

**SISTEMA ELECTRONICO DE MONITOREO Y CONTROL PARA MOVIMIENTO
DE VOLUMENES DE AGUA EN UN LABORATORIO DE AGUAS DE BAJO COSTO
USANDO UNA INTERFAZ WEB Y UN SERVIDOR EMBEBIDO USANDO EL
SHIELD ETHERNET BASADO EN EL CI WI5100 DE WIZNET**

FAVIAN OCAMPO ZAPATA - 0T720121022

**UNIVERSIDAD CATOLICA DE MANIZALES
FACULTAD DE INGENIERIA
PROGRAMA INGENIERIA TELEMATICA
MANIZALES - CALDAS
2015-2**

CONTENIDO DEL INFORME

Título de proyecto	
Resumen.....	
Introducción.....	
Planteamiento del problema.....	
Justificación.....	
Objetivos.....	
Referentes teóricos	
Arduino	
Controlador CI WI5100 DE WIZNET	
Arduino Shield Ethernet	
Electrovalvulas	
Motobombas	
Metodología	
Cronograma	
Bibliografía	

TITULO DE PROYECTO

SISTEMA ELECTRONICO DE MONITOREO Y CONTROL PARA MOVIMIENTO DE VOLUMENES DE AGUA EN UN LABORATORIO DE AGUAS DE BAJO COSTO USANDO UNA INTERFAZ WEB Y UN SERVIDOR EMBEBIDO USANDO EL SHIELD ETHERNET BASADO EN EL CI WI5100 DE WIZNET.

RESUMEN

El movimiento de aguas ha sido una necesidad en ambientes tales como: laboratorios de aguas, sistemas de recolección de agua lluvia, represas y acueductos.

En este proyecto, se busca proponer un prototipo usando lo aprendido en la carrera ingeniería telemática.

La propuesta consiste en diseñar un prototipo integrando un sensor de nivel de agua por cada tanque del sistema, un sistema de reglas para el estado de los diferentes niveles de los diferentes tanques y un conjunto de actuadores que se implementan a partir de motobombas de diferentes caudales dependiendo de la altura que sea necesaria y de acuerdo a las capacidades de las motobombas disponibles.

A pesar de que ya existen prototipos de medición de niveles en diferentes proyectos, la integración en un sistema de control realimentado con afectación a moto válvulas o electroválvulas es muy importante y es un aspecto inexplorado salvo por los acueductos, los cuales tienen sistemas basados en el control de flujo de agua de acuerdo a la gravedad y a las necesidades de consumo de agua de las comunidades a afectar, pero no emplean retorno hacia los tanques superiores.

El presente proyecto no utiliza moto válvulas ni electroválvulas proporcionales sino más bien electroválvulas con solenoide normalmente cerradas, para controlar el flujo de caudal (flujo masivo) de acuerdo a la ley de la gravedad, y motobombas controladas por electrónica para garantizar los flujos de subida hacia tanques superiores.

La alimentación de los componentes electrónicos y la comunicación por Ethernet debe ser provista por transformadores y fuentes de alimentación estables, de acuerdo a los actuadores propuestos.

.

Palabras Clave: Telemetría, Monitoreo, Alarma, Prevención, Laboratorio de Mecánica de fluidos e Hidráulica, agua, nivel, arduino.

INTRODUCCIÓN

En los sistemas que tienen que ver con movimiento de líquidos entre diferentes tanques y /o contenedores es muy importante tener control y monitoreo de:

- Niveles de cada tanque
- Volúmenes
- Rata Másica
- Sensores de nivel
- Controladores (on/off, proporcionales, PID)
- Actuadores (electrovalvulas, motobombas)

El presente trabajo de grado presenta una metodología de diseño a partir de diagramas de bloque de flujo de líquidos (agua para este caso) dado que se inició como un proyecto de control del laboratorio de aguas pero se presenta como un modelo que debe llevar a un prototipo genérico para reutilizar aguas en un sistema cerrado.

Entendiendo un sistema cerrado para movimiento de aguas, un sistema en el cual es posible volver a llevar agua hacia niveles superiores del sistema (tanques) por medio del uso de motobombas, las cuales se encuentran en el mercado con diferentes precios y especificaciones técnicas.

PLANTEAMIENTO DEL PROBLEMA

No existe una metodología genérica para el diseño de sistemas de movimiento de flujo másico a una rata determinada por parte los laboratorios de aguas.

Por tal motivo los desperdicios de agua son altos en las empresas o laboratorios que implementan dichos laboratorios o sistemas.

No solamente es un problema existente a nivel de laboratorios, los mismos sistemas de recolección de aguas lluvias no tienen una metodología de control y monitoreo de los flujos másicos.

Si se lograra integrar tecnologías para diseñar un sistema electromecánico con control por software embebido que tenga como objetivo principal reducir las pérdidas de líquido que existen actualmente. Mejorando el servicio en cuanto a la gestión de la información sobre el comportamiento de los diferentes elementos del sistema a nivel individual, para un sistema con un flujo de volumen que se debe mantener constante, entonces se podría prevenir en mayor porcentaje las pérdidas materiales ocasionadas por desperdicios.

JUSTIFICACIÓN

Con el panorama ya descrito, Se observa la pérdida de líquido (agua) en diferentes sistemas hidráulicos tales como los sistemas de protección de agua lluvia, o el mismo sistema hidráulico de una casa, es tan importante la pérdida de agua y tan sensible como variable ambiental, que inclusive existe un indicador manejado por las diferentes autoridades ambientales de Colombia se llama: Huella Hídrica

Desarrollar una metodología para la el desarrollo de prototipos de reutilización de agua en laboratorios empresas y hogares disminuiría el valor de la huella hídrica por persona, hogar e institución, es decir por la sociedad en su conjunto. Y en momentos como el actual donde el proceso cíclico de fenómenos climáticos como el fenómeno del niño, hacen de los recursos hídricos, recursos escasos, es pertinente plantear un prototipo que permita disminuir la huella hídrica, por medio de la reutilización de las aguas en la medida que se pueda y que tenga la instrumentación básica necesaria, utilizando tarjetas programables programando las reglas propias de cada sistema.

OBJETIVOS

Objetivo General

Diseñar y construir un sistema de monitoreo y control electrónico para sistemas de movimiento de agua, a través de una interfaz web y utilizando un servidor embebido para los actuadores que permita tener control de los flujos entre los diferentes nodos de almacenamiento de agua del sistema.

Objetivos Específicos

- Proponer un prototipo de movimiento controlado de caudales de agua por medio de un controlador programado.
- Usar el protocolo TCP/IP (tarjeta shield ethernet) para controlar un sistema de flujo de lazo cerrado con un sensor de nivel por tanque y un controlador tanto de entrada como de salida para cada tanque.
- Diseñar un bloque funcional que pueda ser utilizado en la creación de modelos y prototipos más complejos y útiles para diferentes necesidades (hogar, aguas lluvias, recolección de aguas, tanques comunales, laboratorios de aguas etc)

REFERENTES TEORICOS

ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.



El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280 y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa lenguaje de programación processing/Wiring y el cargador de arranque que es ejecutado en la placa.

Se programa en el ordenador para que la placa controle los componentes electrónicos.

Desde octubre de 2012, Arduino se utiliza también con microcontroladoras CortexM3 de ARM de 32 bits, que coexistirán con las más limitadas, pero también económicas AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar con el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. Eso sí, las

microcontroladoras CortexM3 usan 3,3V, a diferencia de la mayoría de las placas con AVR, que generalmente usan 5V. Sin embargo, ya anteriormente se lanzaron placas Arduino con Atmel AVR a 3,3V como la Arduino Fio y existen compatibles de Arduino Nano y Pro como Meduino en que se puede conmutar el voltaje.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador.

También cuenta con su propio software que se puede descargar de su página oficial que ya incluye los drivers de todas las tarjetas disponibles lo que hace más fácil la carga de códigos desde el computador.

.

Controlador CI W5100 DE WIZNET

El chip W5100 es un controlador de Ethernet incorporado Cableado de TCP / IP que permite el fácil conexión a Internet para sistemas embebidos.



W5100 se adapta a los usuarios que necesitan una conectividad estable a Internet mejor, utilizando un solo chip para implementar la pila de protocolos TCP / IP 10/100 Ethernet MAC y PHY. Pila TCP / IP cableada compatible con TCP, UDP, IPv4, ICMP, ARP, IGMP, y PPPoE. El W5100 utiliza un 16 bytes buffer interno como su memoria de comunicación de datos.

Mediante el uso de W5100, se puede implementar la aplicación de Ethernet que necesitan utilizando un programa de toma simple en lugar de manejar un controlador de Ethernet complejo.

Es posible utilizar 4 zócalos de hardware independientes simultáneamente. BUS (directa e indirecta) y SPI (Serial Peripheral Interface) se proporcionan para una fácil integración con el MCU externa.

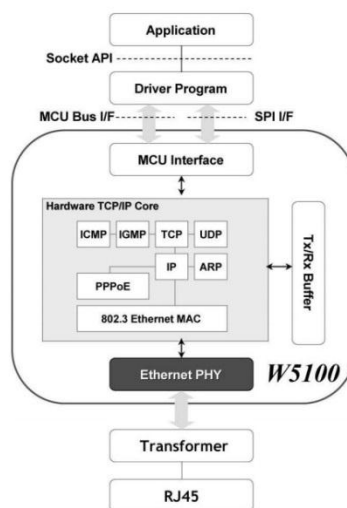
Características principales

- Protocolos TCP / IP cableadas: TCP, UDP, ICMP, IPv4, ARP, IGMP,

PPPoE

- Motor de hardware de red Un-atacable para la prevención de ataques de red tales como inundaciones, spoofing, inyección
- Host Interface: Directo / Indirecto Autobuses y Serial Peripheral Interface (SPI MODO 0, 3)
- 4 Toma de hardware independiente
- Memoria 16Kbytes Interna para el procesamiento de paquetes TCP / IP
- 10BaseT / 100Base TX Ethernet PHY Embedded
- Soporte Auto-Negociación (Full & Half Duplex, 10 y 100 Based)
- Soporte Auto-MDIX
- 3.3V Operación con la tolerancia de la señal de E / S 5V I
- Salidas LED (TX, RX, semidúplex / full, colisión, enlace, velocidad)
- Paquete sin plomo 80LQFP (10x10mm)

Pila de protocolos de CI WI5100 DE WIZNET



ARDUINO SHIELD ETHERNET

La Arduino Ethernet Shield permite que un Arduino se conecte a Internet. Esta Shield está basada en el chip Wiznet W5100, que proporciona un enlace de red (IP) tanto por TCP como por UDP. La Arduino Ethernet Shield soporta la conexión de hasta cuatro elementos de forma simultánea. Puedes usar la librería de ethernet para escribir programas que se conecten a Internet usando esta Shield.



La Ethernet Shield se conecta a una placa Arduino mediante el largo terminal macho situado en sus laterales. Esto mantiene el esquema de pines intacto y permite enchufar otra Shield encima de ella. La última revisión de esta Shield incluye una ranura para tarjetas microSD, que puede usarse para almacenar archivos a los que se podrá acceder a través de la red. Es compatible con la Arduino UNO y Mega 2560 (utilizando la librería Ethernet que incluye la versión 0019 del software de Arduino). Todavía no se incluye una librería para a tarjeta

micro SD en la distribución estándar pero la librería sdfatlib de Bill Greiman funciona muy bien.

La última revisión de esta Shield también incluye un controlador de reset automático para asegurar que el módulo Ethernet W5100 esté correctamente reseteado al iniciarse. Las revisiones anteriores no eran compatibles con la Arduino Mega 2560 y es necesario resetearlas manualmente después del encendido. La revisión original de la Shield incorporaba una ranura para tarjetas SD de tamaño estándar. Esta ya no es compatible.

Arduino se comunica tanto con el módulo W5100 como con la tarjeta microSD utilizando el bus SPI (a través de la clavija ICSP). Es decir, a través de los pines 11, 12 y 13 de la UNO y los pines 50, 51 y 52 de la Mega 2560. En ambas placas el pin 10 se usa para seleccionar el módulo W5100 y el pin 4 para la tarjeta microSD. Estos pines no pueden ser usados para E/S genéricas. En la Arduino Mega 2560 el pin SS (53) no se utiliza para seleccionar ni el módulo W5100 ni la tarjeta microSD. Este pin SS de la Mega 2560 debe mantenerse como salida o la interfaz SPI no funcionará.

Se debe tener en cuenta que como el módulo W5100 y la tarjeta microSD comparten el bus SPI, solo uno de ellos puede estar activo. Si estás usando los dos periféricos en el programa, estos deben ser atendidos por la librería correspondiente.

La Arduino Ethernet Shield proporciona un conector Ethernet RJ45 estándar.

El botón de reset en la Shield resetea tanto el módulo W5100 como la placa Arduino

ELECTROVALVULAS

Una **electroválvula** es una válvula electromecánica, diseñada para controlar el paso de un fluido por un conducto o tubería. La válvula se mueve mediante una bobina solenoide. Generalmente no tiene más que dos posiciones: abierto y cerrado, o todo y nada. Las electroválvulas se usan en multitud de aplicaciones para controlar el flujo de todo tipo de fluidos.



No se debe confundir la electroválvula con válvulas motorizadas, en las que un motor acciona el mecanismo de la válvula, y permiten otras posiciones intermedias entre todo y nada.

Clases de electrovalvulas

Una electroválvula tiene dos partes fundamentales: el solenoide y la válvula. El solenoide convierte energía eléctrica, mediante magnetismo, en energía mecánica para actuar la válvula.

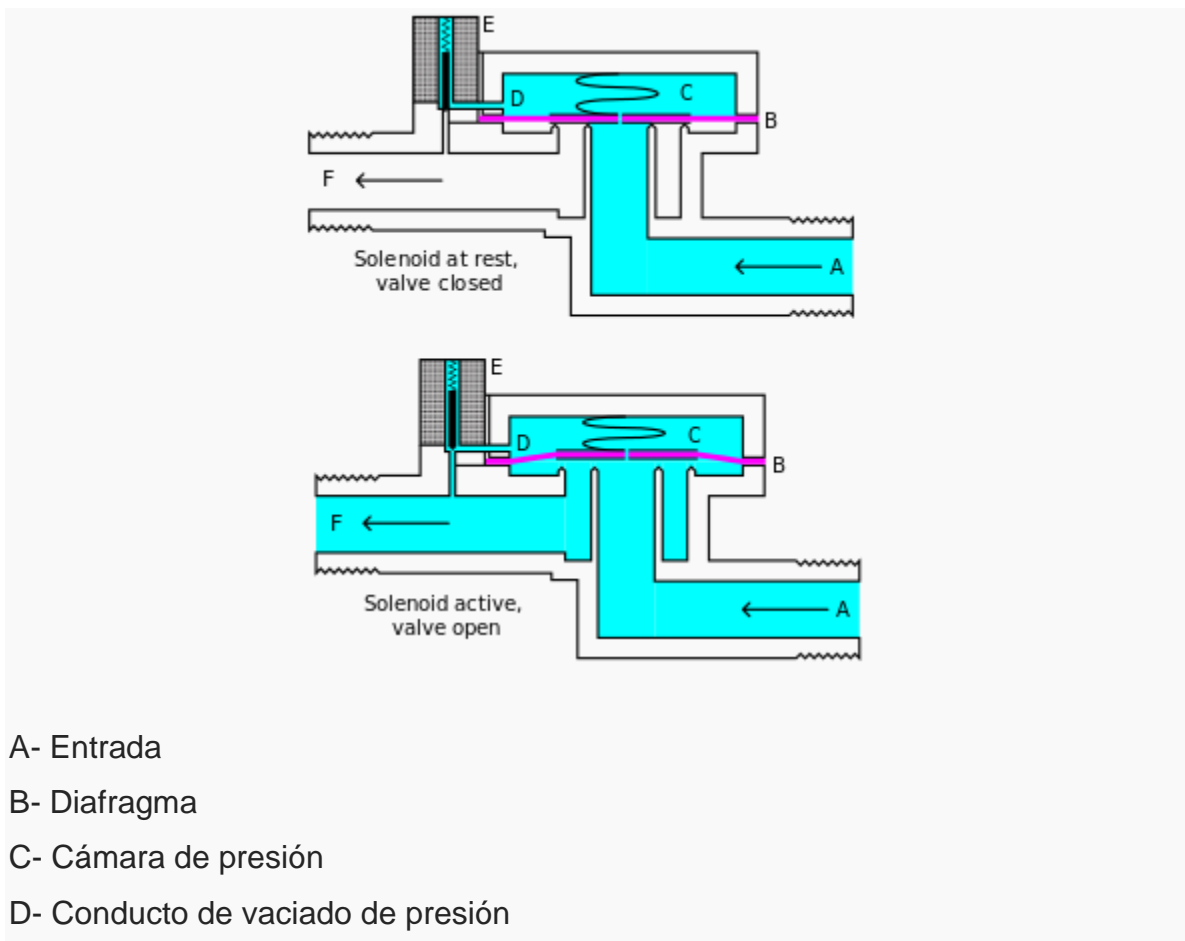
Existen varios tipos de electroválvulas. En algunas electroválvulas el solenoide actúa directamente sobre la válvula dando la energía necesaria para su movimiento. También es posible construir electroválvulas biestables que usan un solenoide para abrir la válvula y otro para cerrar o bien un solo solenoide que abre con un impulso de corriente y cierra con el siguiente. Estas tienen dos contactos eléctricos, de modo que al cambiar de posición la válvula abre uno de ellos y cierra el otro.

Electrovalvulas sencillas

Las electroválvulas de tipo directo pueden ser *cerradas en reposo* o *normalmente cerradas* lo cual quiere decir que cuando falla la alimentación eléctrica quedan cerradas o bien pueden ser del tipo *abiertas en reposo* o *normalmente abiertas* que quedan abiertas cuando no hay alimentación. Es decir, en el primer caso la válvula se mantiene cerrada por la acción de un muelle el solenoide la abre venciendo la fuerza del muelle. Esto quiere decir que el solenoide debe estar activado y consumiendo energía mientras la válvula está abierta. Las *normalmente abiertas*, funcionan al revés.

Este tipo de válvulas se utilizan muy comúnmente en lavadoras, lavaplatos, riegos y otros usos similares.

Electrovalvulas asistidas



E- Solenoide

F- Salida.

En otro tipo de electroválvula el solenoide no controla la válvula directamente sino que el solenoide controla una válvula piloto secundaria y la energía para la actuación de la válvula principal la suministra la presión del propio fluido.

El gráfico adjunto muestra el funcionamiento de este tipo de válvula. En la parte superior vemos la válvula cerrada. El agua bajo presión entra por **A**. **B** es un diafragma elástico y tiene encima un muelle que le empuja hacia abajo con fuerza débil. El diafragma tiene un diminuto orificio, de menos diámetro que el del conducto **D**, en el centro que permite el paso de un pequeño flujo de agua. Esto hace que el agua llene la cavidad **C** y que su presión sea igual en ambos lados del diafragma, pero un poco mayor en la parte superior, debido al empuje del muelle, por lo que presiona hacia abajo sellando la entrada.

Mientras, el conducto **D** está cerrado por el núcleo del solenoide **E** al que un muelle empuja hacia abajo. Si se activa el solenoide, el núcleo sube y permite pasar el agua desde la cavidad **C** hacia la salida; como el caudal que puede pasar por **D**, al ser mayor su sección, es mayor que el que pasa por el orificio del diafragma, disminuye la presión en **C** y el diafragma se levanta permitiendo el paso directo de agua desde la entrada **A** a la salida **F** de la válvula. Esta es la situación representada en la parte inferior de la figura.

Si se vuelve a desactivar el solenoide se vuelve a bloquear el conducto **D**, se equilibran las presiones del agua en los dos compartimentos y el muelle situado sobre el diafragma necesita muy poca fuerza para que vuelva a bajar ya que la fuerza principal la hace la presión del propio fluido en la cavidad **C**.

De esta explicación se deduce que este tipo de válvula depende para su funcionamiento de que haya mayor presión a la entrada que a la salida y que si se invierte esta situación entonces la válvula abre sin que el solenoide pueda controlarla. Por esta razón se emplean principalmente en sistemas en que la salida (**F**) tiene salida directa a un lugar a presión atmosférica.

Electroválvulas de tres vías

Hay electroválvulas que en lugar de abrir y cerrar lo que hacen es conmutar la entrada entre dos salidas, en una válvula de tres vías. Este tipo de electroválvulas a menudo se usan en los sistemas que tienen calefacción y preparación de agua caliente sanitaria lo que permite permutar el calentamiento de uno u otro sistema alternativamente utilizando una sola bomba de circulación.

En los Calentadores de agua circulante, el agua se calienta según va pasando por el calentador en el momento del consumo y es la propia presión del agua la que abre la válvula de gas; pero en los calentadores por acumulación esto no es posible ya que el agua se calienta mientras está almacenada en un depósito y debe hacerlo aunque no haya circulación. Normalmente se utiliza un válvula solenoide, mandada por un termostato que, cuando detecta una temperatura por debajo de la de consigna (normalmente 60 °C), desvía el agua caliente, destinada a la calefacción, por un intercambiador dispuesto en el depósito de agua caliente sanitaria y cuando el termostato determina que el agua ha llegado a la temperatura de acumulación, corta la corriente de la válvula, que vuelve a su posición de reposo, devolviendo el flujo de agua caliente al sistema de calefacción.

MOTOBOMBAS



Una motobomba, es una máquina que se utiliza en circunstancias en las que se necesita mover una gran cantidad de agua de forma, rápida y eficiente. Existen 2 tipos de motobombas según su aplicación:

SEMI-SUCIAS: para usos regulares con agua clara. Ideal para piscifactorías, piscinas o usos similares.

SUCIAS: Para usos intensivos y de emergencia con aguas limpias o sucias (con componentes sólidos como arena, lodo, gravilla, escombros, etc.).

En una motobomba se debe de tener siempre en cuenta los diagramas de caudal/presión y altura total. El caudal es la cantidad de agua que se desea mover, medida en m³/hora o litros/minuto, y la altura

total describe la altura de una columna vertical de agua teórica que una bomba puede producir.

En una de estas máquinas es también importante determinar el caudal que se desea trasegar, conjuntamente con la altura de impulsión, que viene a ser diferencia de cuota entre los niveles de agua que se desea mover y la altura máxima de descarga del agua (la distancia entre la bomba y la altura de descarga, se llama altura de elevación).

Para nuestro prototipo utilizaremos una motobomba sencilla de fácil adquisición en ferreterías y tiendas eléctricas, usadas en pesceras y pequeñas fuentes,



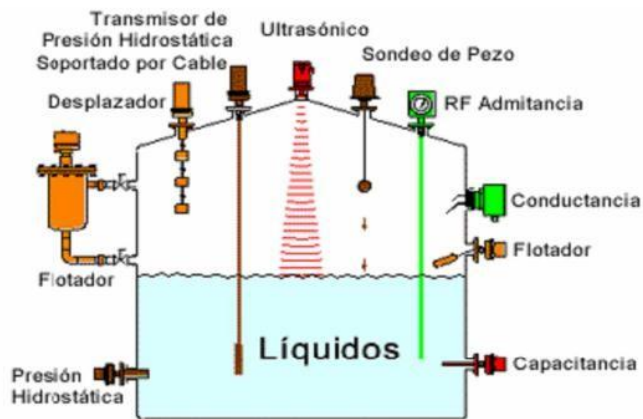
Estas motobombas pueden manejar hasta 2 metros de altura enviando 240 litros hora, y reduciendo su capacidad de entrega dependiendo la altura hacia la cual se desee hacer subir el agua.

La fuente de alimentación es 10 vac, pero aquí en este proyecto será controlado a partir de opto acopladores y triacs (opto acoplador moc3021 triac: bt136)

SISTEMAS DE MONITOREO DE NIVEL DE AGUA

En cuanto a sensores se refiere, se cuenta con una variedad para todo tipo de necesidades, en la Figura 1 se observan los tipos de sensor más utilizados en la medición de nivel en los ríos.

Figura 1 - Sensores de nivel



Sensor por Desplazamiento (flotador)

Según (IIMPI, 2012) consiste en un flotador ubicado en contacto con el fluido y conectado al exterior del depósito indicando directamente el nivel sobre una escala graduada.

Los instrumentos de flotador tienen una precisión de 0,5 %. Son adecuados en la medida de niveles en estanques abiertos y cerrados a presión o a vacío, y son independientes del peso específico del líquido. Por otro lado, el flotador puede agarrarse en el tubo guía por un eventual depósito de los sólidos o cristales que el líquido pueda contener y además los tubos guía muy largos pueden dañarse ante olas bruscas en la superficie del líquido o ante la caída violenta del

líquido en el estanque.

Sensor Capacitivo.

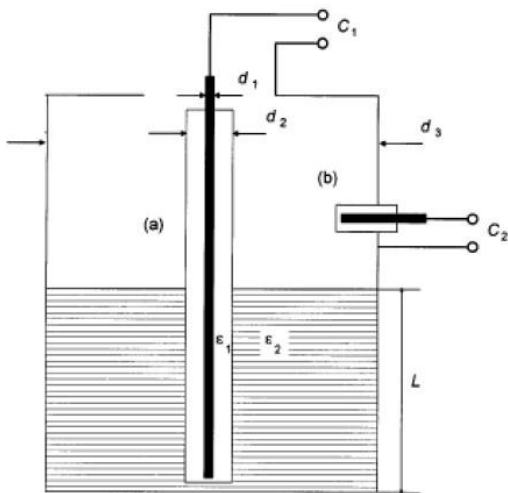
—Se basa en medir la variación de capacitancia de un condensador cuando va variando el medio dieléctrico entre sus placas (IIMPI, 2012). Mide la capacidad del condensador formado por el electrodo sumergido en el líquido y las paredes del tanque (ver Figura 2). La capacidad del conjunto depende linealmente del nivel del líquido.

En fluidos no conductores se emplea un electrodo normal y la capacidad total del sistema se compone de la del líquido, la del gas superior y la de las conexiones superiores.

En fluidos conductores el electrodo está aislado usualmente con teflón interviniendo las Capacidades adicionales entre el material aislante y el electrodo en la zona del líquido y del gas. La precisión de los transductores de capacidad es de $\pm 1 \%$.

Se caracterizan por no tener partes móviles, son ligeros, presentan una buena resistencia a la corrosión y son de fácil limpieza. Su campo de medida es prácticamente ilimitado. Tiene el inconveniente de que la temperatura puede afectar las constantes dieléctricas (0,1 % de aumento de la constante dieléctrica. °C) y de que los posibles contaminantes contenidos en el líquido puedan adherirse al electrodo variando su capacidad y falseando la lectura, en particular en el caso de líquidos conductores. (IIMPI, 2012)

Figura 2 - Funcionamiento sensor capacitivo



Sensor Ultrasónico.

—Se basa en la emisión de un impulso ultrasónico a una superficie reflectante y la recepción del eco del mismo en un receptor. El retardo en la captación del eco depende del nivel del estanquell (IIMPI,2012).

Los sensores trabajan a una frecuencia de unos 20 KHz. Estas ondas atraviesan con cierto Amortiguamiento o reflexión el medio ambiente de gases o vapores y se reflejan en la Superficie del sólido o del líquido.

La precisión de estos instrumentos es de ± 1 a 3 %. Son adecuados para todos los tipos de estanques y de líquidos o fangos pudiendo construirse a prueba de explosión. Presentan el inconveniente de ser sensibles a la densidad de los fluidos y de dar señales erróneas cuando la superficie del nivel del líquido no es nítida como es el caso de un líquido que forme espuma, ya que se producen falsos ecos de los ultrasonidos.

La utilización de la computadora permite, a través de un programa, almacenar el perfil ultrasónico del nivel, y así tener en cuenta las características particulares de la superficie del líquido, tal como la espuma, con lo cual se mejora la

precisión de la medida.

Conductivímetro.

Según (IIMPI, 2012), consiste en uno o varios electrodos y un relé eléctrico o electrónico que es excitado cuando el líquido humedece a estos electrodos. El líquido debe ser lo suficientemente conductor como para excitar el circuito electrónico, y de este modo el aparato puede discriminar la separación entre el líquido y su vapor, tal como ocurre, por ejemplo, en el nivel de agua de una caldera de vapor. La impedancia mínima es del orden de los 20 MΩ/cm, y la tensión de alimentación es alterna para evitar fenómenos de oxidación en las sondas por causa del fenómeno de la electrólisis. Cuando el líquido humedece los electrodos se cierra el circuito electrónico y circula una corriente segura del orden de los 2 mA; el relé electrónico dispone de un temporizador de retardo que impide su enclavamiento ante una ola del nivel del líquido o ante cualquier perturbación momentánea o bien en su lugar se disponen dos electrodos poco separados enclavados eléctricamente en el circuito.

El instrumento se emplea como alarma o control de nivel alto y bajo, utiliza relés eléctricos para líquidos con buena conductividad y relés electrónicos para líquidos con baja conductividad.

Montado en grupos verticales de 24 o más electrodos, puede complementar los típicos niveles de vidrio de las calderas, y se presta a la transmisión del nivel a la sala de control y a la adición de las alarmas correspondientes.

El instrumento es versátil, sin partes móviles, su campo de medida es grande con la limitación física de la longitud de los electrodos. El líquido contenido en el estanque debe tener un mínimo de conductividad y si su naturaleza lo exige, la corriente debe ser baja para evitar la deterioración del producto. Por otro lado, conviene que la sensibilidad del aparato sea ajustable para detectar la presencia de espuma en caso necesario.

sensor ultrasonidos HC-SR04



El sensor de ultrasonidos se enmarca dentro de los sensores para medir distancias o superar obstáculos, entre otras posibles funciones.

En este caso vamos a utilizarlo para la medición de distancias al nivel de agua. Esto lo consigue enviando un ultrasonido (inaudible para el oído humano por su alta frecuencia) a través de uno de la pareja de cilindros que compone el sensor (un transductor) y espera a que dicho sonido rebote sobre un objeto y vuelva, retorno captado por el otro cilindro.

Este sensor en concreto tiene un rango de distancias sensible entre 3cm y 3m con una precisión de 3mm.

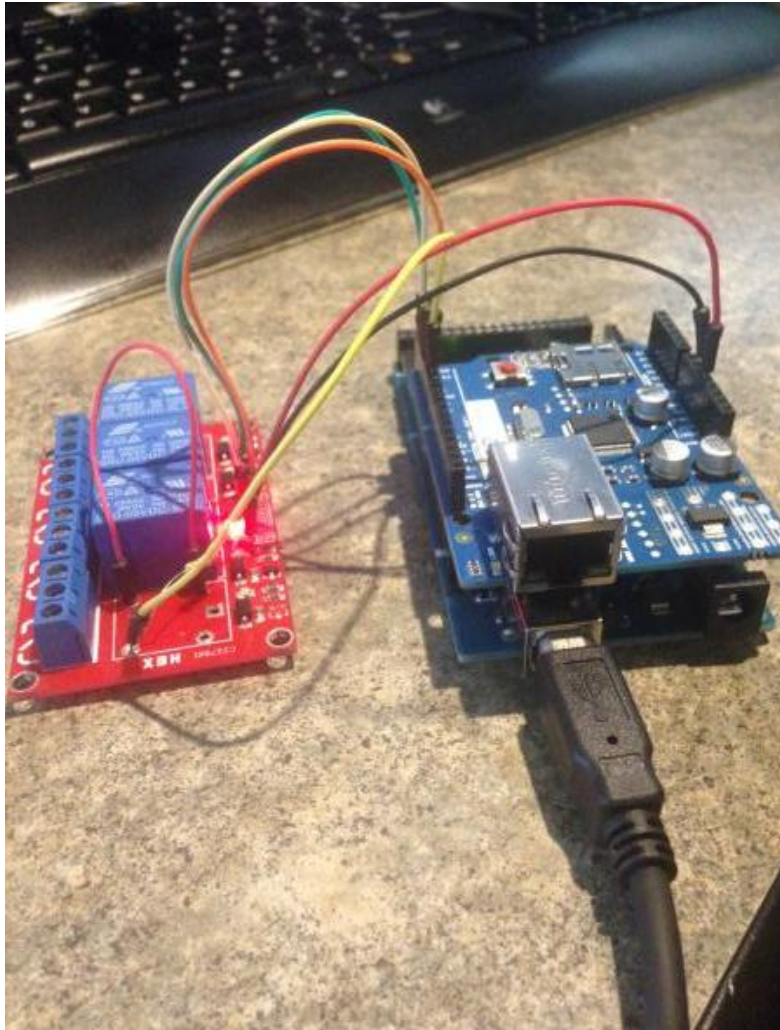
METODOLOGIA

Para explicar la metodología usada vamos a ver por partes la metodología usada para cada uno de los modulos del prototipo

- Diseño del control de la electroválvula
- Diseño del control de la Motobomba
- Configuración de Wampserver en Localhost
- Configuración del Arduino como Web Server
- Programación de los modulos de la aplicación en Arduino

Diseño del control de la electroválvula

Para la implementación del prototipo del sistema electrónico de monitoreo y control para movimiento de volúmenes de agua en un laboratorio de de aguas de bajo costo, se desarrollaron pruebas usando una interfaz web de control para relevos en el caso del control de las electroválvulas como lo muestra la siguiente fotografía:



La electroválvula usada fue la ZE 4F1-80 que trabaja con un solenoide de 12 voltios, la cual la vamos a controlar con relevos, se pueden tener tantas válvulas como permita el sistema en este caso usaremos los pines 2-6 (5 posibles electro válvulas)

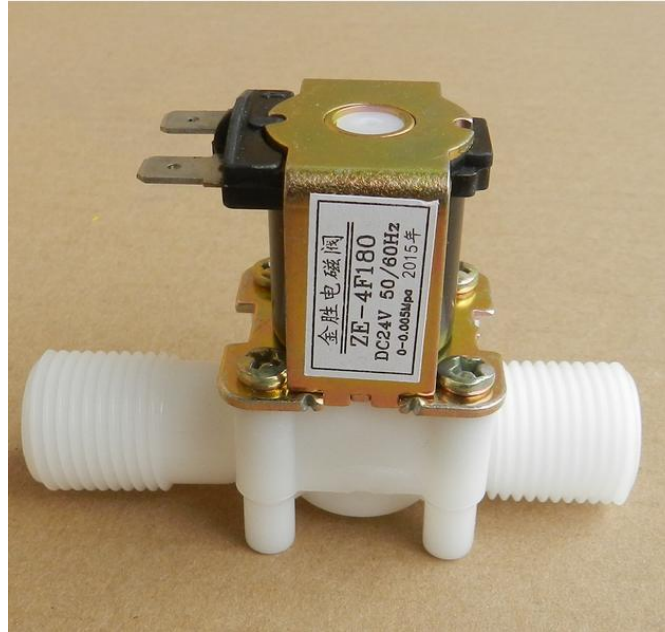
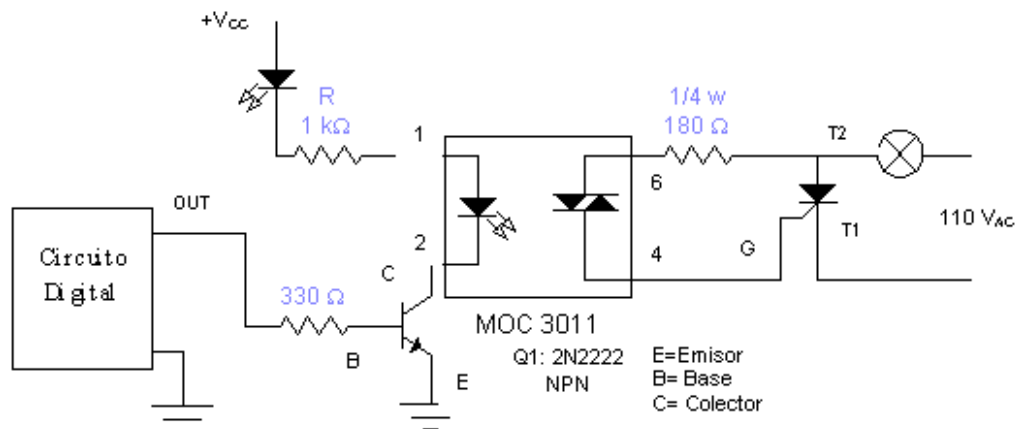
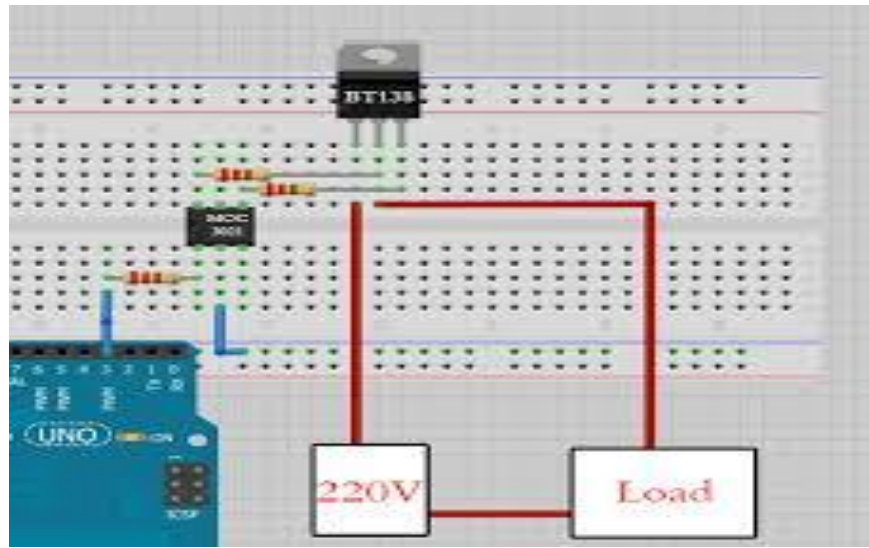


Imagen de la ZE 4F1-80

Control de la Motobomba

Para el caso del control de la motobomba se decidió usar triacs optoacoplados para controlar la potencia entregada a la motobomba, usando el siguiente esquemático:





Y como carga del anterior diseño se coloca la motobomba, la cual consume baja potencia por ser una motobomba de prototipo:



El triac actúa como un interruptor de estado sólido controlado por el opto acoplador el cual a su vez es controlado por la señal proveniente del servidor web Arduino shield Ethernet.

Configuración del WampServer en Localhost



WampServer es un entorno de desarrollo web para Windows en el cual se podrán crear aplicaciones web con Apache, PHP y base de datos en MySQL (*motor de base de datos*). Esta herramienta incluye además con un administrador de base de datos PHPMyAdmin con el cual podremos crear una nueva base de datos e ingresar la data de las tablas creadas en ella, realizar consultas y generar scripts SQL, como exportar e importar scripts de base de datos. WampServer ofrece a los desarrolladores herramientas necesarias para realizar aplicaciones web de manera local, con un sistema operativo (Windows), un manejador de base de datos (MySQL), un software de programación script web PHP. WampServer se caracteriza por que puede ser usado de forma libre es decir no debemos de contar con alguna licencia el cual nos permita el uso de la misma, ya que pertenece a la corriente de "open source".

UTILIDAD

Su utilidad es importante a la hora de desarrollar aplicaciones web, ya que funciona al igual como si cuando trabajamos en un servidor web, ya que podemos ejecutar estas aplicaciones de manera local y ver como seria el funcionamiento antes de ser subidas a un hosting o servidor web. Además de ello podemos gestionar datos con la ayuda del motor de base de datos (MySQL) y su administrador (PHPMyAdmin).

Configuración del Arduino como Web Server

Lo primero que se hizo fue acceder al Router desde el navegador web. Y se le debe configurar la puerta de enlace (o gateway en inglés).

A continuación se muestra como se hace desde Windows.

- Se pulsa el botón de inicio.
- En la barra de búsquedas, se escribe ejecutar y pulsamos la tecla intro.
- En la ventana que te acaba de aparecer escribimos “cmd” y vuelve a pulsar intro.
- Te habrá salido otra ventana de fondo negro. Escribe en ella “ipconfig”.
- Dentro del texto que se acaba de escribir en la ventana hay una sección llamada Adaptador de Ethernet (o similar). Buscamos en ella la “puerta de enlace predeterminada” o “gateway” y apunta la dirección asociada (suele ser algo del tipo 192.168.1.1).

Ahora que ya tenemos la dirección con la que acceder al Router, basta con escribir en el navegador <http://127.0.0.14/> y el servidor web embebido responde con

```
Adaptador de LAN inalámbrica Wi-Fi:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . : domain.name

Adaptador de Ethernet Conexión de red Bluetooth:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de Ethernet Ethernet:
    Sufijo DNS específico para la conexión. . . : domain.name
    Vínculo: dirección IPv6 local. . . . : fe80::3572:1e9e:1220:70ab%3
    Dirección IPv4. . . . . : 192.168.1.33
    Máscara de subred. . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.1.1

Adaptador de túnel isatap.domain.name:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . : domain.name

Adaptador de túnel Teredo Tunneling Pseudo-Interface:
```

Programacion de los Modulos en Arduino

Módulo de control del relés por parte del cliente web:

```
#include <SPI.h>
#include <Ethernet.h>

int pos = 0;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //Direccion Fisica MAC
byte ip[] = { 192, 168, 0, 14 }; // IP Local que se debe configurar
byte gateway[] = { 192, 168, 0, 1 }; // Puerta de enlace
byte subnet[] = { 255, 255, 255, 0 }; //Mascara de Sub Red
EthernetServer server(80); //Se usa el puerto 80 del servidor
String readString;

void setup() {

  Serial.begin(9600); // Inicializa el puerto serial
  while (!Serial) { // Espera a que el puerto serial sea conectado, Solo necesario para el Leonardo
    ;
  }
  pinMode(2,OUTPUT); // Se configura como salidas los puertos del 2 al 6
  pinMode(3,OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  microservo.attach(7); // Se configura como Servo el Puerto 7

  Ethernet.begin(mac, ip, gateway, subnet); // Inicializa la conexion Ethernet y el servidor
  server.begin();
  Serial.print("El Servidor es: ");
  Serial.println(Ethernet.localIP()); // Imprime la direccion IP Local
}
```

```

void loop() {
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();

        //Lee caracter por caracter HTTP
        if (readString.length() < 100) {
          //Almacena los caracteres a un String
          readString += c;
        }

        // si el requerimiento HTTP fue finalizado
        if (c == '\n') {
          Serial.println(readString); //Imprime en el monitor serial

          client.println("HTTP/1.1 200 OK"); //envia una nueva pagina en codigo HTML
          client.println("Content-Type: text/html");
          client.println();
          client.println("<HTML>");
          client.println("<HEAD>");
          client.println("<TITLE>Control de valvulas usando Shield Ethernet</TITLE>");
          client.println("</HEAD>");
          client.println("<BODY>");
          client.println("<hr />");
          client.println("<H1>Control de Valvulas usando Arduino desarrollado por Favian Ocampo</H1>");
          client.println("<br />");
          client.println("<H2>Control on/off 2-6 (electrovalvulas del sistema)</H2>");
          client.println("<br />");

          client.println("<a href='\"/?button2on\"'> Encender valvula 1</a> "); // construye en la pagina cada uno de

```

los botones

```
client.println(" | | ");
client.println("<a href='\"/?button2off\\\"'> Apagar Valvula 1</a><br /> ");
client.println("<br />");

client.println("<br />");
client.println("<a href='\"/?button3on\\\"'> Encender Valvula 2</a> ");
client.println(" | | ");
client.println("<a href='\"/?button3off\\\"'> Apagar Valvula 2</a><br /> ");
client.println("<br />");

client.println("<br />");
client.println("<a href='\"/?button4on\\\"'> Encender Valvula 3</a> ");
client.println(" | | ");
client.println("<a href='\"/?button4off\\\"'> Apagar Valvula 3</a><br /> ");
client.println("<br />");

client.println("<br />");
client.println("<a href='\"/?button5on\\\"'> Encender Valvula 4</a>");
client.println(" | | ");
client.println("<a href='\"/?button5off\\\"'> Apagar Valvula 4</a><br />");
client.println("<br />");

client.println("<br />");
client.println("<a href='\"/?button6on\\\"'> Encender Valvula 5</a> ");
client.println(" | | ");
client.println("<a href='\"/?button6off\\\"'> Apagar Valvula 5</a><br /> ");
client.println("<br />");

client.println("<br />");
client.println("<a href='\"/?left\\\"'>Girar Izq Pin7</a>");
client.println(" | | ");
client.println("<a href='\"/?right\\\"'>Girar Der Pin7</a><br />");
```

```
client.println("<hr />");
client.println("<p>Fabian ocampo Universidad Catolica de Manizales</p>");
client.println("<br />");
client.println("</BODY>");
client.println("</HTML>");
```

```
delay(1);
//detiene el cliente servidor
client.stop();
```

```
//control del arduino si un boton es presionado
```

```
if (readString.indexOf("?button2on") >0){
  digitalWrite(2, HIGH);
}
```

```
if (readString.indexOf("?button2off") >0){
  digitalWrite(2, LOW);
}
```

```
if (readString.indexOf("?button3on") >0){
  digitalWrite(3, HIGH);
}
```

```
if (readString.indexOf("?button3off") >0){
  digitalWrite(3, LOW);
}
```

```
if (readString.indexOf("?button4on") >0){
  digitalWrite(4, HIGH);
}
```

```
if (readString.indexOf("?button4off") >0){
  digitalWrite(4, LOW);
}
```

```
if (readString.indexOf("?button5on") >0){  
    digitalWrite(5, HIGH);  
}  
if (readString.indexOf("?button5off") >0){  
    digitalWrite(5, LOW);  
}
```

```
if (readString.indexOf("?button6on") >0){  
    digitalWrite(6, HIGH);  
}  
if (readString.indexOf("?button6off") >0){  
    digitalWrite(6, LOW);  
}
```

```
// Limpia el String(Cadena de Caracteres para una nueva lectura  
readString="";
```

```
    }  
}  
}  
}  
}
```

Modulo del sensor de nivel usando el sensor de ultrasonido HC SR04

```
#include <SoftwareSerial.h>
#include <URMSerial.h>
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,69,14); EthernetServer server(80);
URMSerial urm;
#define DISTANCE 1
#define ERROR 3
#define NOTREADY 4
void setup() {
  Serial.begin(9600);
  urm.begin(2,3,9600);
  while (!Serial) {
    ;
  }
  Ethernet.begin(mac, ip);
  server.begin();
}
int tacometro;
int distancia[10];
int dvalue;
void loop() {
  delay(100);
  tacometro++;
  if (tacometro>=60000){ //60000 vueltas de 100ms, sería cada 10 minutos
    urm.requestMeasurementOrTimeout(DISTANCE, dvalue);
    dvalue=400-dvalue; //Se debe cambiar dependiendo de la altura de cada tanque y su calibracion.
    for (int i=1; i<10; i++){
      distancia[i-1]=distancia[i];
```



```

}

distancia[9]=dvalue;

tacometro = 0;

}

EthernetClient client = server.available();

if (client) {

Serial.println("new client");

boolean currentLineIsBlank = true;

while (client.connected()) {

if (client.available()) {

char c = client.read();

Serial.write(c);

if (c == '\n' && currentLineIsBlank) {

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println("Refresh: 60");

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.println("<head>");

client.println("<script type='text/javascript' src='https://www.google.com/jsapi'></script>");

client.println("<script type='text/javascript'>");

client.println("google.load('visualization', '1', {packages:['corechart']});");

client.println("google.setOnLoadCallback(drawChart);");

client.println("function drawChart() {");

client.println("var data = google.visualization.arrayToDataTable([");

client.println("<H1>Control de nivel de tanque usando Arduino desarrollado por Favian Ocampo</H1>");

client.println("[['Tiempo', 'Nivel de Agua en cm'],");

client.println("[0, "+String(distancia[0])+"],");

client.println("[10, "+String(distancia[1])+"],");

client.println("[20, "+String(distancia[2])+"],");

```

```

client.println("[30', "+String(distancia[3])+"],");
client.println("[40', "+String(distancia[4])+"],");
client.println("[50', "+String(distancia[5])+"],");
client.println("[60', "+String(distancia[6])+"],");
client.println("[70', "+String(distancia[7])+"],");
client.println("[80', "+String(distancia[8])+"],");
client.println("[90', "+String(distancia[9])+"],");
client.println("]");

client.println("var options = {");

client.println("title: 'Grafico de nivel',");

client.println("hAxis: {title: 'Tiempo en minutos', titleTextStyle: {color: 'red'}}");

client.println("}");

client.println("var chart = new google.visualization.AreaChart(document.getElementById('chart_div'))");
client.println("chart.draw(data, options);");

client.println("}");

client.println("</script>");
client.println("</head>");
client.println("<body>");

client.println("<H1 style='margin-bottom: 0px;margin-top: 0px;'>Sistema de Monitoreo Ultrasonico Remoto</H1>");
client.println("<div id=\"chart_div\" style=\"width: 900px; height: 500px;\"></div>");
client.println("<H2>Medicion actual</H2>");

client.println("Valor de nivel :");

client.println(dvalue);

client.println("<br/>");

client.println("<hr/>");

client.println("Desarrollado por Favian Ocampo Universidad Catolica de Manizales");

client.println("</body>");

client.println("</html>");

break;
}

if (c == '\n') {
currentLineIsBlank = true;
}

```

```
else if (c != '\r') {  
    currentLinesBlank = false;  
}  
}  
}  
delay(1);  
client.stop();  
Serial.println("cliente desconectado ");  
}  
}
```

CRONOGRAMA DE ACTIVIDADES

ACTIVIDAD	DESCRIPCION DE ACTIVIDADES
1	Búsqueda de antecedentes
2	Elección del módulos
3	Diseño y armado del sistema de control de válvulas
4	Diseño y armado del control de motobombas
5	Diseño de interconexión electrónica módulos
6	Programación de la de la tarjeta Arduino-Ethernet
7	Diseño interfaz web
8	Pruebas preliminares
9	Pruebas FINALES

ACTIVIDAD	CRONOGRAMA(semanas)														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	■	■	■												
2			■	■	■										
3					■	■	■								
4							■	■	■	■	■				
5									■	■	■	■	■		
6									■	■	■	■	■		
7									■	■	■	■	■		
8										■	■	■			
9													■	■	■

11																	
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

BIBLIOGRAFIA

1 <http://www.huellahidrica.org/Reports/Arevalo-2012-HuellaHidricaColombia.pdf>
Una mirada a la agricultura colombiana desde su huella hídrica

2 Atmel Corporation. (2009). 8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash.

3 Atmel Corporation. (2009). 8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash.

4 IIMPI. (2012). Universidad de la República de Uruguay. Recuperado el 2012, de Sitio web de Facultad de Ingeniería: <<http://goo.gl/DUwW0y>>

5 Jayr. (7 de Enero de 2012). ENERGÍA CASERA. Recuperado el 7 de 10 de 2012, de Tecnologías para generar tu propia energía: <<http://goo.gl/9B7FG9>>

6 Ministerio de Ambiente, Vivienda y Desarrollo Territorial, IDEAM. (2007). PROTOCOLO PARA EL MONITOREO Y SEGUIMIENTO DEL AGUA. Bogotá (Colombia).