



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE INGENIERÍA
DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“MONITOREO Y CONTROL DE NIVEL REMOTO UTILIZANDO TECNOLOGIA
ARDUINO”

AUTOR (ES): Walter Oliva Neira
Felipe Pincheira Muñoz

SEMINARIO PARA OPTAR AL TÍTULO DE
INGENIERO DE EJECUCIÓN EN ELECTRÓNICA
CONCEPCIÓN – CHILE
2016



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE INGENIERÍA
DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“MONITOREO Y CONTROL DE NIVEL REMOTO UTILIZANDO TECNOLOGIA
ARDUINO”

AUTOR (ES): Walter Oliva Neira
Felipe Pincheira Muñoz

Profesor Guía: Gustavo Sanhueza Garrido

Profesor adjunto 1: Pablo Sáez Srain

Profesor adjunto 2: John Correa Toloza

SEMINARIO PARA OPTAR AL TÍTULO DE
INGENIERO DE EJECUCIÓN EN ELECTRÓNICA
CONCEPCIÓN – CHILE

INDICE

RESUMEN	5
INTRODUCCION	6
OBJETIVO GENERAL	7
OBJETIVOS ESPECIFICOS	7
CAPITULO N°1: CONCEPTOS GENERALES	8
SENSOR:	8
CONTROLADOR:	8
ACTUADOR:	9
SEÑAL DIGITAL:	9
SEÑAL ANALÓGICA:	10
MODULACIÓN PWM:	10
PROTOCOLOS DE COMUNICACIÓN INDUSTRIAL.....	11
<i>HART</i>	11
<i>MODBUS</i>	¡Error! Marcador no definido.
<i>PROFIBUS</i>	12
<i>DEVICENET</i>	12
<i>GPRS</i> :	13
<i>802.11n (WIFI)</i>	13
CAPITULO N°2: DISPOSITIVOS Y VARIABLES	14
ARDUINO	14
VENTAJAS DE ARDUINO	14
SHIELD (MODULOS DE EXPANSION).....	15
LENGUAJE DE PROGRAMACIÓN	16
VARIABLES FÍSICAS INVOLUCRADAS	17
VARIABLES A CONTROLAR	18
<i>NIVEL</i>	18
<i>VELOCIDAD</i>	18
CAPITULO N°3: CONTROLADOR A IMPLEMENTAR	19
CONTROLADORPID	19
DESCRIPCIÓN DE CADA UNA DE SUS ACCIONES	19
DESCRIPCIÓN DE CADA CONSTANTE UTILIZADA.....	20
MÉTODOS PARA SINTONIZAR EL CONTROLADOR.....	20
<i>METODO EMPÍRICO</i>	20
<i>METODO DE SINTONIZACIÓN ZIEGLER-NICHOLS</i>	21
CAPITULO N°4: DESARROLLO DEL HADWARE DE LA PROPUESTA	23
DESCRIPCIÓN DE LA PROPUESTA.....	23
HADWARE PARA DESARROLLO DE LA PROPUESTA.....	24
<i>ARDUINO MEGA 2560</i>	24
<i>SHIELD RELAY</i>	24
<i>SENSOR ULTRASÓNICO HC-SR04</i>	25

CONVERSOR DE PWM A "0V a 10V":	25
VARIADOR DE FRECUENCIA "V1000 YASKAWA"	26
ELECTROBOMBA PK 60	26
SHIELD WIFI EPS8266	27
PROTOTYPE SHIELD	27
CONEXION DE LOS ELEMENTOS INVOLUCRADOS.....	28
DESCRIPCION DE LA MAQUETA CONSTRUIDA	29
CAPITULO Nº5: DESARROLLO DEL SOFTWARE DE LA PROPUESTA	31
DESCRIPCIÓN GENERAL	31
CONSIDERACIONES PARA EL DESARROLLO DEL ALGORITMO	32
PAGINA WEB DESARROLLADA PARA MONITOREO Y CONTROL.....	33
ELEMENTOS DE LA INTERFACE.....	34
MODELAMIENTO MATEMÁTICO DEL PROCESO	35
SINTONIZACIÓN DEL CONTROLADOR	41
CONCLUSIONES	44
BLIBLIOGRAFIA	46
ANEXOS	47
DISEÑO DEL CONVERTOR DE PWM – 0V A 10V	47
CIRCUITOS Y GRAFICOS.....	48
ANÁLISIS DEL CONVERTOR PWM	51
DATA TÉCNICA DE LOS ELEMENTOS UTILIZADOS	52
ARDUINO MEGA 2560.....	52
MODULO WIFI SERIAL EPS8266	53
SHIELD RELE.....	54
SENSOR ULTRASONICO HC-SR04.....	55
VARIADOR DE FRECUENCIA "V1000 YASKAWA"	56
PROTOTYPE SHIELD	57
ELECTROBOMBA PK 60.....	58
COSTO DE LOS ELEMENTOS NECESARIOS PARAREALIZAR EL SISTEMA DE CONTROL.	59
PROGRAMA DESARROLLADO EN ARDUINO	60

RESUMEN

En este proyecto de tesis se desarrolla una propuesta innovadora para rehacer un control de nivel utilizado en grandes empresas, pero a escala más reducida. Para ello se buscarán los diferentes dispositivos capaces de cumplir las funciones necesarias para controlar y monitorear el nivel de agua de forma remota.

Se programará un código en Sketch capaz de realizar un control PID sintonizado según el método de Ziegler-Nichols. Una vez encontrados los valores de las constantes se podrá realizar el control de forma precisa y en tiempo real.

Además, el código deberá poder detectar todos los elementos primarios presentes.

En el capítulo I, se describen los distintos protocolos de comunicación y buses de campo disponibles para el correcto desarrollo de la propuesta a implementar, detallando de forma clara y precisa el funcionamiento de éstos.

El capítulo II, dispone de todos los dispositivos que se utilizarán para la implementación del sistema de control, como también una descripción de las variables controladas y manipuladas en el proceso.

El capítulo III, muestra el tipo de controlador que se implementará, explicando cada variable y los métodos asociados a la sintonización del PID.

A continuación, en el capítulo IV, se desarrolla todo el hardware, incluyendo circuitos de conversión y conexión entre dispositivos. Se entrega toda la información sobre el rango de trabajo de cada elemento utilizado y se muestra el esquema final del hardware armado en su totalidad.

En el capítulo V, se realiza todo el desarrollo del software del arduino, como lo son el diagrama de flujo de la programación, el modelamiento matemático del sistema de control y la sintonización del PID.

Finalmente, se entrega un detalle de las características funcionales de cada dispositivo usado, y también los costos que significa adquirirlos actualmente. Anexo a esto, va incluido el programa con el que trabaja el arduino, explicando cada función de sus comandos.

INTRODUCCIÓN

El Arduino mega 2560 es un dispositivo electrónico que posee el microcontrolador más potente de la familia arduino. Posee 54 pines digitales, los cuales pueden funcionar como entrada y salida; 16 entradas análogas, un cristal oscilador (reloj) de 16Mhz, puerto USB, botón de reseteo y una entrada para la alimentación de la placa.

Para establecer la comunicación entre el arduino Mega y un computador, se necesita conectar el dispositivo mediante UBS (el cual es un convertidor UBS-SERIE). También se encuentran los Shield de arduino, los cuales son placas compatibles con cualquier otro dispositivo de la línea arduino, y se utilizan principalmente para realizar procesos que no están incluidos en la placa base (Arduino Mega), así se puede ahorrar espacio y consumo de energía.

Uno de los Shield más importantes que se utilizará es el Shield WIFI de arduino, el cual será el principal encargado de transmitir los datos de forma inalámbrica a cualquier dispositivo que este en una red de internet.

La tecnología WI-FI, es un mecanismo de conexión de dispositivos de forma inalámbrica a través de un punto de acceso, el cual tiene un alcance de veinte metros en interiores y treinta metros aproximadamente al aire libre.

Wi-Fi es una marca comercial que prueba y certifica equipos que cumplen con los estándares 802.11n relacionados a redes inalámbricas de área local (WLAN).

La transmisión vía WI-FI proporciona un medio que reemplaza el cableado y otros tipos de enlaces entre dispositivos, siendo el método más utilizado en la última década para la transmisión de datos.

Hoy en día se presenta una problemática en la industria sanitaria, siendo ésta la falta de control y monitoreo de los depósitos de los estanques de agua principales, que abastecen a las diferentes zonas urbanas del país. Sin embargo para estos estanques se busca implementar sistemas de control de bajo costo. Para esto se desarrollará un control de nivel con telemetría, el cual será presentado en una maqueta utilizando arduino y sus respectivos shield, los cuales reemplazarán los controladores y sensores que se usan en los procesos industriales actualmente.

OBJETIVO GENERAL.

- Desarrollar un sistema de monitoreo y control de nivel automatizado de forma remota utilizando Arduino.

OBJETIVOS ESPECIFICOS.

- Proponer una solución a escala para el monitoreo y control de nivel de forma remota.
- Proponer un monitoreo y control de nivel a partir de tarjeta Arduino.
- Definir elementos sensores y actuadores necesarios para medir y controlar las variables del proceso.
- Diseñar el proceso (maqueta).
- Programar código fuente, necesario para la tarjeta Arduino Mega 2560

CAPITULO N°1: CONCEPTOS GENERALES

SENSOR: Un sensor es un dispositivo eléctrico y/o mecánico que convierte magnitudes físicas (luz, magnetismo, presión, etc.) en valores medibles de dicha magnitud.



Figura N°01: “Diferentes tipos de sensores”

CONTROLADOR: Es un dispositivo que ejecuta la acción de control sobre el actuador para poder llevar el proceso a la referencia programada. Estos se diseñan dependiendo del proceso que se quiera controlar.



Figura N°02: “Diferentes tipos de controladores (PLC, NOVUS, Arduino)”

ACTUADOR: Dispositivo que recibe una señal eléctrica desde el controlador, para luego cambiar alguna característica, física/mecánica/eléctrica que nos permite generar un cambio en el proceso.



Figura N°03: “Actuadores industriales (motores)”

SEÑAL DIGITAL: Una señal digital es discontinua, y sólo puede tomar dos valores o estados: 0 y 1, que pueden ser impulsos eléctricos de baja y alta tensión, interruptores abiertos o cerrados, etc.

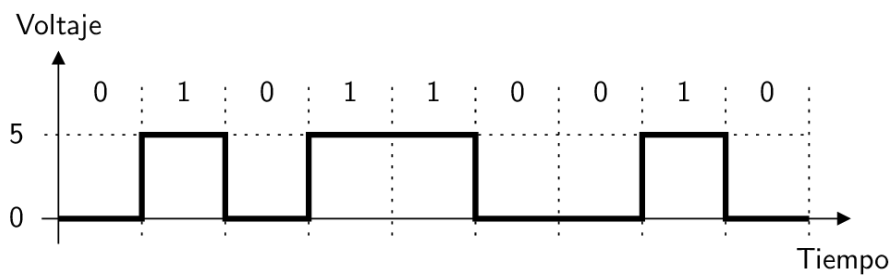


Figura N°04: “Señal digital”

SEÑAL ANALÓGICA: Una señal analógica es continua, y puede tomar infinitos valores.

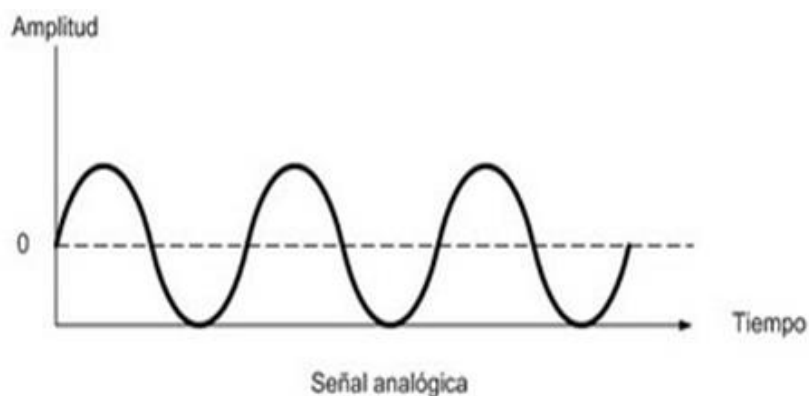


Figura N°05: “Señal analógica”

MODULACION PWM: Es una técnica para obtener resultados análogos con medios digitales. El control digital se utiliza para crear una onda cuadrada que conmuta entre encendido y apagado variando el tiempo para cada uno de los estados. En el caso de arduino esta variación se ejecuta entre 0v y 5v. Siendo capaz de generar 255 condiciones de tiempo diferente.

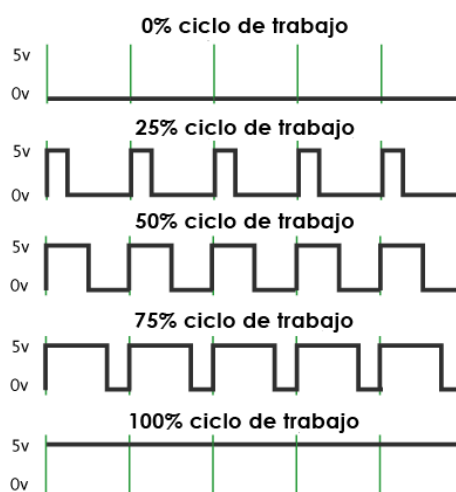


Figura N°06: “Ciclos PWM de un Señal”

PROTOCOLOS DE COMUNICACION INDUSTRIAL.

Normalmente en la industria se utilizan muchos protocolos de comunicación para los diferentes tipos de controladores y poder comunicar las diferentes etapas que componen un proceso.

HART: El protocolo HART (High way-AddressableRemote-Transducer) agrupa la información digital sobre la señal analógica típica de 4 a 20 mA DC. La señal digital usa dos frecuencias individuales de 1200 y 2200 Hz, que representan los dígitos 1 y 0 respectivamente y que en conjunto forman una onda sinusoidal que se superpone al lazo de corriente de 4-20 mA.



Figura N°07: “Lógica del protocolo de comunicación Hart”

MODBUS: Modbus es un protocolo de transmisión para sistemas de control y supervisión de procesos (SCADA) con control centralizado, puede comunicarse con una o varias Estaciones Remotas (RTU) con la finalidad de obtener datos de campo para la supervisión y control de un proceso. Las Interfaces de Capa Física puede estar configurada en: RS-232, RS-485, Ethernet.

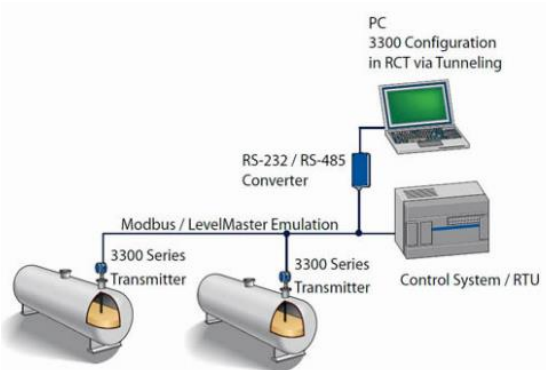


Figura N°08: “Lógica del protocolo de comunicación MODBUS”

PROFIBUS:(Process Field Bus) Norma internacional de bus de campo de alta velocidad para control de procesos.

Donde los más utilizados son:

- Profibus DP (Decentralized Periphery). Orientado a sensores, actuadores enlazados a procesadores (PLCs) o terminales.
- Profibus PA (Process Automation). Para control de proceso, cumple normas especiales de seguridad para la industria química.

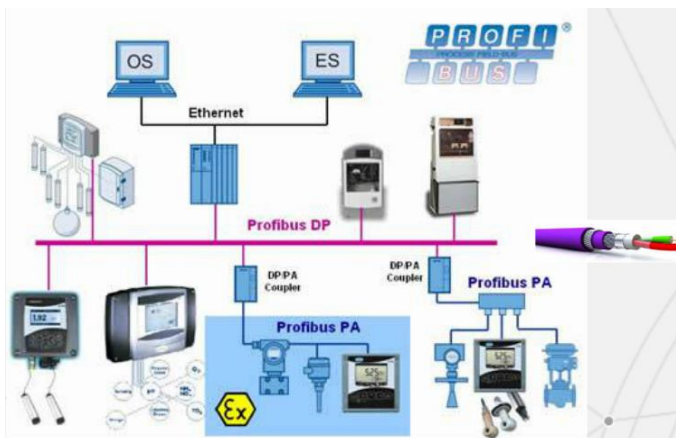


Figura N°09: “Lógica del protocolo de comunicación Profibus”

DEVICENET: Red de bajo nivel adecuada para conectar dispositivos simples como sensores fotoeléctricos, sensores magnéticos, pulsadores, etc. y dispositivos de alto nivel (PLC, controladores, computadores, HMI, entre otros).

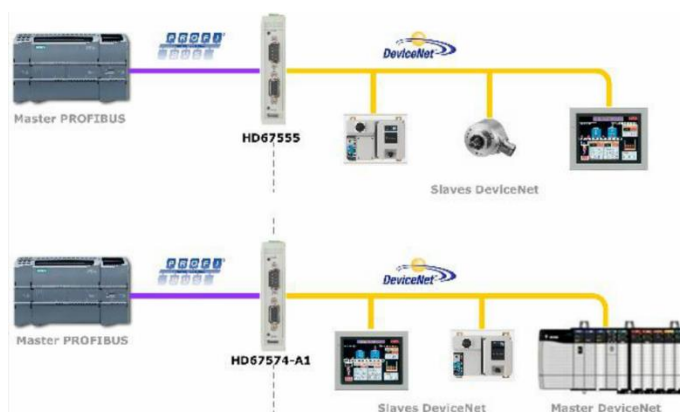


Figura N°10: “Lógica del protocolo de comunicación MODBUS”

GPRS:(*General Packet Radio Services*) es una extensión de la tecnología de comunicaciones móviles GSM. En ella los datos se transmiten en modalidad paquete, con lo cual la información es dividida en pequeños bloques, los cuales posteriormente se reagrupan al llegar a destino. Este tipo de transmisión por paquetes, permite la utilización de diferentes canales en paralelo, sin bloquear la línea, logrando así una mayor capacidad y alcanzando una mayor velocidad. Esta tecnología permite una velocidad de datos de 115 kbps.

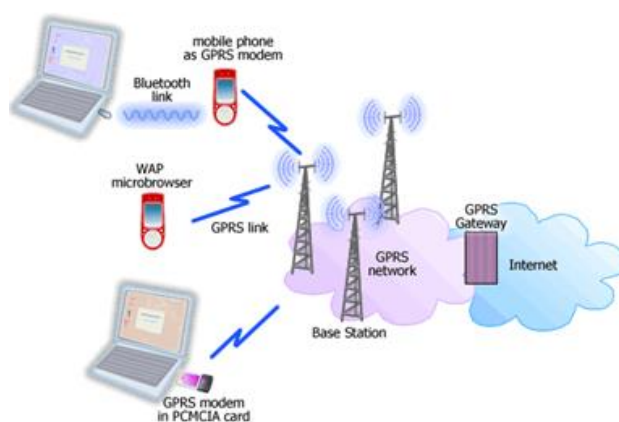


Figura N°11: “Lógica de comunicación GPRS”

802.11n (WIFI): Es un estándar definido por la IEEE, el cual especifica sus normas de funcionamiento en una red de área local inalámbrica (WLAN). Permite un máximo de transferencia de datos de 600 Mbps en un rango de 100 metros aproximadamente en ambientes cerrados y de más de 200 metros al aire libre (o incluso más que eso con el uso de antenas direccionales).

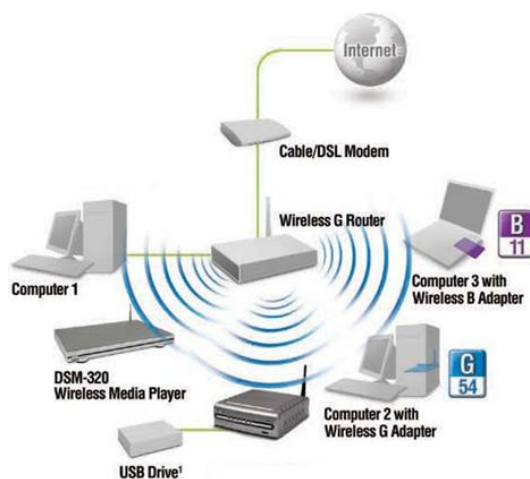


Figura N°12: “Lógica de comunicación 802.11n (WLAN)”

CAPITULO N°2: DISPOSITIVOS Y VARIABLES

ARDUINO

Arduino es una plataforma de desarrollo abierta, concebida para la creación de prototipos y aplicaciones Hardware.

Una gran ventaja de Arduino como un sistema de desarrollo abierto (no necesita licencias) y la constitución de un lenguaje de programación propio, en el que prima la sencillez en la programación, dio como resultado un entorno de desarrollo muy cercano a la gente, que facilita el uso de la electrónica en proyectos de todo tipo, haciendo esta plataforma muy atractiva y constituyendo entorno a ella una de las comunidades más grandes de desarrollo que existen en internet actualmente.

El Hardware de Arduino, consiste en una placa con un microcontrolador Atmel AVR, en función del modelo de placa llevará un microcontrolador u otro, los más usados son el Atmega168, Atmega328, Atmega1280 y el ATmega8. En los últimos años también se han empezado a implementar microcontroladores ARM de 32bits. Una de las características que hacen que Arduino sea un sistema muy extendido y muy valorado, es su alta conectividad, y es que saca al exterior todas las salidas del microcontrolador mediante pines de conexión rápido, permitiendo conectar los cables como lo hacemos con los protoboard. Además se ha creado un estándar en cuanto a la distancia y posición de estos pines, esto es así, para la implementación de todo tipo de “shield”, por ejemplo un shield muy útil es el "4 a 20 mA" el que es de gran ayuda para interconectar sensores de características industriales con la placa arduino.

VENTAJAS DE ARDUINO

- **Barato:** Las tarjetas Arduino son baratas comparadas a otros sistemas. Además, al ser una plataforma de hardware libre, hay muchos clones de tarjetas Arduino a precios realmente bajos.
- **Multiplataforma:** El IDE se puede instalar en Windows, Macintosh OSX y Linux.
- **Ambiente de programación simple y clara:** Como ya se mencionó antes, el IDE es muy fácil de usar, tanto para principiantes como para usuarios avanzados.
- **Amplia comunidad y documentación:** Si buscan la palabra Arduino en google, aparecerán muchísimas páginas con documentación, ejemplos, tutoriales, etc. El IDE también cuenta con una gran variedad de ejemplos que están listos para subirse a una tarjeta.
- **Gran variedad de Shield:** Esto permite interconectar arduino con el mundo real.

SHIELD (MODULOS DE EXPANSIÓN)

Una gran parte del éxito de la plataforma Arduino reside en la gran variedad de shield y sensores que podemos encontrar en el mercado, estos dan realmente la funcionalidad a Arduino y hacen mucho más sencilla la creación de cualquier proyecto electrónico.

Para explicarlo de forma sencilla, podríamos decir que si Arduino es el cerebro del sistema, un shield puede llegar a ser desde el musculo hasta los sentidos del sistema. La placa Arduino tan solo puede trabajar con pequeñas señales, además esta placa no tiene demasiada conectividad aparte de la conexión USB que posee y algún puerto serie interno del propio microcontrolador, por ello y a modo de que cada cual adapte Arduino a sus necesidades, surgen los shields, hay desde shield para controlar motores o elementos de potencia, hasta shields que dan conectividad, wifi, GSM o incluso bluetooth, esto hace que Arduino sea un sistema modular el cual podemos adaptar según las necesidades del proyecto.

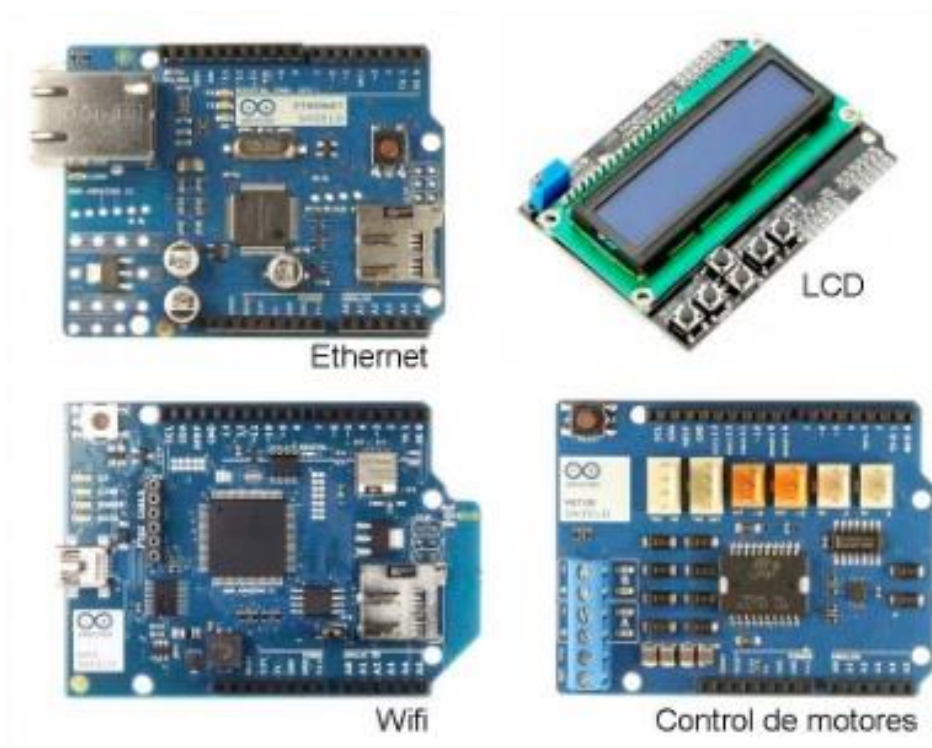


Figura N°13:“Shield para arduino”

LENGUAJE DE PROGRAMACIÓN

Arduino se programa mediante un editor de texto o “sketch” en el cual usa un lenguaje propio y creado expresamente para la plataforma Arduino, este lenguaje, está basado en “Processing”, que es un lenguaje de alto nivel. El Sketch es un entorno de programación sencillo el cual no tiene más que las opciones básicas de programación, verificación del código y carga en la placa Arduino.

```

barGraph | Arduino 1.0.4
Archivo Editar Sketch Herramientas Ayuda

barGraph

const int analogPin = A0; // the pin that the potentiometer is attached to
const int ledCount = 10; // the number of LEDs in the bar graph

int ledPins[] = {
  2, 3, 4, 5, 6, 7,8,9,10,11 }; // an array of pin numbers to which LEDs are attached

void setup() {
  // loop over the pin array and set them all to output:
  for (int thisLed = 0; thisLed < ledCount; thisLed++) {
    pinMode(ledPins[thisLed], OUTPUT);
  }
}

void loop() {
  // read the potentiometer:
  int sensorReading = analogRead(analogPin);
  // map the result to a range from 0 to the number of LEDs:
  int ledLevel = map(sensorReading, 0, 1023, 0, ledCount);

  // loop over the LED array:
  for (int thisLed = 0; thisLed < ledCount; thisLed++) {
    // if the array element's index is less than ledLevel,
    // turn the pin for this element on:
    if (thisLed < ledLevel) {
      digitalWrite(ledPins[thisLed], HIGH);
    }
  }
}
1 Arduino Uno on COM1

```

Figura N°14: “Entorno de programación de arduino”

VARIABLES FÍSICAS INVOLUCRADAS

Las variables físicas dentro de un proceso se dividen en:

- Variables a controlar
- Variable medida
- Variable manipulada
- Perturbaciones

En el siguiente diagrama se muestran las variables involucradas en este tipo proceso.

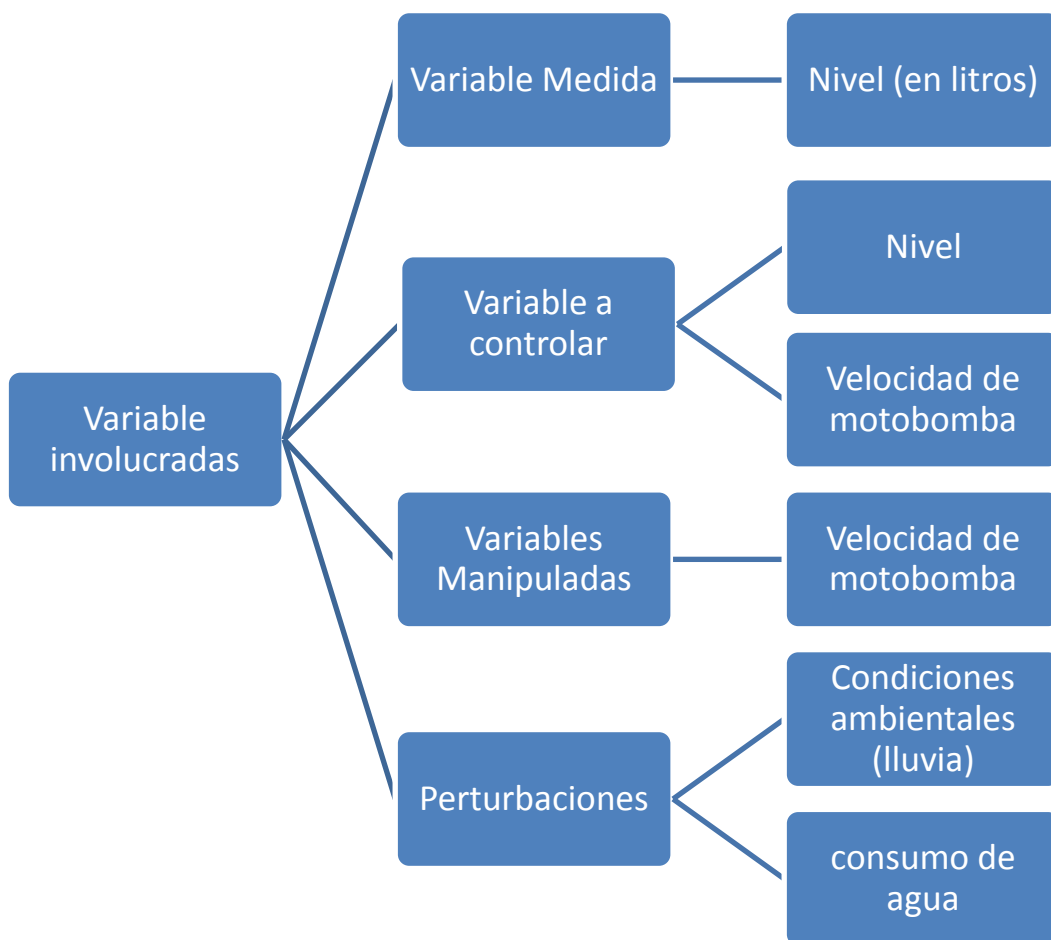


Figura N°15:“Variables involucradas”

VARIABLES A CONTROLAR

Son los parámetros más importantes dentro del proceso industrial, estos se deben mantener fijos, sin alteraciones, ya que sus variaciones podrían alterar las mediciones del sistema.

El monitoreo a través de los sensores es lo primordial para dar inicio al control.

NIVEL

El nivel es el aumento o disminución de alguna sustancia en un contenedor natural o artificial. En el proceso a realizar, es la variable más importante que se necesita conocer.

Esta medida sirve para determinar el contenido de los tanques para accionar dispositivos de alarma y seguridad en los recipientes a presión, para el accionamiento de válvulas y vertederos en la regulación de las centrales hidroeléctricas, para la determinación de la altura de la lámina en los vertederos de medidas, etc. En la industria química la medida de nivel se requiere para determinar la cantidad exacta de líquidos que hay que administrar en un proceso de mezcla.

La medida del nivel puede ser necesaria con mucha o poca precisión, con indicación del nivel instantáneo o con registro continuo de la medida, con medición local o transmisión a distancia de unos centenaes o miles de metros.

En el caso del proceso a realizar, se adquirirá la variable de nivel de forma remota, para el cual hay varios tipos de sensores que se podrían utilizar, pero específicamente se trabajara con un sensor hidrostático.

VELOCIDAD

La velocidad es una magnitud física de carácter vectorial, que expresa la distancia recorrida de un objeto por unidad de tiempo. Su unidad en el sistema internacional de unidades es el metro por segundo (m/s).

En virtud de su carácter vectorial, para definir la velocidad deben considerarse: la dirección del desplazamiento y el módulo, el cual se denomina celeridad o rapidez.

De igual forma que la velocidad es el ritmo o tasa de cambio de la posición por unidad de tiempo, la aceleración es la tasa de cambio de la velocidad por unidad de tiempo.

La variable física “Velocidad”, se deberá regular en el proceso del variador de frecuencia del sistema.

CAPITULO N°3: CONTROLADOR A IMPLEMENTAR

CONTROLADOR PID

Un controlador PID, es un sistema que cuenta con una realimentación, la cual compara el valor medido con el valor deseado. Este tipo de control es usado en la mayoría de los procesos industriales.

$$\text{Controlador PID} = y(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt}$$

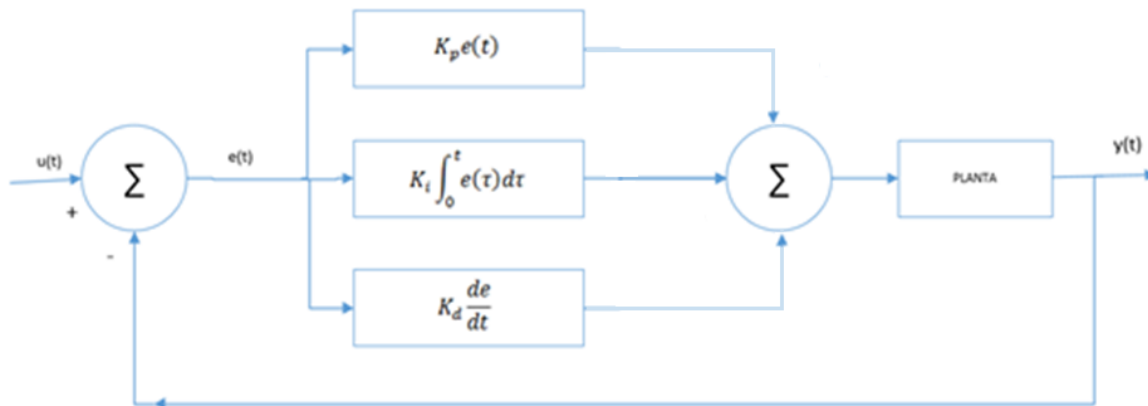


Figura N°16: “Diagrama de bloque del controlador PID”

DESCRIPCIÓN DE CADA UNA DE SUS ACCIONES

$$\text{Acción Proporcional} = K_p e(t)$$

La salida de un controlador proporcional, es el resultado del producto entre la señal de error y la ganancia proporcional.

$$\text{Acción Integral} = K_i \int_0^t e(\tau) d\tau$$

La acción de control integral tiene como propósito eliminar el error en estado estacionario, provocado por la acción proporcional. El control integral actúa cuando hay una desviación entre la variable y el punto de consigna, luego integra esta desviación en el tiempo y la suma a la acción proporcional.

$$\text{Acción Derivativa} = K_d \frac{de}{dt}$$

Cada vez que ocurre un cambio en el valor absoluto del error, la acción derivativa se hará presente. El *error*, es la desviación existente entre el punto de medida y el Set Point. La función de la acción derivativa es mantener el error al mínimo, corrigiéndolo proporcionalmente con la misma velocidad que se produce; evitando el incremento de este.

DESCRIPCIÓN DE CADA CONSTANTE UTILIZADA

K_p "constante de proporcionalidad": se puede ajustar como el valor de la ganancia del controlador o el porcentaje de banda proporcional.

K_i "constante de integración": indica la velocidad con la que se repite la acción proporcional.

K_d "constante de derivación": hace presente la respuesta de la acción proporcional duplicándola, sin esperar a que el error se duplique. El valor indicado por la constante de derivación, es el lapso durante el cual se manifestará la acción proporcional correspondiente a 2 veces el error y después desaparecerá.

MÉTODOS PARA SINTONIZAR EL CONTROLADOR

METODO EMPÍRICO:

Uno de los métodos usados para la sintonización es llamado empírico o heurístico. Este método sigue reglas generales para obtener resultados aproximados o cualitativos. Hoy en día la mayoría de los PID son sintonizados con este método, ya que el procedimiento es más rápido y simple.

Los pasos que se siguen se explican a continuación:

1. Configurar el controlador solamente con acción proporcional, fijando la ganancia en 1 o cercana a 1. No se debe olvidar configurar las acciones integral y derivativa al mínimo.
2. Incrementar la ganancia del controlador hasta obtener oscilaciones estables, para esto se cambia el valor de setpoint de arriba hacia abajo tantas veces como sea necesario para provocar dichas oscilaciones.
3. Cuando la última ganancia sea determinada, fijar la ganancia del controlador a la mitad de ese valor.
4. Ahora para la constante integral se repite el paso 2 y el valor obtenido debe ser 3 veces menor, es decir, dividido en 3.
5. y para la parte derivativa debe se repite el paso 4.

Este método de sintonización PID es de gran ayuda si uno es consciente de cuál es la característica del proceso que se está controlando, se debe tener una idea de la aplicabilidad de las acciones P,I y D para una característica en particular.

METODO DE SINTONIZACIÓN ZIEGLER-NICHOLS

Con este método, la respuesta de la planta a una entrada escalón unitario se obtiene de manera experimental, tal como se muestra en la Figura N°17. Si la planta no contiene integradores ni polos dominantes complejos conjugados, la curva de respuesta escalón unitario puede tener forma de S, como se observa en la Figura N° 18. Este método se puede aplicar si la respuesta muestra una curva con forma de S. Tales curvas de respuesta escalón, se pueden generar experimentalmente o a partir de una simulación dinámica de la planta. La curva con forma de S se caracteriza por dos parámetros: el tiempo de retardo L y la constante de tiempo T . El tiempo de retardo y la constante de tiempo se determinan dibujando una recta tangente en el punto de inflexión de la curva con forma de S y determinando las intersecciones de esta tangente con el eje del tiempo y con la línea $c(t) \%K$, tal como se muestra en la figura N° 18:

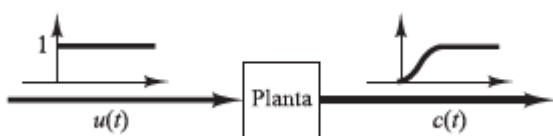


Figura N°17: “Escalón Unitario vs respuesta a un escalón”

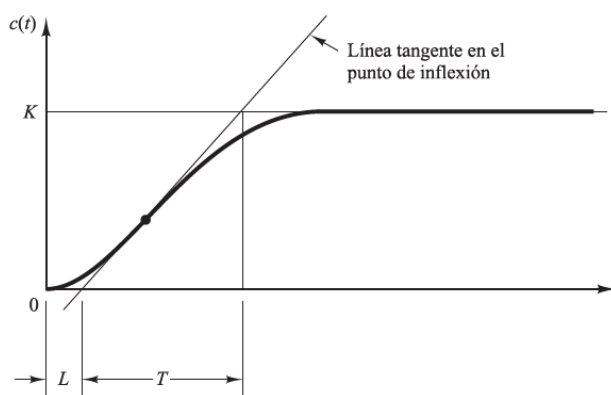


Figura N°18: “Escalón Unitario vs respuesta a un escalón”

Tabla N°1: “Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta K_p , T_i , T_d ”

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

La constante **Kp** corresponde a la ganancia proporcional, **Ti** es la constante de tiempo integral y **Td** es la constante de tiempo derivativa. En el caso de tener el controlador PID configurado con las ganancias integrales **Ki** y derivativa **Kd** en vez de los tiempos **Ti** y **Td**, hay que tener en cuenta las siguientes relaciones:

$$K_i = \frac{K_p}{T_i}$$

$$K_d = K_p * T_d$$

Con lo cual la tabla de valores para ajustar el controlador PID será la siguiente:

Tabla N°2: “Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta Kp, Ki, Kd”

Tipo de controlador	K_p	K_i	K_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{T/L}{L} * 0,27$	0
PID	$1.2 \frac{T}{L}$	$\frac{T/L}{L} * 0,60$	$T * 0,60$

CAPITULO N°4: DESARROLLO DEL HADWARE DE LA PROPUESTA

DESCRIPCIÓN DE LA PROPUESTA

Para desarrollar la solución a la problemática planteada, se utilizan dispositivos electrónicos de bajo costo para reemplazar el controlador (PLC) usado actualmente, ya que este es uno de los elementos más costosos del lazo de control. Además se propone utilizar un nuevo protocolo de comunicación de datos (ESTÁNDAR 802.11), ya que este es más asequible en comparación a los dispositivos GPRS que utilizan actualmente.

Para lo anterior se propone utilizar todo lo que nos entrega la tecnología arduino, sin embargo se debe dejar en claro que estos elementos no poseen características industriales, pero se busca desarrollar el sistema de control lo más cercano a lo que se usa en terreno. Si las pruebas son satisfactorias, el siguiente paso será darle las características físicas necesarias para implementarlo en un ambiente industrial.

El objetivo principal, es que el arduino pueda controlar y monitorear el nivel de un estanque vía WIFI, manipulando un variador de frecuencia y una motobomba. Dicho sistema se encontrará enlazado a una página web, la cual será capaz de monitorear y ejecutar acciones de control en tiempo real.

HADWARE PARA DESARROLLO DE LA PROPUESTA.

ARDUINO MEGA 2560

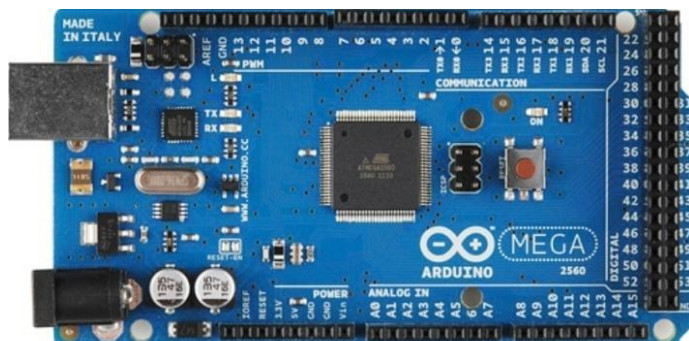


Figura N°19: “Arduino Mega 2560”

FUNCIÓN: Este dispositivo cumplirá la función de controlador del proceso (reemplazara la acción que ejecuta el PLC).

SHIELD RELAY



Figura N°20: “Shield Relay”

FUNCIÓN: Este elemento se utiliza como separador o activador de la etapa de control con la de fuerza. Este poseerá el contacto de “partir-parar” que se implementaran en la página web, los que podrán ser ocupados a criterio del operario.

SENSOR ULTRASÓNICO HC-SR04



Figura N°21: “Sensor de ultrasónico HC-SR04”

FUNCIÓN: Este sensor es capaz de medir distancia a un objeto a través de una señal de ultrasonido que envía y capta. Conociendo esa distancia se puede calcular el volumen del líquido, ya que la geometría del estanque es conocida.

CONVERSION DE PWM A "0V a 10V":



Figura N°22: “Convertor de PWM - 0v a 10v”

FUNCIÓN: Este circuito fue diseñado para poder interconectar el arduino con el Variador de frecuencia (VDF), debido a que este último solo posee entradas analógicas de estándares industriales como los son: "4 a 20 mA", "0v a 10v". Y arduino no las tiene, por esto se debe desarrollar este convertor.

VARIADOR DE FRECUENCIA "V1000 YASKAWA"



Figura N°23: “Variador de frecuencia”

FUNCION: Con este dispositivo se controla la velocidad de giro de la electrobomba a través de la variación de la frecuencia de trabajo de esta última. Gracias a este control de velocidad se logra variar el flujo de líquido que ingresa al estanque, por ende el nivel de este.

ELECTROBOMBA PK 60



Figura N°24: “Motobomba PKM 60”

FUNCION: Este será el elemento actuador del proceso que nos permitirá impulsar el fluido hacia el estanque siendo en mayor o menor cantidad dependiendo de la frecuencia que de trabajo que le entregue el variador de frecuencia.

SHIELD WIFI EPS8266



Figura N°25: “Shield Wifi EPS8266”

FUNCION: Este dispositivo es el encargado de enviar y recibir los datos del servidor para ser mostrados en la página web y modificar el Setpoint de proceso.

PROTOTYPE SHIELD

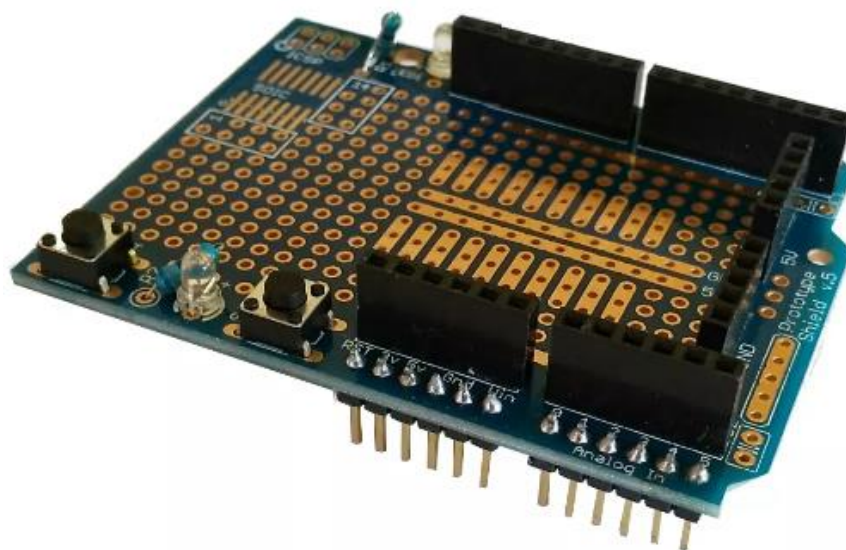


Figura N°26: “Prototype Shield”

FUNCION: Este es una placa base, el cual nos permite agregar conexiones extras.

CONEXION DE LOS ELEMENTOS INVOLUCRADOS.

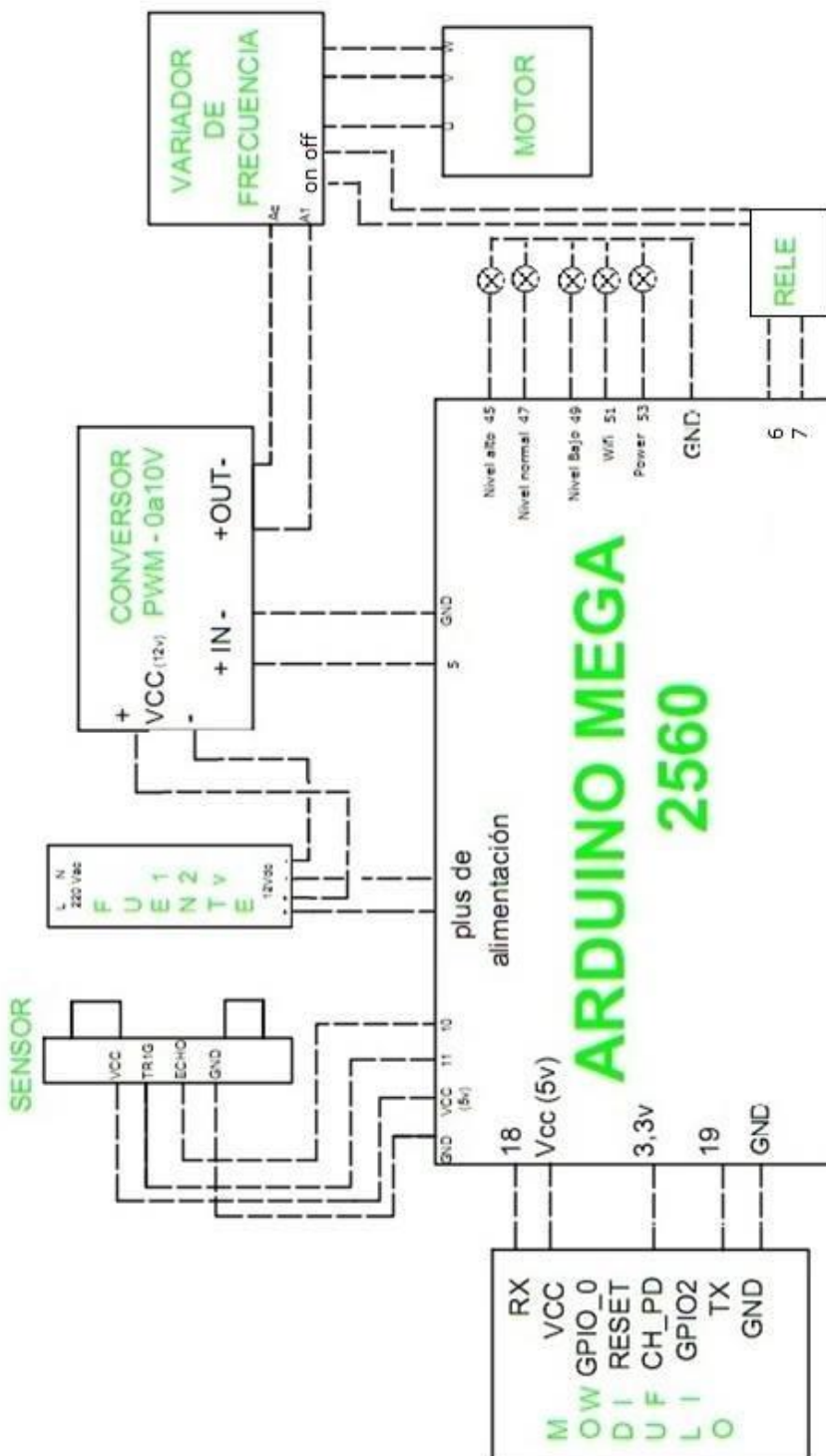


Figura N°27: “Conexión eléctrico de los elementos involucrados”

DESCRIPCION DE LA MAQUETA CONSTRUIDA

La maqueta desarrollada consiste en una representación a escala de un control de nivel de agua potable implementado por la empresas sanitarias de la región.

Este prototipo está compuesto por un estanque superior al que se le controla el nivel y un tanque inferior que será la reserva del líquido del cual se impulsará el fluido con una electrobomba, la que variará su frecuencia de trabajo para mantener el nivel deseado. También posee un sistema de retorno al depósito para simular el consumo de agua, el que es una perturbación en el sistema de control implementado (abierto en un 65%).



Figura N°28: “Maqueta construida”

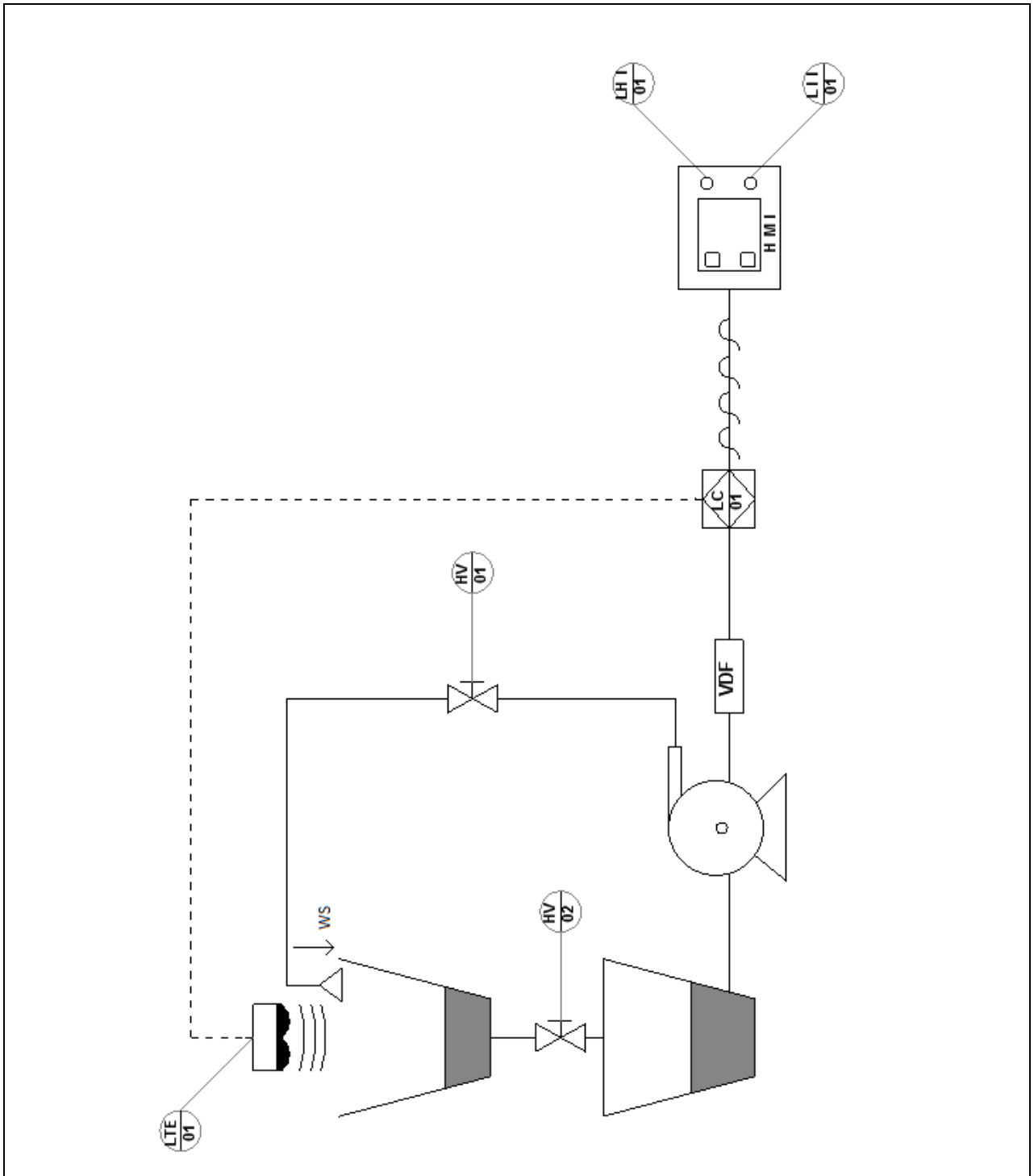


Figura N°29:“P&ID DEL PROCESO”

CAPITULO N°5: DESARROLLO DEL SOFTWARE DE LA PROPUESTA

DESCRIPCIÓN GENERAL

El arduino realiza la petición de lectura del sensor HC-SR04, este le devuelve la lectura del nivel (distancia) en centímetros, el cual se utiliza para calcular (por software) la cantidad de líquido contenido en el estanque. Además con esta información y en conjunto con el control PID se calcula la señal PWM. Una vez realizada esta operación se modifica la salida PWM con el conversor de 0 a 10v para que esta última pueda ser interpretada por el variador de frecuencia y así modificar la velocidad de la bomba.

A su vez, a través de wifi (SHIELD ESP8266), se envían los datos de nivel a una base de datos que se almacenan en un servidor. Cuando se ejecuta esta acción, el microcontralador recibe como respuesta el valor del último SP (Setpoint) y el estado de los relés. Esto se repite mientras el sistema se encuentre energizado.

Por el lado del servidor, la página web desarrollada toma los valores de la base de datos y los muestra de forma amigable para cualquier usuario (gráficos en tiempo real, histórico, Setpoint, botonera, cantidad de líquido en los estanque), pudiendo además modificar el valor de setpoint en todo momento y activar/desactivar el modulo Relé, con un tiempo de respuesta del sistema de 2 segundo aproximadamente.

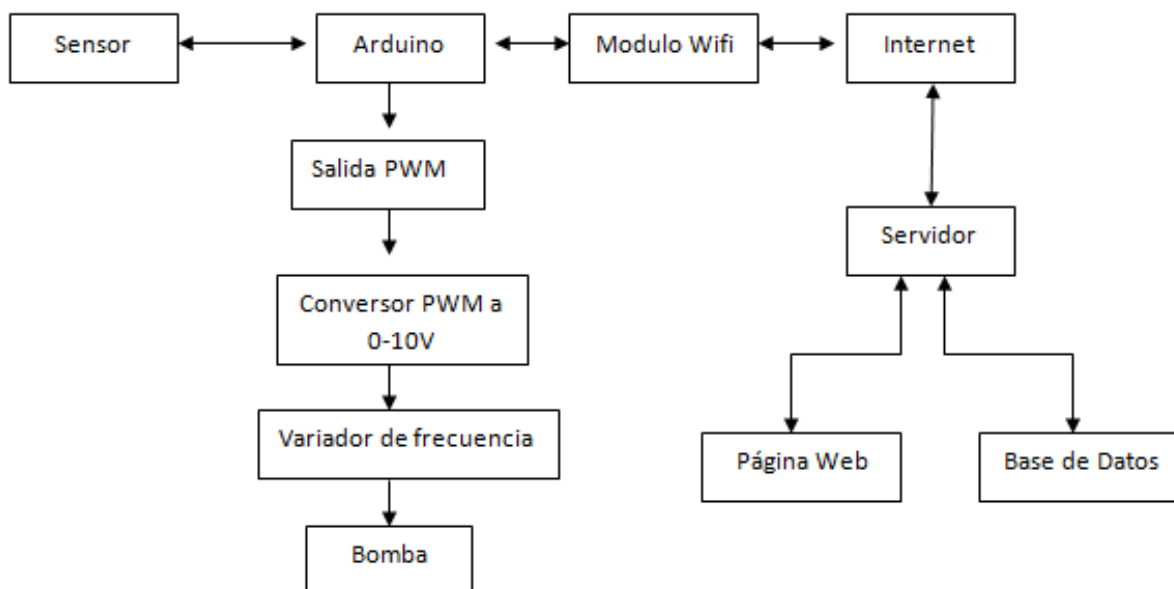


FIGURA N°30:“Diagrama de bloque de los elementos”

CONSIDERACIONES PARA EL DESARROLLO DEL ALGORITMO

- 1- La frecuencia de salida del variador de velocidad no debe ser inferior a 30hz, ya que bajo esta frecuencia según la construcción del motor el ventilador que lleva incorporado no cumple la función de enfriar debido a que la velocidad de este es demasiado baja por lo que los bobinados pueden aumentar su temperatura dañando el motor.
- 2- Bajo el 10% de la capacidad del estanque se considera nivel bajo. Lo que activa una luz de advertencia (roja) y también se ve reflejado en un gráfico en la plataforma web.
- 3- Sobre el 90% de la capacidad del estanque se considera como nivel alto lo que activa una luz de advertencia (roja) y además se ve reflejado en un gráfico en la plataforma web.
- 4- Entre el 10% y 90% de capacidad se considera un nivel normal, bajo esta condición mostrara una luz Verde para indicar el correcto funcionamiento.

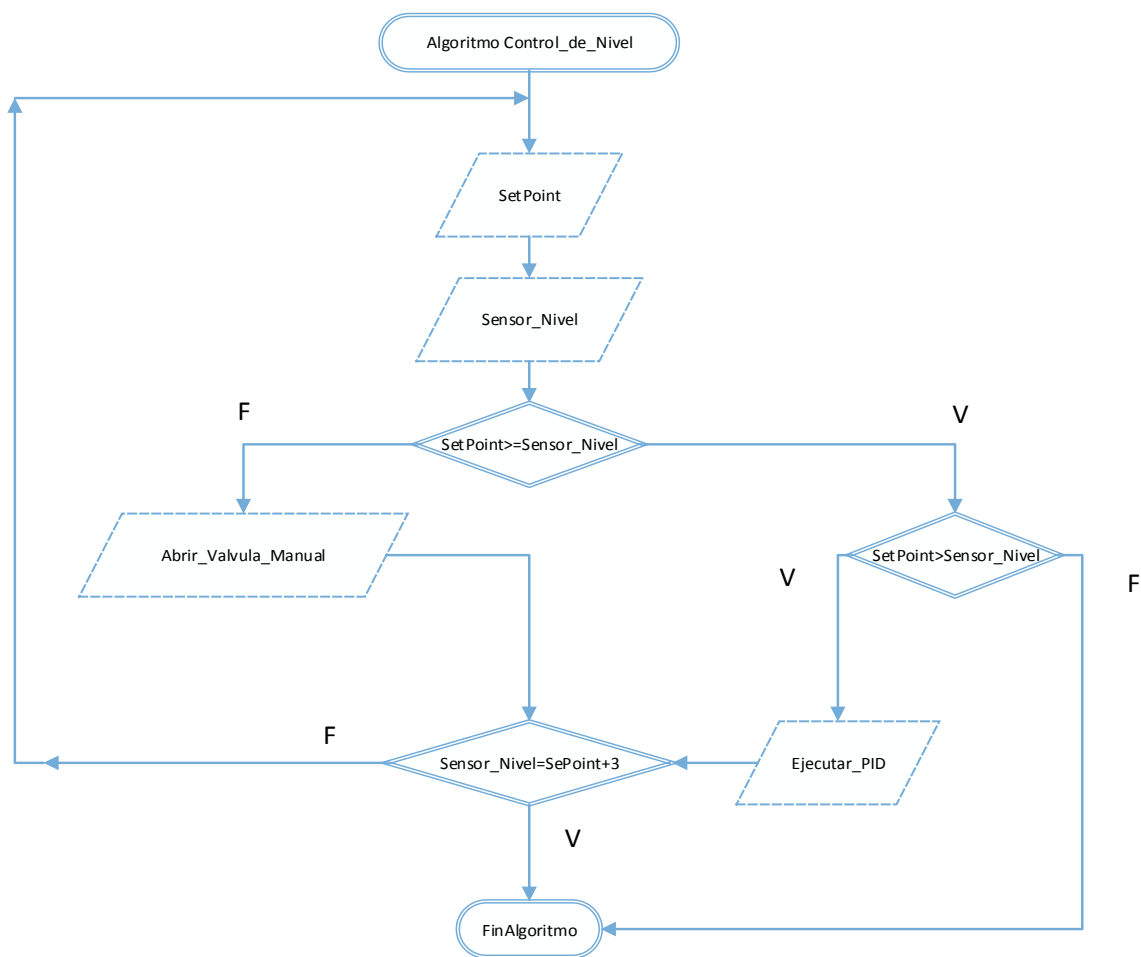


FIGURA N°31: “Diagrama de flujo del programa desarrollado”

PAGINA WEB DESARROLLADA PARA MONITOREO Y CONTROL



Figura N°32: “Interface Web del proceso”

ELEMENTOS DE LA INTERFACE.

- 1-** Lugar para ingresar setpoint deseado (expresado en porcentaje)
- 2-** Ultimo setpoint ingresado (expresado en porcentaje).
- 3-** Nivel del estanque en litros.
- 4-** Nivel del estanque expresado en porcentaje.
- 5-** Botones de partir (verde) y parar (rojo) que controlan los relé de la maqueta.
- 6-** selector para que el grafico que muestra la página web sea en tiempo real o histórico.
- 7-** selector para exportar los datos del grafico historio a un documento (Microsoft Excel).
- 8-** Grafico de la página web (gráfica: setpoint y nivel, individualmente o ambos al mismo tiempo con una capacidad de muestreo de 1 segundo aproximadamente.)

MODELAMIENTO MATEMATICO DEL PROCESO

En este punto se linealiza el estanque, alrededor de un punto de operación estático.

En base al modelo que obtenemos, se realizan diferentes tareas, como por ejemplo diseñar algún tipo de controlador que mantenga las especificaciones de nivel que necesitemos.

Tenemos una entrada y una salida, ambas variables, la entrada está regulada por una electrobomba y la salida, con una válvula (perturbación).

El objetivo es mantener el nivel del estanque constante, para esto se realizara una representación matemática de este.

MODELO ESTÁTICO DE NIVEL

Para determinar la velocidad de flujo que sale por una boquilla lisa y redonda como es mostrado en la figura, se aplica el principio Bernoulli entre el punto de referencia (1) en la superficie y el punto de referencia en la boquilla (2).

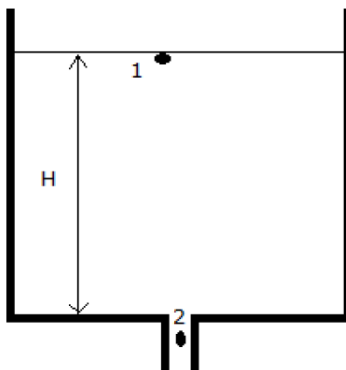


Figura N°33:“Esquema del Estanque”

Considerando la presión manométrica entre los puntos 1 y 2 como cero.

m: como la masa de las partículas de fluido entre el punto 1 y 2. Y

v: es la velocidad de salida de la partícula 1 y 2.

Se debe iniciar del principio de conservación de energía.

$$\frac{1}{2} * mv^2 = mgH.....(1)$$

$$v = \sqrt{2gH}.....(2)$$

El flujo de salida (Q) del tanque en este caso dependerá de la sección y de la velocidad de salida. En consecuencia al reemplazar la ecuación 2 en 3. Se observa que este flujo también varía dependiendo la altura del líquido y por ende de la gravedad que afecta al fluido.

$$Q = S * v \dots\dots\dots(3)$$

$$Q = S * \sqrt{2gH} \dots\dots\dots(4)$$

Analizando el flujo por una válvula:

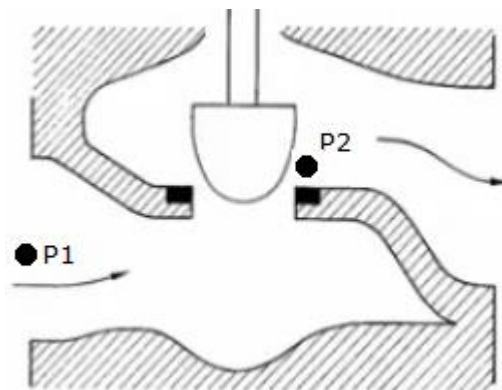


Figura N°34:“Estructura interna de la válvula”

De manera general el flujo que pasa por una válvula en estado estacionario se encuentra dado por:

$$Qv = Kv * As * \sqrt{\Delta P} \dots\dots\dots(5)$$

Donde:

Qv: Flujo a través de la válvula.

Kv: Constante (define el caudal máximo por segundo que pasa por la válvula).

As: Área de paso.

ΔP: Presión diferencial a través de la válvula P2-P, (P2 es la presión de salida y P1 presión de entrada).

Se puede concluir que el flujo que pasa por la válvula es proporcional al área de abertura de esta, en el caso que la diferencia de presión se considera constante. De manera práctica tomamos una válvula con un comportamiento inteligente, donde sea posible hacer una aproximación más o menos lineal entre flujo Qv y la abertura de la válvula (a2).

$$Qs = K_2 a_2 \sqrt{2gH} \dots\dots\dots(6)$$

Ahora se considera que el flujo de entrada Qe es proporcional a la frecuencia de la bomba que suministra el fluido a la entrada considerando un suministro constante tenemos

$$Qe = k_1 * f_1 \dots\dots\dots(7)$$

Entonces, el caudal de entrada está dado por una constante K_1 por la frecuencia de la bomba (ese valor se tomara entre 0-100% (0-50Hz)) y el caudal de salida Q_s está dado por la constante K_2 por la abertura de la válvula y por el fluido dentro del tanque.

Para comenzar a modelar el tanque se debe considerar dos casos:

- 1°: Se debe considerar el flujo de entrada variable y el de salida constante.
- 2°: Se debe considera el flujo de entrada constante y el de salida variable.

A continuación se asignan los valores a las constantes.

Área base= 33cm*33cm

Área base= 0,1089 m²

Hmax del estanque= 30 cm= 0,3m

K1= 0,05m³/s

K2= 0,015m³/s

f1= 0.45 (abertura de equilibrio de f1 (considerado entre 0%-100%)).

A2= 0.65 (abertura de equilibrio de a2 (considerado entre 0%-100%)).

g= 10m/s²

Q_e = Flujo de entrada

Q_s = Flujo de salida

Ahora con los valores dados, se procede a obtener el modelo matemático que representa el comportamiento de nivel dentro del estanque:

$$A \frac{dH}{dt} = Q_e - Q_s \dots \dots \dots (8)$$

$$Q_e - Q_s$$

$$Q_e = k_1 f_1 \dots \dots \dots (9)$$

$$Q_s = k_2 a_2 \sqrt{2gH} \dots \dots (10)$$

Reemplazando 9 y 10 en 8

$$A \frac{dH}{dt} = c f_1 - k_2 a_2 \sqrt{2gH} \dots \dots \dots (11)$$

Se quiere buscar un punto lineal ya que esta solución no es lineal, por esto, se busca un punto de equilibrio.

Donde el punto de equilibrio se logra al hacer el flujo de entrada igual al flujo de salida:

$$Q_e = Q_s \dots \dots \dots (12)$$

$$k_1 f_1 = k_2 a_2 \sqrt{2gH} \dots \dots \dots (13)$$

Nota: H_o = altura en equilibrio.

$$H_o = \left(\frac{k_1 f_1}{k_2 a_2 \sqrt{2g}} \right)^2 \dots \dots \dots (14)$$

$$\mathbf{H_o = 0.26m}$$

- Teniendo el punto de equilibrio H_o . Linealizamos por serie de Taylor ($a_2=cte.$ $Q_e=variable$ y $Q_s=cte.$)

$$f(f_1, h) = A \frac{dh}{dt} \dots \dots \dots (15) \text{ (función de Taylor)}$$

$$f(f_1, h) \approx f(\bar{f}_1, \bar{h}) + \frac{\partial f}{\partial f_1} \Delta f_1 \Big|_{(\bar{f}_1, \bar{h})} + \frac{\partial f}{\partial h} \Delta h \Big|_{(\bar{f}_1, \bar{h})} \dots \dots \dots (16)$$

$$\Delta f_1 = f_1 - \bar{f}_1 \dots \dots \dots (17)$$

$$\Delta h = h - \bar{h} \dots \dots \dots (18)$$

$$A \frac{dh}{dt} = k_1 f_1 - k_2 a_2 \sqrt{2gh} \dots \dots \dots (19) \text{ (ecuación de la dinámica del estanque)}$$

$$f(f_1, h) \approx K_1 \bar{f}_1 - k_2 a_2 \sqrt{2g\bar{h}} + K_1 \bar{f}_1 - \frac{k_2 a_2 \sqrt{2g}}{2\sqrt{\bar{h}}} \Delta h \dots \dots \dots (20)$$

Reemplazando 19 en 20 tenemos

$$f(f_1, h) \approx A \frac{dh}{dt} \Big|_{(\bar{f}_1, \bar{h})} + K_1 \Delta a_1 - \frac{k_2 a_2 \sqrt{2g}}{2\sqrt{\bar{h}}} \Delta h \dots \dots \dots (21)$$

Reemplazando 15 en 21 ahora tenemos

$$A \frac{dh}{dt} \approx A \frac{dh}{dt} \Big|_{(\bar{f}_1, \bar{h})} + K1 \Delta f_1 - \frac{k_2 a_2 \sqrt{2g}}{2\sqrt{\bar{h}}} \Delta h \dots \dots \dots (22)$$

$$A \frac{dh}{dt} - A \frac{dh}{dt} \Big|_{(\bar{f}_1, \bar{h})} \approx K1 \Delta f_1 - \frac{k_2 a_2 \sqrt{2g}}{2\sqrt{\bar{h}}} \Delta h \dots \dots \dots (23)$$

$$A \frac{d\Delta h}{dt} \approx K1 \Delta f_1 - \frac{k_2 a_2 \sqrt{2g}}{2\sqrt{\bar{h}}} \Delta h \dots \dots \dots (24) \text{ (ecuación ya linealizada)}$$

Ahora aplicamos transformada de Laplace para expresar la ecuación 24 como una función de transferencia.

$$A s h(s) = K1 f_1(s) - \frac{k_2 a_2 \sqrt{2g}}{2\sqrt{\bar{h}}} h(s) \dots (25)$$

Se despeja h(s) sobre f₁(s) para obtener la función de transferencia para ese caso.

$$\frac{h(s)}{f_1(s)} = \frac{K1}{\frac{k_2 a_2 \sqrt{2g}}{2\sqrt{\bar{h}}} + A s} \dots \dots \dots (26)$$

Reemplazando los valores de las diferentes constantes en la ecuación 26 se obtiene:

$$\frac{h(s)}{f_1(s)} = \frac{0,459}{s + 0,392} \dots \dots \dots (27)$$

- Con el mismo punto de equilibrio usado en el caso anterior linealizamos por Serie de Taylor considerando ahora (f=cte., por lo tanto Q_e=cte. y Q_s=variable)

$$f(a_2, h) \approx f(\bar{a}_2, \bar{h}) + \frac{\partial f}{\partial a_2} \Delta f_1 \Big|_{(\bar{a}_2, \bar{h})} + \frac{\partial f}{\partial h} \Delta h \Big|_{(\bar{a}_2, \bar{h})} \dots \dots (28)$$

Aplicando las ecuaciones 19 y 15 en 28 se obtiene

$$f(a_2, h) \approx A \frac{dh}{dt} \Big|_{(\bar{a}_2, \bar{h})} - k_2 a_2 \sqrt{2g \bar{h} \Delta a_2} - \frac{K2 \bar{a}_2 \sqrt{2g}}{2\sqrt{\bar{h}}} \Delta h \dots \dots \dots (29)$$

$$A \frac{d\Delta h}{dt} \approx -K2 \sqrt{2g \bar{h} \Delta a_2} - \frac{K2 \bar{a}_2 \sqrt{2g}}{2\sqrt{\bar{h}}} \Delta h \dots \dots \dots (30) \text{ (ecuación linealizada para el caso 2)}$$

Ahora aplicamos transformada de Laplace para expresar la ecuación 30 como una función de transferencia.

$$Ahs \approx -K2\sqrt{2g\bar{h}\Delta a2(s)} - \frac{K2\bar{a}2\sqrt{2g}}{2\sqrt{\bar{h}}} h(s).....(31)$$

Se despeja h(s) sobre a2(s) para obtener la función de transferencia para ese caso.

$$\frac{hs}{F_1(s)} \approx \frac{-K2\sqrt{2g\bar{h}}}{As + \frac{K2\bar{a}2\sqrt{2g}}{2\sqrt{\bar{h}}}} (32)$$

Reemplazando los valores de las diferentes constantes en la ecuación 32 se obtiene:

$$\frac{hs}{a_2(s)} \approx \frac{0,314}{s + 0,392} (33)$$

Ahora se saca la función de transferencia total realizando la suma de la ecuación 27 con la 33. Pero a la ecuación 33 le debe anteceder un signo negativo ya que esta es el caso es cuando el caudal de salida es variable siendo una perturbación (perdida) para el sistema.

$$H(s) = \frac{0,459}{s + 0,392} * F_1(s) - \frac{0,314}{s + 0,392} * a_2(s) (34)$$

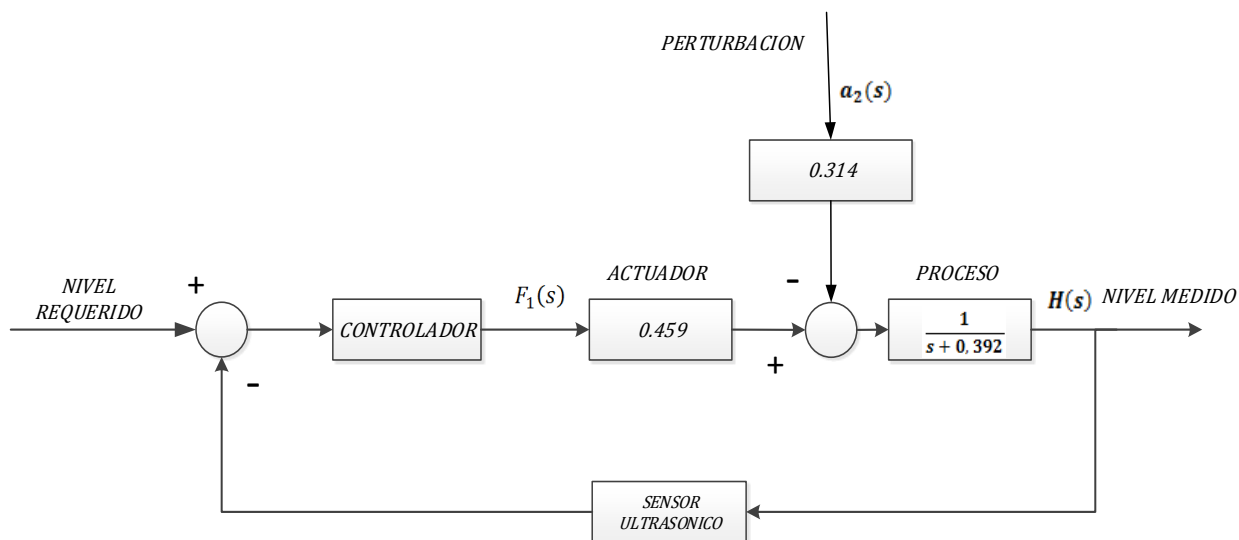


Figura N°35: "Diagrama de Bloques del Proceso"

Nota: La función de transferencia encontrada con el modelo anterior es válida solo para pequeñas variaciones de nivel alrededor del punto de equilibrio calculado.

SINTONIZACION DEL CONTROLADOR

Para la sintonización del sistema de control por el método de Ziegler-Nichols, se aplica un cambio de estado de “Set-Point” (10% a 70%). Con esto se obtiene una curva de transferencia que describe como trabaja el proceso, luego se traza una recta tangente en el punto de inflexión de la curva para obtener los valores “L y T”. A continuación se procede a calcular las constantes K_p , K_i , K_d y se obtiene una nueva curva de transferencia, la que debe ser más eficiente que la anterior.

A continuación se muestran los resultados de los gráficos paso a paso:

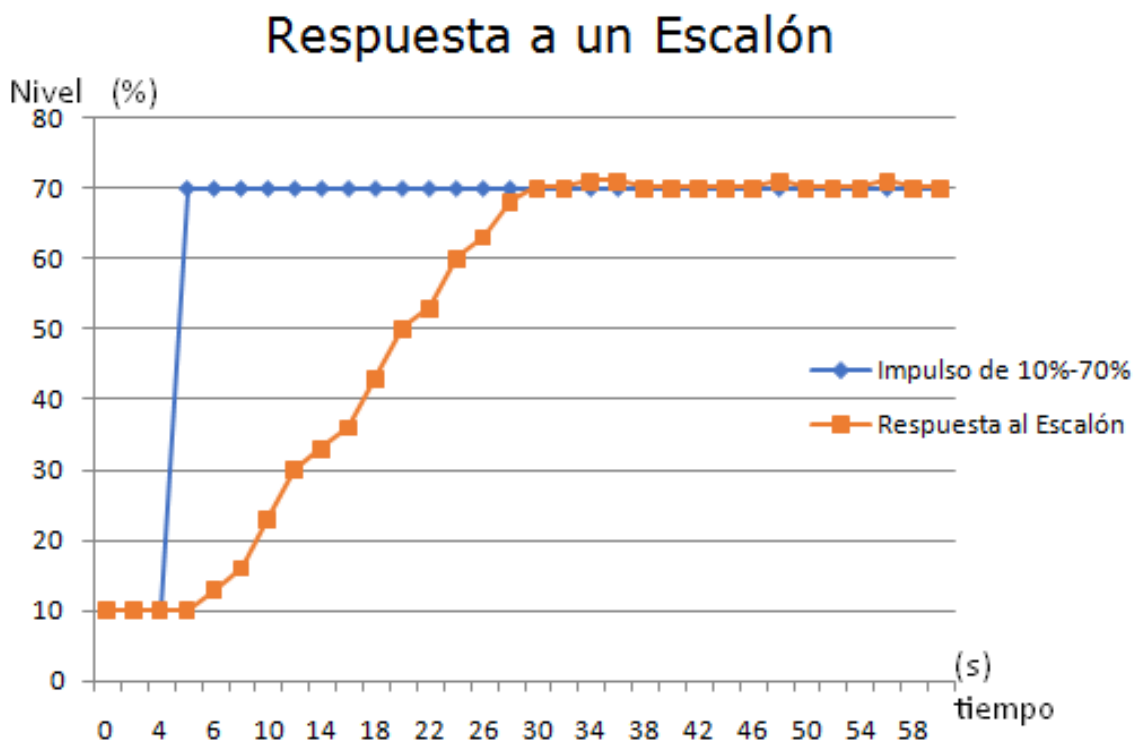


Figura N°36: "Comportamiento del sistema antes de sintonizar"

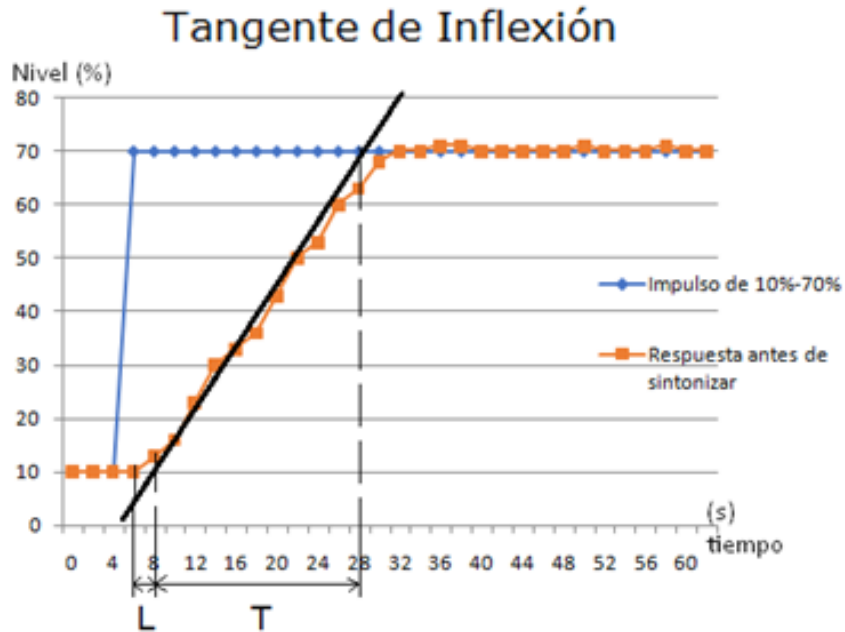


Figura N°37: "Grafico para calcular Parámetros PID"

Nota: Por errores de dispositivos, el punto de inflexión es aproximado, ya que la curva de transferencia es tomada directamente del proceso.

Aplicando el método número dos de sintonización anteriormente mencionado se obtienen las constantes para el controlador PID.

Del grafico anterior(figura N°36) se obtiene:

$$L = 3\text{seg} = 0,05\text{min}$$

$$T = 20\text{seg} = 0,33\text{min}$$

Entonces aplicando la tabla de Ziegler-Nicholsobtenemos:

$$K_p = 1,2 * \frac{T}{L} = 1,2 * \frac{20}{3} = 8$$

$$T_i = 2 * L = 2 * 3 = 6 \qquad K_i = K_p / T_i = 1,33$$

$$T_d = 0,5 * L = 0,5 * 3 = 1,5 \qquad K_d = K_p / T_d = 12$$

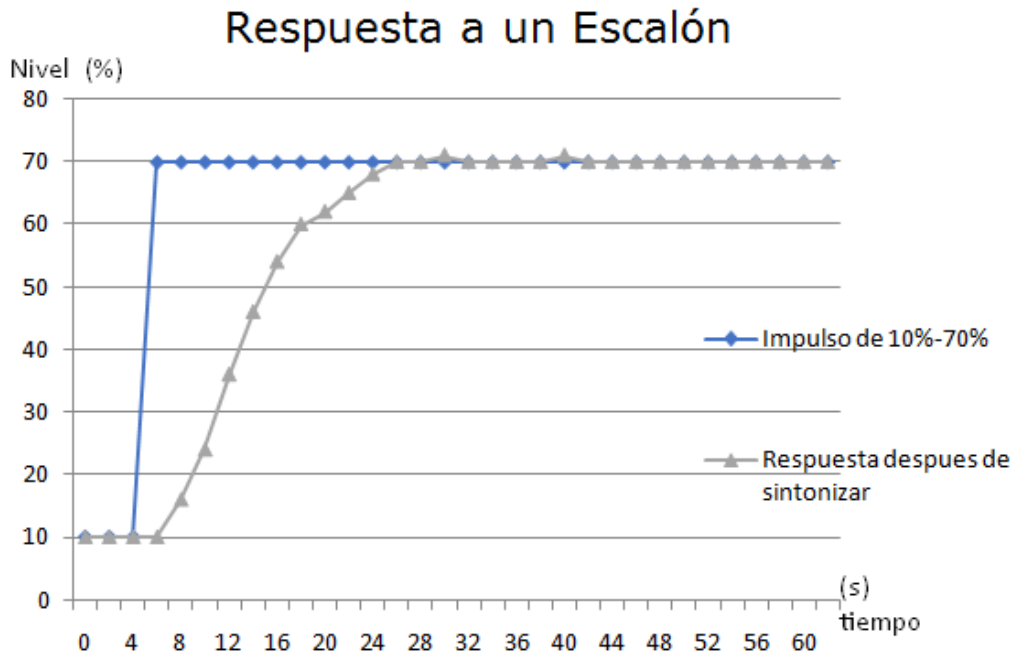


Figura N°38:“Comportamiento del sistema Después de sintonizar”

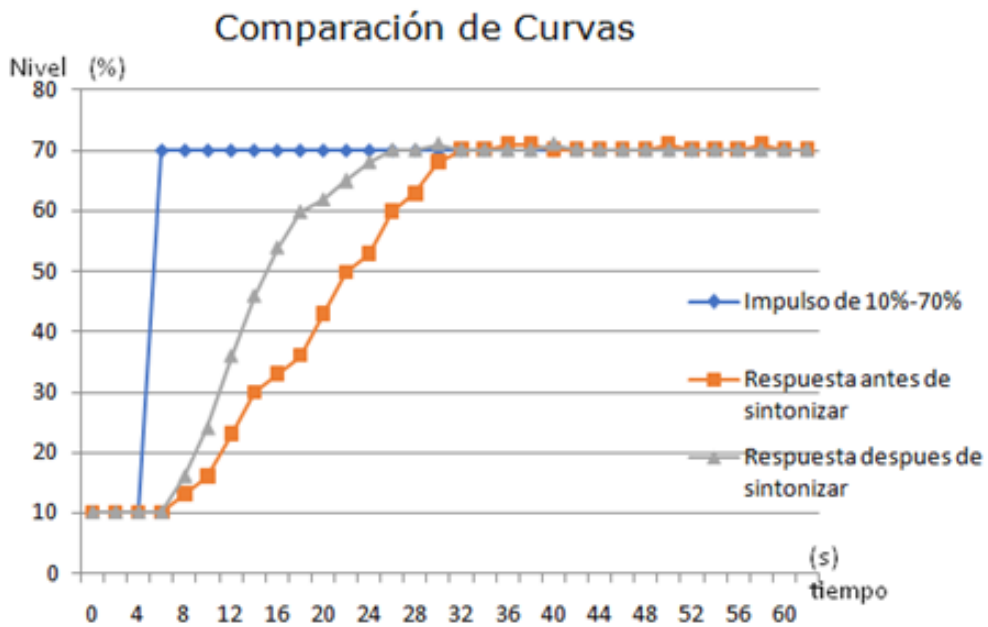


Figura N°39:“Grafico Comparativo antes y después de sintonizar”

Nota: Tras aplicar los parámetros del PID cálculo se observa en la figura anterior una gran mejora en el tiempo de repuesta del proceso para llegar al “Set point” establecido.

CONCLUSIONES

En el presente seminario de título, se desarrolló una propuesta innovadora de como sustituir el actual sistema de control de nivel que poseen algunas empresas sanitarias de la región, de una manera más sencilla y de bajo costo monetario. Los dispositivos utilizados por el momento no están estandarizados para la industria o ambientes hostiles. Sin embargo, su funcionalidad no deja indiferente a las compañías desarrolladoras de nuevas tecnologías.

Una de las propuestas que se implementan, es el uso del estándar 802.11, es decir, WI-FI. Este medio de comunicación es el más utilizado en la última década y hoy en día está tomando fuerza en los diferentes procesos industriales, ya que la utilización de cableado genera costos extras y necesitan una mantención frecuente. Además, implica menos costos de implementación que un sistema de comunicación que utilice por ejemplo protocolos GPRS.

El dispositivo principal con el que se trabaja es el Arduino, que presenta muchas ventajas, como lo son su fácil programación y la gran gama de módulos, pero tiene ciertos problemas, específicamente el rango de trabajo, el cual es acotado y esta propenso a sufrir daños, por ejemplo, a variaciones de la tensión de alimentación. Por otro lado su salida PWM necesita un conversor para poder relacionarse con otro tipo de dispositivos industriales, como por ejemplo variadores de frecuencia y sensores industriales.

Para el control realizado se calcularon las constantes K_p , K_i y K_d para la sintonización del PID, esto fue con el método de Ziegler-Nichols. Aunque ciertamente se podría utilizar el método empírico o “prueba y error”, este último es el más utilizado en la industria debido a la facilidad para encontrar los valores para un óptimo funcionamiento de algún sistema de control. A pesar de su facilidad, está sujeto a la experiencia del operador, debido a que es bastante visual el cálculo de las constantes de control, dicho método fue objeto de investigación y estudio, pero no se realizó debido a que no se considera un método con un respaldo científico.

Si se deseara implementar esta propuesta en un proceso industrial real, habría que encontrar un sustituto para el elemento primario en terreno, como lo es el sensor de nivel (que actualmente es solo un sensor ultrasónico sin estándares industriales), por un sensor de 4-20mA. Para ello la tecnología arduino si está capacitada, ya que posee un módulo de expansión que es capaz de procesar este tipo de señales para ser interpretadas por el microcontrolador. Cabe destacar que esta prueba fue realizada con éxito, sin embargo, para propósitos del proyecto, el sensor industrial no se pudo utilizar, ya que su rango de operación era muy sobredimensionado para el tamaño de nuestro estanque.

Ahora, si se quisiera controlar una electrobomba de mayor potencia no existiría inconveniente, ya que la salida PWM del arduino fue adaptada con éxito a un protocolo de comunicación industrial (0V-10V). Además, no existiría problema con el aumento de potencia del motor, dado que la señal de control que entrega arduino es transmitida a un variador de frecuencia, el cual actúa como puente entre la etapa de control y la etapa de potencia.

Sin embargo, todas estas ventajas se ven opacadas por la falta de protección física del arduino, ya que actualmente no posee algún tipo de carcasa que lo aisle de ruidos electromagnéticos, polvo, humedad y variables que puedan afectar su funcionamiento, pero es un problema menor, lo que se solucionaría desarrollando una placa de protección o tableros adecuados para su instalación en ambientes hostiles.

Con lo mencionado anteriormente, podemos concluir que la plataforma está capacitada para poder controlar procesos reales industriales no críticos (cambios abruptos de la variable de control), debido a que no es tan preciso en el proceso de adquisición y transmisión de datos.

BLIBLIOGRAFIA

1. Katsuhiko Ogata. (2010). Modelado matemático de sistemas de fluidos y sistemas térmicos. En Ingeniería de Control Moderna (101-105). España: Pearson.
2. Katsuhiko Ogata. (2010). Controladores PID y controladores PID modificados. En Ingeniería de Control Moderna (567-577). España: Pearson.
3. www.emb.cl/ “Protocolos de programación industrial”
4. www.arduino.cc “software de programación y librerías”
5. www.ardumania.es/ “Manual para programar arduino”
6. www.controlautomaticoeducacion.com “Modelamiento de sistemas”
7. www.tiendaarduino.com/datasheet “manuales arduino y shield”

ANEXOS

DISEÑO DEL CONVERTOR DE PWM – 0V A 10V

Para diseñar este convertor debemos conocer la tensión y frecuencia de trabajo de la salida PWM del arduino mega, esto está especificado en la data técnica del micro controlador o bien se puede obtener al realizar una medición con un osciloscopio.

Siendo estos valores los siguientes $f_{pwm} = 490Hz$

$$V_{pwm} = 5v_{max}$$

Lo que se busca es transformar los cambios de nivel de tensión media de la salida Pwm en una salida continua en el tiempo (señal analógica).

Entonces lo primero es establecer una frecuencia de corte para un filtro pasa Bajos esto es para eliminar todas aquellas frecuencias altas del sistema. Para ello debemos usar una frecuencia por lo menos 20 veces menor que la frecuencia Pwm.

$$f_{corte} = \frac{f_{pwm}}{20} = \frac{490Hz}{20} = 24,5Hz$$

Ahora conociendo esta frecuencia de corte, se utiliza la ecuación.

$$f_{corte} = \frac{1}{2 * \pi * R * C}$$

De la ecuación nos damos un valor de **condensador** “**el que debe ser pequeño para que pueda seguir rápidamente la dinámica de carga y descarga que producida la frecuencia de la señal PWM**”. Para este caso se da un valor de 4,7uF. Quedando de incógnita solo el valor de la resistencia

Nota: se da el valor del condensador ya que técnicamente es más fácil realizar una malla resistiva que una malla capacitada.

$$R = \frac{1}{2 * \pi * f_{corte} * 4,7uF} = 1382 \Omega$$

Una vez diseñado el filtro pasa bajos. Se aplica un amplificador no inversor de ganancia 2 para obtener una salida de 0v a 10v. Esta salida debe ser filtrada por un pequeño banco de condensadores para eliminar el rizado que se amplifica desde el filtro pasa bajos.

Teniendo en consideración que la ganancia para un amplificador no inversor está establecido por la ecuación

$$G = 1 + \left(\frac{R2}{R1}\right)$$

Se usa dos resistencia del mismo valor por ejemplo $R2=R1= 1k\Omega$, para obtener una ganancia de dos.

CIRCUITOS Y GRAFICOS

Para comprobar el funcionamiento del circuito conversor PWM, se procede a simular en Multisim. Se añade también algunos casos particulares para poder comparar la respuesta.

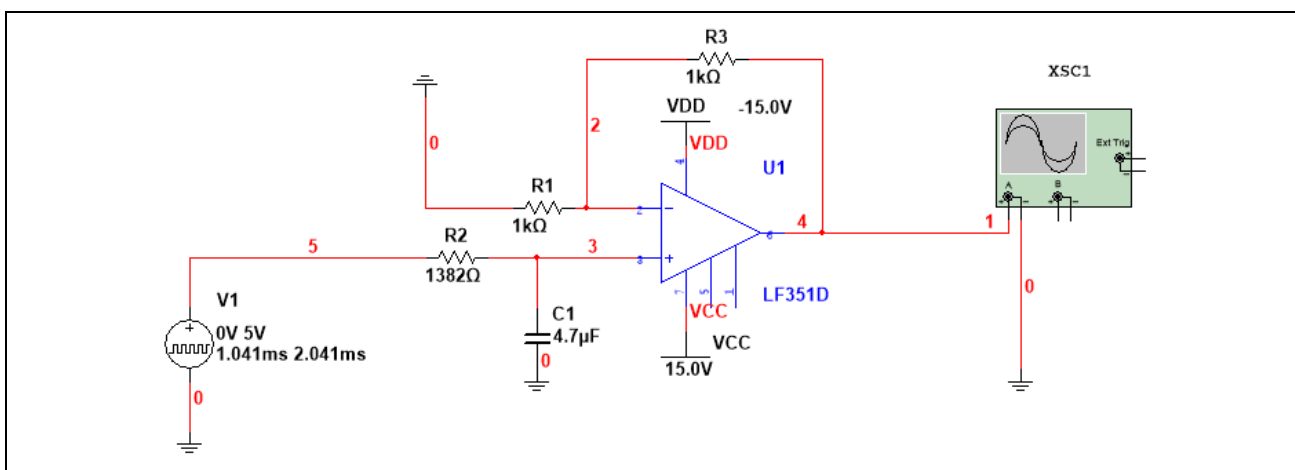


Figura N°40: “Circuito sin filtro en la salida del A.O”

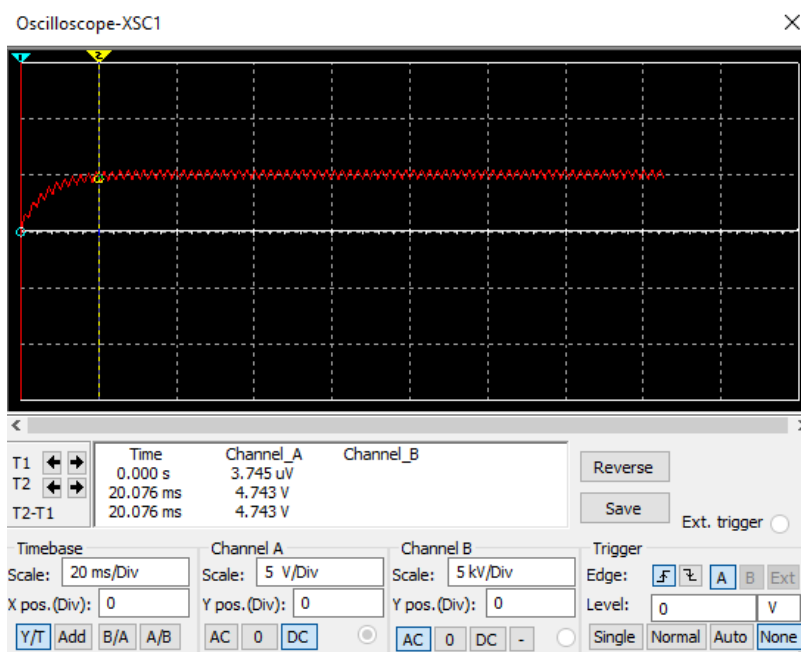


Figura N°41: “Grafico del circuito sin filtro en la salida del A.O.”

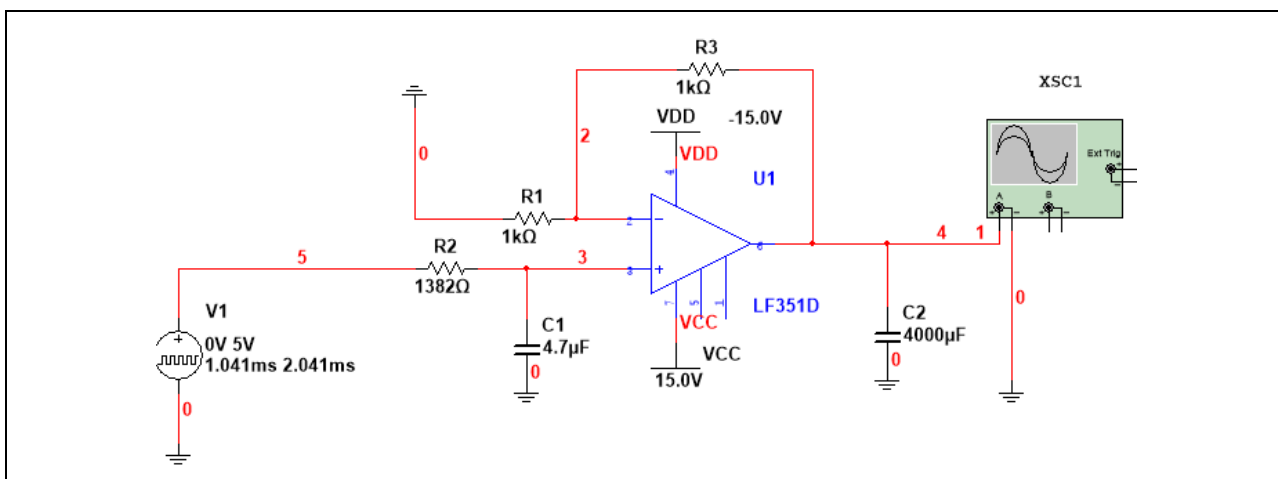


Figura N°42:“Circuito con un filtro a la salida del A.O.”

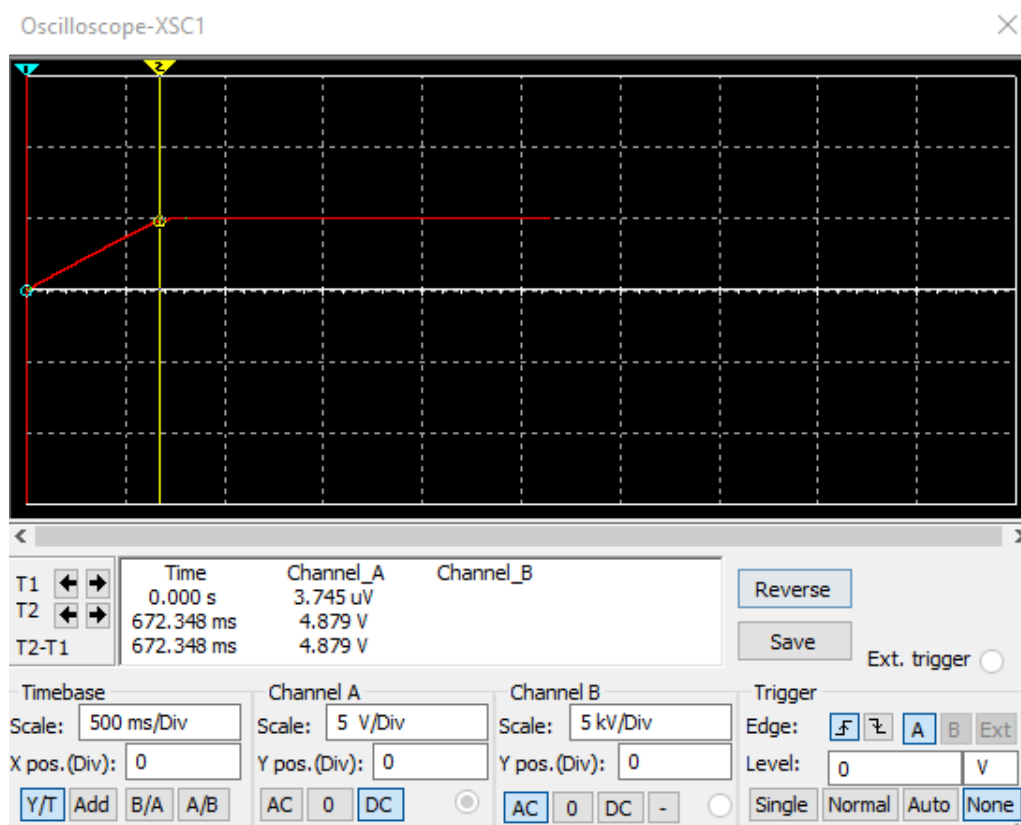


Figura N°43:“Grafico del circuito con filtro a la salida del A.O.”

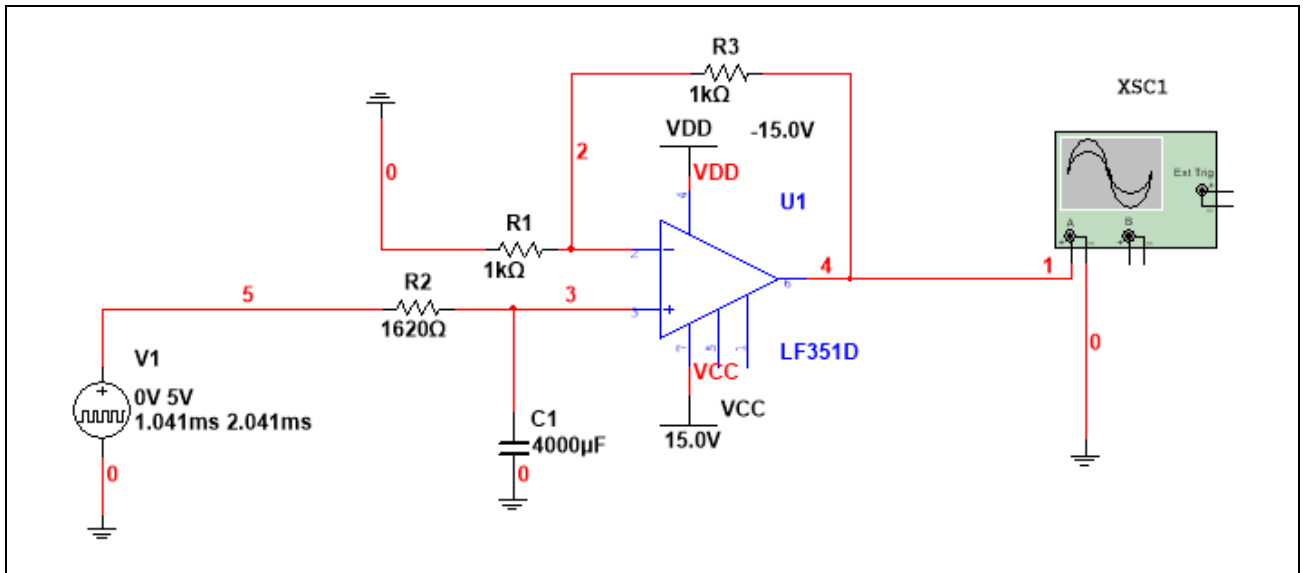


Figura N°44: “Circuito con capacitancia aumentada antes del A.O y sin filtro a la su salida”

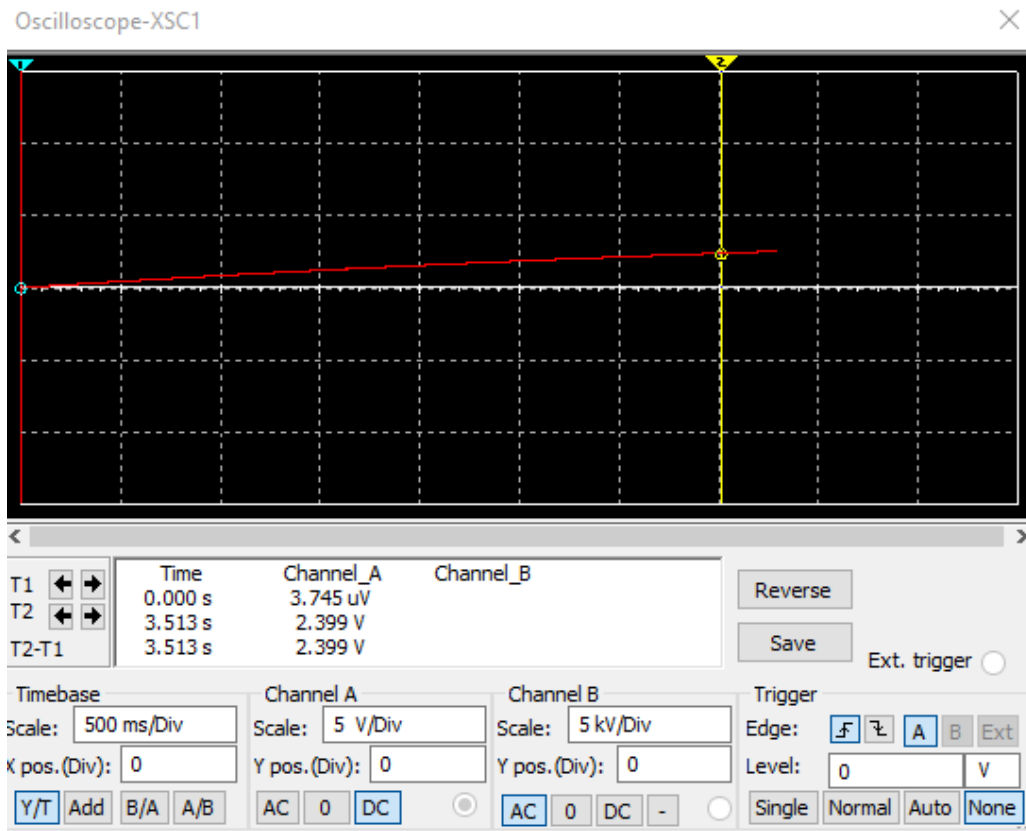


Figura N°45: “Grafico del circuito con capacitor grande antes del A.O y sin filtro a la su salida”

ANALISIS DEL CONVERTOR PWM

En la “figura N°40” se justifica el porqué del uso de un condensador pequeño en el filtro de entrada del amplificador operacional, donde se observa que el tiempo que toma la señal en llegar al valor de 5volts aproximadamente es de 20ms, sin embargo la señal de salida queda con un rizado el cual se debe eliminar a través de un filtro de salida como se muestra en la “figura N°41”. El resultado de esto se refleja en la “figura N°42”. Donde el tiempo que tarda en obtener un valor estable y sin rizado es de tan solo 600ms, lo que es bajo para la respuesta que se necesita.

En la “figura N°43” se aplica directamente el filtro de mayor capacidad a la entrada, Sin embargo al simular este caso se observa en la “figura N°44” que no es una buena opción ya que el tiempo que toma el circuito en llegar a la valor deseado es demasiado extendido. Como se ve en la gráfica, para llegar a la mitad del valor deseado (5volts) toma 3,5 segundos lo que desfavorece el proceso.



Figura N°46:“Circuito Conversor PWM - 0 a 10 Implementado”

DATA TÉCNICA DE LOS ELEMENTOS UTILIZADOS

ARDUINO MEGA 2560

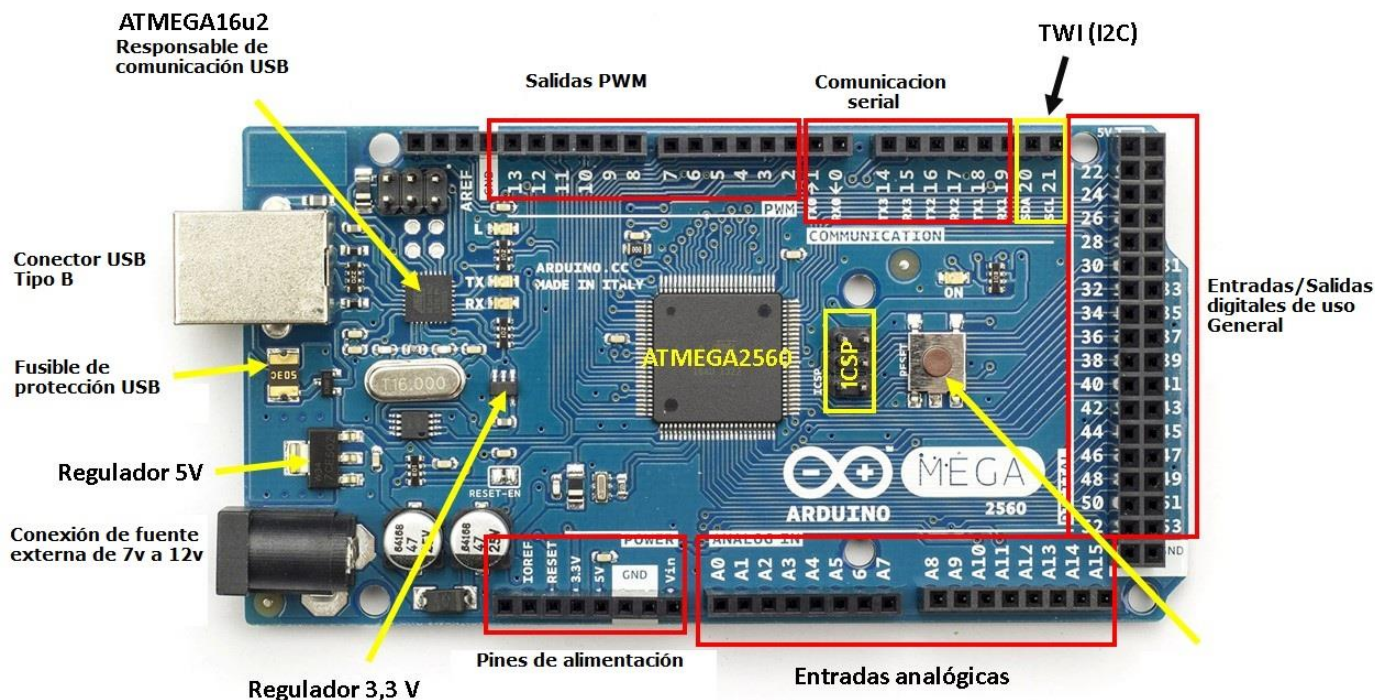


Figura N°47: “Diferentes partes del arduino Mega 2560”

ARDUINO MEGA POSEE LAS SIGUIENTES ESPECIFICACIONES:

Microcontrolador: ATmega2560

Voltaje Operativo: 5V

Voltaje de Entrada (límites): 6-20V

Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)

Pines análogos de entrada: 16

Corriente DC por cada Pin Entrada/Salida: 40 mA

Corriente DC entregada en el Pin 3.3V: 50 mA

Memoria Flash: 256 KB (8KB usados por el bootloader)

ClockSpeed: 16 MHz

VIN: A través de este pin es posible proporcionar alimentación a la placa.

5V: Podemos obtener un voltaje de 5V y una corriente de 40mA desde este pin.

3.3V: Podemos obtener un voltaje de 3.3V y una corriente de 50mA desde este pin.

GND: El ground (0V) de la placa.

MODULO WIFI SERIAL EPS8266

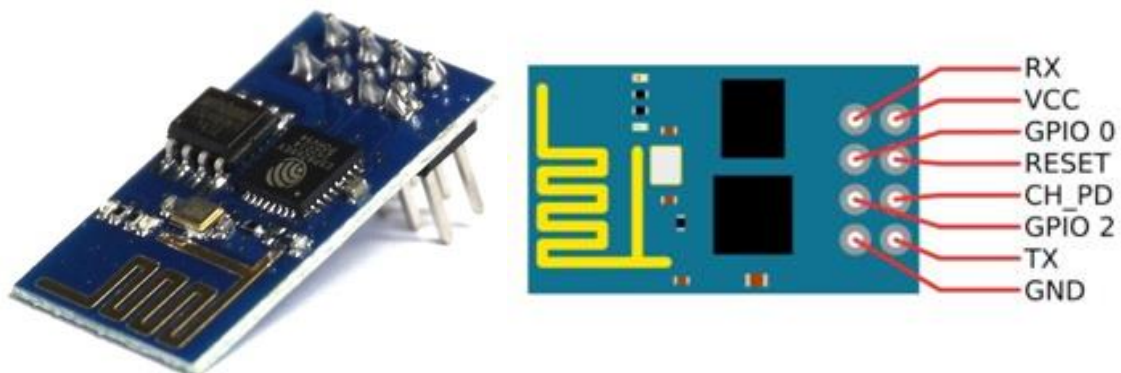


Figura N°48: “Modulo EPS8266 y su disposición de pines”

Características del módulo Wifi Serial ESP8266

- Protocolos soportados: 802.11 b/g/n
- Wi-Fi Direct (P2p), Soft Access Point
- Stack TCP/IP integrado
- PLL, reguladores y unidades de manejo de energía integrados
- Potencia de salida: +19.5dBm en modo 802.11b
- Sensor de temperatura integrado
- Consumo en modo de baja energía: <10 uA
- Procesador integrado de 32 bits, puede ser utilizado como procesador de aplicaciones

SHIELD RELE

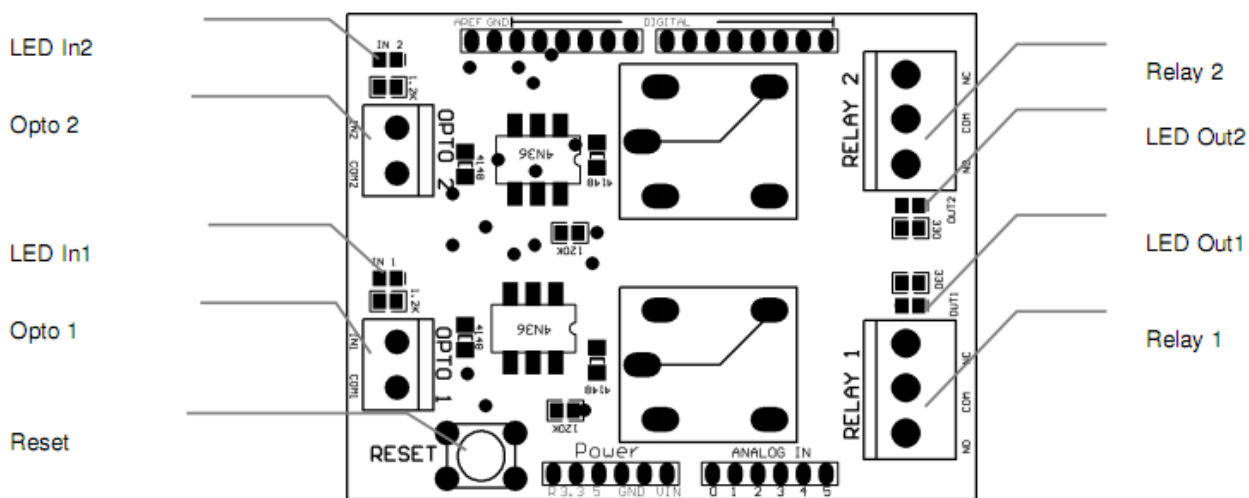


FIGURA N°49: “Partes del shield”

- Opto 1: Señal de entrada 1 opto-acoplada
- Opto 2: Señal de entrada 2 opto-acoplada
- LED In1: LED indicador de señal conectada a Opto 1 está en Alto
- LED In2: LED indicador de señal conectada a Opto 2 está en Alto
- Relay 1: Conexión a Relay 1
- Relay 2: Conexión a Relay 2
- LED Out2: LED indicador de salida Relay 2 está en Alto
- LED Out1: LED indicador de salida Relay 1 está en Alto
- Reset: Botón Reset placa base Arduino

Características específicas

- Salida con relay de hasta 10A/250VAC; 10A/30VDC
- Entrada opto-acoplada 30 VDC máx. conectadas a puertos de interrupción 0 y 1 de Arduino.
- Consumo máximo de 200 [mA]
- LED indicador de entrada optoacoplada en alto
- LED indicador de salida con relay en alto
- Botón de Reset de Arduino
- Dimensiones: 71 x 54 [mm]

SENSOR ULTRASONICO HC-SR04



Figura N°50: “Sensor ultrasónico HC-SR04”

- El módulo incluye transmisores ultrasónicos, el receptor y el circuito de control.
- Número de pines:

VCC: Alimentación +5V (4.5V min – 5.5V max)
TRIG: Trigger entrada (input) del sensor (TTL)
ECHO: Echo salida (output) del Sensor (TTL)
GND

- Corriente de reposo: < 2mA
- Corriente de trabajo: 15mA
- Ángulo de medición: 30°
- Ángulo de medición efectivo: < 15°
- Detección de 2cm a 400cm o 1" a 13 pies (Sirve a más de 4m, pero el fabricante no garantiza una buena medición).
- “Resolución” La precisión puede variar entre los 3mm o 0.3cm.
- Dimensiones: 45mm x 20mm x 15mm
- Frecuencia de trabajo: 40KHz

VARIADOR DE FRECUENCIA "V1000 YASKAWA"

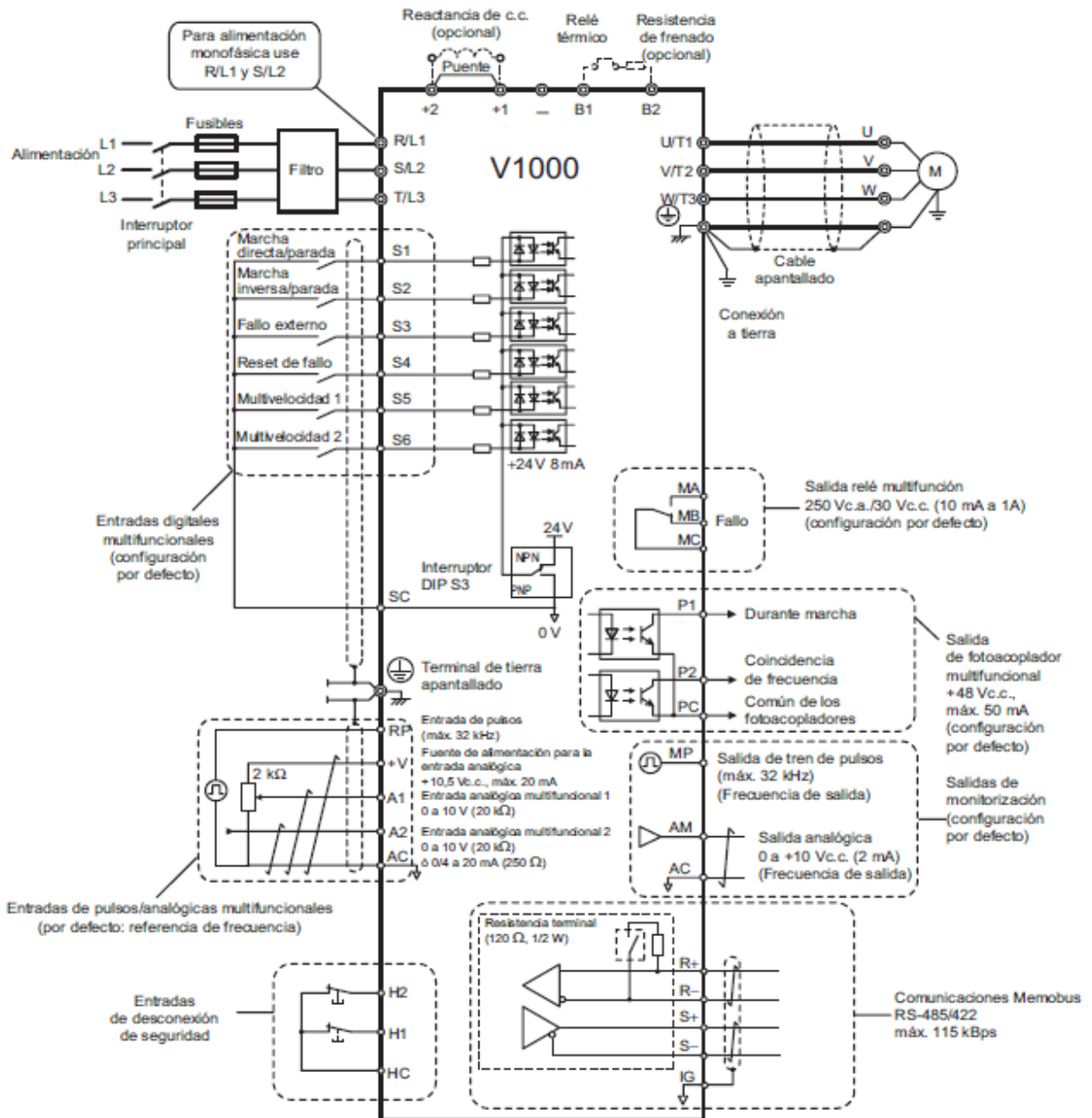


Figura N°51: "Conexión Eléctrica"

PROTOTYPE SHIELD

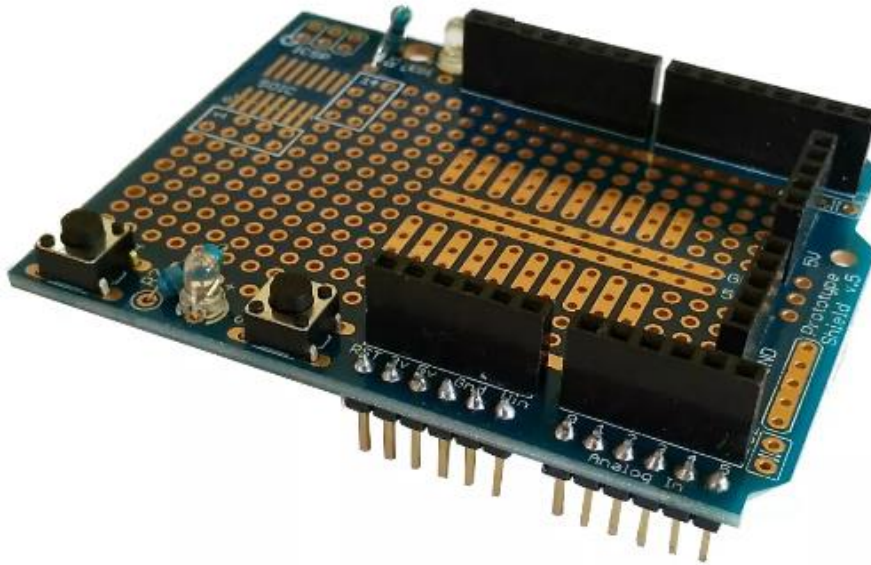


Figura N°52: “Prototype Shield”

Características

- Todos los pines de Arduino son visibles en la superficie.
- Se muestran los pines 5V, 3,3V, GND y Vin.
- Hay un socket para comunicación entre Arduinos mediante BlueSMiRF.
- 2 LEDs de uso general.
- 1 Botón de uso general.
- 1 Botón de Reset en la superficie de arduino.

ELECTROBOMBA PK 60



Figura N°53: “Electrobomba PKM 60”

Características

- capacidad de 40ltr/min
- altura máxima de aspiración de 38 metros
- temperatura del liquido -10°C a 60°C
- Presión máxima del cuerpo de la bomba de 6 bar
- Tensión de operación de 220v.
- Potencia de 0,5 HP

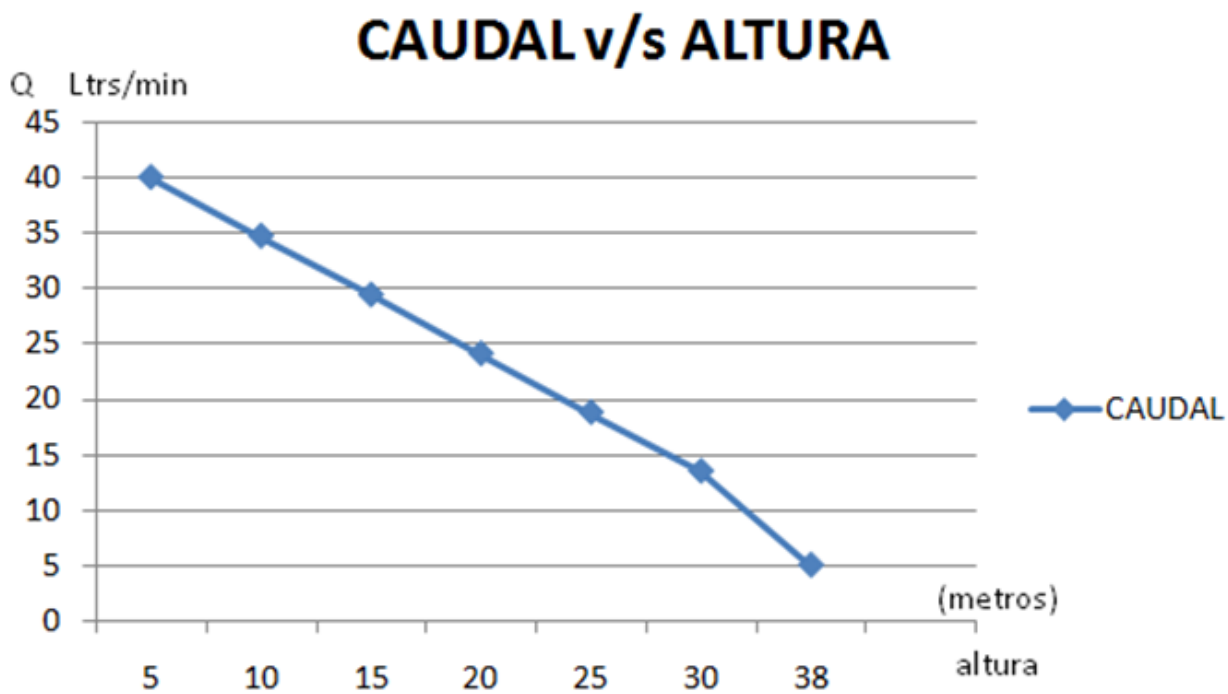


Figura N°54: “Curva de prestaciones”

COSTO DE LOS ELEMENTOS NECESARIOS PARAREALIZAR EL SISTEMA DE CONTROL.

Tabla N°3: “Costos de los elementos necesarios para desarrollar el lazo de control”

ITEM	CANTIDAD	PRECIO
Arduino Mega 2650	1	\$15000
Módulo Wifi ESP8266	1	\$7000
Relé Shield	1	\$2500
Prototype Shield	1	\$5000
Sensor HC-SR04	1	\$2000
Conversor PWM- 0va10v	1	\$7000
COSTO TOTAL		\$38500

PROGRAMA DESARROLLADO EN ARDUINO

```

#include<PID_v1.h>           //Librería PID //
#include<NewPing.h>         //Librería de sensor ultrasónico//

#define TRIGGER_PIN 11 //definición de Constantes y pines //
#define ECHO_PIN 10
#define MAX_DISTANCE 35

#define PINRELE 6

#define RUN 7

NewPingsonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // Inicia sensor//

// PID -----
#define PIN_OUTPUT 5
//Define Variables we'll be connecting to
double Setpoint, Input, Output;
double Kp=8, Ki=1,3, Kd=12;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
//-----

// WiFi -----
#define SSID "clown" // SSID
#define PASS "comida1234" // Contraseña Internet
#define HOST "192.168.43.154" // Webhost
//-----

// Variables -----
String comando; // Cadena de caracteres
String bufer;
String codigo;
String sp_string;
String estado_string;
intestado;
intch_pd = 5;
longsensor_nivel;
longnivel_porcentaje;
intsp;
//-----

// Leds
intledPower = 53;
intledWifi = 51;
intledNivelBajo = 49;

```

```

intledNivelNormal = 47;
intledNivelAlto = 45;

void setup() {
  Serial.begin(38400); // Inicializacomunicación serial con PC
  Serial1.begin(38400); //inicializa la comunicación serial y la velocidad con el módulo wifi
  pinMode(ledPower, OUTPUT); // define el modo de operación del pin
  pinMode(ledWifi, OUTPUT);
  pinMode(ledNivelBajo, OUTPUT);
  pinMode(ledNivelNormal, OUTPUT);
  pinMode(ledNivelAlto, OUTPUT);
  pinMode(PINRELE, OUTPUT);
  pinMode(RUN, OUTPUT);
  digitalWrite(ledPower, HIGH); // escritura de 0 o 5 V
  digitalWrite(ledWifi, LOW);
  digitalWrite(ledNivelBajo, LOW);
  digitalWrite(ledNivelNormal, LOW);
  digitalWrite(ledNivelAlto, LOW);

  sensor_nivel = sonar.ping_cm(); // devuelve en centímetros la variable sensor
  Input = mapear(sensor_nivel, 0, 30, 0, 100);
  Setpoint = 50;

  pinMode(ch_pd, OUTPUT); // configurapinMode como salida
  digitalWrite(ch_pd, LOW);
  delay(1000); // retardo 1 seg
  digitalWrite(ch_pd, HIGH);

  //Conecta wifi //
  conectar_wifi();
}

voidloop() {
  sensor_nivel = 30 - sonar.ping_cm(); //lee el sensor //
  nivel_porcentaje = mapear(sensor_nivel, 0, 30, 0, 100);
  Input = nivel_porcentaje; // traspaso de variables //
  revisarNivel();
  enviar_db();
  Serial.print("Nivel: ");
  Serial.print(sensor_nivel);
  Serial.print("cm - ");
  Serial.print(nivel_porcentaje);
  Serial.print("%. ");
  Serial.print("SP: ");
  Serial.print(Setpoint);
  Serial.print("% PWM: ");

```

```

myPID.Compute();
  Output = map(Output, 0, 255, 127, 212);
analogWrite(PIN_OUTPUT, Output);

Serial.print(mapear(Output, 0, 255, 0, 100));
Serial.println("%.");
if (Input > (Setpoint + 5)){
digitalWrite(RUN, LOW);
  }
if (Input < (Setpoint - 5)){
digitalWrite(RUN, HIGH);
  }
}

floatmapear(float x, float in_min, float in_max, float out_min, float out_max)
{
return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

voidenviar_db() { // Configura la conexión TCP
comando = "AT+CIPSTART=\"TCP\",\""; // TCP al host en el Puerto 80 //
comando += HOST;
comando += "\",80";
Serial1.println(comando); // envía el comando //
delay(200);
  // Envía los datos
comando = "GET /arduino/estanques/simple/cargar_nivel.php";
comando += "?sensor=";
comando += nivel_porcentaje;
comando += " HTTP/1.1\r\n";
comando += "Host: ";
comando += HOST;
comando += "\r\nConnection: close\r\n\r\n";
Serial1.print("AT+CIPSEND="); // envía el comando AT al módulo WIFI //
Serial1.println(comando.length()); //Envía el largo del comando//
if (Serial1.find(">")) { // signo enviado por el módulo wifi para avisar que está recibiendo //
Serial1.print(comando); // ahora que el módulo wifi recibe, se envía el comando complete //
  }
while (Serial1.available()) {
if (Serial1.find("sp=")) {
sp_string = Serial1.readStringUntil(".");
estado_string = Serial1.readStringUntil(".");
  Setpoint = sp_string.toInt();
estado = estado_string.toInt();
sp_string = "";
estado_string = "";
}
}
}

```

```

if (estado == 1){
digitalWrite(PINRELE, HIGH);
}
else digitalWrite(PINRELE, LOW);
}
// if (Serial1.find("estado=")){
//   sp_string = Serial1.readStringUntil(".");
//   estado = sp_string.toInt();
//   sp_string = "";
//   if (estado == 1){
//     digitalWrite(PINRELE, HIGH);
//   }
//   else digitalWrite(PINRELE, LOW);
// }
}
}

booleanconectar_wifi() {
Serial.println("Conectando WiFi."); // escribe conectando wifi por el Puerto serie de pc//
Serial1.println("AT+CWMODE=1"); //envía comando para configurar el módulo wifi como
estación //
delay(4000);
comando = "AT+CWJAP=\""; // comienza a escribir el comando para conectarse a la red SSID
con PASS //
comando += SSID;
comando += "\",\"";
comando += PASS;
comando += "\"";
Serial1.println(comando); // envía el comando al módulo WIFI //
for(int i = 0; i < 10; i++){ // Hace que el led parpadee 150mseg encendido y 150mseg apagado //
digitalWrite(ledWifi, HIGH);
delay(150);
digitalWrite(ledWifi, LOW);
delay(150);
}
if (Serial1.find("OK")) { //si el módulo está conectado, se deja encendido el led WIFI//
Serial.println("Conectado.");
digitalWrite(RUN, HIGH);
digitalWrite(ledWifi, HIGH);
return true;
}
else {
Serial.println("No Conectado."); // si el modulo no se conecta, el led se apaga//
digitalWrite(ledWifi, LOW);
return false;
}
}
}

```

```
voidrevisarNivel(){
if (Input > 10 && Input < 90){ // si input esta entre 10 y 90 se considera nivel normal //
nivelNormal();
}
else {
if (Input <= 10){
nivelBajo();
}
if (Input >= 90){
nivelAlto();
}
}
}

voidnivelAlto(){ // Enciende led si el nivel es alto y apaga las demás //
digitalWrite(ledNivelAlto, HIGH);
digitalWrite(ledNivelNormal, LOW);
digitalWrite(ledNivelBajo, LOW);
}

voidnivelNormal(){ // Enciende led si el nivel es normal y apaga las demás //
digitalWrite(ledNivelAlto, LOW);
digitalWrite(ledNivelNormal, HIGH);
digitalWrite(ledNivelBajo, LOW);
}

voidnivelBajo(){// Enciende led si el nivel es bajo y apaga las demás //
digitalWrite(ledNivelAlto, LOW);
digitalWrite(ledNivelNormal, LOW);
digitalWrite(ledNivelBajo, HIGH);
}
}
```