérdidas de Carga (1-10)	Principio de Funcionamiento	Coste Máx. (1-10)	Coste Min. (1-10)	Más Utilizados (1-10)	Menor Complejidad (1-10)	General
másico	9	7	8	Ultrasónicos, Coriolis	8	8	6	9	7.85
volumétrico	6	3	5	Rodete	5	6	8	6	5.57

De esta breve comparativa se puede sacar en claro, primero que no existe una gran diferencia de costes entre una tecnología y otra. Los costes Máximos en el caso de caudalímetros másicos, se corresponden a caudalímetros ultrasónicos, en nuestro caso no nos interesan ya que las pérdidas de carga superan los márgenes de funcionamiento. En el caso de los volumétricos, son todos de rodete. Sin embargo poseen diferencias sustanciales en cuanto a precisión y funcionalidad. En general los caudalímetros de rodete están muy limitados en cuanto a precisión, ya que incluso los de mejor comportamiento presentan un error relativo en torno a un 2 % , esto nos deja sin margen de error respecto de las especificaciones iniciales del proyecto, lo que teniendo en cuenta el aumento de complejidad practicamente descarta de oficio esta opción.



Medidor mecánico de gasto del tipo turbina (cortesía de Potter Aeronautical Corp., Union, New Jersey).

figura 8

4.4.1.- Arquitectura de un sistema con medidores volumetricos

Para implementar el sistema con caudalímetros de tipo volumétrico, se ha de tener en cuenta que trabajamos con dos caudales por cada motor, uno de alimentación y uno de refrigeración. el caudal de refrigeración se devuelve a los tanques y por tanto se ha de restar

del caudal de alimentación inicialmente medido. Sin embargo, teniendo en cuenta que el caudal de retorno se encuentra a unos 30-50 °C por encima del caudal de alimentación hay que hacer una compensación por temperatura con respecto al caudal volumétrico de entrada. Esta compensación se haría mediante el coeficiente de expansión térmica del gasoil.

Coeficiente de expansión térmica del
combustible Diesel:

*0.00083 m³/°C (per degree
Celsius)*

Fte: Diesel Fuel Technical Review(Chevron,2007)

Sin embargo, esta temperatura de retorno no es estacionaria, depende de las condiciones ambientales y del estado operativo del buque, entre otros factores, luego es absolutamente necesario medir la temperatura a tiempo real mediante un sensor de temperatura con el objeto de realizar una compensación correcta. Este aumento de la complejidad y el coste de implementación deben ser tenidos en cuenta de cara a la elección de este tipo de configuración. Los caudalímetros de rodete de gama más alta son de la empresa Contoil e introducen un error relativo mínimo del 1% sobre el caudal a medir y puede detectar retrocesos.

4.4.2.- Arquitectura de un sistema con medidores de caudal másico



figura 9. Caudalímetro másico Optimass 1000
(Fte. Kröhne Ibérica)

En el caso de un caudalímetro másico de coriolis la complejidad de la arquitectura se reduce sustancialmente, dado que estamos trabajando directamente con caudales másicos no es necesario colocar sensores de temperatura para compensar en el caudal de retorno, esto supone una ventaja tanto en el coste, como en la complejidad y de cara al acarreo de errores. el error relativo introducido por estos caudalímetros en la medición oscila entre el 1% y el 0,15 % en los modelos de gama más alta. lo cual aumenta nuestro margen de error y cumple con las especificaciones iniciales del proyecto. Los caudalímetros contemplados para este proyecto corresponden a los modelos MicroMotion F300 de EMERSON y el OPTIMASS Plus 1000 de la marca Kröhne. Ambos Poseen la posibilidad de detectar caudales retroceso y poseen tanto salidas analógicas como digitales. El coste es algo superior a los caudalímetros de rodete.

Con todos estos datos hemos elaborado una comparativa funcional más extensa con los modelos de caudalímetros existentes en el mercado.

figura 9. Comparativa Funcional Caudalímetros Presentes en el Mercado

Modelo	Tipo	Princ de Func.	Origen	Precision	Perd. De carga	Coste	Complej.	Docum. Téc.	Func.	Robustez	Adap. de Señal	Total
Contoll VZF 50	Volum.	Rodete	suiza	7	6	8	6	8	6	7	6	6,75
Emerson Micromotion F300	Másico	Coriolis	EE.UU.	7	8	7	8	8	9	8	9	8
Kröne Optimass 1000	Másico	Coriolis	Alemania	9	8	8	8	8	8	8	8	8,125

4.5.- Caudalímetro másico kröne modelo optimass 1000

Para la ejecución de este proyecto, y teniendo en cuenta la comparativa anterior se ha optado por el caudalímetro Másico de Corioliss Marca Kröne modelo Optimass 1000.



OPTIMASS 1000 Hoja de datos técnica

Sensor para caudal másico

- La mejor elección para aplicaciones universales
- La mejor relación precio-rendimiento
- Una amplia gama de opciones disponibles sin restricciones

La documentación sólo está completa cuando se usa junto con la documentación relevante del convertidor.

© KRÖHNE 06/2011 - 4001381001 - TD OPTIMASS 1000 R05 es

KRÖHNE

figura 10. Caudalímetro Optimass 1000 de Kröhne

4.5.1.- Precisión Optimass 1000

Masa	
Líquido	±0,15% de rango del caudal real de medición + estabilidad de cero
Gas	±0,5% de rango del caudal real de medición + estabilidad de cero
Repetibilidad	Mejor que 0,05% más estabilidad de cero (incluye los efectos combinados de repetibilidad, linealidad e histéresis)
Estabilidad de cero	
Acero inoxidable	±0,01% de rango de caudal máximo con el tamaño de sensor correspondiente
Condiciones de referencia	
Producto	Agua
Temperatura	20 °C / 68 °F
Presión de funcionamiento	1 barg / 14,5 psig
Efecto en el punto cero del sensor provocado por un cambio en la temperatura de proceso	
Acero inoxidable	0,001% por 1 °C / 0,00055% por 1 °F
Efecto en el punto cero del sensor provocado por un cambio en la presión de proceso	
Acero inoxidable	0,00012% del rango de caudal máximo por 1 bar _{rel.} / 0,0000083% del rango de caudal máx. por 1 psig
Densidad	
Rango de medición	400-2500 kg/m ³ / 25-155 lbs/ft ³
Precisión	±2 kg/m ³ / ±0,13 lbs/ft ³ (S15: ±5 kg/m ³ / ±0,33 lbs/ft ³)

figura 11. Tabla Precisión Optimass 1000
fte: Hoja de Datos Técnica(kröhne)

4.5.2.- Arquitectura Optimass 1000

En caudalímetro másico de tubo doble Coriolis Optimass 1000 está formado por dos tubos de medida 1, una bobina conductora 2, y dos sensores (3 y 4) que están colocados a ambos lados de la bobina conductora.

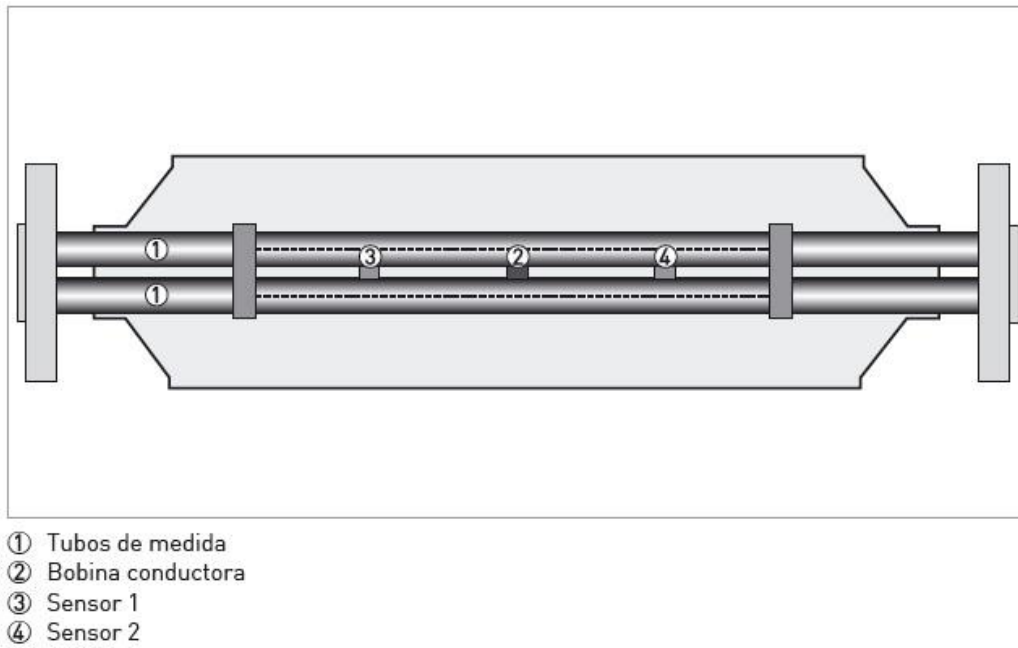


figura 12. Medidor estático no excitado y sin caudal
 fte: Hoja de Datos Técnica(kröhne)

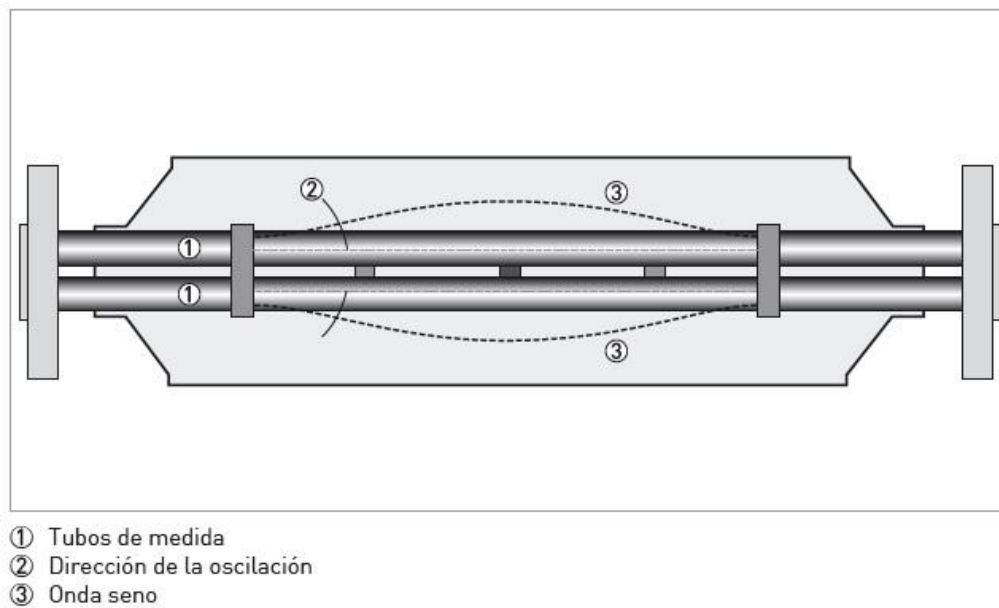


figura 13. Medidor excitado
 fte: Hoja de Datos Técnica(kröhne)

Cuando el medidor está excitado, la bobina conductora hace vibrar los tubos de medida haciendo que oscilen y produzcan una onda seno 3. La onda seno está monitorizada por los dos sensores.

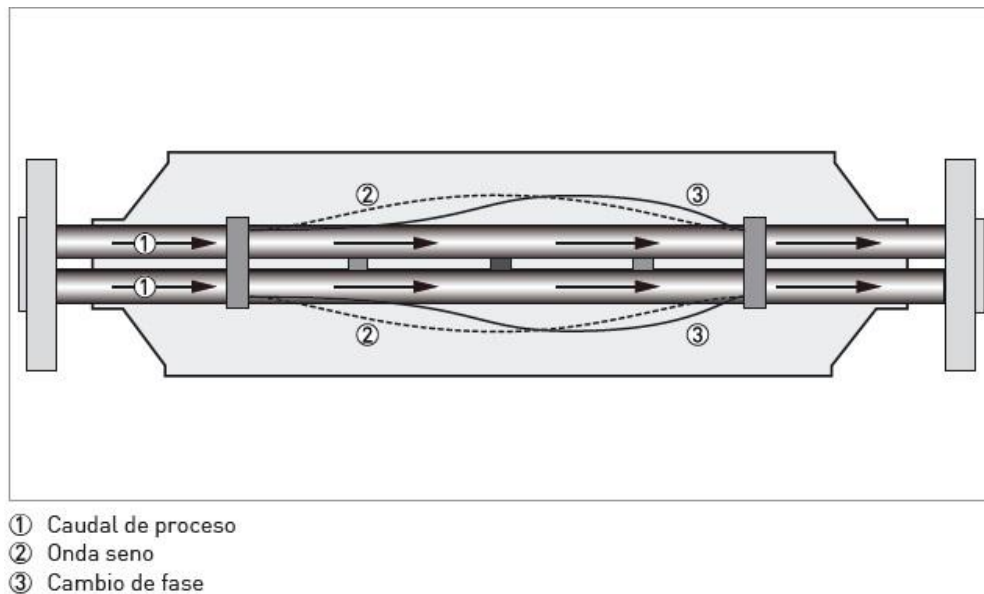


figura 14. Medidor excitado con caudal en proceso
fte: Hoja de Datos Técnica(kröhne)

Cuando un fluido o un gas pasa a través del tubo, el efecto Coriolis provoca un cambio de fase en la onda seno que es detectada por los dos sensores. Este cambio de fase es directamente proporcional al caudal másico. La medida de la densidad se realiza mediante la evaluación de la frecuencia de vibración.

4.5.3.- Pérdidas de carga

Como se puede observar en la gráfica siguiente las pérdidas de carga dentro del rango de aplicación están muy por debajo del máximo (0,39 bar) estipulado por las especificaciones del proyecto.

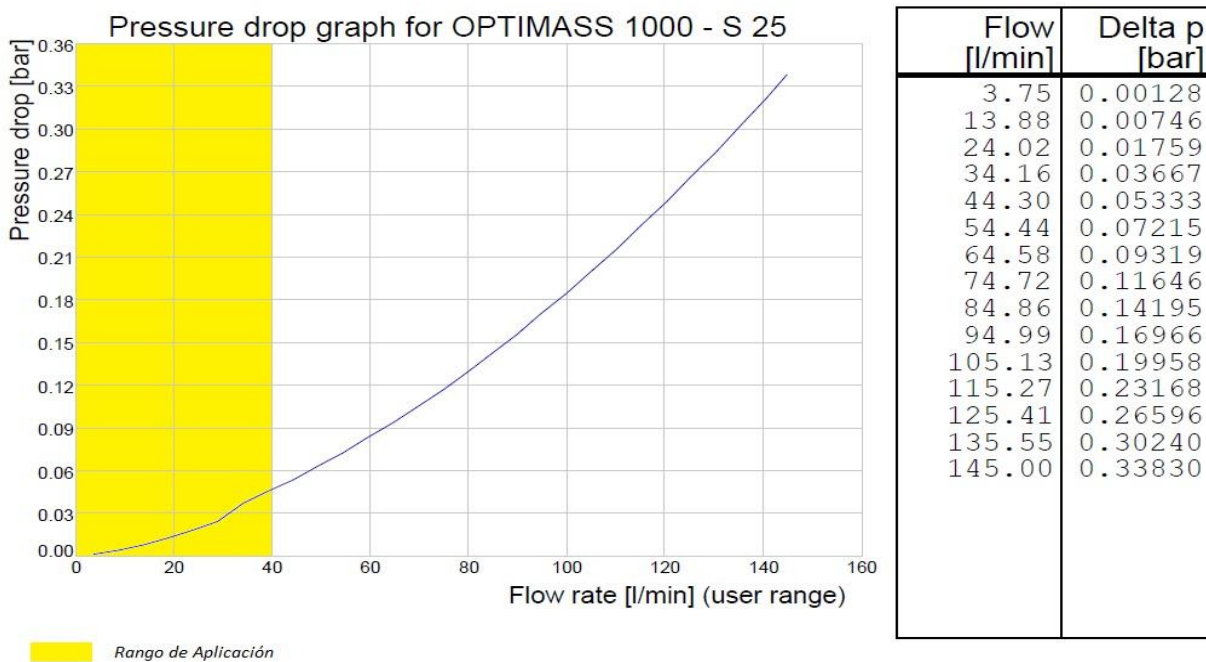
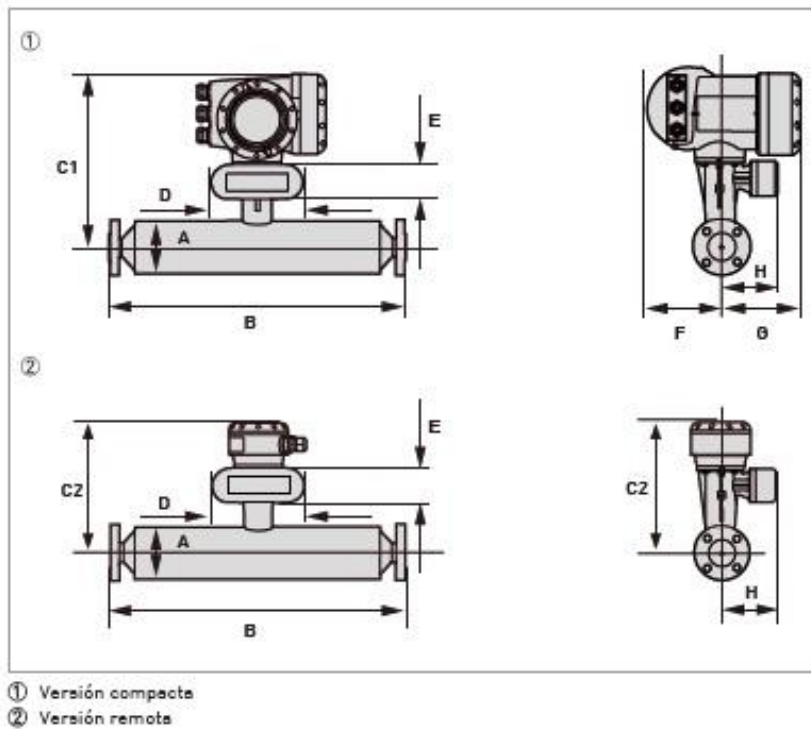


figura 15. Pérdidas de Carga Optimass 1000
(fte: Kröhne Ibérica)

4.5.4.- Dimensiones y peso



Pesos del medidor (todas las bridas)

	Peso [kg]			
	S15	S25	S40	S50
Aluminio (compacto)	13,5	16,5	29,5	57,5
Acero inoxidable (compacto)	18,8	21,8	34,8	62,8
Aluminio (remoto)	11,5	14,5	25,5	51,5
Acero inoxidable (remoto)	12,4	15,4	26,4	52,4

	Dimensiones [mm]			
	S15	S25	S40	S50
A	101,6	114,3	168,3	219,1
C1 (compacto)	311	317	344	370
C2 (remoto)	231	237	264	290
D	160			
E	60			
F	123,5			
G	137			
H	98,5			

figura 16. Esquema Dimensiones y Pesos
(Fte: Kröhne Ibérica)

4.5.5.- Condiciones de operación

Condiciones de operación	
Rangos de caudal máximo	
S15	6500 kg/h / 240 lbs/min
S25	27 000 kg/h / 990 lbs/min
S40	80 000 kg/h / 2935 lbs/min
S50	170 000 kg/h / 6235 lbs/min
Temperatura ambiente	
Versión compacta con convertidor de aluminio	-40+60 °C / -40+140 °F Rango de temperaturas ampliado: +65 °C / +149 °F para algunas opciones de E/S. Para más información, póngase en contacto con el fabricante.
Versión compacta con convertidor de acero inoxidable	-40+55 °C / -40+130 °F
Versiones remotas	-40+65 °C / -40+149 °F
Temperatura de proceso	
Conexión embreadada	-40+130 °C / -40+266 °F
Conexión higiénica	-40+130 °C / -40+266 °F
Presión nominal a 20 °C/68 °F	
Tubo de medida	
Acero inoxidable	-1-100 barg / -14,5-1450 psig
Cilindro exterior	
Sin homologación PED/CRN	Presión de rotura típica > 100 barg/1450 psig a 20 °C
Contención secundaria con homologación PED/CRN	-1-63 barg / -14,5-910 psig
Contención secundaria con homologación PED	-1-100 barg / -14,5-1450 psig
Propiedades de fluido	
Condición física admitida	Líquidos, gases y lodos
Contenido en gases admitido (volumen)	Para más información, póngase en contacto con el fabricante
Contenido en sólidos admitido (volumen)	Para más información, póngase en contacto con el fabricante
Grado de protección (según EN 60529)	IP67/NEMA 4X

figura 17. Condiciones de Operación Optimass 1000

Fte: Guía Técnica Kröhne

Capítulo 5.-

PREPROCESADO Y ACONDICIONAMIENTO DE SEÑALES (CAUDALÍMETRO OPTIMASS 1000)

El Caudalímetro Optimass 1000 posee un sistema de acondicionamiento de señal incorporado con cuatro funciones de salida disponibles:

- ✚ Emisión de pulsos de volumen programables para totalizadores externos
- ✚ Lazo de corriente 4-20 mA correspondiente al caudal
- ✚ Salida de frecuencia 0-100 KHz correspondiente al caudal
- ✚ Rele limitador programable para caudales superior e inferior(Alarmas)

Excepto para el lazo de corriente, cualquier pareja de las restantes 3 funciones se pueden utilizar simultáneamente. Esto nos permite dos modelos de conexionado:

Opción 1

- ✚ Una salida Analógica (4-20mA)
- ✚ Una Salida Dígital
Parametrizable
- ✚ Alarmas (Retroseso, Máx, Mín)

Opción 2

- ✚ No Salidas Analógicas
- ✚ Alarmas (Retroseso, Máx, Mín)
- ✚ Dos Salidas digitales
Parametrizables
Simultaneamente

La alarma de retroseso tiene lugar cuando el caudal fluye en sentido inverso al estipulado, dicho caudalímetro no posee la capacidad de restar estos retrosesos, situación que tendremos que subsanar desde nuestro controlador.

Como el proyecto contempla un procesador digital para recabar y procesar los datos escogemos el modelo de dos salidas digitales parametrizables, además la salida dos no utilizada en este proyecto se programará para transmitir datos de la densidad del fluido.

Esto nos permite disponer de este dato en algún proyecto posterior.

5.1.- Acondicionamiento De La Salida Digital Para El Caudalímetro**Optimass 1000**

Siguiendo el estandar industrial las salidas digitales del caudalímetro Optimass 1000 son señales de 24V, sin embargo dado que nuestro sistema se basa en lógica TTL, se ha de adaptar la señal a la entrada de nuestro controlador.

Para este propósito, comprobado que la entrada a nuestro controlador es de alta impedancia, se propone un circuito partidor de tensión estándar ya que es el que introduce menos limitaciones en cuanto a ancho de banda.

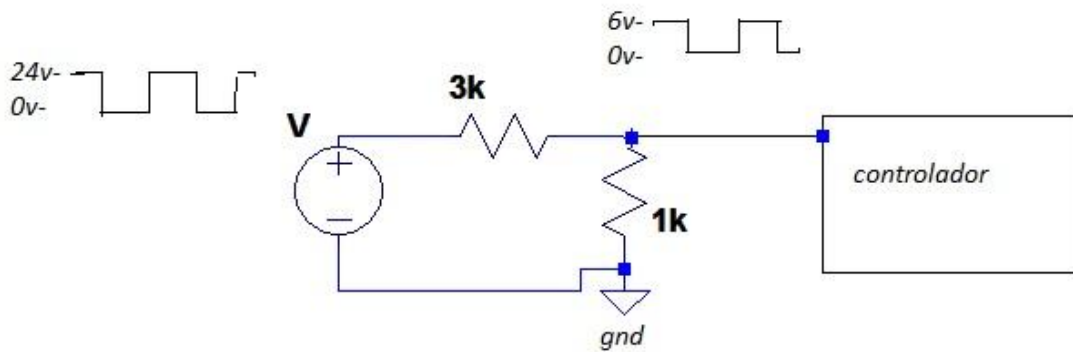
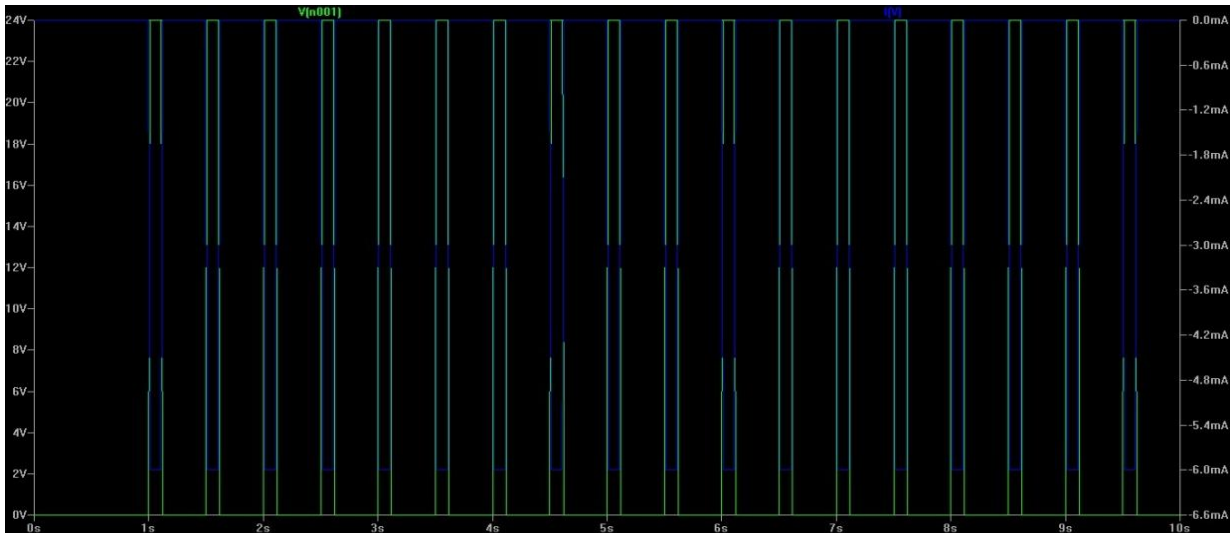


figura 18. circuito de acondicionamiento de señal

Para nuestro proyecto, nos interesa que los caudalímetros funcionen en modo totalizador, asignando un pulso a una cantidad de combustible determinada que atraviesa el sensor. Esta configuración es la más conveniente porque nos interesa medir con la máxima exactitud la cantidad de combustible que ha pasado por el circuito, y no tanto el caudal instantáneo en sí.

5.2.- Simulaciones Circuito Con LT-Spice

Entrada



Salida

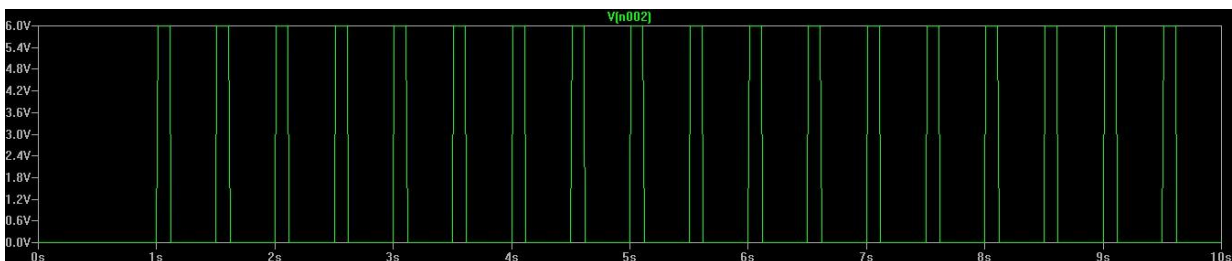


figura 19. resultados de simulación de circuito con LT-Spice

A la salida del Sensor, el ancho del pulso es programable y se puede ajustar a nuestras necesidades, nos interesa que el tiempo de señal sea pequeño ya que vamos a trabajar con frecuencias altas, pero estable para no introducir señales espurias.

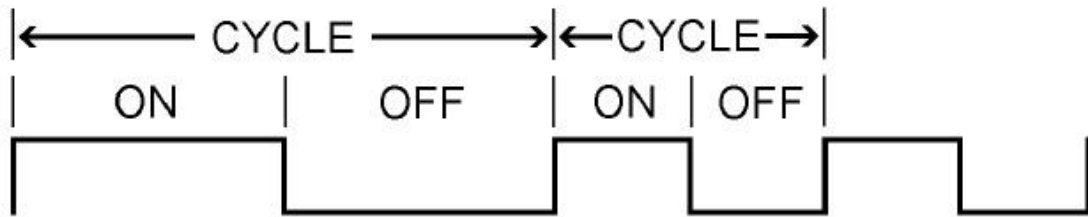


figura 20. Regulación Del Ancho De Pulso
(fte: Guía técnica de configuración Optimass 1000)

Siguiendo las instrucciones del fabricante, para una frecuencia máxima de 100 KHz definiremos un ancho de pulso que sea un 50% del tiempo de ciclo, es decir, 5 μ s.

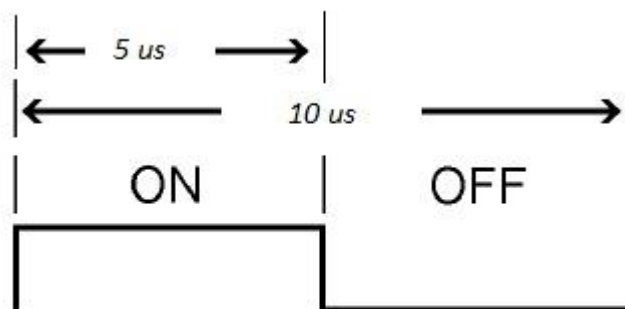


figura 21. Ancho de pulso definido para nuestra aplicación
(fte: kröhne ibérica)

Capítulo 6.-

SISTEMA DE ADQUISICIÓN Y PROCESADO DE DATOS

En este capítulo trataremos el diseño e implementación del controlador digital para el procesado de los datos procedentes de los caudalímetros de entrada y salida de cada motor. El objetivo es establecer a la salida del controlador una comunicación vía MODBUS/TCP para la transmisión del consumo acumulado por motor, el consumo instantáneo, caudales de entrada y salida, así como otros parámetros de control.

6.1.- El microcontrolador AVR mega 2560, arquitectura y características



figura 22. Microcontrolador AVR-Mega 2560

fte: Atmel Corporation

El **AVR Mega 2560, de Atmel** es un micro-controlador de baja potencia y altas prestaciones basado en RISC. Posee un Microprocesador de 8 bits a 16 MHz que combina con una memoria ISP flash de 256KB, 8KB de SRAM, 4 KB de EEPROM, 86 líneas I/O (Entrada/Salida) de propósito general, 32 registros de trabajo de propósito general, contador de tiempo real, seis temporizadores/contadores con posibilidad de trabajar en modo de comparación, PWM, 4 UARTs, interfaz serie de 2 hilos, convertidor A / D de 10 bits de 16 canales, y una interfaz JTAG para depuración on-chip.

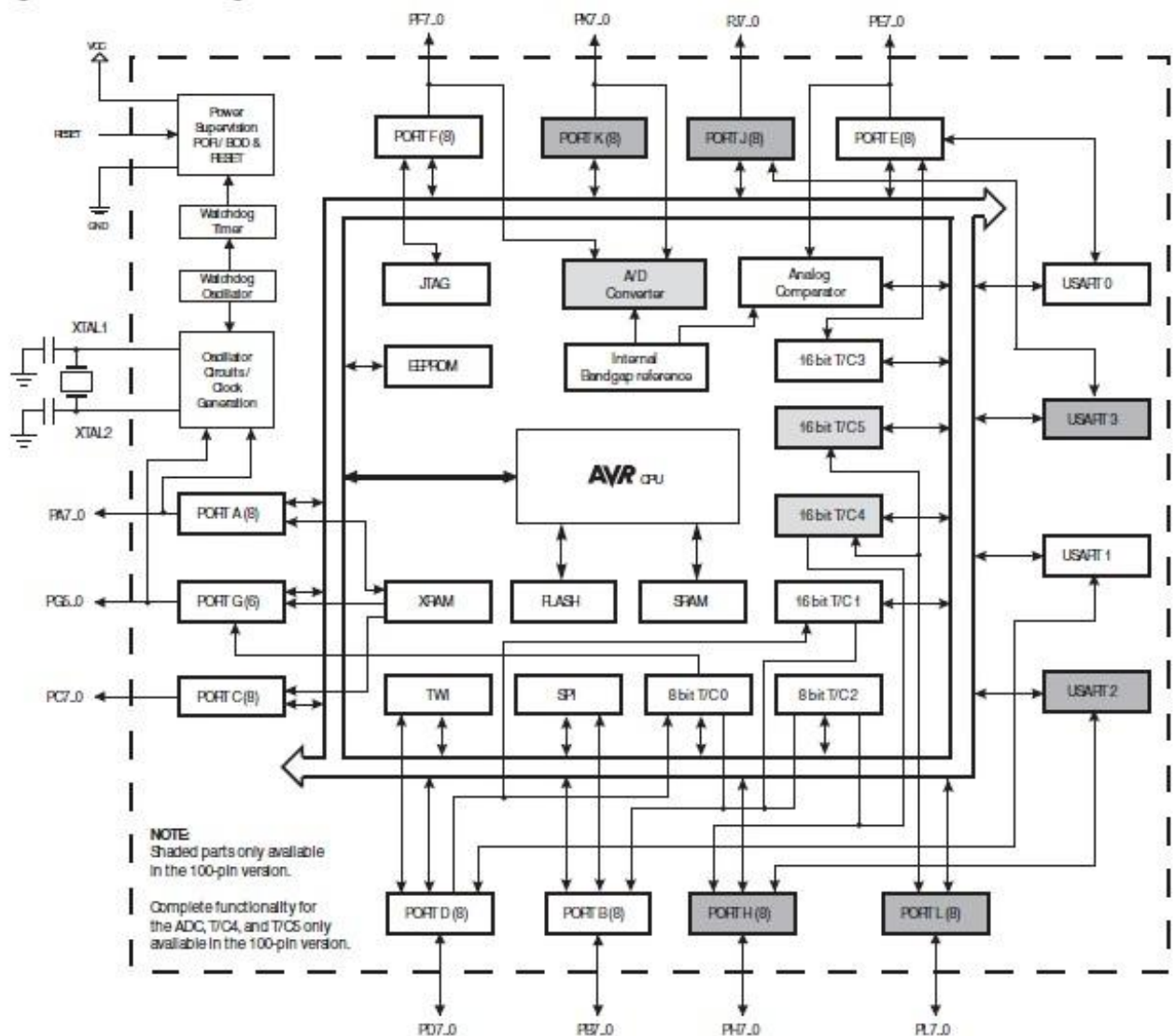


figura 23. Diagrama de Bloques AVR2560
fte: Atmel Corporation

El dispositivo logra un rendimiento de 16 MIPS a 16 MHz y funciona entre 4.5 a 5.5 voltios.

Mediante la ejecución de instrucciones potentes en un solo ciclo de reloj, el dispositivo alcanza un rendimiento cercano a 1 MIPS por MHz, equilibrando el consumo de energía y velocidad de procesamiento.

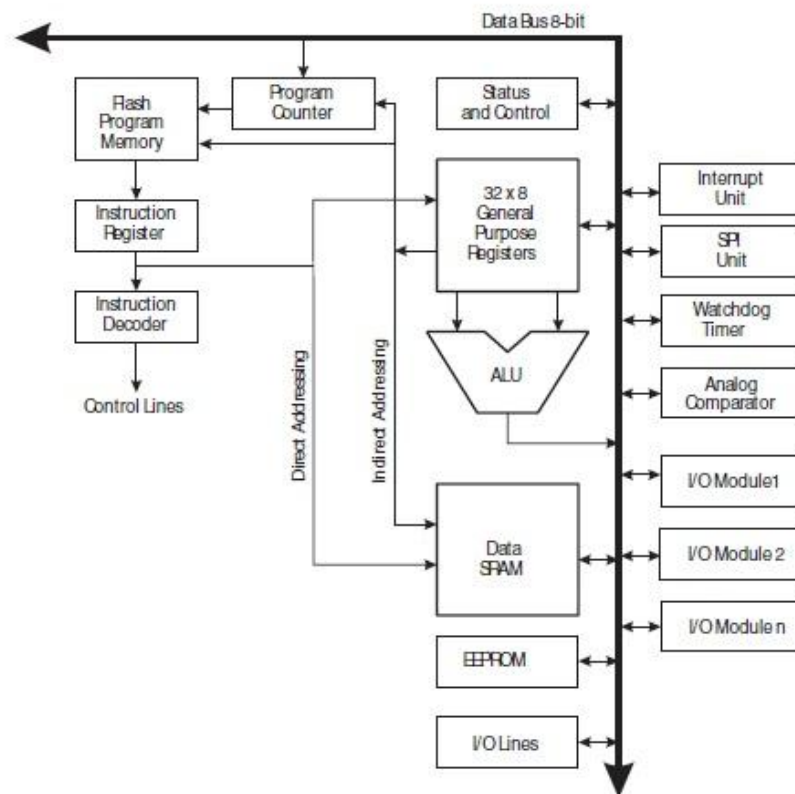


figura 24. CPU Core Harvard Architecture - avr 2560
fte. Atmel Corporation

6.2.- Arduino AVR mega 2560

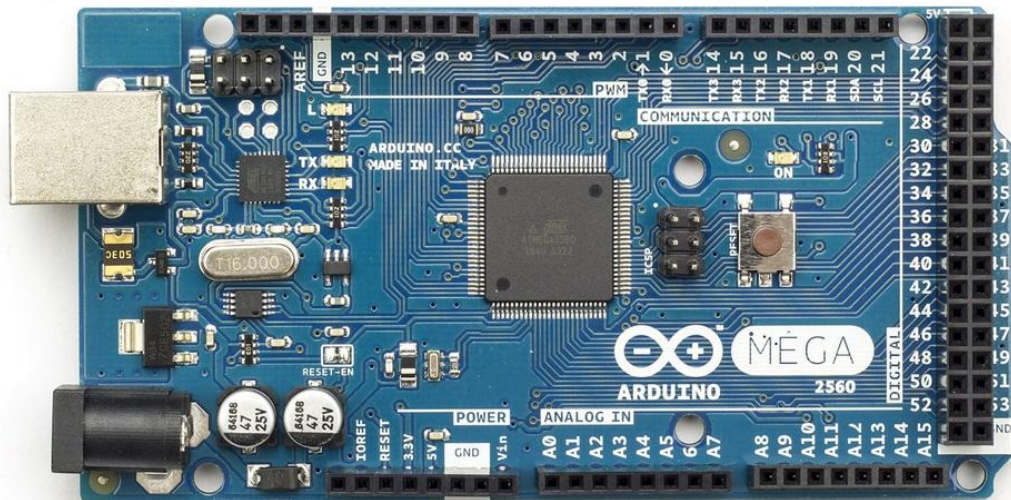


figura 25. Arduino Mega 2560
(Fte. Arduino.cc)

El Arduino Mega 2560 es una placa electrónica basada en el microprocesador Avrmega2560 diseñada para el control de procesos. Es un hardware de propósito general cuya circuitería simplifica el conexionado y el acceso a las funcionalidades del microcontrolador. Además de proporcionar al circuito de robustez estructural, cuenta con sistemas adicionales de protección contra sobretensiones y sobrecargas, Posee un regulador de voltaje, y resistencias de carga en todas sus líneas de entrada-Salida. Cuenta con 54 pines digitales de entrada / salida ,de los cuales 15 se pueden utilizar como salidas PWM, 16 entradas analógicas, acceso a las 4 UARTs (puertas seriales), una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio.

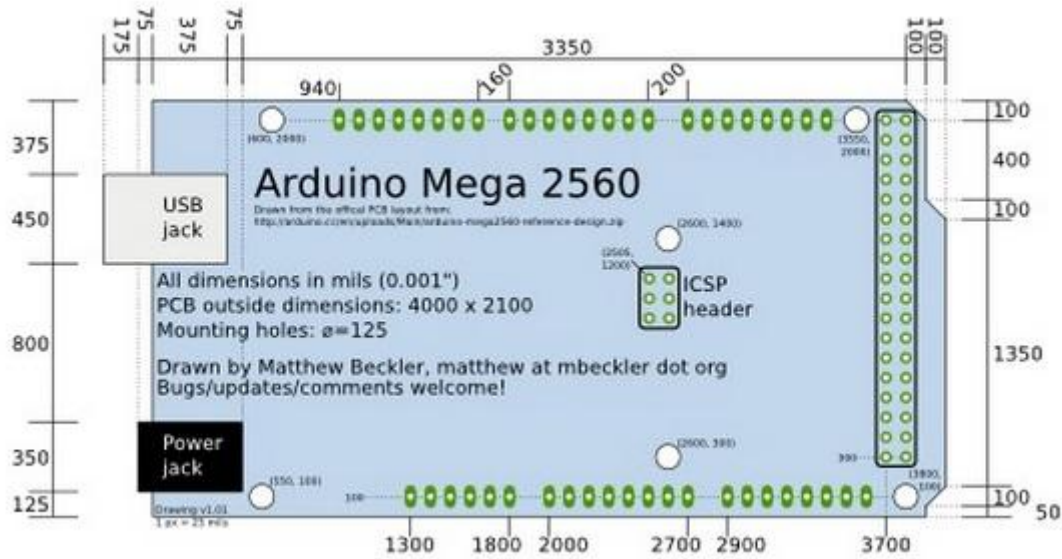


figura 26. Esquema Posterior Arduino Mega 2560
Fte: Makezine.Com

Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o a un adaptador o batería DC para empezar a funcionar. Además posee Indicadores de estado, luminosos para la monitorización de la actividad. Es uno de los más utilizados de los llamados hardware abiertos, y posee abundante documentación debido a su alcance global.

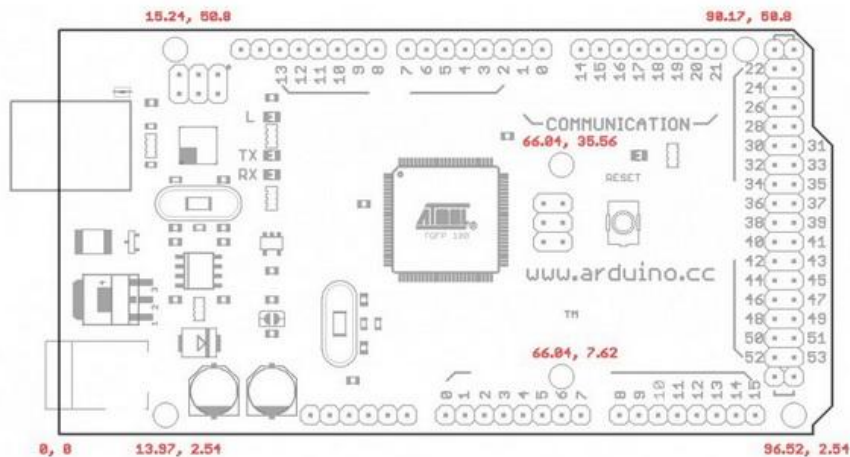


figura 27. Esquema anterior Arduino Mega 2560
Fte: hwkitchen.com

6.2.1.- Entorno de programación Arduino 1.5.5-R2

Además de su capa física, la Arduino proporciona un entorno de programación amigable y sencillo, así como potentes librerías, con un alto nivel de depuración.

El entorno de programación Arduino 1.5.5-r2 permite revisar y compilar el código fuente, y descargar automáticamente a través de un puerto serie el archivo de compilación en la memoria flash de la Arduino. Además el propio entorno posee potentes herramientas de depuración como el "serial monitor", que permite comunicarse con la Arduino mediante el puerto serie durante la ejecución del programa.

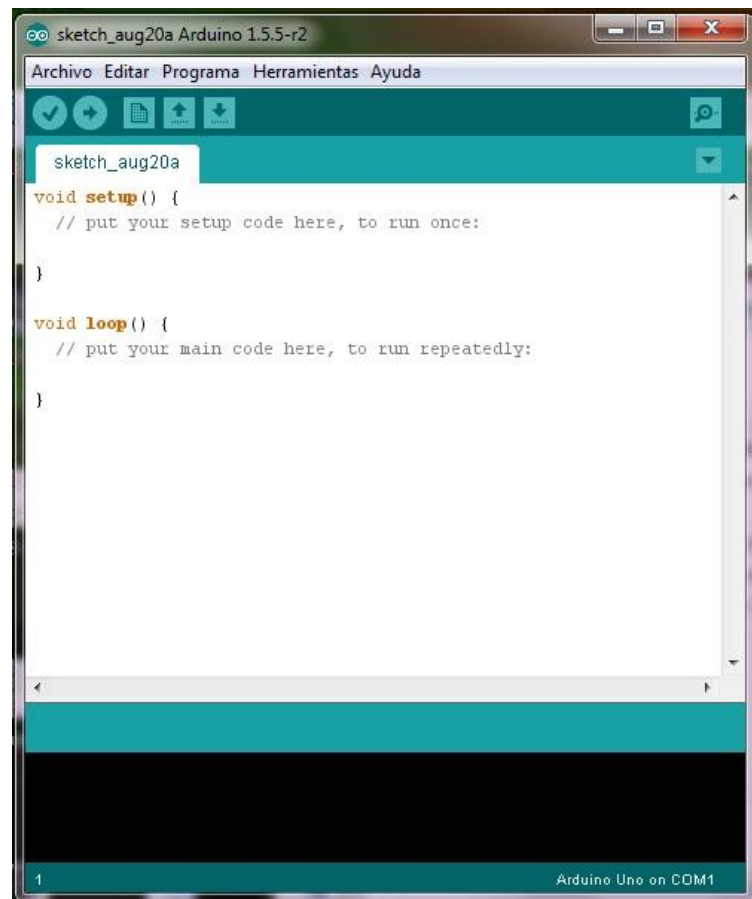


figura 28. Software de Programación Arduino
Fte. Arduino.cc

El lenguaje de alto nivel más utilizado para los desarrollos en Arduino ha sido desarrollado específicamente para este fin y está basado en Processing. Se puede entender como una adaptación del lenguaje java al entorno del micro-controlador.

A pesar de sus limitaciones, este compilador es una potente herramienta de desarrollo para los módulos Arduino.

Para el desarrollo de este proyecto se intercalará el uso de librerías y la programación de alto nivel con el uso de registros de bajo nivel para el control de los relojes.

6.3.- La ethernet shield de Arduino



figura 29. Ethernet Shield de Arduino

Fte. Arduino.cc

La Arduino Ethernet Shield permite a una placa Arduino conectarse a internet. Está basada en el chip ethernet Wiznet W5100.

El Wiznet W5100 provee de una pila de red IP que admite TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas.

La ethernet shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino.

Arduino usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la ethernet shield. Estos pines no pueden ser usados para e/s genéricas.

La shield provee un conector ethernet estándar RJ45.

El botón de reset en la shield resetea ambos, el W5100 y la placa Arduino. Además la shield contiene un número de LEDs para información sobre su estado.

Para usar la Ethernet Shield solo hay que montarla sobre la placa Arduino. se puede usar la librería Ethernet para escribir programas que se conecten a internet usando la shield.

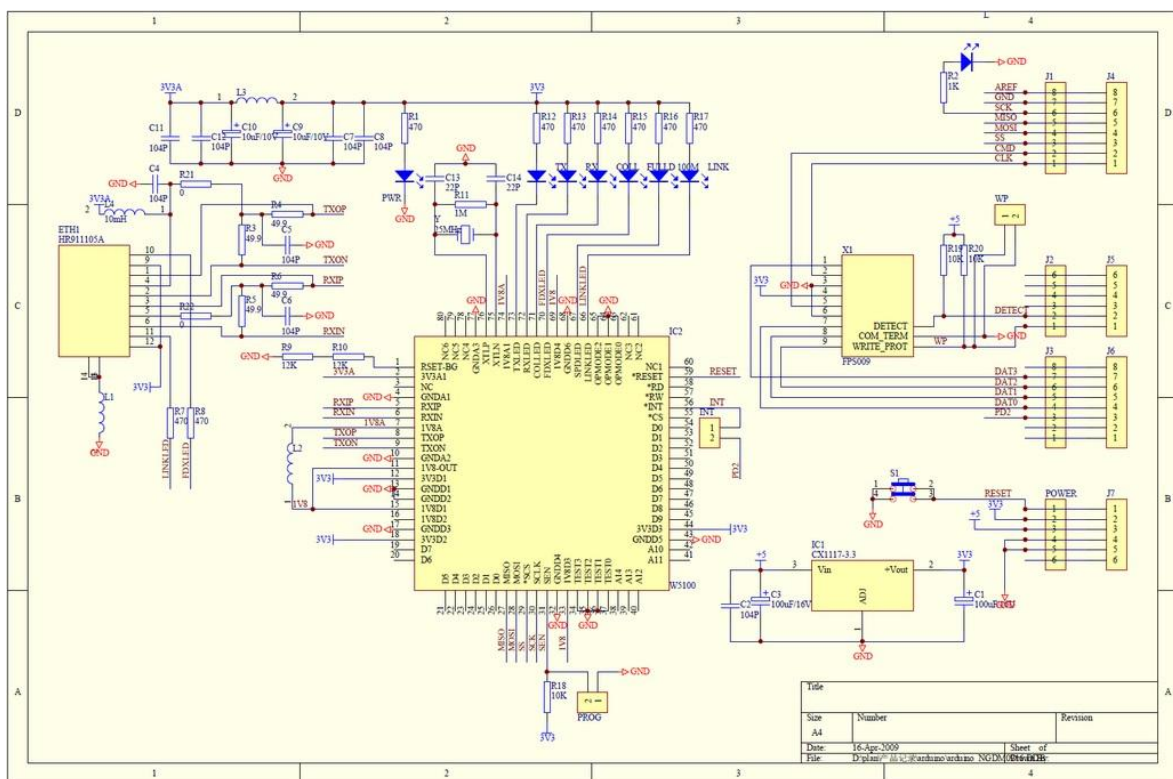


figura 30. Esquema Ethernet Shield

Fte. blog.kedairobot.com

6.4.- Implementación

A la hora de plantearnos implementar un sistema de captación y procesamiento de los datos recibidos por los caudalímetros, lo primero que debemos hacer es aclarar cual va a ser el formato de transmisión de datos desde los propios sensores hacia el sistema Arduino, en qué forma debemos procesar esos datos de acuerdo al resultado que queramos obtener, y de que forma presentaremos esos datos a los usuarios. A continuación se presentará un sencillo boceto de la solución propuesta:

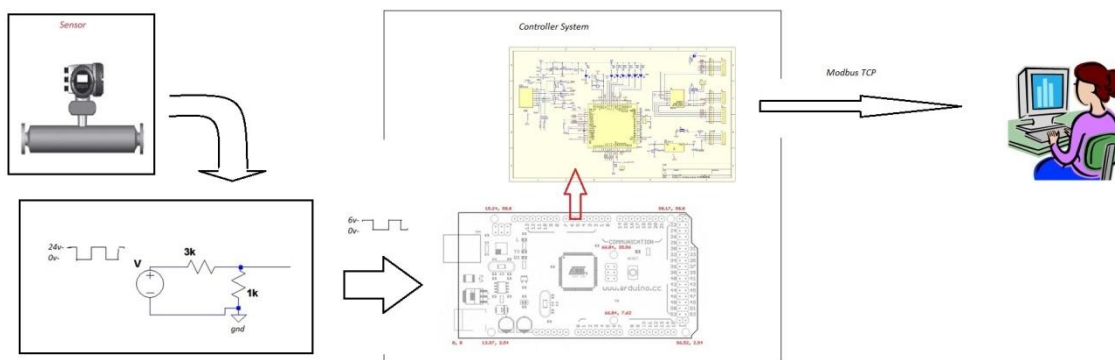


Figura 31. Esquema De Flujo De Datos Del Sistema

6.4.1.- Entradas

Tal y como propusimos en un plantemaiento anterior las entradas provenientes del sistema de acondicionamiento de señal será un tren de pulsos TTL. que se corresponderá con el flujo totalizado de combustible que pase por el sensor. Por razones prácticas se decide programar la salida del sensor para un régimen de 10 pulsos por kilo de combustible. Esto nos otorga una resolución de 100g/pulso, y un rango dinámico de hasta 1 tonelada por segundo, a 10 Khz.

Además, el controlador recibirá una señal de alarma por cada caudalímetro, si estos detectan que el flujo esta invertido.

El controlador también recibirá una señal de activación proveniente del operador para indicarle que empiece la monitorización.

6.4.2.- Salidas

Las salidas se transmitirán Vía Modbus TCP a través del puerto ethernet, para ser interpretados por el ordenador.

6.4.3.- Cálculos y procedimientos

Para calcular el consumo de combustible a tiempo real, y con la máxima precisión posible, debemos utilizar un controlador por cada motor, calculando individualmente la contribución de cada uno de ellos al consumo total del buque. Posteriormente, será el software del operador quien sume todas esas contribuciones para obtener el consumo total.

Cada controlador recibirá pues 2 señales digitales de entrada compuestas por sendos trenes de pulsos correspondientes a los caudales de alimentación y retorno del motor en cuestión, en el instante en que reciba dichas señales, es misión del controlador sumarlas a un cómputo general, y hallar la diferencia entre estos caudales, obteniendo así el combustible utilizado en un intervalo de tiempo dado.

El problema entonces radica en la resolución temporal, y en no perder ningún pulso proveniente de los sensores, ya que perder un sólo pulso no solo significaría aumentar el error puntualmente en una medición, sino que la acumulación de errores y el acarreo de estos en un sistema que trabaja a altas frecuencias distorsionaría los cálculos de un modo inaceptable.

Garantizar esta situación con un controlador digital, con un funcionamiento puramente secuencial se constituye como todo un desafío por muy rápido que este sea. incluso trabajar con interrupciones no garantiza el no solapamiento de dos o más pulsos durante el proceso de llamada, además de otros factores de incertidumbre, como las interrupciones asociadas al puerto ethernet.

No obstante existe una característica muy de destacar en estos controladores que nos ayuda a resolver eficazmente esta situación, y está presente en los relojes, o timers del sistema.

Ante las alarmas provenientes de los caudalímetros el controlador debe mediante interrupciones invertir el conteaje.

6.4.4.- Configuración de los temporizadores del AVR mega 2560

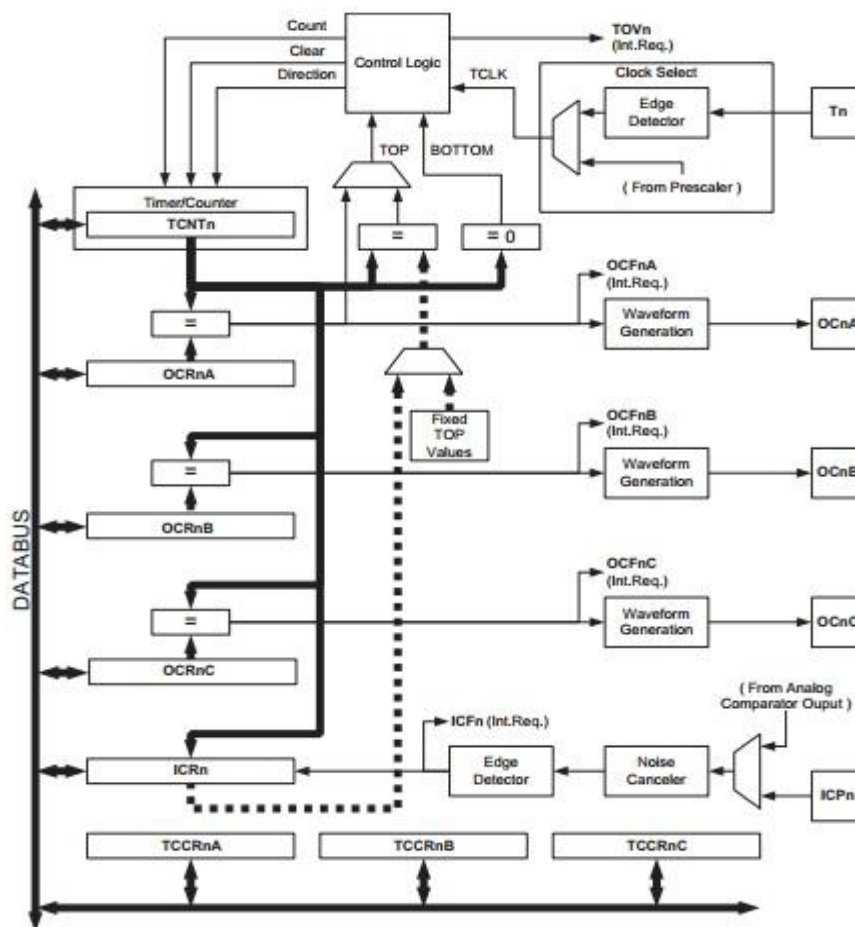


figura 32. Diagrama de Bloques Temporizadores/Contadores
Fte: Atmel Corporation

El AVR Mega 2560 posee 6 temporizadores/Contadores de los cuales el T1,T3,T4,T5 son temporizadores/contadores de 16 bits, los temporizadores /contadores T0 y T2 son de 8 bits.

El temporizador/contador T0 es usualmente utilizado como contador de sistema por las librerías arduino, es el responsable de funciones como milis, micros y también es utilizado en la librería ethernet.h, luego no es recomendable su uso a bajo nivel.

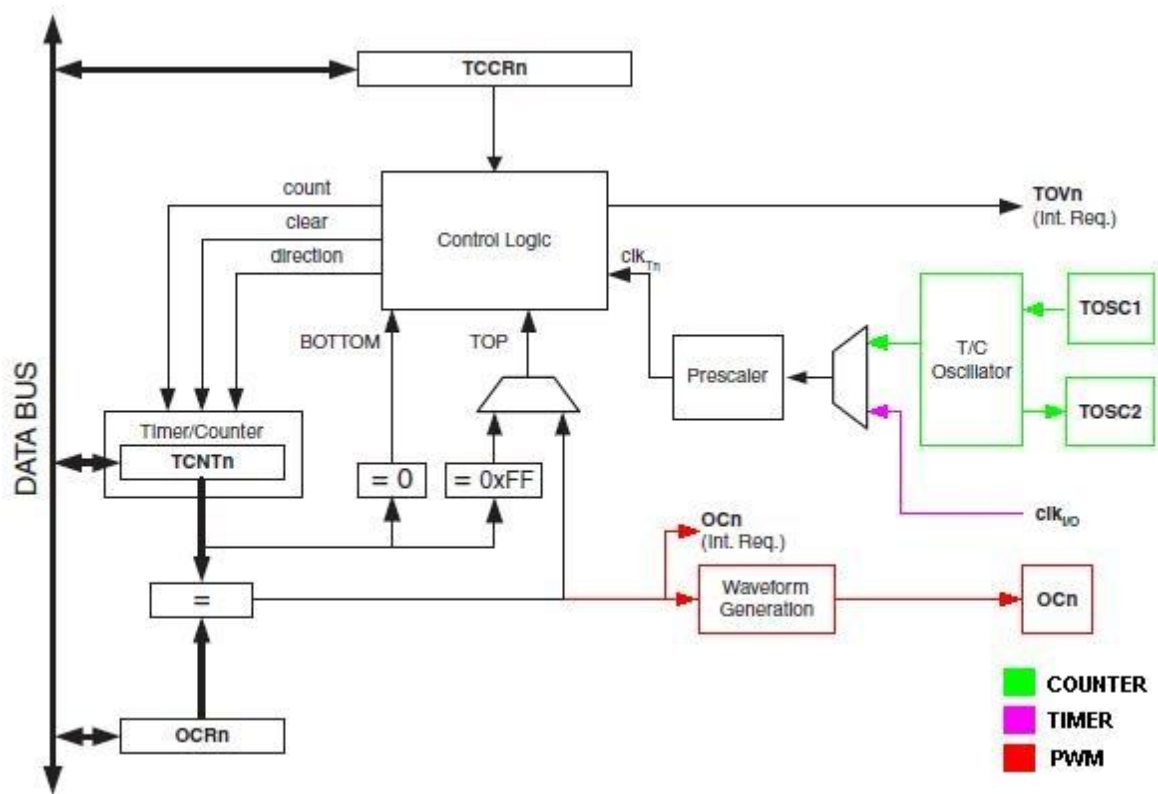


figura 33. Configuración del Timer como contador, Temporizador o generador de pulsos PWM
fte: <http://dubworks.blogspot.com/es/>

Un timer es sencillamente un contador controlado por registros internos. La ventaja es que tanto el oscilador de entrada como las operaciones del timer son completamente independientes de la ejecución del programa. Esto hace posible medir el tiempo midiendo los ciclos de reloj transcuridos, pero además el AVR cuenta con un circuito de recuento interesante. Este circuito cuenta con dos tipos de entradas, de manera que podemos asociar el timer al reloj interno del controlador, es un cristal de cuarzo que oscila a 16Mhz, o como más

interesante nos resulta en nuestro caso, se pueden asociar a un reloj externo, en cuyo caso funcionaría como contador de pulsos.

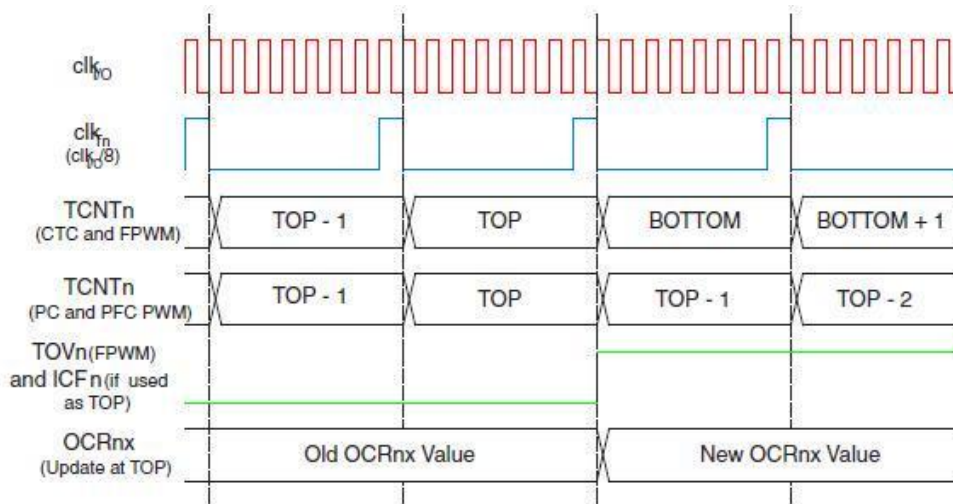


figura 34. Diagrama lógico de funcionamiento de un timer en avr
fite: <http://dubworks.blogspot.com.es>

Cuando la lógica de control recibe un impulso del prescaler se incrementa / decrementa el registro TCNT_n, ("n" simboliza el numero de timer que stamos utilizando). El registro TCNT_n se compara con el registro OCR_n. Cuando el registro TCNT_n alcanza el mismo valor que el valor OCR_n o por overflow, también es configurable, la lógica de control borra el registro TCNT y activa el bit tovn (Timer Overflow), que permanece fija hasta que se restablece por el usuario.

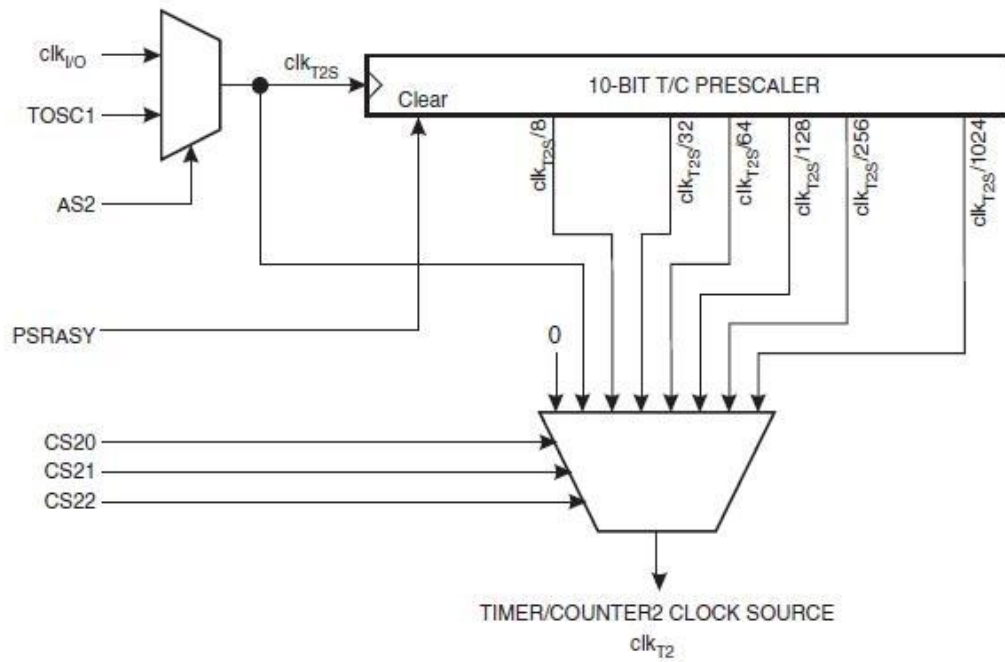


figura 35. Arquitectura Pre-scaler
 fte: : <http://dubworks.blogspot.com.es>

El pre-scaler cuenta el número de pulsos de entrada (desde el reloj interno o de fuentes externas) y cuando el número alcanza el número predeterminado (0, 8, 32, 64, 128, 256 o 1024) genera un pulso en la salida. Esto es necesario porque el AVR posee solamente dos registros de 8 bits y 3 registros de 16 bits asignados a sus cinco relojes, a la velocidad que funciona por ejemplo el reloj interno, el registro TCNTn (16 bits) se desbordaría en apenas unos 4 segundos.

Para configurar los Timers debemos modificar los registros TCCRnA y TCCRnB:

TCCR1A

Bit 7							Bit 0
COM1A1	COM1A0	---	---	---	---	PWM11	PWM10

TCCR1B:

Bit 7							Bit 0
ICNC1	ICES1	---	---	CTC1	CS12	CS11	CS10

COM1A1	COM1A0	Compare Output Mode
0	0	Disconnect Pin OC1 from Timer/Counter 1
0	1	Toggle OC1 on compare match
1	0	Clear OC1 on compare match
1	1	Set OC1 on compare match
PWM11	PWM10	PWM Mode
0	0	PWM operation disabled
0	1	Timer/Counter 1 is an 8-bit PWM
1	0	Timer/Counter 1 is a 9-bit PWM
1	1	Timer/Counter 1 is a 10-bit PWM

figura 35. Configuración De Bits Del Registro Tccrna

Fte: Avrbeginners.Net

El registro TCCRnA controla la salida OC1 para generación de ondas o señales PWM, en nuestro caso esto no nos interesa y debemos desactivarlo.

CS12	CS11	CS10	Mode Description
0	0	0	Stop Timer/Counter 1
0	0	1	No Prescaler (Timer Clock = System Clock)
0	1	0	divide clock by 8
0	1	1	divide clock by 64
1	0	0	divide clock by 256
1	0	1	divide clock by 1024
1	1	0	increment timer 1 on T1 Pin falling edge
1	1	1	increment timer 1 on T1 Pin rising edge

figura 36. Configuración Del Pre-Scaler En El Registro Tccrnb

Fte: Avrbeginners.Net

CTCn: setea el timer n en modo Compare-Match , al activar este bit podremos controlar hasta donde contará el contador antes de resetearse.

OCRn: en este registro de 16 bit debemos poner el número hasta el cual queramos que cuente el timer, previamente este ha de ser configurado como Compare-Match.

TIMSKn

Bit 7							Bit 0
TOIE1	OCIE1A	---	---	TICIE1	---	TOIE0	---

Con el Timer Interrupt Mask Register (TIMSK), podemos habilitar y deshabilitar las interrupciones que nos interesan.

TOIE_n: Timer Overflow Interrupt Enable (Timer 1); activa interrupción por desbordamiento.

OCIE_{nA}: Output Compare Interrupt Enable 1 A; activa la interrupcion cuando TCNT_n es igual a OCR_n

TICIE_n: Timer n Input Capture Interrupt Enable; Activa interrupciones externas

TOIE₀: Timer Overflow Interrupt Enable (Timer 0); activa interrupción por desbordamiento del Timer 0

6.4.5.- Limitaciones del sistema Arduino

Si bien el sistema Arduino nos aporta grandes ventajas en el desarrollo de aplicaciones con el microcontrolador avr, también es cierto que por desuso o por mantener la características del hardware, también limita muchas funcionalidades del mismo. En nuestro caso, para el desarrollo de nuestro proyecto necesitamos hacer uso de 2 contadores de impulsos externos de 16 bit, y sabemos que mientras el microcontrolador nos ofrece nada menos que 4 contadores de estas características, arduino solo nos provee de uno.

Como se puede observar en el mapa de conexionado del AVR2560 (figura 37), una patilla puede tener varios usos dependiendo de la configuración interna del chip, y siendo las de nomenclatura de color rojo las conexiones en el circuito de la Arduino, se puede observar perfectamente que las patillas 8, 27 y 49 correspondientes a las entradas de reloj para T3, T4 y T1 respectivamente están desconectadas.

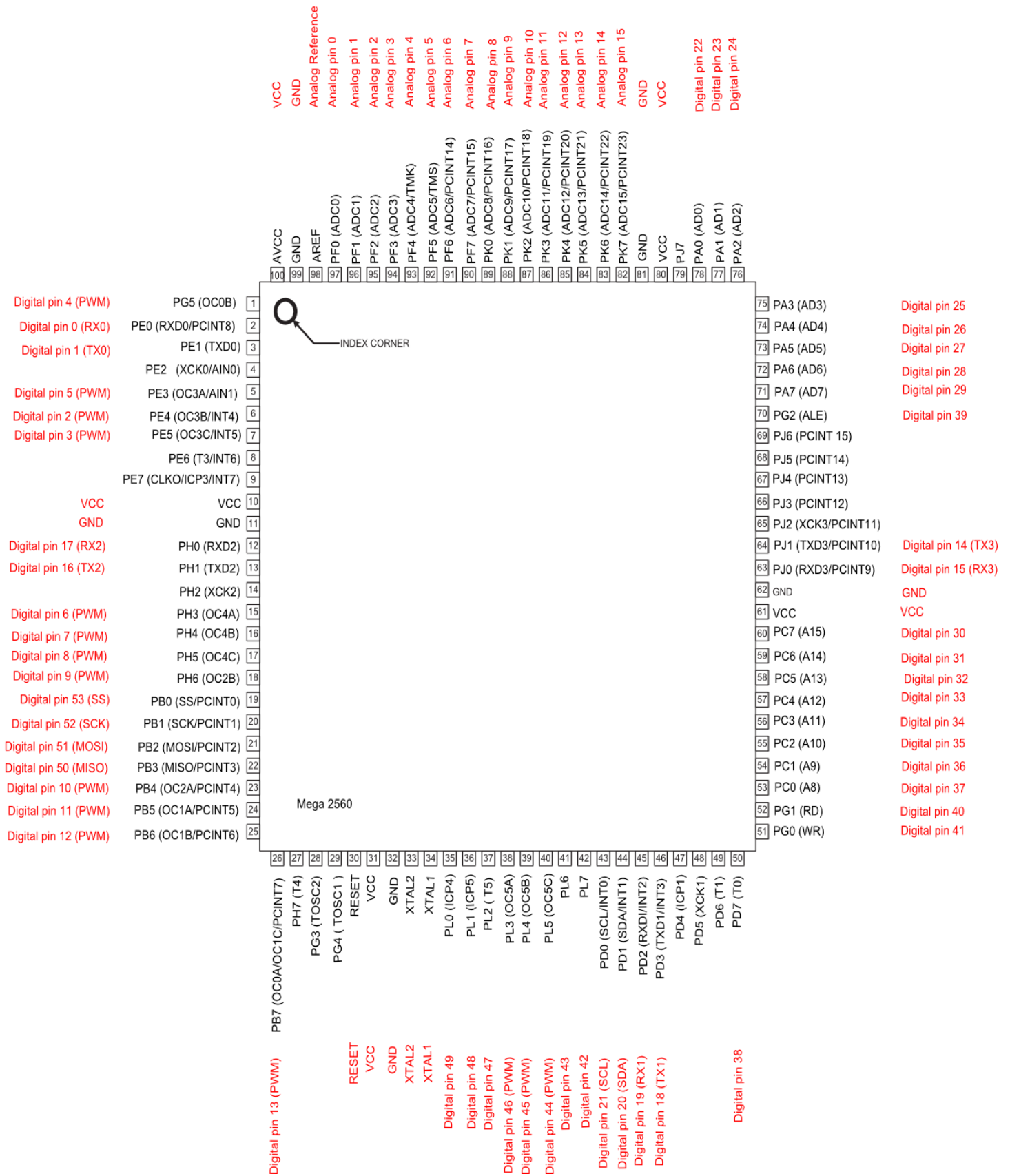


figura 37. Mapa de conexionado AVR 2560

fte: Atmel Corporation

Ante esta situación se planteo una solución simple que además mantiene la coherencia estructural del circuito.

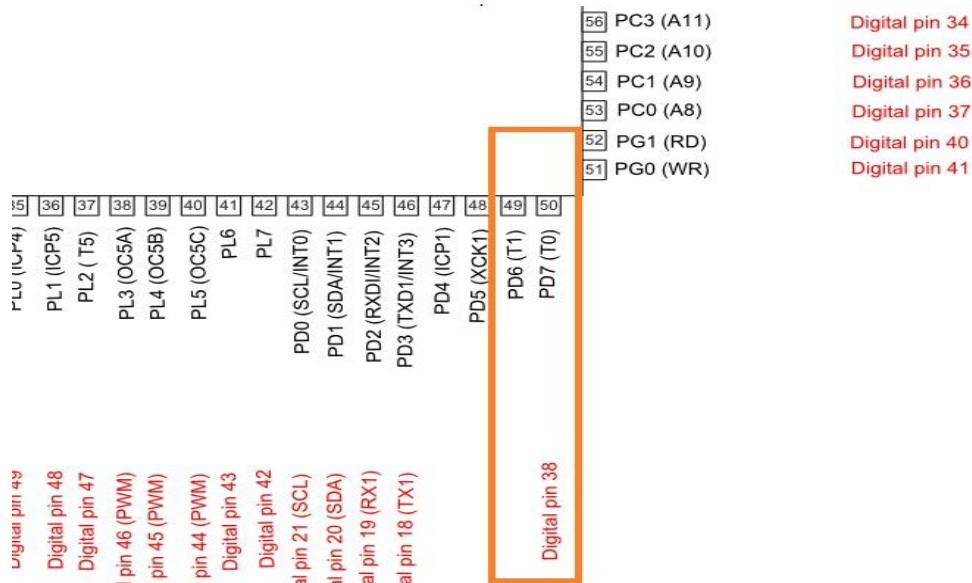


Figura 38. Ampliación Sectorial Figura 37

En esta ampliación podemos observar la proximidad a nivel de circuito entre la patilla 49 que corresponde a la excitación del temporizador T1 y la 50 conectada a la entrada/salida digital de propósito general número 38 del circuito de la Arduino.

Esta proximidad entre ambas y el hecho de estar en una esquina del chip hace más sencillo cortocircuitar ambas patillas minimizando el riesgo de afectar a las demás.

Para llevar a cabo esta operación debemos tener en cuenta que para que el sistema funcione correctamente la patilla 50 debe estar siempre en estado de alta impedancia, esto significa que:

- ✚ La Entrada/Salida Digital 38 no debe volver a ser usada como tal, quedaría anulada su función.

- ✚ Como puede observarse la patilla 50 también puede servir como reloj externo para el T0, cosa que no nos interesa porque es un contador de 8 bits, y porque como dijimos anteriormente T0 trabaja con funciones Internas de Arduino, luego debemos asegurarnos que T0 trabaje siempre con reloj Interno y no externo.

Tomando esas precauciones podemos cortocircuitar mediante soldadura de estaño las patillas 49 y 50 convirtiendo la entrada 38 del circuito del arduino en nuestro segundo reloj externo (T1), y sin que por ello penalice ninguna otra funcionalidad en el circuito.

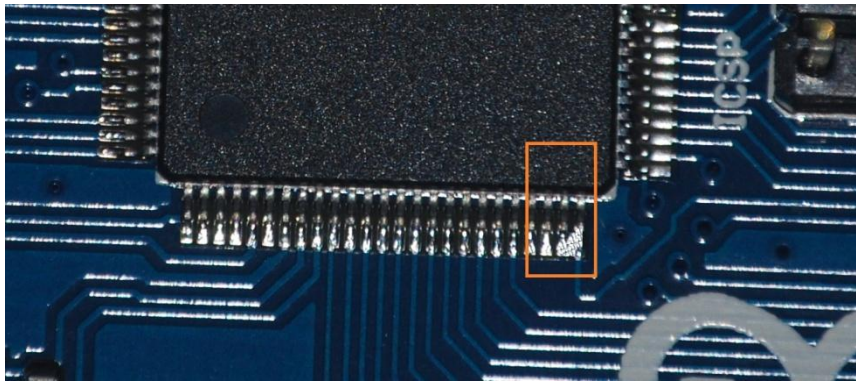


Figura 39. Arduino Modificada Para Disponer De Un Segundo Contador Externo (T1)

Una vez modificado apropiadamente el circuito ya podemos empezar a programar el sistema.

6.4.6.- Programación del controlador en Arduino

Para poder programar en el entorno arduino debemos tener en cuenta dos bloques de programa. El primero es el bootloader, es un firmware que proporciona las instrucciones básicas para el arranque del microcontrolador, normalmente este código viene de fábrica en nuestro chip (Built-in), y solo se puede acceder a él desde un programador. El segundo es el Bloque de programa, donde va nuestro archivo compilado y ejecuta las operaciones y comandos que nosotros programamos desde nuestro entorno. Ambos bloques se guardan en el

bloque de memoria flash de nuestro controlador, que en nuestro caso es de 256 Kbyte, de los cuales 4 Kbyte son ocupados por el bootloader.

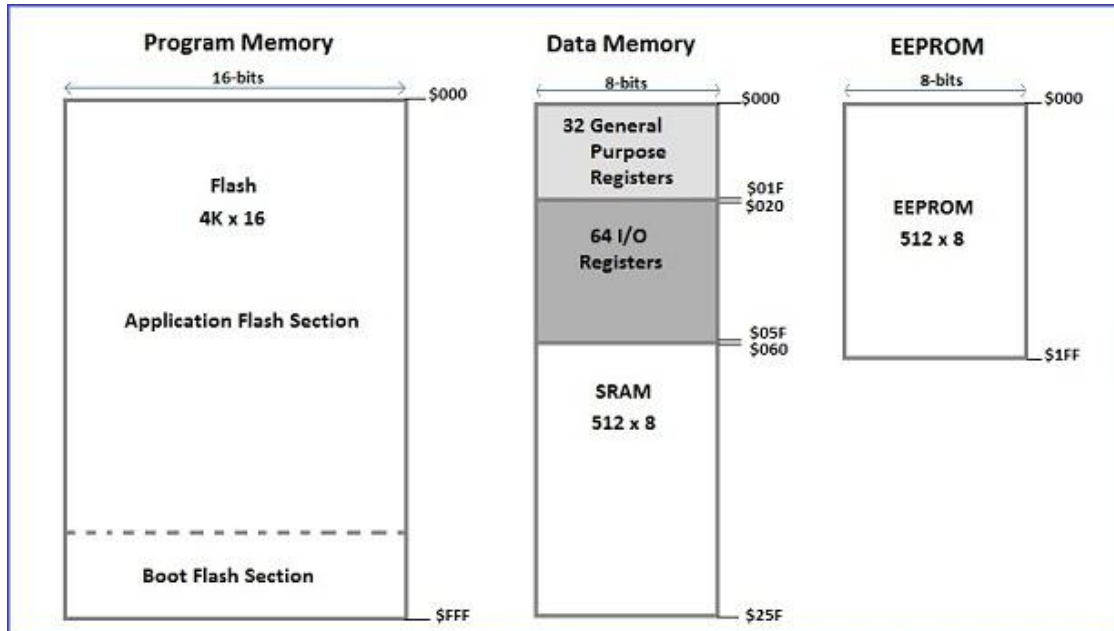


figura 40. Mapa de memoria Avr-Mega 2560
fte: www.avr-tutorials.com

6.4.7.- Desarrollo De un servidor de comunicación vía Modbus/TCP con Arduino

Además de los ya expuestos, uno de los principales motivos para desarrollar un sistema embebido que lea los datos de los sensores de caudal, es para que, una vez calculados los consumos por motor, estos puedan ser enviados a un operador mediante un protocolo de comunicación universal. Para cumplir con este propósito se ha establecido que por su sencillez, robustez y amplia adopción en el ámbito de las comunicaciones industriales el protocolo Modbus/TCP se perfila como el más apropiado.

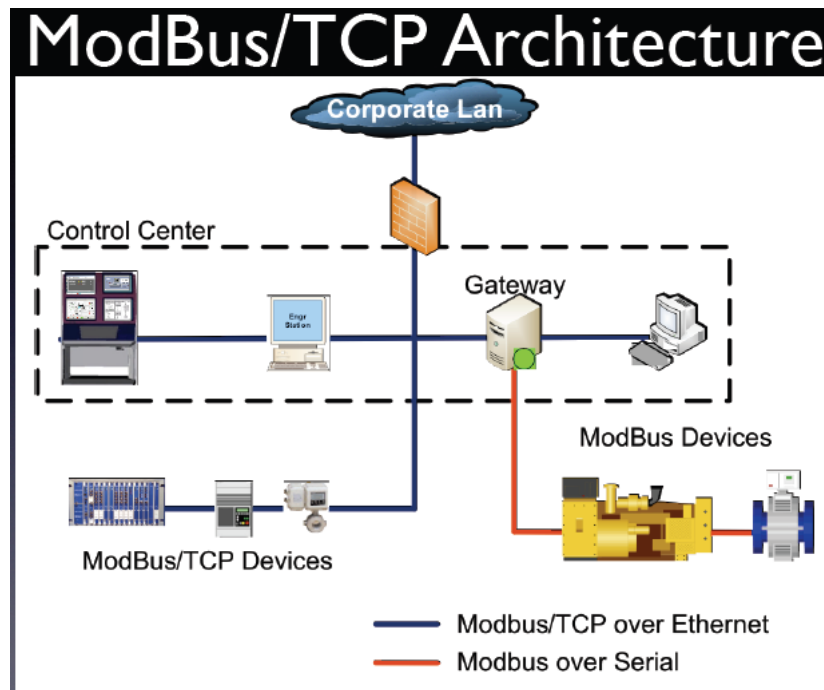


figura 41. Arquitectura Modbus Y Modbus/TCP
Fte: uhu.es

6.4.7.1.- El protocolo Modbus

El protocolo de Comunicaciones Industriales MODBUS fue desarrollado en 1979 por la empresa MODICON, es un protocolo abierto, sencillo de implementar y flexible lo que lo ha convertido en uno de los protocolos de comunicación más extendidos. Se puede implementar sobre muchos tipos de soportes físicos, sirviendo para las comunicaciones a través de puertos serie, como RS-485, RS-232, o Ethernet.

La comunicación se basa en una arquitectura "Maestro-Esclavo", donde el Maestro es quien inicia la comunicación, ordena, y el esclavo quien ejecuta y responde. En esta configuración puede haber un Maestro, y hasta 248 esclavos direccionados desde el 0 al 247. Cada solicitud del maestro es tratada de forma independiente por el esclavo, sin relación con las anteriores. Esto facilita proveer transacciones de datos resistentes a rupturas, requiriendo

mínima información de recuperación para mantener una transacción en cualquiera de los dos terminales.

El Maestro también puede enviar información a los esclavos.

MODBUS utiliza una representación "big-endian" para direcciones y datos, esto significa que cuando un registro más grande que un byte es transmitido, el byte más significativo es enviado primero. por ejemplo, para enviar el número "0x1234", se enviaría primero "0x12" y luego "0x34".

En el MODBUS serial los datos se envían como una serie de bits, cada bit se envía como una tensión, los ceros se envían como voltajes positivos (+5V) y los 1 como voltajes negativos (-5V). La velocidad de transmisión típica para una conexión serie es de 9600 baudios.

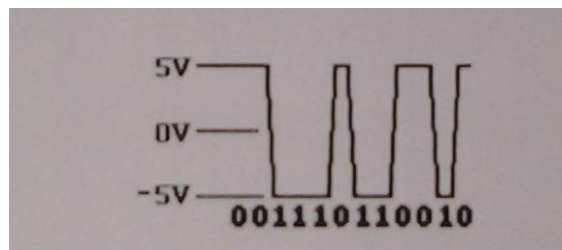


figura 41. Codificación En Niveles De Voltaje (MODBUS RS-232)
Fte: Servidor Modbus/TCP Con Microcontrolador Pic (PFC,2012)

Para el almacenamiento de datos en el dispositivo esclavo se utilizan cuatro tablas diferentes.

Dos tablas almacenan valores lógicos "COILS" (se traduce como bobinas), y dos tablas almacenan registros de 8 bits, para cada tipo de dato, una tabla es de sólo lectura, y la otra es de lectura /escritura.

Cada tabla posee hasta 9999 valores. y se direccionan como sigue:

Dirección de la Tabla	dirección de datos (Hex)	tipo	Nombre de la tabla
00001-09999	0000 - 270E	Lectura/Escritura	Output Coil
10001-19999	0000 - 270E	Sólo Lectura	Input Contact
30001-39999	0000 - 270E	Sólo Lectura	Input Register
40001-49999	0000 - 270E	Lectura/Escritura	Input Holding register

Figura42. Direccionamiento De Tablas De Almacenamiento
Fte: Servidor Modbus/Tcp Con Microcontrolador Pic (Pfc,2012)

Cuando el Maestro envía una orden, el primer byte que envía es la correspondiente al Id. de esclavo, el Segundo byte corresponde al código de función, aquí se incluye a que tabla tiene que acceder, y si el acceso es de sólo lectura o de lectura/escritura.

El Código de Operación es un número de un solo byte que usa MODBUS para diferenciar entre una operación y otra, este número indica al esclavo qué función o que operación tiene que hacer, A continuación se muestran los códigos de operación para cada una de las tablas de datos que maneja MODBUS y una pequeña descripción de lo que hace cada función.

Output Coils

Código de operación	Acción	Descripción
01 (01 Hexadecimal)	Lectura de uno o más "coil's"	Sirve para leer qué valor tiene el coil (un 1 o un 0)
05 (05 Hexadecimal)	Escritura de un solo "coil"	Sirve para enviar un uno o un cero al coil elegido
15 (0F Hexadecimal)	Escritura de múltiples "coil's"	Envía unos o ceros a varios coil.

Input Contact

Código de operación	Acción	Descripción
02 (02 Hex)	Lectura de los "Input Contact"	Lee los valores que tiene almacenado el o los "input contact"

Input Register

Código de operación	Acción	Descripción
04 (04 Hex)	Lectura de los "Input Registers"	Lee los valores que tiene almacenado el o los "input register"

Holding Registers

Código de operación	Acción	Descripción
03 (03 Hex)	Lectura de los holding	Lee qué valor tiene almacenado uno o varios Holding Registers
06 (06 Hex)	Escritura de un holding	Escribe un valor entre 0 65535 en un solo holding
16 (10 Hex)	Escritura de múltiples holding	Escribe un valor entre 0 65535 en varios holdings

El tercer y cuarto Byte es una palabra (16 bits) que corresponde a la dirección del primer coil/registro de la trama a la que se desea acceder , y el quinto y sexto byte corresponden a la longitud de la trama.

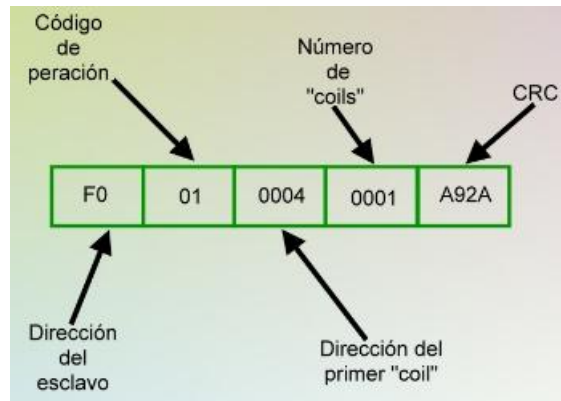


figura43. Descripción de la trama de información enviada por el Maestro
 fte: <http://www.tecdigitaldelbajio.com/>

Existen dos bytes más en el protocolo MODBUS original correspondiente al CRC (Código de Redundancia Cíclica) para la detección de errores en la transmisión. No es así en la variante Modbus/TCP que nosotros utilizamos, ya que en este caso el protocolo MODBUS está embebido en la capa de datos del Protocolo TCP/IP, que posee su propio código de detección de errores.

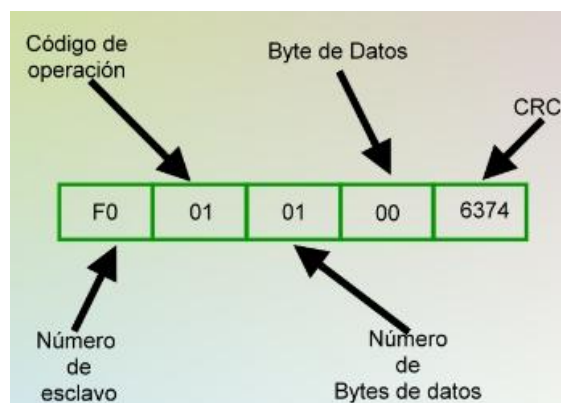


figura 44. Descripción de la trama de información enviada por el Esclavo
 fte: <http://www.tecdigitaldelbajio.com/>

La trama de respuesta del esclavo contiene tanto más bytes cuanto mayor sea la cantidad de datos pedidos por el maestro. Esto se explicará mejor con un ejemplo.

Supongamos que el maestro realiza una petición del valor de 7 coils, la trama enviada sería:

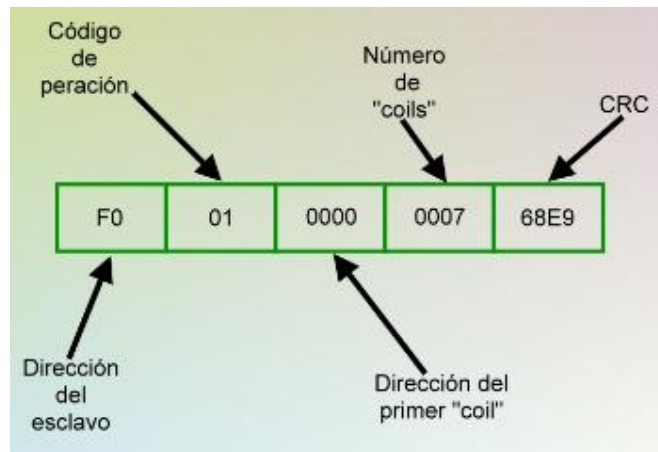


figura 45. Descripción de la trama de información enviada por el Maestro
 fte: <http://www.tecdigitaldelbajio.com/>

y supongamos que los valores de la tabla para los **coil's #** son:

- #1=0
- #2=1
- #3=0
- #4=1
- #5=0
- #6=1
- #7=1

en este caso la respuesta del esclavo sería:

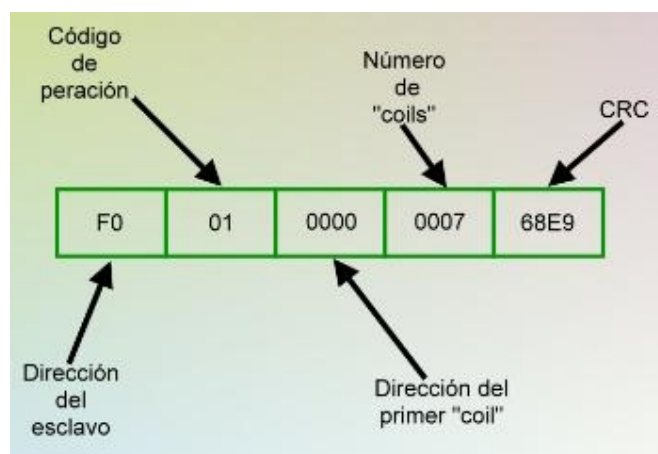


figura 46. Descripción de la trama de información enviada por el Maestro
 fte: <http://www.tecdigitaldelbajio.com/>

Donde '6A' es la representación hexadecimal de cada uno de los coils.

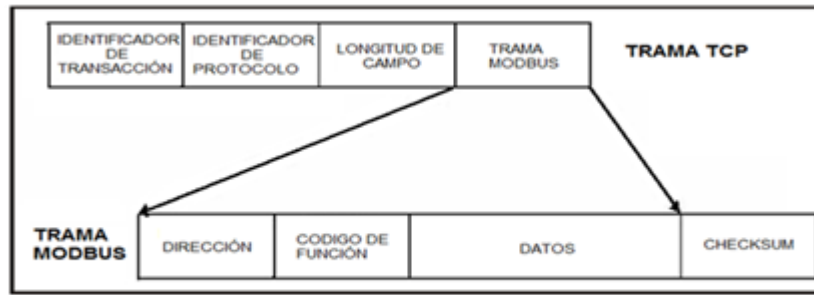


figura 47. Representación hexadecimal de los bits enviados
fte: <http://www.tecdigitaldelbajio.com/>

6.4.7.2.- El Protocolo MODBUS TCP/IP

El protocolo MODBUS/TCP es un protocolo de comunicación diseñado por Schneider Automation como una variante de la familia de protocolos MODBUS, ampliamente usada para la supervisión y el control de equipo de automatización. Específicamente el protocolo define el uso de mensajes MODBUS en un entorno intranet o internet usando los protocolos TCP/IP.

La especificación Modbus/TCP define un estándar interoperable en el campo de la automatización industrial, el cual es simple de implementar para cualquier dispositivo que soporte sockets TCP/IP. Todas las solicitudes son enviadas vía TCP sobre el puerto registrado 502 y normalmente usando comunicación half-duplex sobre una conexión dada. Modbus/TCP básicamente encapsula una trama MODBUS dentro de una trama TCP en una manera simple como se muestra en la figura a continuación.



Encapsulamiento de la trama Modbus en TCP.

figura 48

fte: monografias.com

Esta aplicación se fundamenta en que las operaciones de programación esperan una comunicación orientada a la conexión, es decir, las máquinas de origen y de destino deben establecer un canal de comunicaciones antes de transferir datos.

En Modbus/TCP una conexión se establece inicialmente en la capa de aplicación y esta única conexión puede llevar múltiples transacciones independientes. En Modbus/TCP se usa el protocolo orientado a la conexión TCP en lugar del protocolo orientado a datagramas UDP.

El campo "Id. esclavo" de MODBUS es reemplazado por un byte "identificador de unidad" el cual puede ser usado para comunicar a través de dispositivos tales como puentes y gateways, los cuales usan una dirección IP única para soportar múltiples unidades terminales independientes.

Los mensajes de solicitud y respuesta en Modbus/TCP poseen un prefijo ó encabezado compuesto por seis bytes como se aprecia en la tabla.

Ref.	Ref.	00	00	00	Len.
------	------	----	----	----	------

figura 49. Estructura del prefijo de Modbus/TCP

fte: monografias.com

El "Ref. Ref." anterior son los dos bytes del campo "referencia de transacción", un número que no tiene valor en el servidor pero son copiados literalmente desde la solicitud a la respuesta a conveniencia del cliente. Este campo se utiliza para que un cliente Modbus/TCP

pueda establecer simultáneamente múltiples conexiones con diferentes servidores y pueda identificar cada una de las transacciones.

El tercer y cuarto campo del prefijo representan el identificador de protocolo, un número el cual debe ser establecido a cero.

El "Len" especifica el número de bytes que siguen. La longitud es una cantidad de dos bytes, pero el byte alto se establece a cero ya que los mensajes son más pequeños que 256. De esta forma, un mensaje Modbus/TCP completo posee una estructura como se muestra en la tabla:

Posición de Byte	Significado
Byte 0	Identificador de transacción. Copiado por el servidor- normalmente 0.
Byte 1	Identificador de transacción. Copiado por el servidor -normalmente 0.
Byte 2	Identificador de protocolo = 0.
Byte 3	Identificador de protocolo = 0.
Byte 4	Campo de longitud (byte alto) = 0. Ya que los mensajes son menores a 256.
Byte 5	Campo de longitud (byte bajo). Número de bytes siguientes.
Byte 6	Identificador de unidad (previamente *dirección esclavo*).
Byte 7	Código de función MODBUS.
Byte 8 a más	Los datos necesarios.

figura 50. Estructura de mensajes en Modbus/TCP
fte: monografias.com

6.4.7.3.- Ventajas Del Protocolo MODBUS/TCP

- ✚ Es simple para administrar y expandir. No se requiere usar herramientas de configuración complejas cuando se añade una nueva estación a una red Modbus/TCP. No es necesario equipo o software propietario de algún vendedor. Cualquier sistema de cómputo con una pila de protocolos TCP/IP puede usar Modbus/TCP.
- ✚ Puede ser usado para comunicación con una gran base instalada de dispositivos MODBUS, usando productos de conversión los cuales no requieren configuración.

- ✚ Es de muy alto desempeño, limitado típicamente por las capacidades de comunicación del sistema operativo del computador. Se pueden obtener altos ratios de transmisión sobre una estación única y la red puede ser configurada para lograr tiempos de respuesta garantizados en el rango de milisegundos.

- ✚ Realizar reparaciones o mantenimiento remoto desde la oficina utilizando un PC, reduciendo así los costes y mejorando el servicio al cliente.

- ✚ El ingeniero de mantenimiento puede entrar al sistema de control de la planta desde su casa, evitando desplazamientos.

- ✚ Permite realizar la gestión de sistemas distribuidos geográficamente mediante el empleo de las tecnologías de Internet/Intranet actualmente disponibles.

- ✚ La amplia adopción del Protocolo MODBUS/TCP a escala global pone a nuestra disposición amplio material y documentación para su implementación.

- ✚ En lo que respecta a Arduino existen gran cantidad de librerías de software abierto con mayor y menor grado de depuración para la comunicación vía Modbus TCP.

En nuestro caso nos decidimos, basándonos en su fiabilidad, nivel de depuración y amplia adopción , a trabajar con la librería "Mudbus.h" que proporciona las funciones necesarias para la implementación de un terminal esclavo de comunicaciones vía MODBUS/TCP.

6.4.8.- Funciones Básicas De La Librería Ethernet

Con el Arduino Ethernet Shield, esta biblioteca permite que una placa Arduino se conecte a una red a través del puerto ethernet. Puede servir como un servidor para establecer conexiones entrantes o como cliente que realiza las salientes. La biblioteca admite hasta cuatro conexiones simultáneas (entrantes, salientes o una combinación de ambas).

Las funciones básicas que Utilizaremos en nuestro proyecto son:

- ✚ [IPAddress\(\)](#): Asigna una dirección IP a la Arduino.

- ✚ [EthernetServer\(\)](#): Crea un servidor que atiende las conexiones entrantes en el puerto especificado se ha de llamar mediante la clase server(Puerto).

- ✚ [begin\(\)](#): Inicia la conexión.

Una vez conectado la librería [#mudbus.h](#) toma el control. Esta librería a su vez hace uso de la librería Ethernet.

6.4.9.- Funciones Básicas De La Librería 'mudbus.h'

La librería Mudbus.h contiene un conjunto necesario de estructuras de Datos e instrucciones para realizar la conexión de un terminal esclavo mediante el protocolo MODBUS/TCP. Las ordenes Básicas que usaremos en nuestro proyecto son:

- ✚ Clase Mudbus: se ha de crear un objeto de la clase mudbus para empezar.
- ✚ Run(): activamos la escucha, el esclavo queda a la espera de las instrucciones del Maestro.
- ✚ Mb.R[Dirección de registro]: Direcciona la tabla de registros de Lectura/Escritura.
- ✚ Mb.C[Dirección de bit]: Direcciona la tabla de bits.

6.4.10.- Descripción del código fuente:

En esta parte haremos una descripción detallada de la implementación del código fuente, dicha descripción se hará siguiendo un orden lógico y mediante capturas de fragmentos del código original desarrollado en el entorno Arduino.

Al Igual que en C++, lo primero que debemos hacer para empezar a programar con arduino es declarar que librerías vamos a utilizar, para lo que utilizamos la directiva `#Include`. Las librerías `'avr/io'` y `'avr/interrupt'` nos permiten acceder a los registros a bajo nivel y programar interrupciones, también podemos programar directamente en código ensamblador.

```
#include <SPI.h>
#include <Ethernet.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>
#include "mudbus.h"
```

Directivas de Compilación - librerías-

A continuación definimos la dirección MAC unívoca que asignaremos a nuestro sistema, y una ip que esté en el rango de la red a la cual queremos conectarlo, declaramos el

puerto de acceso (80) ya que es el que por defecto se asigna a internet, creamos un objeto tipo mudbus, utilizando los comandos como explicamos anteriormente.

```
// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(172,16,201,4);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):

EthernetServer server(80);
Mudbus Mb;
```

Configuración de red y conexiones Modbus

En el setup inicializamos los registros y las variables del programa. Este Código sólo se ejecutará una vez al arrancar el dispositivo.

El WatchDog nos permite vigilar que el dispositivo no se cuelgue o entre en bucles infinitos, de ser así, reiniciaría inmediatamente el sistema para interrumpir la monitorización el menor tiempo posible.

Ponemos todos los registros TCCRxA a cero puesto que nos interesa deshabilitar las funcionalidades PWM de los Timers. En cambio en los TCCRxB sólo es una forma de inicializarlos, como puede verse más abajo a estos se les asigna el número 15 para activar los bits que configuran los relojes 1 y 5 como contadores externos(Compare Match). También inicializamos la cuenta mediante el registro TCNTx. y colocamos los límites de cuenta que nos interesan en los registros OCRxA. Finalmente con TIMSKx habilitamos las interrupciones.

```

void setup() {

  wdt_disable(); // deshabilitamos el WatchDog

  // Inicializamos los contadores T1 y T5

  cli();          // Deshabilitamos interrupciones globalmente
  TCCR1A = 0;     // ponemos el registro TCCR1A a 0
  TCCR1B = 0;     // lo mismo para TCCR1B
  TCCR5A = 0;     // lo mismo para TCCR5A
  TCCR5B = 0;     // lo mismo para TCCR5B
  TCCR3A = 0;
  TCCR3B = 0;
  TCNT5 = 0;
  TCNT1 = 0;
  TCNT3 = 0;

  // Seteamos los limites de las cuentas(compare match)
  OCR1A = 9;
  OCR5A = 9;
  OCR3A = 3125;

  // cambiamos a modo CTC
  TCCR1B |= 15;  //WGM12
  TCCR5B |= 15;  //WGM12

  TCCR3B |= 13;  //WGM12 // cambiamos Timar 3 en modo reloj

  // habilitamos interrupciones timer compare interrupt:
  TIMSK1 |= (1 << OCIE1A);
  TIMSK5 |= (1 << OCIE5A);
  TIMSK3 |= (1 << OCIE3A);

```

Set-Up de Temporizadores

Configuramos las entradas con las que vamos a trabajar:

```

pinMode(19, INPUT);
pinMode(18, INPUT);
pinMode(47, INPUT);
pinMode(38, INPUT);

```

entradas

Configuramos las interrupciones externas. Para nuestra aplicación necesitamos 2 interrupciones externas (2 y 3): una para cuando el caudal de alimentación está en retroceso, y otra cuando el caudal de retorno retrocede. Las interrupciones externas en Arduino pueden ser síncronas o asíncronas, las síncronas se pueden configurar a alto nivel o a bajo nivel. Sin embargo a las asíncronas, las que interesan en nuestro proyecto, sólo se puede acceder mediante registros. Las interrupciones externas asíncronas se configuran mediante los

registros EICRA(Interrupciones 2 y 3), y EIMSK es la máscara de habilitación de las interrupciones.

La siguiente tabla nos muestra las posibles configuraciones del registro EICRA.

EICRA – External Interrupt Control Register A

The External Interrupt Control Register A contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
(0x69)	ISC31 ISC30 ISC21 ISC20 ISC11 ISC10 ISC01 ISC00								EICRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

figura 51. Registro EICRA
fte: <http://www.jmnlab.com/>

ISCn1	ISCn0	Description
0	0	The low level of INTn generates an interrupt request.
0	1	Any edge of INTn generates asynchronously an interrupt request.
1	0	The falling edge of INTn generates asynchronously an interrupt request.
1	1	The rising edge of INTn generates asynchronously an interrupt request.

figura 52. configuración de eventos Registro EICRA
fte: <http://www.jmnlab.com/>

Como puede extraerse del código presentado a continuación se han configurado las interrupciones externas para dispararse ante cualquier cambio en el nivel de entrada.

```
// definimos las interupciones que controlan el retorno

EICRA &= ~(1<<ISC21); //INT2
EICRA |= (1<<ISC20);
EICRA &= ~(1<<ISC31); //INT3
EICRA |= (1<<ISC30);
EIMSK |= ((1<<INT2)|(1<<INT3)); //Habilitar interrupciones externas 2,3.
sei(); //Habilitar todas las interrupciones

// terminamos interrupciones
```

Configuración Interrupciones Externas

Finalmente iniciamos la conexión Modbus/TCP configuramos los pines de salida y activamos el Watchdog.


```

// Inicializamos Conexion Modbus/TCP

Serial.begin(9600); // Open serial communications
Ethernet.begin(mac, ip); // start the Ethernet connection and the server
server.begin();
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(13, OUTPUT);
pinMode(22, INPUT);

// Terminamos Set Up de modbus/TCP

wdt_enable(WDTO_4S); // Habilitamos WatchDog

} /// final del Setup

```

El siguiente paso es declarar las variables que se usarán en todo el programa, hay que tener en cuenta que al trabajar con interrupciones las variables deben ser declaradas como de acceso directo a memoria ("**volatile**").

```

volatile unsigned long gin,retin=0;
volatile unsigned long gout,retout=0;
volatile unsigned long parcial=0;
volatile unsigned long Cin,Cout,ginpre,goutpre = 0;
volatile byte pulsos_ret_e=0;
volatile byte pulsos_ret_s=0;
volatile byte pulsos_ent=0;
volatile byte pulsos_sal=0;

```

Las instrucciones que se ejecutan en la función `loop()` se ejecutan cíclicamente mientras el controlador permanezca activado. En esta parte del código calculamos la diferencia entre caudales de entrada y salida, y preparamos los datos para su colocación en orden en las tablas modbus.

Caben destacar dos cosas:

- ✚ Como estamos trabajando con precisiones elevadas nuestros variables son de 16 bits, no así los registros modbus, con lo cual tenemos que fraccionar los datos en dos bytes para poder incluirlos en las tablas.

- ✚ Al tener en cuenta los retrocesos esto puede dar lugar a consumos negativos, esta situación que puede darse debido al sesgo de una medición, no es real en la práctica, por lo tanto debemos eliminar este tipo de situaciones si es que llegan a producirse.

Finalmente se consulta en la tabla de bits de entrada/salida si el maestro ha enviado la instrucción de reiniciar el equipo, si es así se ejecuta la Función Reset, si no es así se refresca el watchdog y todo sigue ejecutándose.

```

void loop() {
  unsigned long tmp1,tmp2,tmp=0;
  Mb.Run();
  if (gin > retin)    /// si el retroceso es mayor que el consumo ej. arranque, consumo debe ser 0
  {
    tmp1 = (gin-retin)*100;    /// y permanecera a cero hasta que que el consumo acumulado supere el :
  }
  else                /// momento en que empezará a contar, manteniendo siempre el registro
    tmp1=0;
  Mb.R[0] = 65535 & tmp1; // g entrada Parte Menos Significativa
  Mb.R[1] = (tmp1>>16); // g entrada Parte más Significativa

  if (gout > retout)    /// si el retroceso es mayor que el consumo ej. arranque, consumo debe se:
  {
    tmp2 = (gout-retout)*100;
  }
  else
    tmp2=0;
  Mb.R[2] = 65535 & tmp2; // g Salida Parte - significativa
  Mb.R[3] = (tmp2>>16); // g Salida Parte + significativa

  if (tmp1 > tmp2)    /// si la salida es mayor que la entrada es por un error
    tmp = tmp1-tmp2;
  else
    tmp=0;

  Mb.R[4] = 65535 & tmp; // Consumo Total Acumulado g parte - sign.
  Mb.R[5] = (tmp>>16); // Consumo Total Acumulado g parte + sign.

  if (Mb.C[0])
    parcial=tmp; // Reset del Parcial
  tmp=tmp-parcial; // restamos el parcial fijado con el boton de reset para dar el consumo parcial
  Mb.R[6] = 65535 & tmp; // Consumo Parcial Acumulado g parte - sign.
  Mb.R[7] = (tmp>>16); // Consumo Parcial Acumulado g parte + sign.

  Mb.R[8] = (65535 & Cin); // Caudal de entrada g/s - Parte - sign
  Mb.R[9] = (Cin>>16); // Caudal de salida g/s - Parte + sign

  Mb.R[10] = (65535 & Cout); // Caudal de salida g/s parte - sign.
  Mb.R[11] = (Cout>>16); /// Caudal de salida g/s parte + sign.

  long Cinst = Cin-Cout;

  if (Cinst >= 100) {
    tmp=Cinst*3600;
    Mb.R[12] = 65535 & tmp; // consumo Instant. g/s parte - sign.
    Mb.R[13] = (tmp>>16); // Consumo Instant g/s parte + sign.
  }
  else {
    Mb.R[12] = 0; // consumo Instant. g/s parte - sign.
    Mb.R[13] = 0; // Consumo Instant g/s parte + sign.
  }

  unsigned long MotorON=1;//digitalRead(22); //MotorON;
  //digitalWrite(13,MotorON);
  Mb.R[14] = MotorON;

  if (Mb.C[0]==true)
    resetFunc();

  wdt_reset(); // punto de control Watchdog

} /// final de Loop

```

Rutina **loop()**

Rutina de atención a las interrupciones (Relojes 1 y 5):

Valga decir que ambas rutinas son similares en su ejecución, por lo cual se ha decidido explicar el funcionamiento sólo del Timer_1, pudiendo comprobar el lector en el anexo_1 que la rutina de atención a la interrupción del Timer_5 tiene idéntico comportamiento.

```

boolean retroc_in=0, retroc_out=0;

ISR(TIMER1_COMPA_vect) /// cuenta 10 pulsos, se resetea y entra aqui
{
  if (retroc_in){
    retin++;/// si esta en modo retroceso
    if (pulsos_ret_e != 0) {
      TCNT1 = TCNT1+pulsos_ret_e; // volcamos los pulsos anteriores
      pulsos_ret_e = 0; //limpiamos pulsos anteriores
    }
  }
  else {
    gin++;/// si esta en modo normal
    if (pulsos_ent != 0){
      TCNT1 = TCNT1+pulsos_ent;/// volcamos los pulsos anteriores
      pulsos_ent = 0; //limpiamos pulsos anteriores
    }
  }
}

```

Rutina de atención a la interrupción Timer_1

Al contar 10 pulsos significa que han pasado 100g de combustible a través del caudalímetro, momento en que se ejecuta esta rutina.

Lo primero que hacemos es comprobar si esta en modo retroceso o en modo normal y en función de ello incrementamos una u otra variable. Sin embargo, una vez hecho esto comprobamos que al cambiar de modo retroceso a modo normal y viceversa no se nos haya quedado ningún pulso por el camino, si es así, este queda guardado en las variables "pulsos_ret_e" y "pulsos_ent", lo que debemos hacer entonces es sumarlo a la cuenta actual "TCNTx" y borrarlo de esas variables.

Rutina de atención a la interrupción (Reloj 3):

```
ISR(TIMER3_COMPA_vect) // cuenta 1 segundo y calcula el caudal instantáneo

{
  if ((gin-ginpre) == 0)
    Cin=0;
  else
    Cin=round(((gin-ginpre)*500.0)+Cin)/2); // 100.0 es para que haga la división con decimales
    ginpre = gin;

  if ((gout-goutpre) == 0)
    Cout=0;
  else
    Cout=round(((gout-goutpre)*500.0)+Cout)/2); // 100.0 es para que haga la división con decimales
    goutpre = gout;
}
```

ISR TIMER_3

En el caso del reloj 3, éste está asociado al reloj interno del controlador, su objetivo es simplemente contar 0,2 segundos y calcular el caudal instantáneo. Como mencionamos anteriormente nuestro mayor interés radica en la precisión del caudal acumulado, debido a la configuración elegida, la única manera de calcular el caudal instantáneo con cierto grado de precisión es ésta.

Calculamos la diferencia en el caudal acumulado a intervalos de 0,2 segundos. Multiplicando esa cifra por 5 podemos hacer una buena aproximación del caudal instantáneo en Kg/s.

No obstante, dado que estamos haciendo una aproximación por redondeo, para mejorar la presentación de los datos y reducir el error absoluto, calculamos la media aritmética entre el valor actual y el anterior, presentando un valor intermedio.

Rutinas de Control de Interrupciones externas (Retrocesos):

En este caso también tenemos dos rutinas con funcionamientos similares, baste pues con explicar el funcionamiento de una de ellas.

```
volatile boolean semaforointerrupciones=false; // mediante esta variable evitamos conflicto entre las interrupciones externas 2 y 3

ISR(INT2_vect){

if (!semaforointerrupciones) {
    semaforointerrupciones=true; // tomamos el control del semaforo
    if (digitalRead(19)) {
        pulsos_ent=TCNT1;
        digitalWrite(13,1);
        retroc_in=true;
    }
    else {
        pulsos_ret_e=TCNT1;
        digitalWrite(13,0);
        retroc_in=false;
    }
    semaforointerrupciones=false; // soltamos el control del semaforo
}
}
```

ISR Int_2

Lo primero es establecer un semáforo para evitar conflictos de acceso a los importantísimos registros "TCNTx". Si salta esta interrupción es porque ha dado lugar un retroceso, en ese caso se toma el control del semáforo, se guardan los pulsos y se activa la variable "retroc_in" Indicando que los contadores deben girar en sentido inverso.

Cuando el Pulso vuelve a bajar, se entra otra vez por aquí, y esta vez se guardan los pulsos en sentido inverso y se vuelve al estado inicial.

6.5 Topología física de la red (LAN)

Para la transmisión de datos de los controladores a la estación se ha establecido una red de área local con **topología en estrella**.

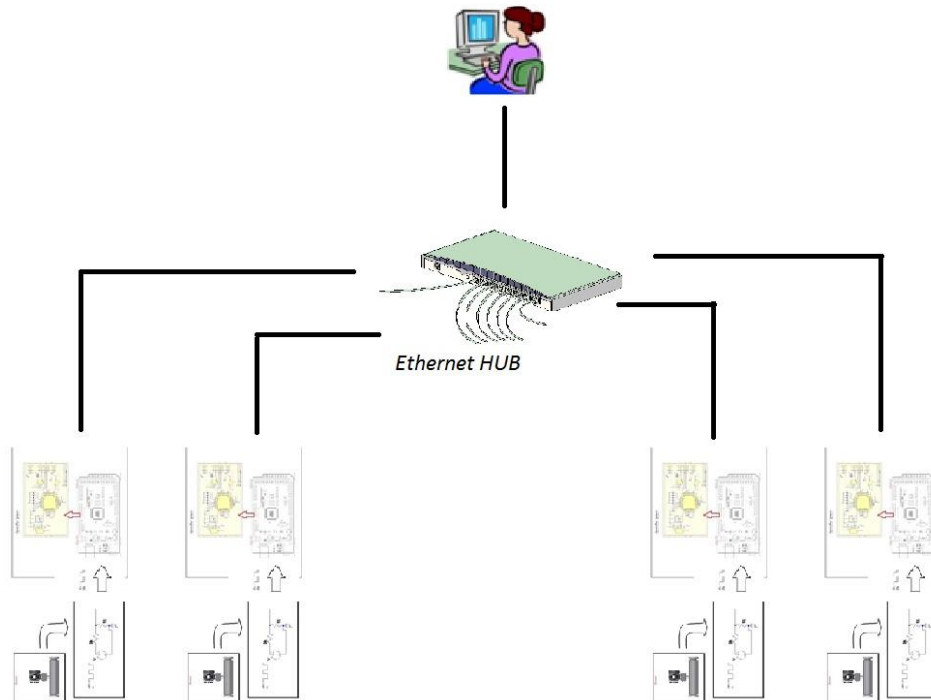


Figura 53 - Topología de red

Así, todos los controladores se conectarán a un concentrador (ethernet HUB), y este a su vez a la estación de trabajo en el centro del buque.

Capítulo 7.-

SISTEMA SCADA

En este capítulo abordaremos el desarrollo de una GUI (Graphical User Interface) utilizando para ello un potente sistema SCADA (software de control, supervisión y adquisición de datos)

7.1.- El sistema SCADA IGSS Free 50

IGSS FREE50 es una edición gratuita de IGSS V11 (Revisado en Mayo de 2014), software de control, supervisión y adquisición de datos (SCADA), El cual representa el estandarte de la corporación danesa Seven Technologies, recientemente adquirida por Shneider Electric. Se trata de una versión Gratuita con todas las funcionalidades del software comercial pero limitando el desarrollo de aplicaciones a un máximo de 50 objetos.

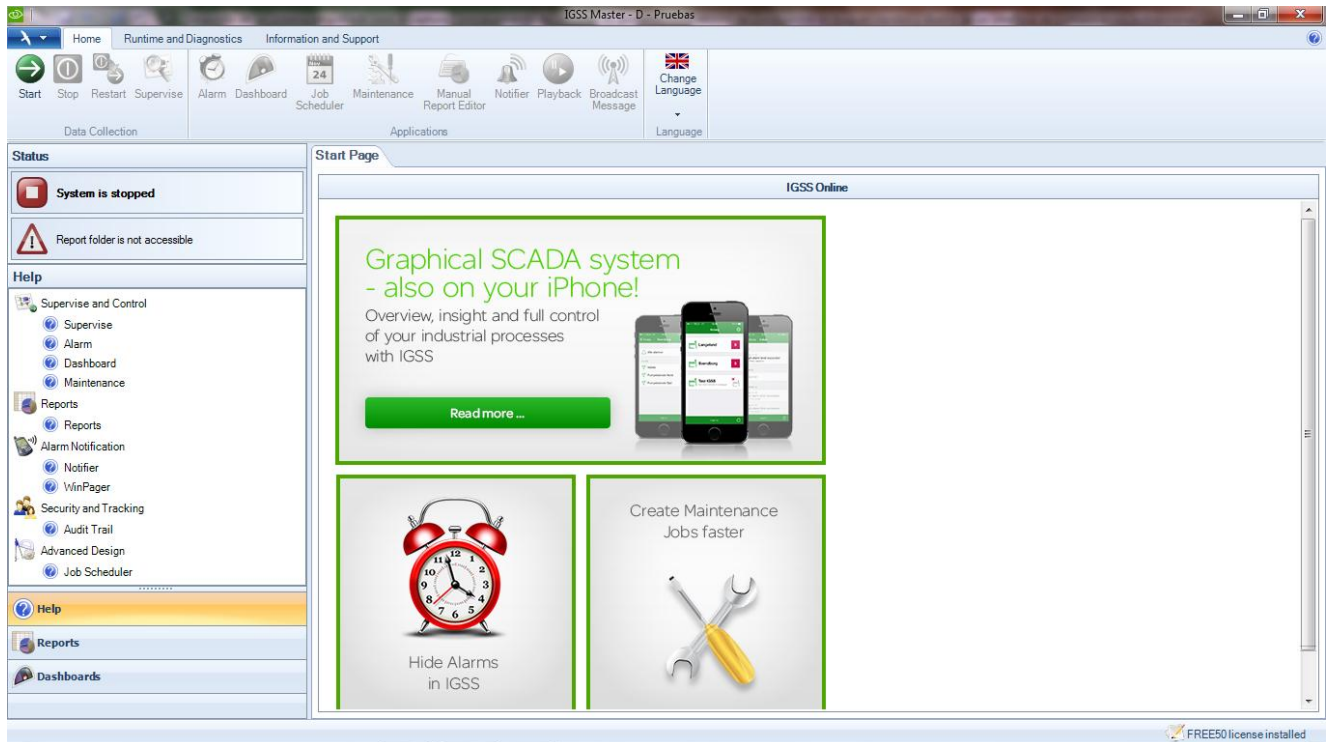


figura 54. Captura de pantalla de bienvenida IGSS (free 50)

Para nuestro cometido debemos desarrollar una interfaz de adquisición de datos y supervisión, estableciendo una comunicación "half-duplex" con nuestros cuatro controladores y utilizando el protocolo MODBUS/TCP. Para esto debemos configurar un Mestro MODBUS/TCP que sea capaz de requerir datos a los controladores y enviar comandos cuando sea necesario.

Para ello en el apartado configuración del sistema, creamos la interfaz lógica de nuestros controladores, IGSS nos facilita muchísimo esta tarea, aportándonos los drivers necesarios para la comunicación mediante el protocolo MODBUS/TCP.

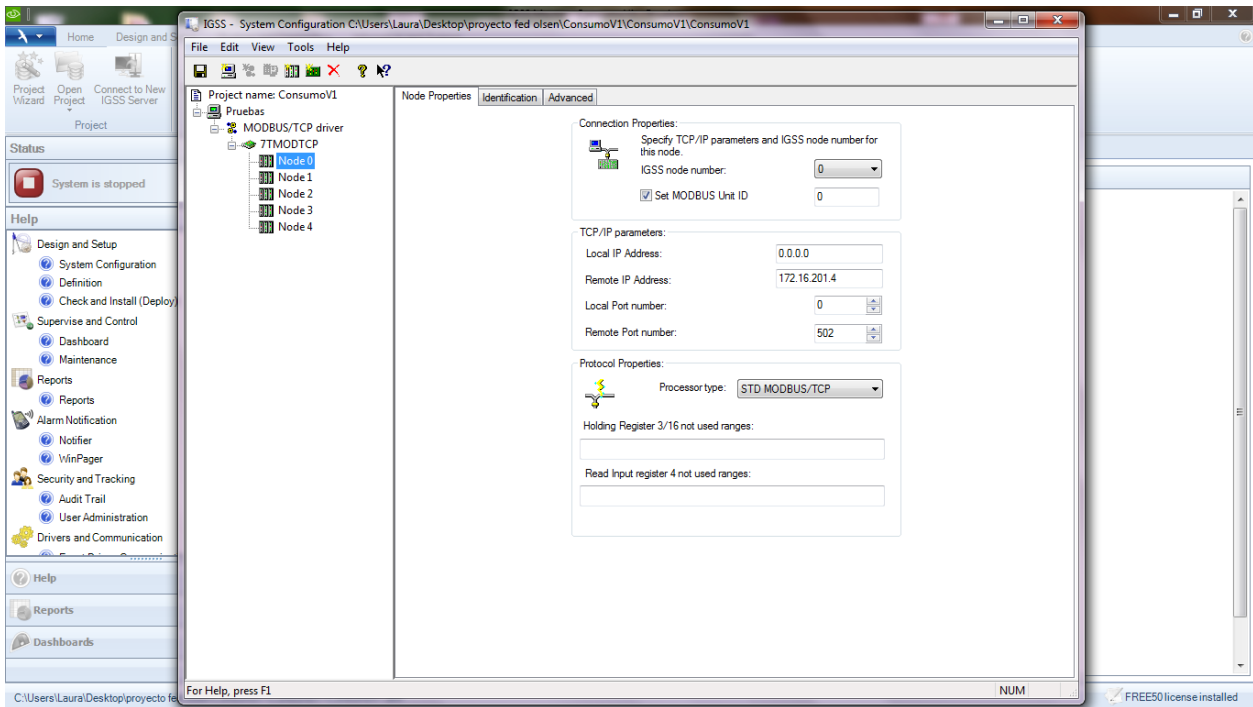


figura 55. captura pantalla de configuración de controladores

Una vez configurados los terminales esclavo, podemos acceder a ellos desde el programa mediante un direccionamiento lógico, abstrayéndonos del resto de parámetros de comunicación.

7.2.- Modos de direccionamiento

Para direccionar un registro del controlador primero debemos crear un descriptor al que asociarlo, una vez hecho esto en las propiedades del mismo, en la pestaña direccionamiento seleccionamos la dirección lógica de nuestro controlador.

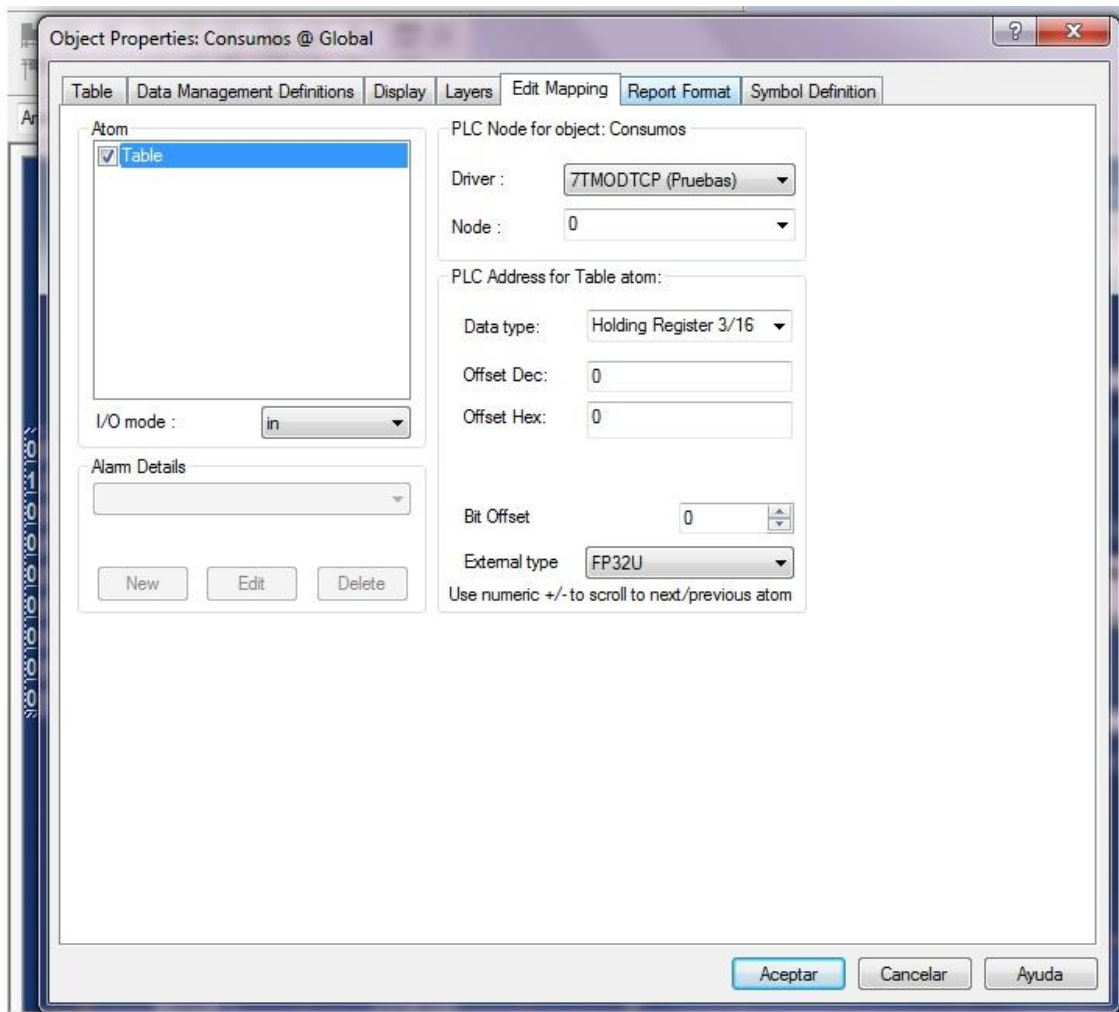


Figura 56. Direccionamiento De Objetos

7.4.- Diseño de un GUI(Graphical User Interface) para la representación de datos en pantalla con Scada de IGSS.

El sistema GUI para la presentación de datos en pantalla incluye control del consumo pormenorizado por motor, consumo total de la nave, indicadores de estado de marcha y reposo de la nave. indicadores de potencia, relación potencia-consumo, velocidad-consumo, consumo medio por milla náutica, y un contador de consumo parcial. Además mediante la entrada de la densidad del combustible, dato que se otorga al repostar, es capaz de calcular el consumo volumétrico.

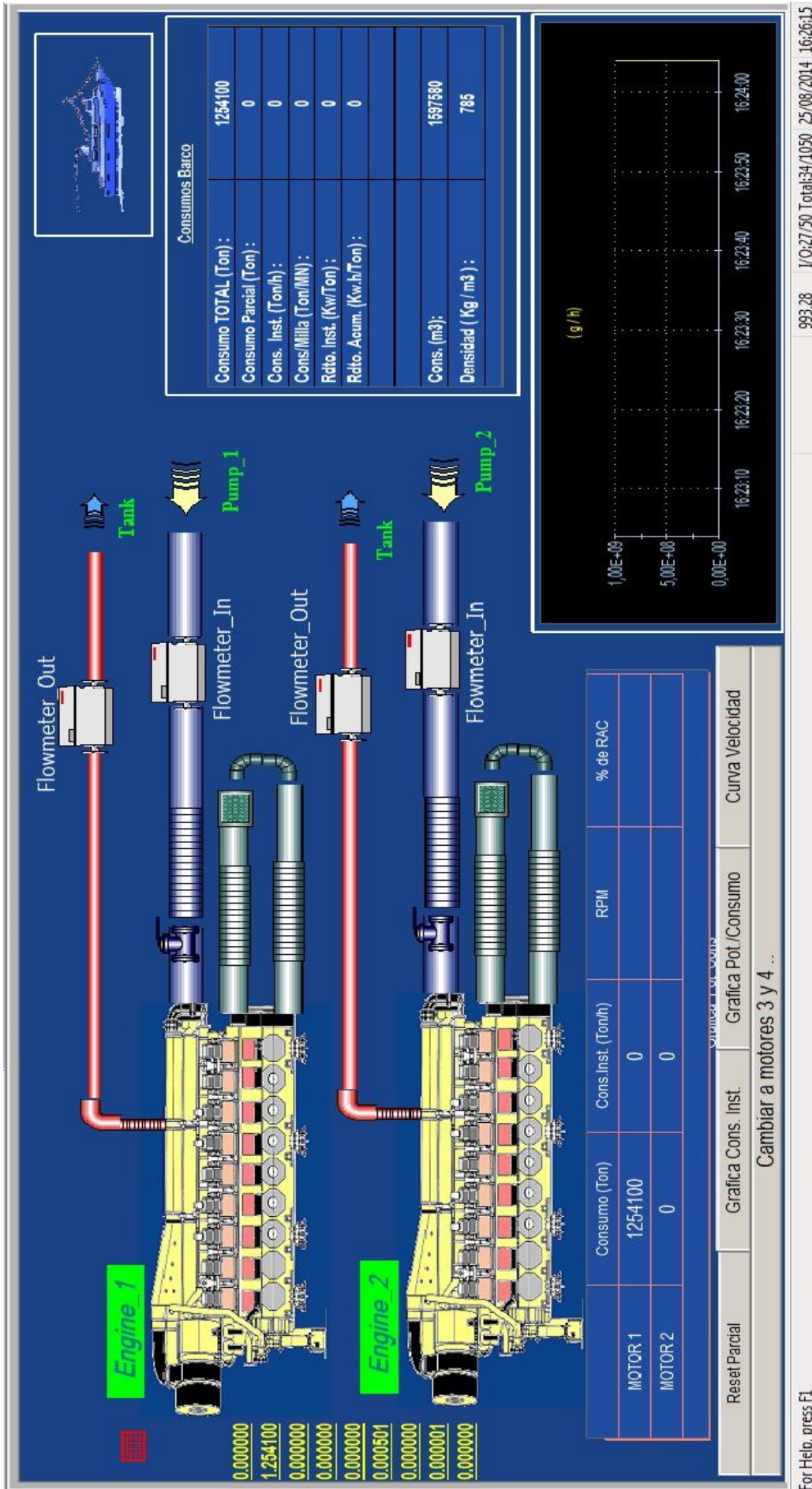


figura 57. Graphical User Interface

Los indicadores de estado de marcha y reposo, así como la velocidad del buque están disponibles actualmente en formato NMEA y estarán disponibles en formato MODBUS/TCP en el momento de la instalación. Los parámetros de potencia y RPM por motor, si bien están contemplados en el sistema GUI, no están aún disponibles en las naves, y se relegan a una segunda fase del proyecto, dado que requiere una importante inversión en sensores ópticos de potencia y torque, así como una compleja instalación.

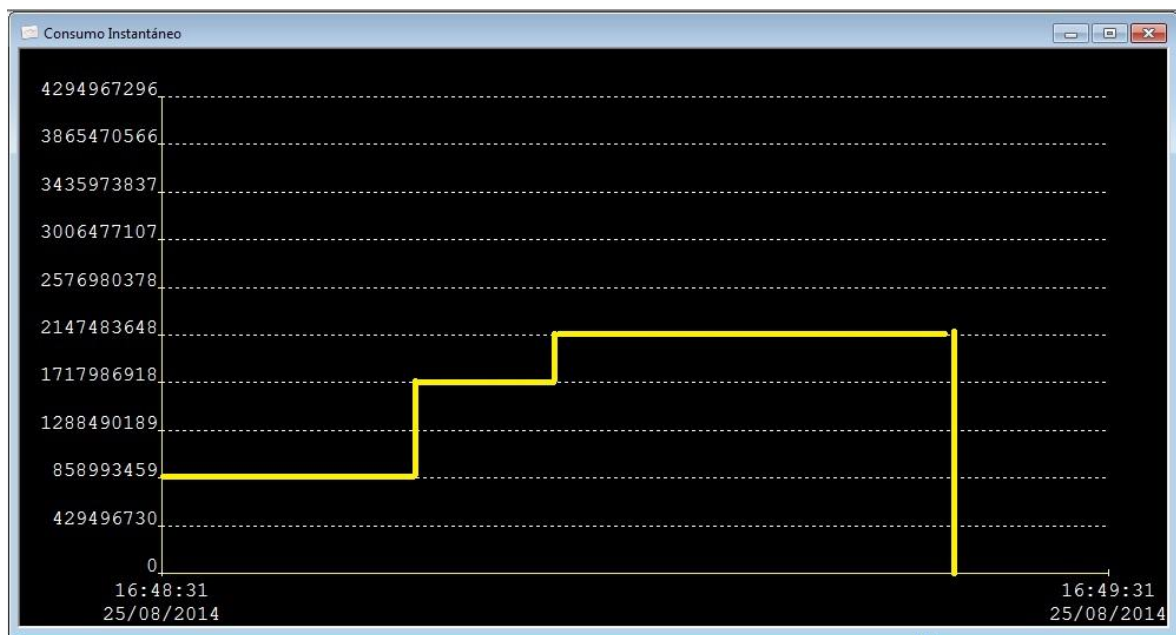


figura 59. Consumo Instantáneo

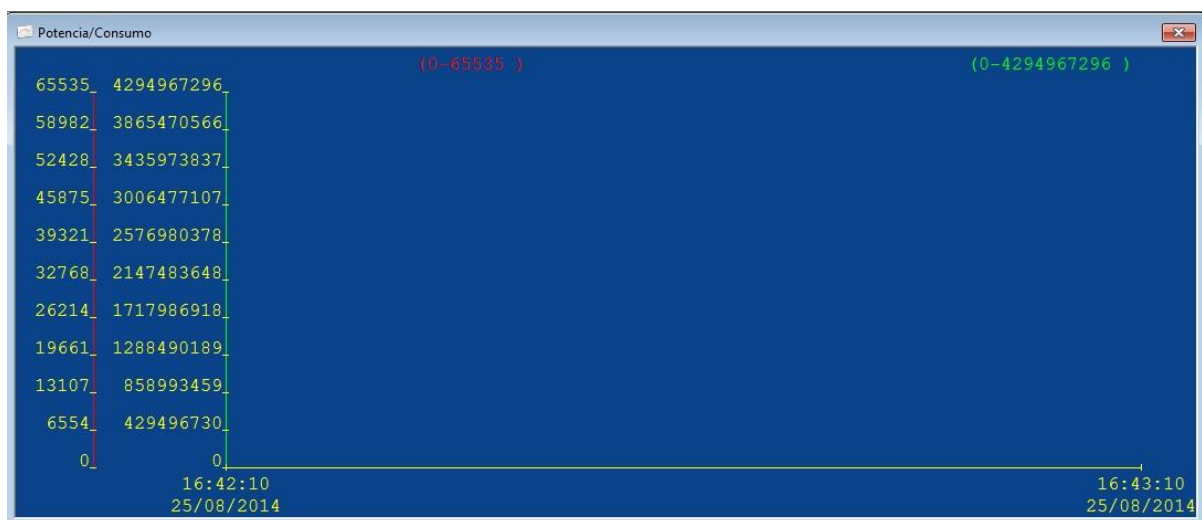


figura 58. Curva Potencia-Consumo

Capítulo 8.- PRUEBAS Y TEST

En este capítulo resumiremos las pruebas efectuadas sobre nuestro sistema de captación, procesamiento y visualización de datos a lo largo del proceso de diseño.

8.1.- Monitorización de las comunicaciones mediante WireShark

El software gratuito WireShark, es una potente herramienta de monitorización de redes y comunicaciones. es un software flexible capaz de detectar y trabajar con múltiples protocolos.

Mediante este software comprobamos que los mensajes son enviados y recibidos del operador a los controladores y viceversa, y que todo se lleva acabo de forma correcta.

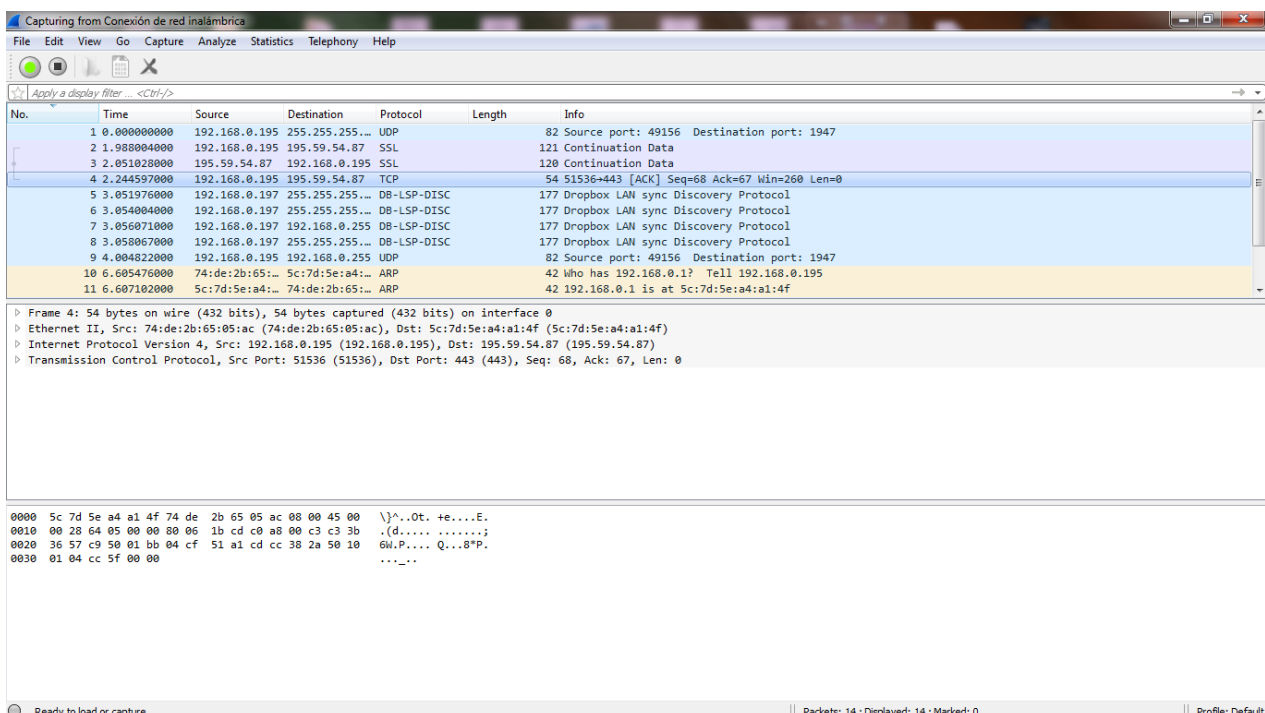


figura 60. captura pantalla wireshark

Dado que el prototipo aún no se encuentra instalado y aún no disponemos de los caudalímetros, para la realización de la pruebas se utilizará un generador de frecuencias portable Fluke Modelo 743B. Con esta herramienta simularemos el tren de pulsos generado por los caudalímetros.

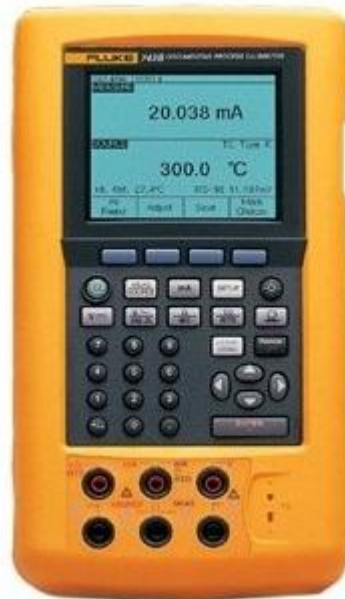


figura 61. Fluke 743B

En estas condiciones se llevaron a cabo la siguientes pruebas de funcionamiento:

- ✚ Pruebas de funcionamiento continuado durante 48 horas seguidas recopilando datos.

Resultados: No se registraron incidencias.

- ✚ Pruebas de estrés a la máxima frecuencia de operación durante diez minutos, recordar que la nave nunca operará en esos rangos de frecuencias.

Resultados: No se registraron Incidencias.

- ✚ Pruebas de funcionamiento también a ultra bajas frecuencias.

Resultados: Se hizo patente la necesidad de estimar el caudal Instantáneo cada 0.2 segundos como máximo. se observó que el salto en el redondeo genera un error que puede minimizarse estimando el valor mediante una media aritmética.

- ✚ Se han conmutado las entradas de retroceso en varias ocasiones para comprobar el correcto funcionamiento de la aritmética del sistema de retroceso.

Resultado: Se ha comprobado el correcto desempeño de la aritmética del sistema de retroceso.

- ✚ Se han generado retardos artificialmente para comprobar el desempeño del WatchDog.

Resultado: Se ha comprobado el correcto desempeño del watchdog.

- ✚ Se envió un numero discreto de pulsos a altas frecuencias, simultáneamente a cada contador para comprobar el conteo.

Resultado: Se ha comprobado el correcto desempeño del contador de pulsos, demostrando que todos eran contabilizados.

✚ Overflow de Registros: se ha forzado el Overflow en los contadores de pulsos

Resultado: Se ha comprobado la capacidad de los registros de reiniciar el contaje al llegar al overflow, con el desempeño normal de un barco, se dispone de una autonomía de nueve meses y medio antes de reiniciar la cuenta. Los periodos de control de este sistema son naturalmente mucho menores.

Capítulo 9.- DISCUSIÓN

9.1.- Resultados

Los datos arrojados por las pruebas de funcionamiento hacen prever un buen desempeño en todas las condiciones de trabajo. A excepción de las limitaciones impuestas por el sensor, la transmisión digital de los datos hacen posible minimizar la propagación del error, volcando datos con un error 2 órdenes de magnitud por debajo del umbral estipulado para el caudal acumulado.

9.2.- Líneas abiertas:

Una vez implementado el software del controlador, y el sistema GUI para la monitorización de los datos, aún queda pendiente la adquisición de los caudalímetros, la parte que requiere una inversión más fuerte, así como la instalación del sistema completo en los buques, y las pruebas de campo para verificar su correcto desempeño.

Una segunda fase del proyecto contemplaría un sistema para la monitorización de parámetros de potencia y RPM, lo que requeriría la implantación de sensores de potencia y torque en la transmisión del buque. Dicho sistema podría arrojar los datos necesarios (potencia, RPM y RAC) y ser conectado vía MODBUS/TCP a nuestro SCADA para contrastarlos con el consumo de combustible.

9.3.- Alternativas comerciales

Cabe destacar que las empresas proveedoras de los caudalímetros ofrecen sus propias soluciones para el control de consumo de combustible. Todas con salidas para la conexión vía **ModBus/TCP** y **sin incluir** el sistema **SCADA**, el cual debería ser contratado a alguna empresa desarrolladora.

Resumiremos brevemente las ofertas en la siguiente tabla:

Proveedor	Contoil	Emerson	Kröhne
Tipo de Caudal	Volumétrico(Medición Directa) ó Másico(Extrapolado)	Másico(Medición Directa) ó Volumétrico(Extrapolado)	Másico(Medición Directa) ó Volumétrico(Extrapolado)
Requiere Medición de Temperatura	Sí	Sí	Sí
Salidas	1 Salida Parametrizable	1 Salida Parametrizable	2 Salidas Parametrizables
Opciones Salida	Caudal Instantáneo ó Acumulado	Caudal Instantáneo ó acumulado	Caudal Instantáneo y/o Acumulado
Tipo Salida	Modbus/TCP	Modbus/TCP	Modbus/TCP
Sobrecoste en euros (Sobre el precio de los caudalímetros)	824,89	513,44	718,12

figura 62. Tabla de Alternativas Comerciales

Lo anterior deja de manifiesto que además de ser una solución mucho más versátil y el 'Know-How' que nos aporta el desarrollo de este proyecto, también es una alternativa económicamente más competitiva.

También conviene mencionar que aunque existen muchas ofertas en el mercado de dispositivos genéricos de control de consumo, las características de estos buques, morfología del circuito de combustible, consumo, máximos umbrales de pérdida de carga,..., hacen inapropiados a estos dispositivos.

Capítulo 10 .-

CONCLUSIONES

Para concluir revisaremos brevemente los hitos alcanzados a lo largo de este proyecto:

- ✚ Hemos estudiado y seleccionado el caudalímetro más apropiado considerando todas las especificaciones críticas de nuestro sistema.
- ✚ Se ha seleccionado el microcontrolador más apropiado basándonos en el coste, amplia implantación y funcionalidad.
- ✚ Se ha desarrollado el software necesario para el correcto desempeño del controlador
- ✚ Se ha desarrollado una interfaz gráfica de usuario basada en un sistema SCADA para hacer posible la monitorización del consumo de combustible en tiempo real, así como parámetros de potencia y velocidad del buque.

Finalmente, con el aporte de los datos recopilados y el trabajo de campo expuesto en la presente memoria, podemos concluir que el desarrollo de un sistema para la captura y procesamiento de caudales de alimentación y retorno de combustible en buques rápidos es factible, y se plantea como una buena solución para el control de consumos y el rendimiento de las operaciones llevadas a cabo en las naves.

Indirectamente, la adopción de sistemas de control es un paso esencial para la toma de decisiones que puedan reducir el consumo, aumentar la eficiencia de las maniobras, y en consecuencia aumentar la rentabilidad y sostenibilidad ambiental del transporte marítimo.

Chapter 10 .-

CONCLUSSIONS

To conclude we briefly review the landmarks achieved in this project:

- ✚ We have studied and selected the most appropriate flowmeter considering all critical parameters of our system.
- ✚ it has selected the most properly micro-controller. We based our decision in cost, broad range of application and functionality.
- ✚ Software needed for the good performance of the controller was developed.
- ✚ A graphical user interface was developed. It's based on a SCADA System and it makes possible to monitoring consumption, power and speed parameters in the ferry.

Finally, the contribution of the collected data and the fieldwork discussed herein, we conclude that it is feasible to develop a fuel flows monitoring and processing system, and it's seen like a good solution for consumption control and operations performance designed on ships.

Indirectly, adoption of control systems is an essential step for making decisions that may reduce consumption, increase maneuvers efficiency, and thus increase profitability and environmental sustainability of maritime transport.

Capítulo 11.-

BIBLIOGRAFÍA

- [1]. Directiva 2012/33/UE del parlamento europeo y del consejo, de 21 de noviembre de 2012 , por la que se modifica la directiva 1999/32/ce del consejo en lo relativo al contenido de azufre de los combustibles para uso marítimo
- [2]. Review Of Maritime Transport, 2013, UNCTAD
<http://unctad.org/es/paginas/pressrelease.aspx?originalversionid=166>
- [3]. Asociación De Navieros Españoles(ANAVE)
www.anave.es
- [4]. e-navales
www.enavales.com
- [5]. Directiva 2012/33/UE del parlamento europeo
- [6]. Lloyd's Register
<http://www.lr.org/>
- [7]. Maersk Sustainability Report 2013 - Maersk International, 2013
<https://my.maerskline.com/link/?page=news&path=/news/news20130318>
<http://www.maerskline.com/en-us/sustainability-progress-update-2013/trends-and-challenges>
- [8]. Plataforma Oceana
<http://oceana.org/es/eu/>

- [9]. Diesel Fuel Technical Review (Chevron,2007)

- [10]. Consumo De Energía Y Emisiones Asociadas Al Transporte Por Barco - grupo de investigación del transporte marítimo de la fundación de la Universidad de Oviedo, grupo gestor del proyecto ENERTRANS, 2008
www.enertrans.es

- [11]. Spec. Sheet Caterpillar Cat3618, Caterpillar,2006

- [12]. Product Data Sheet Micro Motion® F-Series Coriolis Flow And Density Meters, Emerson, 2013

- [13]. Product Configuration Manual Micro Motion® F-Series, Emerson, 2013

- [14]. Catalogo Oficial Contoil, 2014

- [15]. Hoja De Datos Técnica Optimass 1000, Kröhne, 2011

- [16]. <http://www.atmel.com/devices/atmega2560>

- [17]. 8-Bit Atmel Microcontroller With 64k/128k/256k Bytes In-System Programmable Flash, © 2012 Atmel Corporation.

- [18]. arduino.cc

- [19]. <http://maxembedded.com/>

- [20]. <http://www.avrbeginners.net/>

- [21]. <http://uhu.es/antonio.barragan/content/modbus-tcp>

- [22]. www.avr-tutorials.com

- [23]. <http://www.tecdigitaldelbajio.com>
- [24]. Red Modbus/Tcp Con Un Ordenador Y Cuatro Plc's, Miguel Ángel Tavera Farfán
<http://www.monografias.com/trabajos75/red-modbus-tcp-ordenador/red-modbus-tcp-ordenador2.shtml#ixzz3bm0xlds2>
- [25]. <http://www.jmnlab.com/interrupciones>
- [26]. Manual De Programación De Igss, Seven Technologies,2004
- [27]. <http://harperjiangnew.blogspot.com.es/>
- [28]. Servidor Modbus/TCP Con Microcontrolador Pic, Eric Alonso Lindell, Julio 2012

Anexo I

Anexo 1.-**PRESUPUESTO**

Se ha elaborado un presupuesto con el fin de otorgar una dimensión más concreta sobre el coste de implantación de este sistema en los buques. Se han tenido en cuenta los costes de cada uno de los componentes del sistema, así como la mano de obra necesaria y los gastos derivados de la instalación.

Presupuesto

Coste Aproximado de la instalación

Producto	Cantidad	Precio Unitario	Total
OPTIMASS 1300C Caudalímetro Másico Optimass, Krohne Descripción: KROHNE Mass Flow Sensor OPTIMASS 1000-S 25 Sistema de doble tubo recto de acero inoxidable AISI 318 ° Presión de proceso (máx. 40 bar @ 20°C/35.2 bar @ 130°C 580 PSI @ 68°F/510 PSI @ 266°F Temperatura del proceso -40°C to 130°C conexión DN 40 PN 40 según DIN 2501 Cara sellante Tipo B1 según EN 1092-1 Alojamiento Material Acero Inoxidable 304L Sin contenedor secundario de presión. Presión típica de rotura > 100 bar clase de protección IP 67 EN 60 529 Longitud de montaje 541 mm calibración calibr. lineal en 3 punt. del caudal distribuidos a lo largo del caudal nominal Tipo convertidor de señal MFC 300 C APPLICATION DATA Producto MDO Rango de Caudal 3.74 ... 75.6 l/min Densidad 890 kg/m ³ Viscosidad 11 mm ² /s Temperatura 60 °C Presión 3.5 bar g VEBH0800V00G000000000000	8	3403,08	27224,64
Arduino Mega 2560 R3 Descripción: Características: <ul style="list-style-type: none"> • Microcontrolador: ATmega2560 • Tensión de alimentación: 5V • Tensión de entrada recomendada: 7-12V • Límite de entrada: 6-20V • Pines digitales: 54 (14 con PWM) • Entradas analógicas: 16 • Corriente máxima por pin: 40 mA • Corriente máxima para el pin 3.3V: 50 mA • Memoria flash: 256 KB • SRAM: 8 KB • EEPROM: 4 KB • Velocidad de reloj: 16 MHz 	4	27,46	109,84
Estación de trabajo ASUS X551C B Descripción: PORTATIL ASUS X551C Procesador i3-3217 1.8 GHz Memoria Ram de 4096 GB Ram, Disco Duro de 500 HDD Sistema Operativo Windows 8 Pantalla de 15.6'' Dvdrw	1	446,49	446,46
Bridas acero inox, DN40, PN 40 según DIN 304L	16	0,76	12,16
Mano de Obra Estimada(Instalación) Incluye : Soldadura, Cableado, Conexiones, Puesta en marcha y pruebas	30	30	900
Cable Trensado Apantallado alpha wire	2,84	100	284
Hub Behringer Powerplay P16D Ultratnet	109	1	109
Cable Ethernet Carrete de 30 m.	1	148,53	148,53
Coste Total			29234,63

Anexo II

Anexo II.-

Código Fuente

```

/*          --- ANEXO 2 ---
*****
Trabajo de Fin de Grado

Título: Sistema empotrado para la captura y procesamiento de caudales de
       alimentación y retorno de combustible en buques rápidos
Autor: Jorge Sebastián Visser Ortiz
Fecha: 27/08/2014
Compañía: Fred Olsen S.a.

Instrucciones:

Este software ha sido desarrollado para la captura y procesamiento de
datos procedientes de Caudalímetros de alimentación y retorno
en buques de la neaviera Fred Olsen S.A.

Entradas: pulsos digitales (10 pulsos por kilogramos de combustible)
Entradas Modbus/TCP: Reinicio del Sistema.
Entrada Alarmas: Retroceso

Salidas: Modbus/TCP (caudales acumulados de alimentación y retorno,
Consumo Acumulado/Motor, Caudales instantáneos de Alimentación y
retorno, Consumo Instantáneo/Motor)

Se debe compilar y descargar en Arduino Mega 2560 con Ethernet Shield.

Conexiones:
        Conectar mediante puerto Ethernet a la Estación de Trabajo
        y conectar las entradas 38 y 47 a las salidas digitales de
        los caudalímetros. además las salidas de estado de retroce
        de ambos caudalímetros debe ser conectada a las entradas
        digitales 18(alimentación) y 19(retorno) .

Descripción:

Este programa calcula el consumo acumulado e instantáneo por motor conta
provenientes de los caudalímetros másicos de corioliss instalados a ese
circuitos de alimentación y retorno de los buques. Además envía estos dat
protocolo Modbus/TCP por puerto Ethernet.

Inserte Dirección IP en el Apartado IP Adress.

*/

```

```

#include <SPI.h>
#include <Ethernet.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>
#include "C:\Documents and Settings\Practicas.tecnico\Mis documentos\Ardui

////////////////////////////////// CONFIGURACIÓN LIBERÍA ETHERNET //////////////////////////////////

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };// La dirección MAC
// ser única por
// provocar conf:

IPAddress ip(172,16,201,4);// Insertar Aquí Dirección IP

EthernetServer server(80);// inicializa la libreria Ethernet server,
//se puede configurar otro puerto

////////////////////////////////// TERMINAMOS LA CONFIGURACIÓN DE CONEXIÓN ETHERNET //////////////////////////////////

////////////////////////////////// CREAMOS OBJETO MUDBUS, LIBRERÍA "mudbus.h" //////////////////////////////////

Mudbus Mb;

/*
//////////////////////////////////

cODIGOS DE FUNCIÓN ACCESO A TABLAS MODBUS

1 (LEER COIL'S)
3 (LEER REGISTROS)
5 (ESCRIBIR COILS)
6 (ESCRIBIR REGISTRO)

ACCESO A TABLAS

int Mb.R[0 to 125]
bool Mb.C[0 to 128]

```

```

PUERTO DE ACCESO: 502 (DEFINIDO EN Mudbus.h)

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
*/

void(* resetFunc) (void) = 0; // DECLARAMOS "*resetFunc" COMO UN PUNTERO A DIRECCIÓN 0 DE LA MEMORIA DE

void setup() {

  wdt_disable(); // deshabilitamos el WatchDog

  // Inicializamos los contadores T1 y T5

  cli(); // Deshabilitamos Interrupciones

  TCCR1A = 0; // Ponemos todos los registros a cero
  TCCR1B = 0;
  TCCR5A = 0;
  TCCR5B = 0;
  TCCR3A = 0;
  TCCR3B = 0;
  TCNT5 = 0;
  TCNT1 = 0;
  TCNT3 = 0;

  // seleccionamos los límites del conteo

  OCR1A = 9; // Cuenta 10 pulsos (0-9) y resetea el contaje y salta la interrupción
  OCR5A = 9;
  OCR3A = 3125;

  // Cambiamos los timers al modo COMPare Match, y los configuramos como
  // contadores externos

  TCCR1B |= 15; // WGM12
  TCCR5B |= 15; // WGM12

  // Timer 3 en modo compare match pero con reloj interno (Temporizador)
  // Escalamos a 15625 cuentas por segundo (16Mhz/1024)

```

```

TCCR3B      |=13;//WGM12

// Habilitamos las interrupciones de los timers por compare match

TIMSK1 |= (1 << OCIE1A);
TIMSK5 |= (1 << OCIE5A);
TIMSK3 |= (1 << OCIE3A);

sei();      //Habilitar todas las interrupciones

pinMode(19, INPUT);
pinMode(18, INPUT);
pinMode(47, INPUT);
pinMode(38, INPUT);

////////// Terminamos Set up de los contadores //////////

digitalWrite(19,LOW); //con esto nos cuidamos de que las entradas esten
digitalWrite(18,LOW); // cuando no están conectadas convendría quitarla:
digitalWrite(47,LOW); // cuando las entradas esten conetadas, para evita:
digitalWrite(38,LOW);

// definimos las interupciones que controlan el Retorno

// ~ significa complemento a 1 del resultado, es decir negamos el resul:

EICRA &= ~(1<<ISC21);// activamos la INT2 asincrona, ponemos un 0
EICRA |= (1<<ISC20);// ponemos un 1 en el bit isc20
EICRA &= ~(1<<ISC31);//activamos la INT3 asincrona, ponemos un 0
EICRA |= (1<<ISC30);// ponemos un 1 en el bit isc30
EIMSK |= ((1<<INT2)|(1<<INT3));//Habilitar interrupciones ext
sei();//Habilitar todas las interrupciones

// terminamos interrupciones

pinMode(13, OUTPUT); // habilitamos pin 13 para monitorizar las interrupc:

// Inicializamos Conexion Modbus/TCP

Serial.begin(9600); // abrimos el puerto serial
Ethernet.begin(mac, ip); // inicializamos comunicación ethernet

```



```

server.begin(); // abrimos la comunicación

pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(13, OUTPUT);
pinMode(22, INPUT);

// Terminamos Set Up de modbus/TCP

wdt_enable(WDTO_4S); // Habilitamos WatchDog

} /// final del Setup

volatile unsigned long gin,retin=0; // para trabajar con interrupciones
volatile unsigned long gout,retout=0; // usar variables de acceso directo
volatile unsigned long parcial=0;
volatile unsigned long Cin,Cout,ginpre,goutpre = 0;
volatile byte pulsos_ret_e=0;
volatile byte pulsos_ret_s=0;
volatile byte pulsos_ent=0;
volatile byte pulsos_sal=0;

/* como modbus traba con registros de 8 bits debemos cortar ls registros
controlador y enviarlos en 2 trozos */

void loop() {
  unsigned long tmp1,tmp2,tmp=0;
  Mb.Run();
  if (gin > retin)/// si el retroceso es mayor que el consumo
    {/// ej. arranque, consumo debe ser 0
      tmp1 = (gin-retin)*100;/// y permanecerá a cero hasta que que
    }/// el consumo acumulado supere el retroceso
  else
    {/// momento en que empezará a contar,
      tmp1=0;/// manteniendo siempre el registro del f.
  }
  Mb.R[0] = 65535 & tmp1; // g entrada Parte Menos Significativa
  Mb.R[1] = (tmp1>>16); // g entrada Parte más Significativa

  if (gout > retout)/// si el retroceso es mayor que el consumo
    {/// ej. arranque, consumo debe ser 0
      tmp2 = (gout-retout)*100;
    }
}

```

```

    }
else
    tmp2=0;
Mb.R[2] = 65535 & tmp2; // g Salida Parte - significativa
Mb.R[3] = (tmp2>>16); // g Salida Parte + significativa

if (tmp1 > tmp2) // si la salida es mayor que la entrada es por
    tmp = tmp1-tmp2;
else
    tmp=0;

Mb.R[4] = 65535 & tmp; // Consumo Total Acumulado g parte - sign.
Mb.R[5] = (tmp>>16); // Consumo Total Acumulado g parte + sign.

if (Mb.C[0])
    parcial=tmp; // Reset del Parcial
tmp=tmp-parcial; // restamos el parcial fijado con el boton de reset ;
Mb.R[6] = 65535 & tmp; // Consumo Parcial Acumulado g parte - sign.
Mb.R[7] = (tmp>>16); // Consumo Parcial Acumulado g parte + sign.

Mb.R[8] = (65535 & Cin); // Caudal de entrada g/s - Parte - sign
Mb.R[9] = (Cin>>16); // Caudal de salida g/s - Parte + sign

Mb.R[10] = (65535 & Cout); // Caudal de salida g/s parte - sign.
Mb.R[11] = (Cout>>16); // Caudal de salida g/s parte + sign.

long Cinst = Cin-Cout;

if (Cinst >= 100) {
    tmp=Cinst*3600;
    Mb.R[12] = 65535 & tmp; // consumo Instant. g/s parte - sign.
    Mb.R[13] = (tmp>>16); // Consumo Instant g/s parte + sign.
}
else {
    Mb.R[12] = 0; // consumo Instant. g/s parte - sign.
    Mb.R[13] = 0; // Consumo Instant g/s parte + sign.
}

unsigned long MotorON=1; //digitalRead(22); //MotorON;
//digitalWrite(13,MotorON);
Mb.R[14] = MotorON;

if (Mb.C[0]==true)
    resetFunc();

```

```

    wdt_reset();// punto de control Watchdog

}    ///    final de Loop

boolean retroc_in=0, retroc_out=0;

ISR(TIMER1_COMPA_vect) /// cuenta 10 pulsos, se resetea y entra aqui
{
if (retroc_in){
    retin++; // si esta en modo retroceso
    if (pulsos_ret_e != 0) {
        TCNT1 = TCNT1+pulsos_ret_e; // volcamos los pulsos anteriores
        pulsos_ret_e = 0; //limpiamos pulsos anteriores
    }
}
else {
    gin++; // si esta en modo normal
    if (pulsos_ent != 0){
        TCNT1 = TCNT1+pulsos_ent; // volcamos los pulsos anteriores
        pulsos_ent = 0; //limpiamos pulsos anteriores
    }
}
}

ISR(TIMER5_COMPA_vect) // igual que con el timer 1 pero para el T5 mide caudal
{
if (retroc_out){
    retout++;
    if (pulsos_ret_s != 0){
        TCNT5 = TCNT5+pulsos_ret_s;
        pulsos_ret_s = 0;
    }
}
else {
    gout++;
    if (pulsos_sal != 0){
        TCNT5 = TCNT5+pulsos_sal;
        pulsos_sal = 0;
    }
}
}

```

```

}

ISR(TIMER3_COMPA_vect) // cuenta 0.2 segundos y calcula el caudal instan

{
if ((gin-ginpre) == 0)
  Cin=0;
else
  Cin=round((((gin-ginpre)*500.0)+Cin)/2); // 100.0 es para convertir a
ginpre = gin;// y para que haga la división con decimales, si no
      /// multiplicamos por cinco para obtener el caudal en g/s
if ((gout-goutpre) == 0)
  Cout=0;
else
  Cout=round((((gout-goutpre)*500.0)+Cout)/2); // 100.0 es para que haga
goutpre = gout;
}

volatile booleansemaforointerrupciones=false; // mediante esta variable
// conflicto entre las interrupciones externas 2 y 3 que podrían bloquear
// contadores debido a rebotes o cosas extrañas

ISR(INT2_vect){

if (!semaforointerrupciones) {
  semaforointerrupciones=true; // tomamos el control del semaforo
  if (digitalRead(19)) {
    pulsos_ent=TCNT1;
    digitalWrite(13,1);
    retroc_in=true;
  }
  else {
    pulsos_ret_e=TCNT1;
    digitalWrite(13,0);
    retroc_in=false;
  }
  semaforointerrupciones=false; // soltamos el control del semaforo
}
}

```

```
ISR(INT3_vect) {  
  
    if (!semaforointerrupciones) {  
        semaforointerrupciones=true; // tomamos el control del semaforo  
        if (digitalRead(18)) {  
            pulsos_sal=TCNT5;  
            digitalWrite(13,1);  
            retroc_out=true;  
        }  
        else {  
            pulsos_ret_s=TCNT5;  
            digitalWrite(13,0);  
            retroc_out=false;  
        }  
        semaforointerrupciones=false; // soltamos el control del semaforo  
    }  
}
```