

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Diseño de un sistema de depuración de moluscos bivalvos con recirculación, a nivel de laboratorio.

PROYECTO INTEGRADOR

Previo la obtención del Título de:

**Ingeniero en Electricidad especialización Electrónica y
Automatización Industrial**

Presentado por:

Carlos Harley Cardoso Chamaidán

GUAYAQUIL - ECUADOR

Año: 2020

DEDICATORIA

El presente proyecto se lo dedico a mis compañeros de semestres inferiores, para que puedan servirse de este y más trabajos que se desarrollarán en beneficio de la comunidad.

AGRADECIMIENTOS

Agradezco a Dios, a mi familia y a las amistades que con su apoyo y colaboración en cada semestre me permitieron llegar hasta el final de la carrera.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponden conforme al reglamento de propiedad intelectual de la institución; Por lo cual Yo *Carlos Harley Cardoso Chamaidán* doy mi consentimiento para que la ESPOI realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Carlos Harley
Cardoso Chamaidán

EVALUADORES

~~MSc. Dennys Dick Cortez Álvarez~~

PROFESOR DE LA MATERIA

~~MSc. Efrén Herrera Muentes~~

PROFESOR TUTOR

RESUMEN

Durante este proyecto se van a enunciar los principales parámetros que permiten a los moluscos bivalvos un filtrado activo al momento de automatizar el proceso de depuración mediante un sistema de recirculación de agua referenciado en las características determinadas por la FAO en Depuración de bivalvos: aspectos fundamentales y prácticos, en el cual se establece límite por tipo de bivalvo tanto para la temperatura, salinidad, pH y el Oxígeno disuelto en el agua con los cuales permitirá elaborar en futuro un prototipo para ventas masivas a bajo costo con los cuales ya no se deberá verificar estos parámetros puesto que el prototipo estableció un régimen de funcionamiento para que esas variables se mantengan en óptimas condiciones y de esta manera generar un producto fiable y económico. Por otro lado, se enuncia el control a realizar mediante los microcontroladores con sus distintos drivers, características de funcionamiento lógico y la interfaz gráfica del sistema; esta última se implementa con la finalidad de elaborar un sistema amigable y empático con los operadores finales.

Palabras Clave: Bivalvos, Microcontroladores, Recirculación, Depuración

ABSTRACT

During this project, the main parameters that allow bivalve molluscs an active filtering at the time of automating the purification process by means of a water recirculation system referenced in the characteristics determined by the FAO in Bivalve purification: fundamental aspects will be stated. and practical, in which a limit is established by type of bivalve both for temperature, salinity, pH and dissolved oxygen in the water with which it will allow to elaborate in the future a prototype for mass sale at low cost in which it will no longer be necessary Verify these parameters since the prototype established an operating regime so that these variables are kept in optimal conditions and thus generate a reliable and economical product. On the other hand, the control to be carried out by means of the microcontrollers with their different drivers, characteristics of logical operation and the graphical interface of the system is stated; the latter is implemented in order to develop a friendly and empathetic system with the final operators.

Keywords: Bivalves, Microcontrollers, Recirculation, Purification.

ÍNDICE GENERAL

Contenido

EVALUADORES	I
RESUMEN	II
ABSTRACT	III
ÍNDICE GENERAL	IV
ABREVIATURAS	VII
SIMBOLOGÍA.....	VIII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	XI
CAPITULO 1.....	1
1. INTRODUCCION.....	1
2.1. Descripción del Problema	1
2.2. Justificación del Problema	2
2.3. Objetivos.....	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Específico.....	3
2.4. Marco Teórico.....	4
1.4.1. Máquina de Inducción	4
1.4.2. Inverso para Motor de CA	5
1.4.3. Arduino.....	7
1.4.4. Sensores	8
1.4.5. Pantallas Nextion	13
1.4.6. Válvulas solenoides.....	14
1.4.7. Barniz Tropicalizador	14
1.4.8. Comunicación I2C.....	14
Capítulo 2.....	15

2.	Metodología	15
2.1.	Diagrama de bloques	15
2.2.	Diagrama de comunicación	16
2.3.	Diagrama PI&D del sistema de depuración	17
2.4.	Diagrama de Proceso del sistema de depuración	17
2.5.	Metodología de trabajo	18
2.5.1.	Construcción de la interfaz HMI.....	18
2.5.2.	Construcción de la programación del controlador	20
2.5.3.	Diseño del Rectificador Monofásico	21
2.5.4.	Control del módulo de conmutación	22
2.5.5.	Construcción de simulación del sistema	23
2.5.6.	Diseño de Construcción	27
2.5.6..1.	Esquema eléctrico	27
2.5.6..2.	Circuitos Impresos	28
2.5.7.	Selección del sistema UV	32
2.5.8.	Dimensionamiento de bomba para recirculación	32
2.5.9.	Dimensionamiento de bomba para tanque de alimentación	33
2.5.10.	Diseño físico del tablero eléctrico (Control y Potencia)	35
CAPÍTULO 3.....		36
3.	RESULTADOS Y ANÁLISIS.....	36
3.1.	Resultados de la simulación	36
3.2.	Costos de Ingeniería	38
3.3.	Catálogo de conceptos	38
CAPÍTULO 4.....		40
4.	Conclusiones Y Recomendaciones	40
4.1.	Conclusiones.....	40
4.2.	Recomendaciones.....	40

Bibliografía	40
APÉNDICES	42
Apéndice A.....	42
Fuentes de alimentación 5VDC, 15VDC.	42
Apéndice B.....	44
TIMER/CONTADOR de 8bits con PWM.....	44
ANEXOS	45
Código de programa Arduino Maestro	45
Código de programación Arduino Esclavo SPWM (NANO)	51
Código de programación Matlab para la generación de los vectores SPWM a diferentes frecuencias	53
Código de programación Arduino Esclavo Flujo (Micro)	54
Programación en Simulink Matlab para un control continuó en el variador de frecuencia de la bomba de recirculación	55

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
V	Voltaje
A	Amperaje
DC	Direct Current (Corriente Directa)
AC	Alternating current (Corriente Alterna)

SIMBOLOGÍA

mil	Milésima de pulgada
m	Metro
cm	Centímetro
mm	Milímetro
mV	Milivoltio

ÍNDICE DE FIGURAS

Ilustración 1. Rectificador de Puente de Diodos	5
Ilustración 2. Rectificador de Puente completo con filtro capacitivos	6
Ilustración 3. Pulsador de Clase E.....	6
Ilustración 4. Conmutación mediante SPWM	7
Ilustración 5. Especificaciones técnicas del Arduino Mega	7
Ilustración 6. Especificaciones técnicas del Arduino Nano	8
Ilustración 7. Sensor de nivel de líquido mediante boya	9
Ilustración 8. Diferencias y similitudes entre a) MOSFET y b) ISFET	12
Ilustración 9. Diagrama de Bloques.....	15
Ilustración 10. Diagrama de comunicación del sistema	16
Ilustración 11. Diagrama PI&D	17
Ilustración 12. Diagrama de Proceso.....	17
Ilustración 13. Pantalla de arranque del sistema	18
Ilustración 14. Pantalla de monitoreo de accionamientos	19
Ilustración 15. Pantalla de monitoreo de las variables de control del sistema.....	19
Ilustración 16. Pagina Nextion para observar el caudal y potencia del motor	20
Ilustración 17. Sistema de control simulado mediante una comunicación virtual.	24
Ilustración 18. Simulación de los Microcontroladores esclavos.	25
Ilustración 19. Conexión de optoacoplador 6N137	26
Ilustración 20. Conexión de Driver de disparo IR2110 para IGBTs.....	26
Ilustración 21. Diagrama esquemático de los Arduino Esclavo, y el sistema rectificador de potencia	27
Ilustración 22. Esquemático del Arduino Maestro para realizar las conexiones pertinentes	28
Ilustración 23. Dimensiones de la placa PCB para el Arduino Maestro	29
Ilustración 24. Vista Posterior Arduino Maestro	29
Ilustración 25. Vista Superior Arduino Maestro.....	30
Ilustración 26. Dimensión de la placa PBC para los Arduino Esclavo y sistema rectificador de potencia	30
Ilustración 27. Vista Posterior Arduino Esclavos.....	31
Ilustración 28. Vista Superior Arduino Esclavos	31

Ilustración 29. Sistema Filtro UV	32
Ilustración 30. Curvas de operación Bomba Pentair PM21	33
Ilustración 31. Bomba Pentair, modelo PM21	33
Ilustración 32. Curva de operación de las bombas Pentair modelo PM42	34
Ilustración 33. Modelo seleccionado Acuamain Pentair PM42.....	34
Ilustración 34. Diseño del tablero físico de Control y Potencia	35
Ilustración 35. Diagrama unifilar de Potencia y Control	35
Ilustración 36. Señal SPWM generado por Arduino en la plataforma Proteus	36
Ilustración 37. Simulador de la Pantalla Nextion comunicada con la simulación de Arduino	37
Ilustración 38. Programación de Arduino mediante Simulink-Matlab	38
Ilustración 39. Señales de Control SPWM obtenidas por medio de la plataforma de Simulink	38
Ilustración 40. Fuentes de alimentación DC	42
Ilustración 41. Diagrama en Bloques de Timer/Counter 0	44
Ilustración 42. Control de Arduino mediante la plataforma Simulink	55

ÍNDICE DE TABLAS

Tabla 1. Configuración de válvulas del sistema	21
Tabla 2. Costos del Personal que Intervino en el proyecto.....	38
Tabla 3. Catálogo de conceptos.....	39

CAPITULO 1

1. INTRODUCCION

El mercado ecuatoriano se ha visto altamente impulsado por productos marinos entre los que resaltan los mariscos, los cuales pasan por diferentes etapas y cuidados, antes de ser empacado para su posterior venta. En muchos lugares del mundo se emplea la depuración con moluscos que tengan un porcentaje de contaminación microbiana baja para que no sea contaminado el consumidor. Durante el tiempo de depuración los moluscos son colocados en agua de mar libre de agentes fecales.

2.1. Descripción del Problema

La comercialización de los mariscos se lleva a cabo de manera internacional a diferentes países del mundo, el presente proyecto estudia a los moluscos bivalvos los cuales se comercializan y se consumen en su gran mayoría de manera cruda o semi-cruda lo que genera un alto riesgo de contagio de enfermedades en sus consumidores, enfermedades tales como salmonela, gastroenteritis y fiebre tifoidea.

Para poder evitar que estos moluscos sean un agente transmisor de enfermedades se necesita depurar constantemente el medio en el que están desarrollándose que en este caso es el agua de mar, como estamos en un nivel industrial es indispensable automatizar estos procesos de depuración y realizarlos de manera óptima.

Durante el proceso de depuración es de vital importancia aplicar 2 filtros uno de sólidos y otro UV, el filtro de sólido ayudará a limpiar las heces fecales y cualquier otro objeto ajeno a los moluscos, mientras que el filtro UV eliminará cualquier microorganismo presente en el agua de mar de los moluscos.

2.2. Justificación del Problema

Los mariscos son una fuente de alimento popular y nutritiva en todo el mundo y su consumo ha aumentado en los últimos años significativamente. Los bivalvos se alimentan por filtración, por lo tanto, son capaces de concentrar los contaminantes a un nivel mucho más alto que los del agua circundante (Cosson, 2000; Fang y col., 2003). Estos contaminantes pueden causar enfermedades en los seres humanos, especialmente contaminantes microbianos, los cuales son responsables de enfermedades entéricas, porque los mariscos suelen comerse crudos o ligeramente cocidos (Rippey, 1994; Croci y col., 2002; Formiga-Cruz y col., 2003, FAO 2010). Para reducir este riesgo, se recomienda un tratamiento adecuado después de la cosecha. La mejor, más fácil y económica estrategia que se ha desarrollado para la gestión de riesgos de los bivalvos es utilizar su capacidad de eliminar los microorganismos patógenos y las sustancias tóxicas cuando los bivalvos se mantienen en agua de mar limpia y desinfectada (Wong y col., 1997; Sobsey y Jaykus, 1999; El-Shenawy, 2004; Montiel y col., 2009; FAO, 2010; Martínez-Albores y col., 2020). El método anterior (depuración) ayuda a los bivalvos a expulsar y separar los contaminantes de sus branquias y su tracto intestinal durante un período de tiempo y evita su recontaminación. Algunos autores, como Arnold (1991) restringieron la función de este tipo de depuración para eliminar la contaminación bacteriana, pero otros lo destacaron y alentaron para reducir la toxicidad de los metales pesados (El-Shenawy, 2004; Katayon y col., 2004; Martínez-Albores y col. 2020) y de los hidrocarburos del petróleo (Clement y otros, 1980; Rainio y col., 1986; Rantamaki, 1997). Son muchos los factores que influyen en el grado de depuración: el diseño del sistema, la calidad inicial del agua, la oxigenación y los caudales, la salinidad, la temperatura, la proporción entre mariscos y agua, la remoción y el asentamiento de los sedimentos, entre otras (Martínez-Albores y col. 2020). En investigaciones previas, investigadores del equipo de trabajo del presente proyecto han logrado estandarizar el proceso de depuración para las especies *Polymesoda sólida* y *Rangia cuneata*. Existen diferentes formas y sistemas de depuración en el mundo, desde los muy avanzados tecnológicamente, que depuran cientos de kilogramos de molusco por día, hasta los artesanales. La FAO en 2010, publicó el Documento Técnico de Pesca 511, donde señala los aspectos fundamentales y prácticos de la depuración de moluscos bivalvos. La técnica requiere el paso de agua

limpia por el tracto digestivo de los moluscos, para que estos puedan eliminar los contaminantes, los cuales van desde microorganismos como bacterias y virus, arena e incluso metales. En Ecuador la práctica de la depuración no es exigida para la comercialización y consumo de moluscos bivalvos, se visualiza la necesidad de establecer sistemas de depuración que permitan garantizar la inocuidad de estos alimentos. Este proyecto permitirá el diseño a escala para ser probado en el laboratorio.

2.3. Objetivos

1.3.1. Objetivo General

- Diseñar un modelo de depuración con recirculación de agua, a nivel de laboratorio.

1.3.2. Objetivos Específico

- Diseñar el proceso y ensamblaje de elementos de un sistema de depuración para moluscos bivalvos. (Biología y Mecánica)
- Calcular las capacidades de bombas, filtros, tuberías, etc. Para la movilización de las aguas en el sistema. (Mecánica)
- Diseñar la automatización del sistema de depuración a nivel de laboratorio. (Automatización y Mecánica)
- Diseñar un sistema de alarma con indicadores visuales para los parámetros preponderantes del sistema. (Automatización)
- Simular el sistema de depuración automatizado. (Automatización)
- Calcular los costos de producción de un sistema de depuración a nivel de laboratorio y extrapolación a escala comercial. (Biología y Economía)

2.4. Marco Teórico

1.4.1. Máquina de Inducción

Para el desarrollo de este proyecto, se escogió la técnica de variación de velocidad del motor de inducción a través de la variación de la frecuencia de entrada debido a que el motor seleccionado es una maquina asíncrona.

Hablando de la máquina de inducción asíncrona, se debe saber que su funcionamiento se basa en el principio de interacción electromagnética entre el campo magnético giratorio el cual es generado por las corrientes suministradas desde la red al devanado del estator y las corrientes que se inducen en el devanado del rotor cuando el campo giratorio cruza sus hilos conductores (Hall, 2007).

Cuando el campo giratorio del motor dispone de un carácter sinusoidal su velocidad de rotación se expresa mediante la siguiente ecuación 1:

$$n_s = \frac{120f}{P} \quad (1.1)$$

Ecuación 1

Donde:

n_s = velocidad del campo magnético r/min

f = frecuencia del suministro del voltaje aplicado en Hz

P = número de polos del estator

Mientras que la velocidad del rotor se expresa mediante la ecuacion2:

$$n_r = (1 - s)n_s \quad (1.2)$$

Ecuación 2

Donde:

n_r = velocidad del rotor r/min

n_s = velocidad r/min

s = deslizamiento o resbalamiento de la maquina asíncrona

Como se puede observar en los manifiestos anteriores que la forma más sencilla de poder realizar el control sobre una máquina de inducción asíncrona es controlando la frecuencia de la tensión suministrada al campo giratorio.

1.4.2. Inverso para Motor de CA

Para la realización de un variador de velocidad se puede separar en diferentes sub procesos que interfieren para la elaboración de las ondas de alimentación del motor.

La rectificación de la señal proporcionada por la red de la empresa de suministro eléctrico es la etapa principal de este proceso, esta misma rectificación nos permite obtener una fuente de alimentación continua (DC), después de esto se ingresa a la etapa de conmutación en la cual modificamos la frecuencia de la señal generada dependiendo del control implementado, con la finalidad de poder variar la velocidad de funcionamiento del motor.

1.4.2.1. Rectificador de onda completa

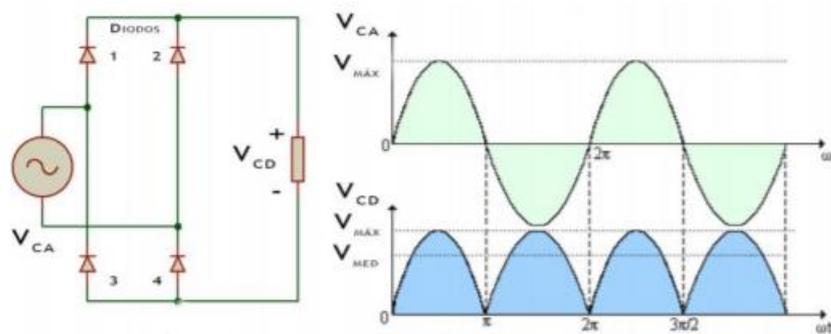


Ilustración 1. Rectificador de Puente de Diodos

Uno de los rectificadores más implementados es la configuración de puente de diodos en el cual solo conducen dos diodos por semiciclo debido que es el más adecuado para aplicaciones de alta tensión puesto que generar una menor tensión de pico en los terminales de los diodos, se suele hacer uso de un filtro capacitivo después de la salida del rectificador con la finalidad de reducir el rizado de la señal

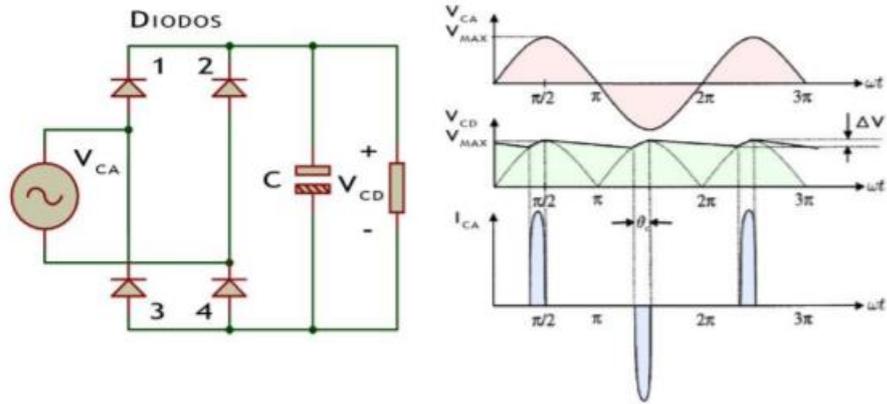


Ilustración 2. Rectificador de Puente completo con filtro capacitivos

Fuente:

1.4.2.2. Método de control

Para la elaboración de este proyecto el tipo de configuración del pulsador es Clase E, con lo cual se asegura que la corriente de la carga puede ser positiva o negativa; este tipo de configuración suele ser denominada Tipo puente completo.

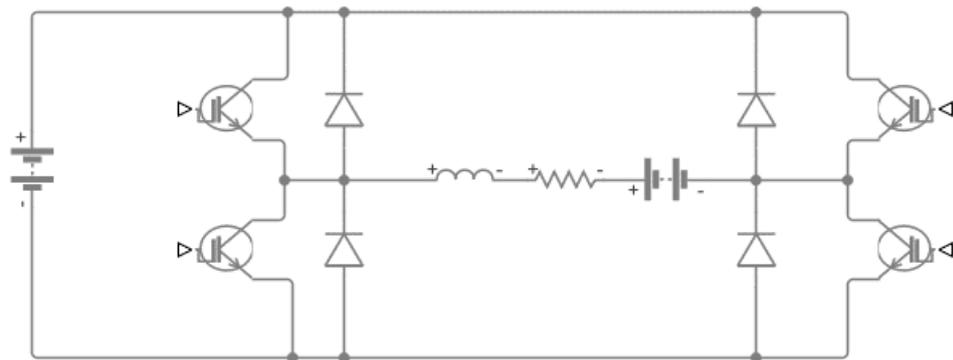


Ilustración 3. Pulsador de Clase E

Haciendo uso del tipo de configuración de puente completo (Pulsador de clase E), se ha decidido implementar un control de conmutación mediante modulación SPWM.

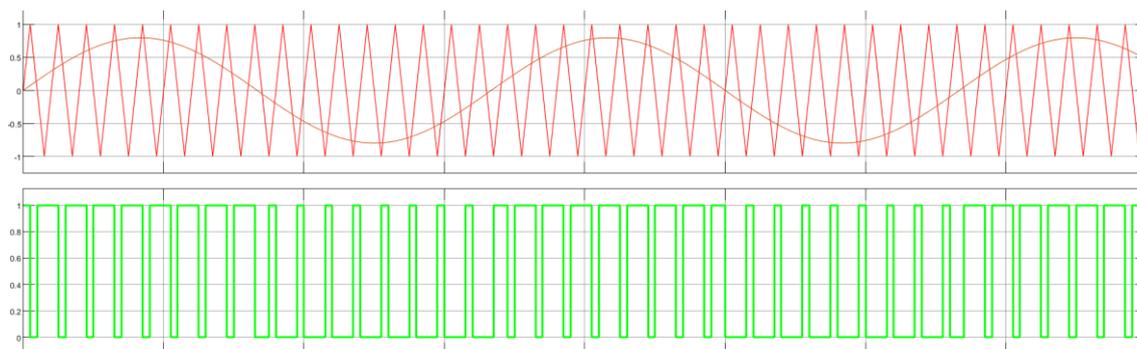


Ilustración 4. Conmutación mediante SPWM

1.4.3. Arduino

Arduino es una plataforma electrónica diseñada para el aprendizaje de software y hardware con diferentes niveles de complejidad. Este microcontrolador dispone de una ventaja al momento de desarrollo de proyectos, pues dispone de una gran variedad de entradas y salidas ya sean estas analógicas o digitales.

Para este proyecto se implementan 2 versiones de la plataforma Arduino enunciadas a continuación:

- Arduino Mega 2560

Microcontrolador	ATmega2560
Tensión de trabajo	5V
Tensión de entrada (recomendada)	7-12V
Tensión de entrada (límite)	6-20V
Pines Digitales I/O	54 (de los cuales 15 proporcionan salida PWM)
Pines de entradas Analógicas	16
DC Corriente por Pin I/O	20 mA
DC Corriente por Pin 3.3V	50 mA
Memoria Flash	256 KB de los cuales 8 KB se usan por el bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16 MHz
Largo	101.52 mm
Anchu	53.3 mm
Peso	37 g

Ilustración 5. Especificaciones técnicas del Arduino Mega

- Arduino Nano

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

Ilustración 6. Especificaciones técnicas del Arduino Nano

1.4.4. Sensores

1.4.4.1. Sensor de Nivel

Los sensores de nivel son dispositivos electrónicos que permiten medir la altura del material en cuestión, generalmente se implementan sobre líquidos dentro de un tanque u otros recipientes. Para la selección de un sensor de nivel se debe considerar el medio en el cual va a trabajar y la aplicación para la cual será implementada, debido a la gran variedad de modelos y diversidad de precios disponibles. Existen los sensores de nivel ultrasónicos sin contacto, sensores de nivel ultrasónicos con contacto, sensores de nivel capacitivo y sensores de nivel de flotador; estos sensores se dividen en dos principales tipos: Los sensores de nivel de punto que se utilizan para marcar una altura pre establecida y los sensores de nivel continuo que permiten realizar el seguimiento del nivel de todo un sistema.



Ilustración 7. Sensor de nivel de líquido mediante boya

Entre los principales se encuentran los Interruptores de flotador el cual será utilizado durante la elaboración de este proyecto, pues es similar a un interruptor por lo cual es denominado sensor de nivel de punto, además pueden ser de un solo polo o de doble tiro.

1.4.4.2. Sensor de Temperatura

Dentro de la medición de la temperatura existen diferentes tipos de sensores, mas todos ellos infieren dicho valor al detectar el cambio de una característica física. Existen seis tipos de sensores: los termopares, dispositivos de temperatura resistivos (RTD y termistores), radiadores infrarrojos, dispositivos bimetálicos, dispositivos de dilatación de líquido y dispositivos de cambio de estado.

- Los sensores Termopar son esencialmente dos alambres de diferentes metales que son unidos en un extremo. En los cuales los cambios de temperatura generan una FEM entre los extremos, dichas variables son proporcionalmente directas, pero no necesariamente en forma lineal.
- Los sensores por resistencia hacen uso de las propiedades resistivas de un material (por los generales metálico) debido que esta característica varía dependiendo de los cambios de temperatura, estos cambios entre dichas variables suelen ser más o menos lineal; mientras que los termistores se basan en el cambio de resistencia de un material semiconductor cerámico, se recalca que para los termistores la resistencia cae en forma no lineal con el aumento de la temperatura.
- Los sensores Bimetálicos no necesitan de una alimentación pues su funcionamiento viene dado por la dilatación térmica entre diferentes metales provocando una curvatura que se traduce como temperatura

mediante una articulación mecánica en el indicador, este tipo de sensores no suelen ser muy sensibles.

- Los sensores por dilatación de fluido son los que generalmente usamos como termómetros de mercurio, dichos sensores no requieren de alimentación eléctrica, nos permiten utilizarlos cíclicamente manteniendo su estabilidad y en algunas ocasiones en vez de líquidos se usa gas. Pero los datos que proporcionan no son transmisibles con facilidad, además de no poder realizar mediciones puntuales.
- Los sensores por cambio de estado por lo general no poseen la capacidad de regresar a sus estados anteriores y se los implementan para determinar posibles alarmas, a diferencia de los enunciados anteriormente no son muy sensibles a los cambios de temperatura relativamente cercanos es por ello que su función es determinar temperaturas excesivas para una máquina.

1.4.4.3. Sensores de Conductividad

Dicho sensor se implementa con la finalidad de poder identificar la salinidad del medio, debido que la salinidad es una medida de la concentración total de iones en el agua los cuales permiten una conducción. Se recalca que la salinidad se puede medir con bastante precisión mediante medios tradicionales, pero al ser medidos mediante un factor de correlación entre salinidad y conductividad se pierde dicha precisión. Además, “El factor que relaciona la conductividad y la salinidad es aproximadamente de 0.7 en aguas oceánicas en todo el mundo” (Claude E. Boyd, 2017)

La unidad de conductividad es “Siemens por metro” y los sensores que disponen de la capacidad de medir esta variable son utilizados en numerosas industrias, la selección del tipo de sensor depende de la aplicación y el rango de conductividad. Para medir la conductividad en agua pura y ultra pura se hace uso de sensores conductivos, y en productos con una alta conductividad se implementan los sensores toroidales o bien los sensores de 4 electrodos.

Los sensores conductivos hacen uso de dos electrodos situados uno frente a otro, su funcionamiento es a través de someter a los electrodos a una diferencia de tensión alta con la finalidad de generar una corriente en el producto.

Los sensores de 4 electrodos como se enuncio anteriormente son implementados en medio altamente conductores por lo cual 2 de sus electrodos no disponen de corriente

y gracias a esto también se evita el efecto de polarización el cual podría terminar influyendo en la medición correcta de la conductividad.

Las sondas toroidales o también denominadas inductivas están compuestas por 2 bobinas una para la transmisión y la otra de recepción, permitiéndoles obtener el valor de la conductividad a través de diversos pasos:

Un oscilador genera un campo magnético en la bobina de transmisión, que induce una tensión en el producto. Los cationes y aniones del producto comienzan a moverse generando una corriente alterna. De esta forma se induce un campo magnético alterno y, por consiguiente, una corriente que circula en la bobina de recepción. (Endress+Hauser, 2019)

1.4.4.4. Sensor de pH

Los medidores de pH son instrumentos utilizados con la finalidad de medir la acidez o la alcalinidad de una solución a través de la actividad de los iones de hidrogeno [H+] e hidroxilo [OH-], la unidad de medida es el pH y toma valores dentro de un rango de 0 a 14. Se recalca que los electrodos de pH y las mediciones son sensibles a la temperatura, por lo que se realizan compensaciones de temperatura de dos modos (Manual y Automático).

a) Sensores electroquímicos:

Los ISFET requieren calibración y limpieza periódica con la finalidad de mantener una precisión de 0.1 unidades de pH, se observa que su principio de funcionamiento es similar al del transistor MOS.

A diferencia de éste, el ISFET no posee una compuerta metálica y el dieléctrico de esta misma región se expone directamente al analito. Es decir, utilizan como principio de medida la modulación del canal del transistor mediante la diferencia de tensión que se establece entre el analito y la puerta del dispositivo. La tensión de salida del transistor es función del pH de la disolución. (Métodos de medida del pH para estudios invasivos prolongados: ISFET y fibra óptica, 2006, p.3)

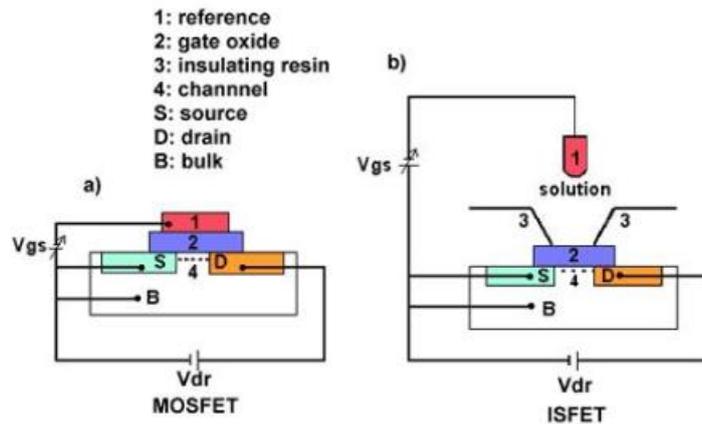


Ilustración 8. Diferencias y similitudes entre a) MOSFET y b) ISFET

Los sensores de pH de vidrio son un estándar a nivel industrial y de laboratorio. Dichos modelos se basan en la implementación de un electrodo de vidrio con una resistencia de salida en el orden de $M\Omega$ la cual entra en contacto con el analito a través de la membrana sensible al ion de hidrogeno por lo cual es necesario mantener dicha membrana húmeda. “Este tipo de modelo permite obtener una precisión de 0.01 unidades de pH, además de poseer un tiempo de vida relativamente cortos (1 año) con costos que varían entre 30 y 300 dólares” (Métodos de medida del pH para estudios invasivos prolongados: ISFET y fibra óptica, 2006)

Los electrodos de membrana líquida tienen un funcionamiento similar al electrodo de vidrio, la membrana implementada para estos electrodos son un polímero orgánico con intercambiador. “La membrana es un polímero neutro o saturado con un intercambiador iónico líquido que es capaz de formar iones complejos de forma reversible y transferirlos a través de una membrana orgánica mediante la translocación del portador” (Wojciech Wroblewski ,2006)

Los electrodos de estado sólido consisten en metales cuya superficie están recubiertas parcialmente por una capa de oxido del mismo metal, se hace uso de iones aliovalentes en la red con la finalidad de aumentar el tiempo de respuesta de las membranas.

b) Sensores ópticos

Este grupo de sensores cambian sus propiedades ópticas en función del analito. Dicho grupo se puede clasificar por absorbancia o luminiscencia. La relación de la intensidad de la luz incidente y la luz reflejada es directa con el valor de pH para los

sensores por absorbancia, mientras que los sensores de luminiscencia determinan los valores de pH mediante el cambio de onda entre la luz incidente y la reflejada.

1.4.4.5. Sensor de Oxígeno Disuelto

Este tipo de sondas son capaces de determinar el porcentaje de oxígeno disuelto en el medio analizado. Las sondas contienen sensores de fluorescencia óptica, sensores galvánicos o sensores polarográficos.

Los sensores de fluorescencia óptica cuentan con una recubierta fluorescente, y por medio de una luz expuesta al recubrimiento se genera una luz y un resplandor de fluorescencia, este último es el cual nos permite obtener el valor del nivel de oxígeno basándonos en la duración de dicho resplandor.

Los sensores galvánicos no necesitan de una alimentación de voltaje externa, dichos sensores contienen un ánodo y cátodo dentro de un electrolito; en el cual el oxígeno ingresa por medio de una membrana al electrolito el cual genera una diferencia de potencia entre el ánodo y cátodo, y dicho diferencial se usa para medir la cantidad de oxígeno disuelto.

Los sensores polarográficos disponen de un funcionamiento similar a los sensores galvánicos, a diferencia que estos necesitan hacer uso de una fuente de alimentación externa.

1.4.5. Pantallas Nextion

Las pantallas Nextion son una alternativa económica a las pantallas HMI implementadas para los PLC, debido que Nextion es una solución de interfaz hombre-máquina (HMI) que combina un procesador integrado y una pantalla táctil de memoria con el software Nextion Editor para el desarrollo de proyectos HMI GUI.

El software Nextion Editor, puede desarrollar rápidamente la interfaz gráfica de usuario de HMI mediante componentes de arrastrar y soltar (gráficos, texto, botón, control deslizante, etc.) e instrucciones basadas en texto ASCII para codificar cómo interactúan los componentes en el lado de la pantalla.

La pantalla Nextion HMI se conecta a la MCU periférica a través de la serie TTL (5V, TX, RX, GND) para proporcionar notificaciones de eventos en las que la MCU periférica puede actuar, la MCU periférica puede actualizar fácilmente el progreso y el

estado a la pantalla Nextion utilizando instrucciones de texto ASCII simples. (Nextion,2019)

1.4.6. Válvulas solenoides

Las válvulas solenoides son de acción instantánea, también son denominadas ON/OFF. Su funcionamiento erradica en que cada válvula dispone de una bobina que al ser energizada genera un campo magnético que levanta un embolo en caso de ser normalmente cerrada, o en caso de ser normalmente cerrado se ejecuta la acción contraria. Para este proyecto se hará uso de válvulas solenoides de 1" de diámetro.

1.4.7. Barniz Tropicalizador

El barniz Tropicalizador es una base acrílica que se aplica sobre los circuitos y conjuntos electrónicos sometidos a entornos agresivos como es la humedad, el calor, la niebla y el aire salina. Este producto es recomendado para aplicaciones cerca de zonas costeras debido a su resistencia climática de hasta 500 horas al 95% de humedad en el aire a 55°C y protege a los componentes de ataques micro biológicos.

1.4.8. Comunicación I2C

Para la comunicación entre la interfaz HMI y el microcontrolador se implementó el protocolo de comunicación serial, por lo cual se definieron tramas de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales.

Capítulo 2

2. Metodología

Este capítulo va a explicar el proceso a realizar para la correcta implementación de la simulación del sistema depurador con recirculación. Para la simulación del sistema depurador se utilizaron los softwares: Proteus 8.9, aplicación Nextion, la IDE de Arduino y Configure Virtual Serial Port Driver.

2.1. Diagrama de bloques

En la ilustración 2 permite observar un diagrama de bloques del sistema, los cuales permiten percibir los pasos a seguir de manera general para llegar al objetivo planteado.

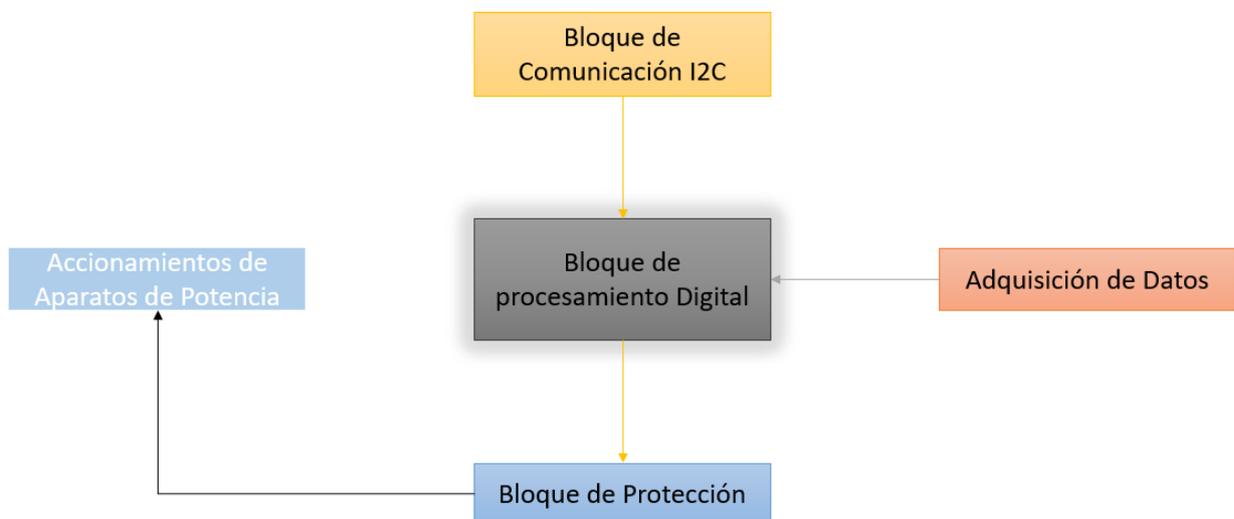


Ilustración 9. Diagrama de Bloques

Dentro del bloque de adquisición de datos se encuentran tanto las señales analógicas y digitales implementadas para obtener una respuesta de cómo se encuentra el sistema.

El bloque de protección se refiere a los optoacopladores que se requieran utilizar con la finalidad de poder mantener aislada la parte de control con la de potencia y evitar posibles averías que comprometan al sistema completo.

2.2. Diagrama de comunicación

El diseño de la comunicación está dado con la finalidad de cumplir con los requerimientos de funcionamiento.



Ilustración 10. Diagrama de comunicación del sistema

Dentro de la comunicación I2C implementada entre la pantalla y el controlador se hizo uso de tramas sencillas codificadas por el mismo diseñador, esto se debe que la velocidad de procesamiento sería un poco más extensa al transmitir las tramas específicas por ID de bloque puesto que estas llegan a ser mucho más complejas.

Como unidad de control Maestro se decidió implementar un Arduino Mega por la simple razón que permite readecuar el diseño en caso de requerir modificaciones al momento de ser escalado. Mientras que para las unidades de control de los esclavos se diseñó mediante la implementación de Arduino Mini.

2.3. Diagrama PI&D del sistema de depuración

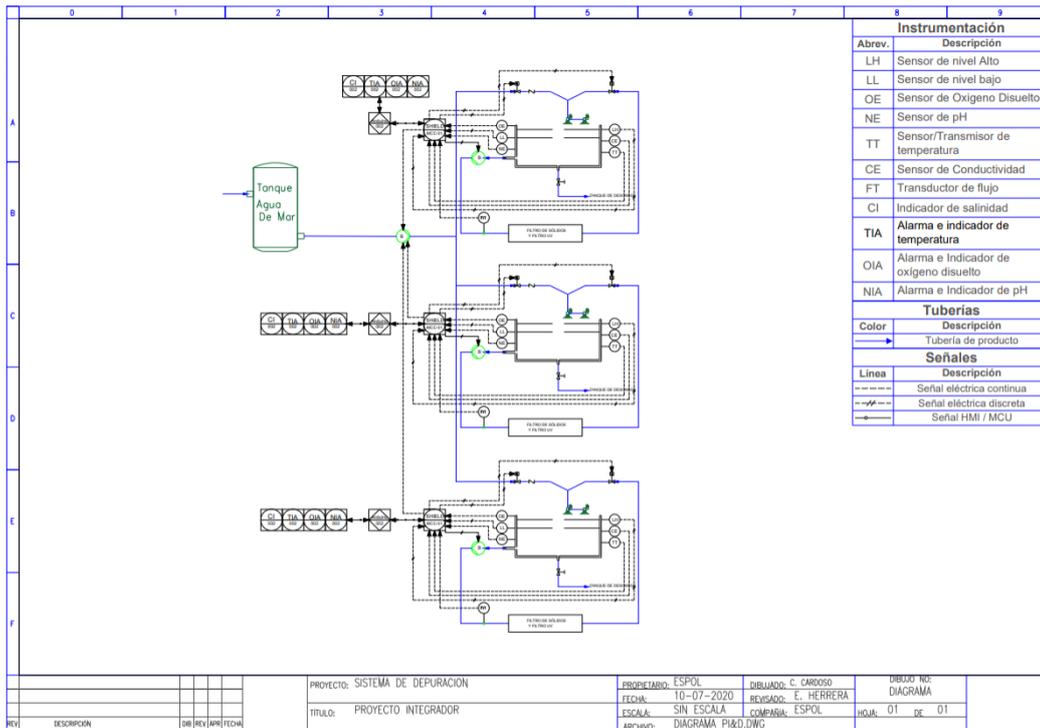


Ilustración 11. Diagrama PI&D

Para esto se establecen ciertas normas, ya sean el tipo de señal, las abreviaciones y el material que contiene la tubería.

2.4. Diagrama de Proceso del sistema de depuración

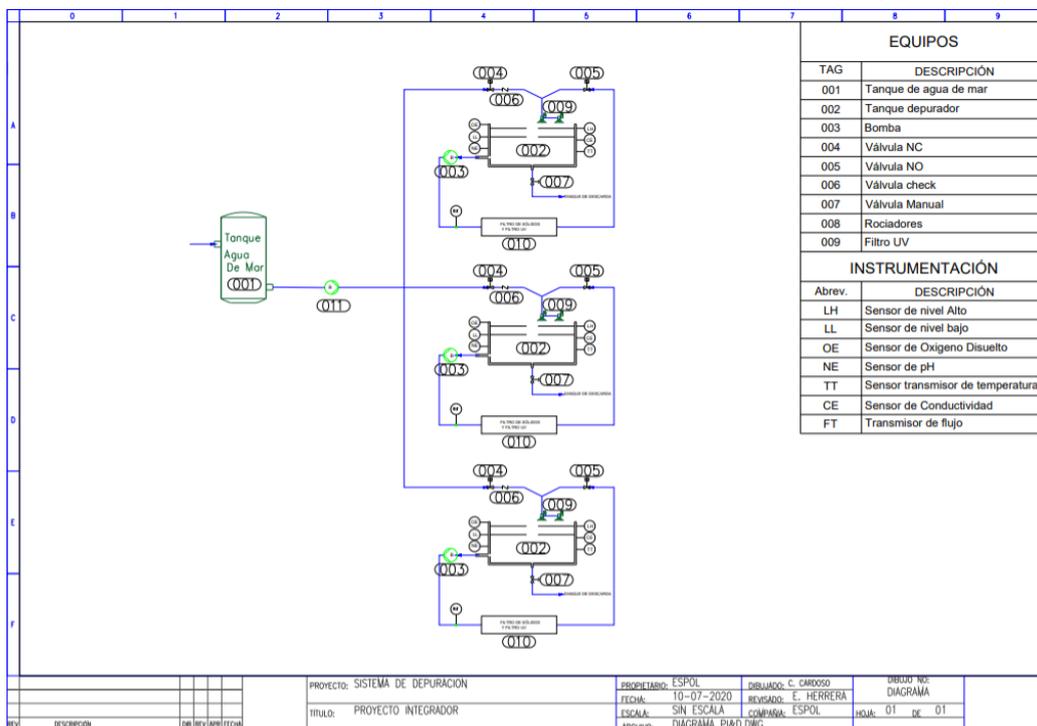


Ilustración 12. Diagrama de Proceso

2.5. Metodología de trabajo

2.5.1. Construcción de la interfaz HMI

Al ser un sistema relativamente económico el cual no será operado por gente experimentada en el ámbito de la automatización se ha decidido implementar una interfaz de lo más sencilla y empática con el usuario, debido a ello constara de 3 pantallas las cuales serán mostradas a continuación:



Ilustración 13. Pantalla de arranque del sistema

Esta pantalla se encuentra compuesta por 2 imágenes y un botón el cual envía al ser presionado una trama por medio de la comunión I2C al controlador (Arduino). Por otro lado, al ser presionado el botón de arranque el sistema completo se puede dar por iniciado, esto se puede observar por medio del tablero eléctrico implementado.

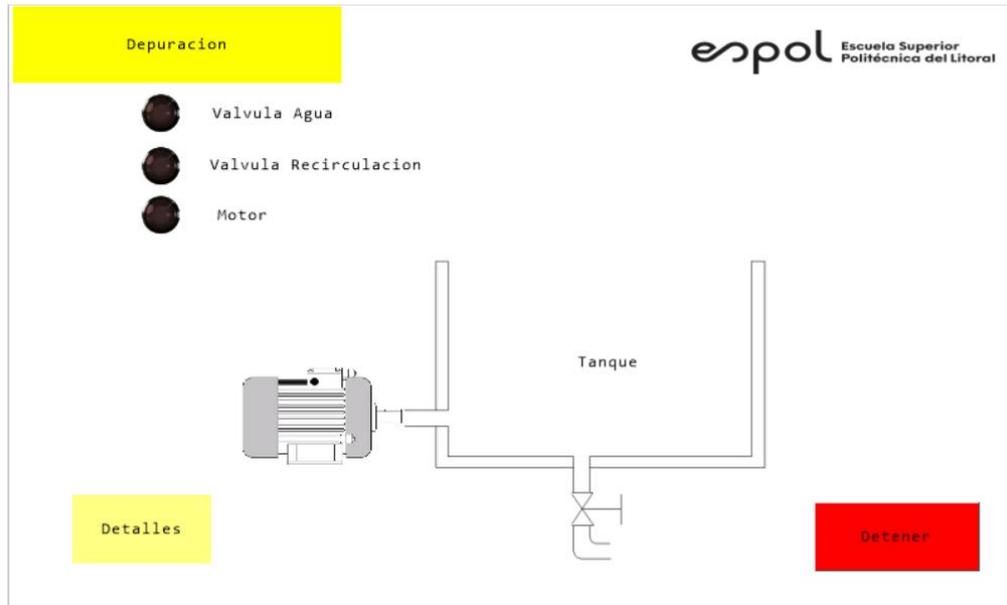


Ilustración 14. Pantalla de monitoreo de accionamientos

De igual manera dentro de esta pestaña se puede observar el estado lógico en el cual se encuentran los accionamientos, como los niveles establecidos dentro del tanque. Además, consta de 2 botones uno para acceder a la siguiente pantalla de la cual se hablará a continuación y el otro que paraliza el proceso en caso que el sistema requiera (Por problemas o simplemente porque el proceso de depuración ha culminado), en este caso ambos botones envían una trama sencilla al microcontrolador una diferente de la otra.

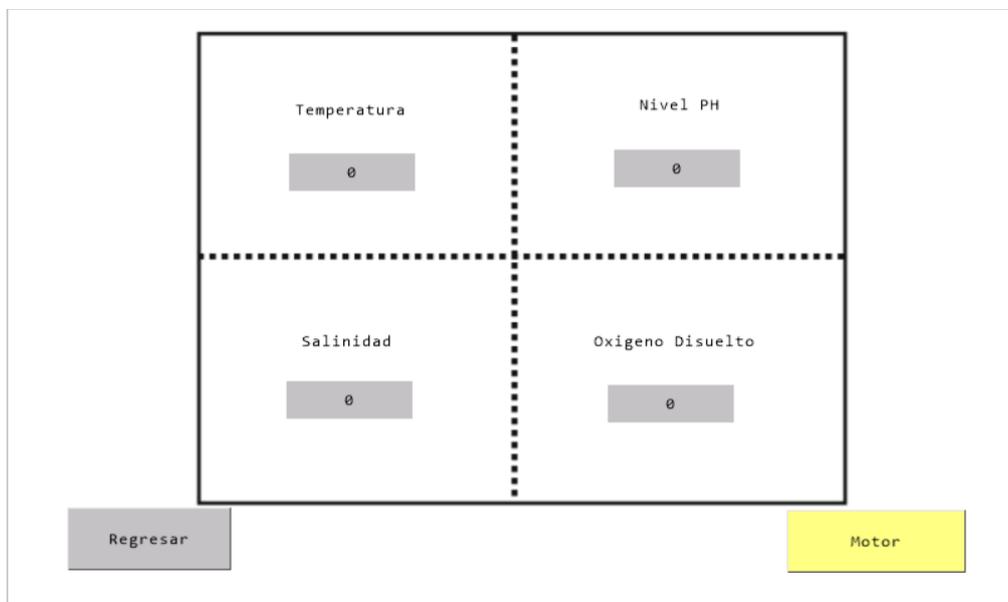


Ilustración 15. Pantalla de monitoreo de las variables de control del sistema

Dentro de esta pantalla de la Nextion, se transmiten las tramas de las señales analógicas por comunicación serial que han sido procesadas por el microcontrolador con sus respectivos índices de ganancia y un botón con una trama sencilla que permita regresar a la pantalla de los accionamientos digitales.

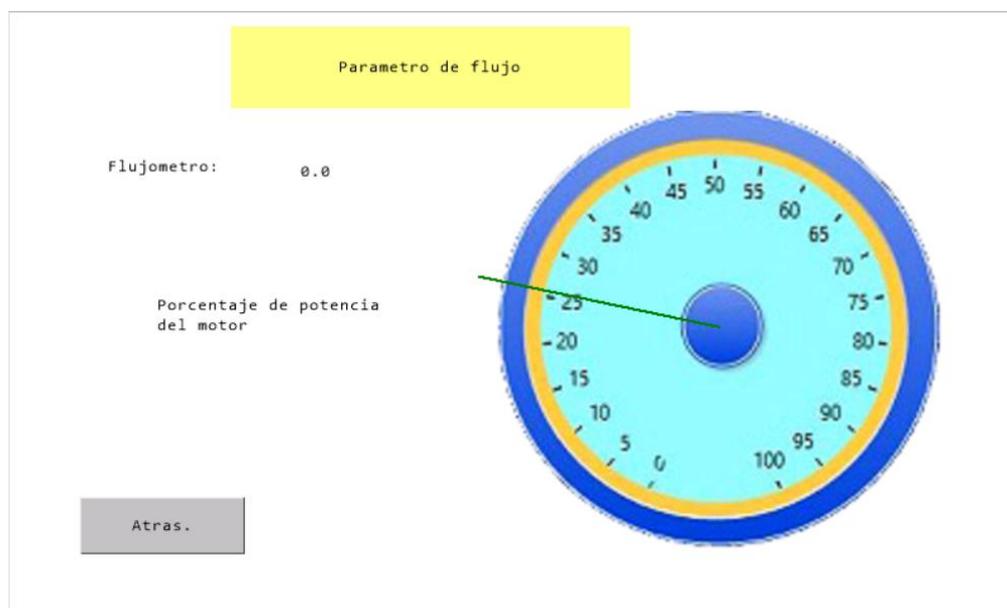


Ilustración 16. Pagina Nextion para observar el caudal y potencia del motor

Esta página dispone de su propio sistema de comunicación entre Arduino, esto se debe a que el Arduino Maestro solicita los valores censados por el Arduino Micro encargado de las interrupciones enviadas por el flujómetro.

2.5.2. Construcción de la programación del controlador

Existen 3 archivos de programación implementados para la ejecución de este proyecto, los cuales son para el controlador principal (Master) y los archivos restantes corresponde los controladores esclavos.

Para el microcontrolador Master se han implementado comunicaciones seriales simples con las cuales se detecta el final de una trama mediante los caracteres "FFF", por otro lado, se implementó una arquitectura por secciones al hacer uso de la función SWITCH CASE la cual permite tener un comportamiento establecido cuando se cumple cierto parámetro.

El código del programa se encuentra conformado por diferentes partes, al inicio del programa se han declarado las variables que van a ser utilizadas, luego se define las funciones con la cual se declaran los eventos seriales, además de eso se define la

función que recibe y decodifica las tramas enviadas por la Nextion en variables utilizadas dentro de la función principal.

A continuación, en la función “setup” se declara el modo de uso de los distintos pines del controlador Arduino y la frecuencia de trabajo a utilizar (esta se encuentra dada en baudios).

Finalmente, en la función principal “Loop” se realiza la lectura de las distintas señales analógicas con sus respectivas condiciones de ganancias y dependiendo de las circunstancias se ira comportando dependiendo del caso en el cual se asigne. Dentro de esta misma función son declaradas las señales de alarma del sistema, solo que necesitan ciertas consideraciones para ser tomadas en cuenta esto se debe a que al arranque del sistema los sensores obtendrán datos que no serán relevantes puesto que aún no se encuentran en óptimas condiciones de funcionamiento.

Los archivos de los esclavos disponen de una misma arquitectura de programación, las diferencias radican en los distintos procesos o instrucciones a controlar.

Tabla 1. Configuración de válvulas del sistema

Configuración	Válvula de Alimentación	Válvula de Recirculación	Válvula de Desagüe
Llenado del tanque de recirculación	Activada	Desactivada	Desactivada
Llenado del tanque + recirculación	Activada	Activada	Desactivada
Periodo de depuración	Desactivada	Activada	Desactivada
Periodo de extracción	Desactivada	Desactivada	Activada
Inicio y Final del sistema	Desactivada	Desactivada	Desactivada

2.5.3. Diseño del Rectificador Monofásico

Selección de Puente rectificador

$$\text{Tensión media en la carga } V_{med} = \frac{2 \cdot V_{max}}{\pi} \quad (2.1)$$

$$\text{Tensión eficaz en la carga } V_{RMS} = \frac{V_{max}}{\sqrt{2}} \quad (2.2)$$

$$\text{Tensión pico inverso } V_{RRM} \geq \sqrt{2} * (110 - 1.1) = 171.12 \text{ V} \quad (2.3)$$

Rendimiento 81%

Por esta razón se seleccionó el puente rectificador kGBPC802 el cual permite:

$$V_{RRM} = V_R = 200V \quad \text{y una corriente de salida } I_o = 8 A$$

Filtro Capacitivo para el rectificador

Se debe considerar que los capacitores electrolíticos implementados en los filtros van perdiendo algo de la capacitancia a lo largo del tiempo, por lo que se recomienda considerar una tolerancia al valor de la capacitancia inicial al momento de la selección.

$$C_{min} = 2 * \frac{P_{in}}{(V_{max}^2 - V_{min}^2) * f_{rect}} \quad (2.4)$$

Donde

P_{in} = Potencia de la carga, en Watts

V_{max} = Voltaje de pico en la línea

V_{min} = Voltaje mínimo permitido

f_{rect} = Frecuencia de rectificación

$$C_{min} = 2 * \frac{370}{(120^2 - 110^2) * 120} = 2.68 * 10^{-3} uF \quad (2.5)$$

De acuerdo con el cálculo se consideró un capacitor de 3000uF a 200v.

Como parte de protección del circuito rectificador se decidió implementar un fusible de 10A.

2.5.4. Control del módulo de conmutación

Como se enuncio anteriormente, para este proyecto se implementa la configuración de inversor Clase E de puente completo, mediante una modulación (Control de conmutación) de señal SPWM.

Esto es posible gracias a que el Arduino permite modular señales sinusoidales a través de diferentes señales PWM. Posibilitando realizar inversores sinusoidales puros con diferentes frecuencias. Como suele ser de conocimiento general, algunos pines del Arduino pueden generar señales PWM a frecuencias muy altas, lo cual resulta beneficioso pues se puede adatar para la ecuación sinusoidal.

La generación de la onda completa se realizará mediante dos pines uno para el flanco positivo y el otro para el flanco negativo. Debido que se desea generar una señal de salida suave elegimos la configuración Phase Correct PWM en la cual se dispone de una frecuencia de 31372 Hz.

Por lo cual mediante las siguientes ecuaciones se puede saber el número de pasos en grados con el cual se obtienen las distintas PWM.

$$T_{PCP} = \frac{1}{f_{PCP}} = \frac{1}{31372} \quad (2.6)$$

Numero de pulsos para un medio ciclo $N = \frac{T_{Seno}}{T_{PCP}}$

Es indispensable conocer el ancho de pulso en grados, debido que la señal sinusoidal está dada por $y = \sin(x)$

$$Ancho_{pulso} = \frac{180}{N} \quad (2.7)$$

Para agilizar el cálculo del vector con el cual se trabajará en el Arduino, se decidió implementar un código en Matlab que me permite obtener dicho vector de manera más fácil y permitiendo cambiar la frecuencia de la señal a generar.

2.5.5. Construcción de simulación del sistema

Al no disponer de los elementos para un montaje a menor escala del sistema automatizado se ha realizado una simulación del mismo, mediante la interfaz de Proteus, el simulador de las pantallas Nextion y un creador de puertos virtuales.

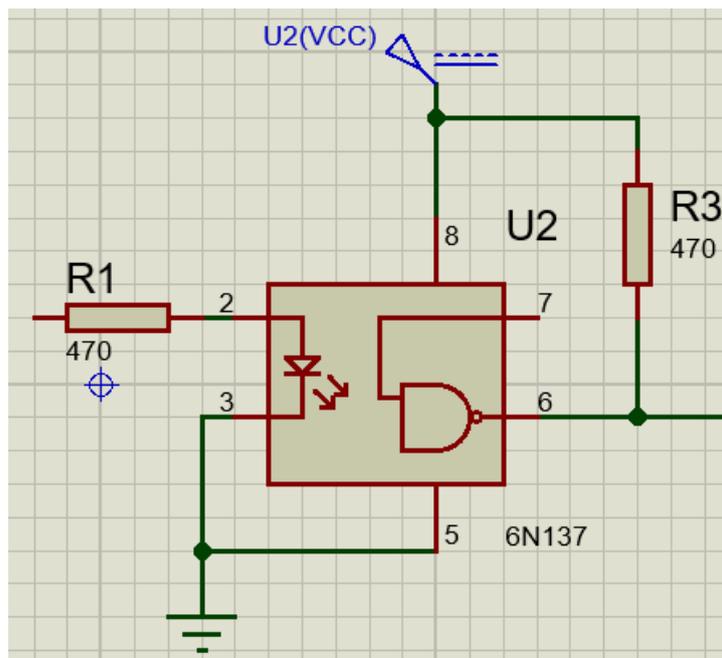


Ilustración 19. Conexión de optocoplador 6N137

Además de implementar los Optocopladores, para que los IGBTs conduzcan es necesario generar una señal de alrededor de 12V. Por lo cual al cambiar de estado los IGBTs generan tierras flotantes en las fuentes de disparo, obligando a implementar un circuito de disparo que sea capaz de generar los 12V independientemente para cada IGBT. Para la ejecución de dicha tarea se ha decidido implementar el CI IR2110 de International Rectifier.

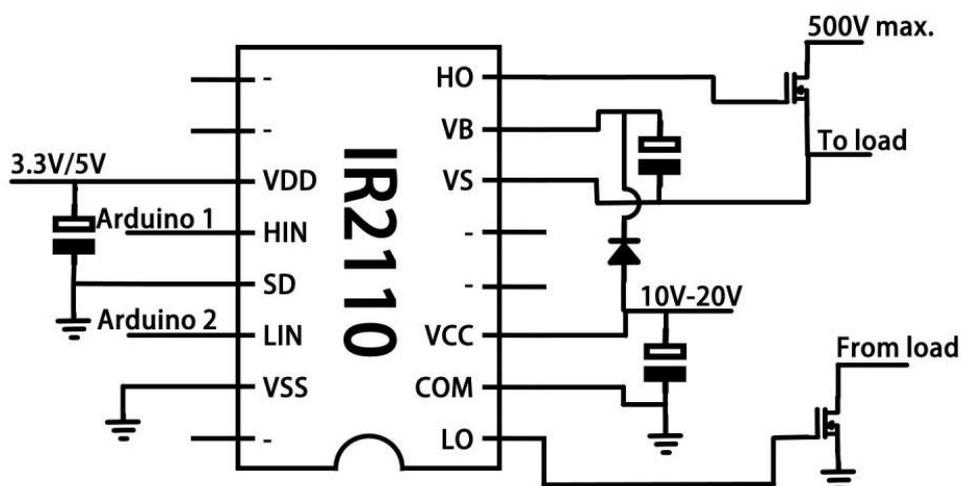


Ilustración 20. Conexión de Driver de disparo IR2110 para IGBTs

Mientras que el microcontrolador ARD2 se implementó con la finalidad de determinar el caudal del sistema de recirculación del agua. Estos procesos se ha

realizado en diferentes controladores debido que en ambos se hace uso de interrupciones con deltas de tiempo muy pequeñas, por lo cual al encontrarse los dos procesos en un solo controlador las medias de ambos se verían afectadas por las interrupciones del otro.

2.5.6. Diseño de Construcción

La elaboración del prototipo requiere el conglomerado de todos los equipos enunciados anteriormente los cuales haciendo uso de sus diferentes hojas técnicas se logra obtener un equipo el cual será expuesto a continuación.

2.5.6.1. Esquema eléctrico

Basándose en los diferentes módulos expuestos anteriormente en la sección 2.5.5, se pueden elaborar en 2 diagramas eléctricos que representen los sistemas configurado en Maestro y Esclavo, controlados por medio de su Arduino correspondiente, en el caso del Arduino Nano (Esclavo) y los IR2110 representan el variador de velocidad en la parte de control la cual acciona la etapa de potencia por los MOSFET's.

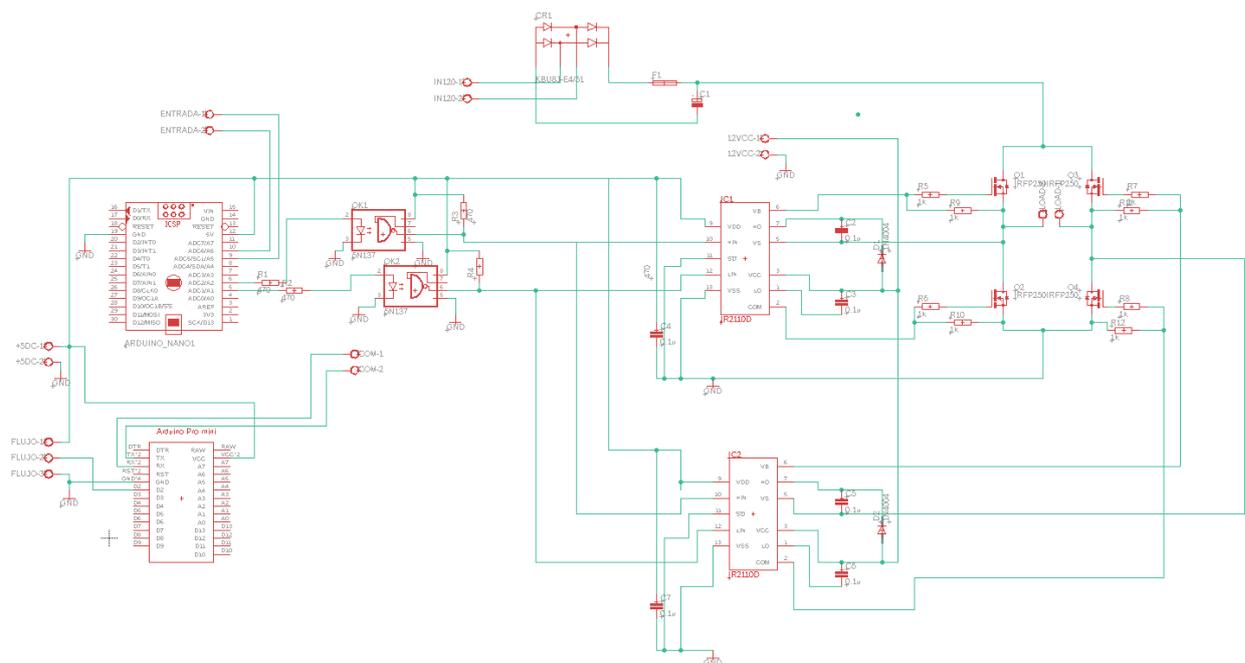


Ilustración 21. Diagrama esquemático de los Arduino Esclavo, y el sistema rectificador de potencia

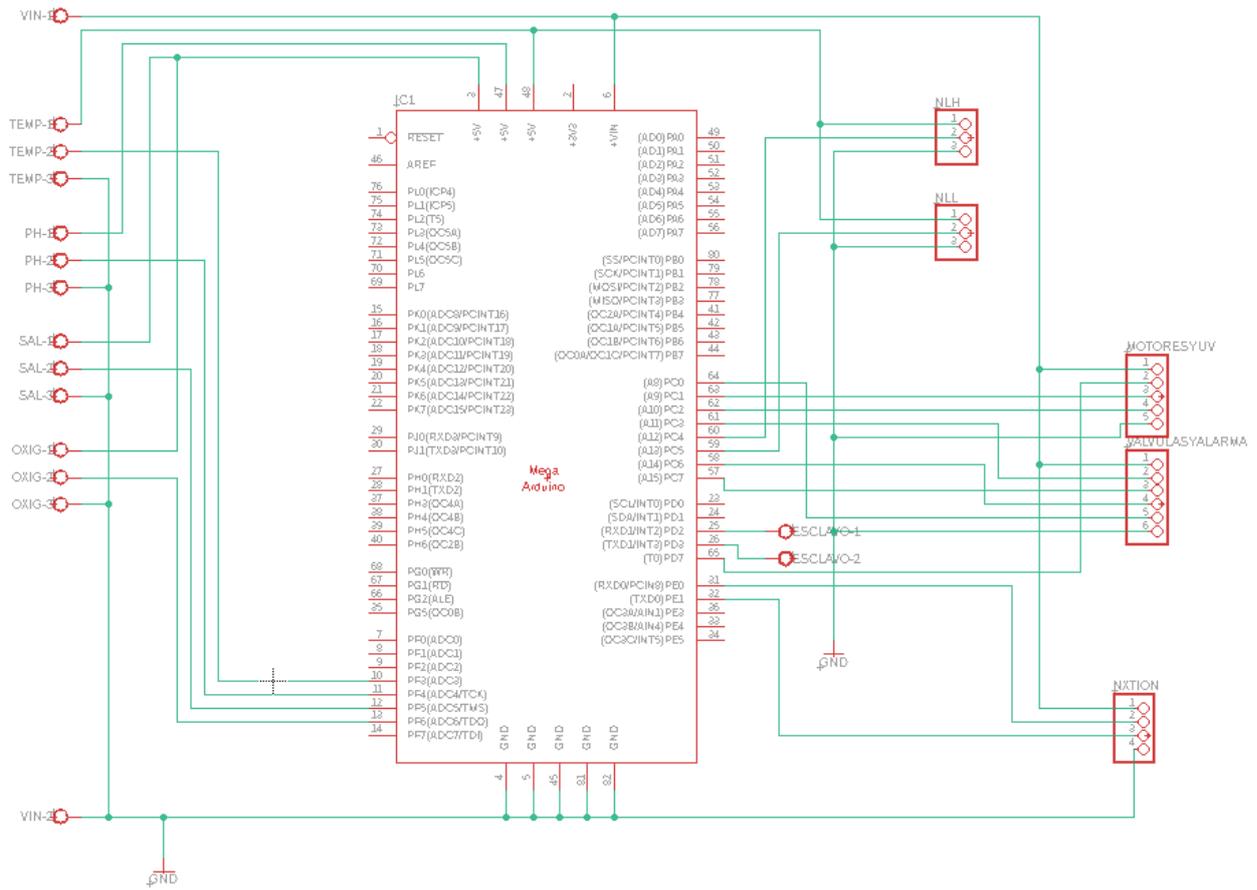
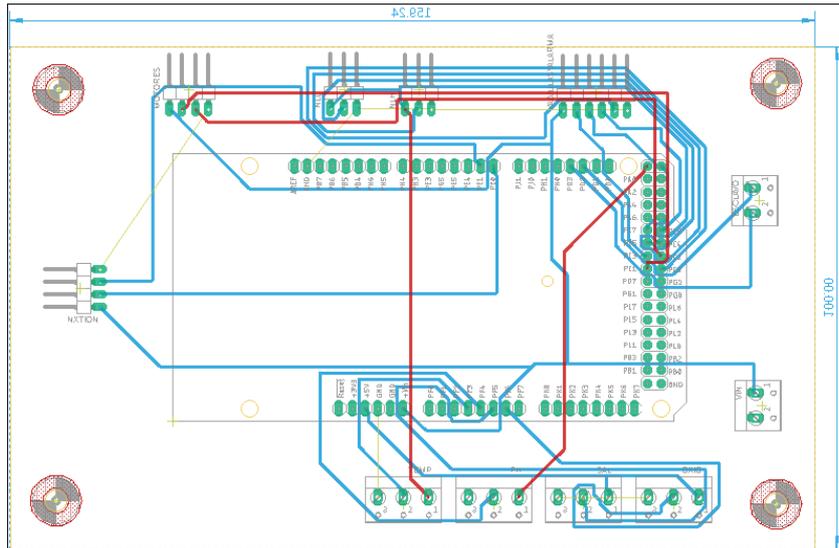


Ilustración 22. Esquemático del Arduino Maestro para realizar las conexiones pertinentes

2.5.6..2. Circuitos Impresos

Una vez que se ha concluido la definición de la electrónica básica para el correcto funcionamiento del sistema se procede a diseñar los circuitos impresos o comúnmente conocidos como PCB, tanto para la operación de control como potencia.

Por tanto, mediante la ayuda del software Eagle se procedió a realizar las placas de circuitos impresos por cuestión que los diagramas esquemáticos fueron elaborados en dicho software y la otra razón por la cual se seleccionó Eagle es que permite la elaboración de nuestros propios empaquetados en nuestro caso las librerías de Arduino.



Size: 6524 x 4273 mil

Ilustración 23. Dimensiones de la placa PCB para el Arduino Maestro

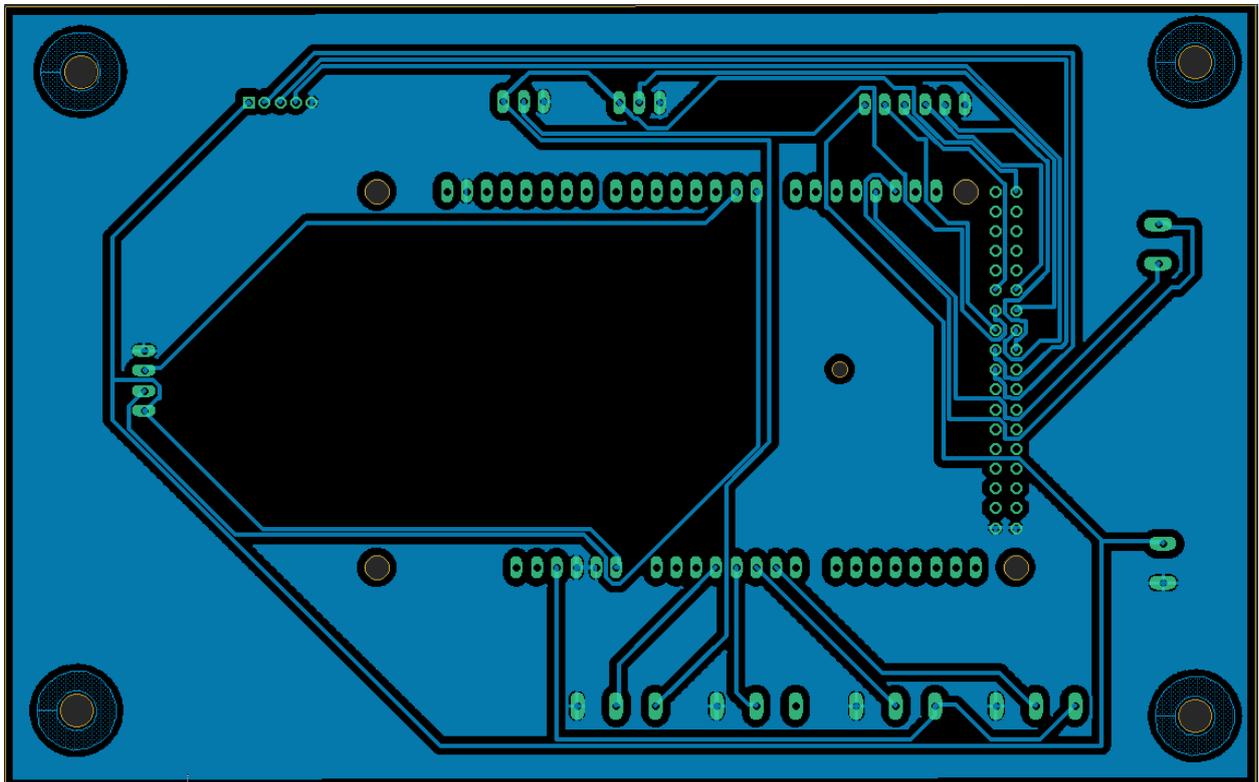


Ilustración 24. Vista Posterior Arduino Maestro

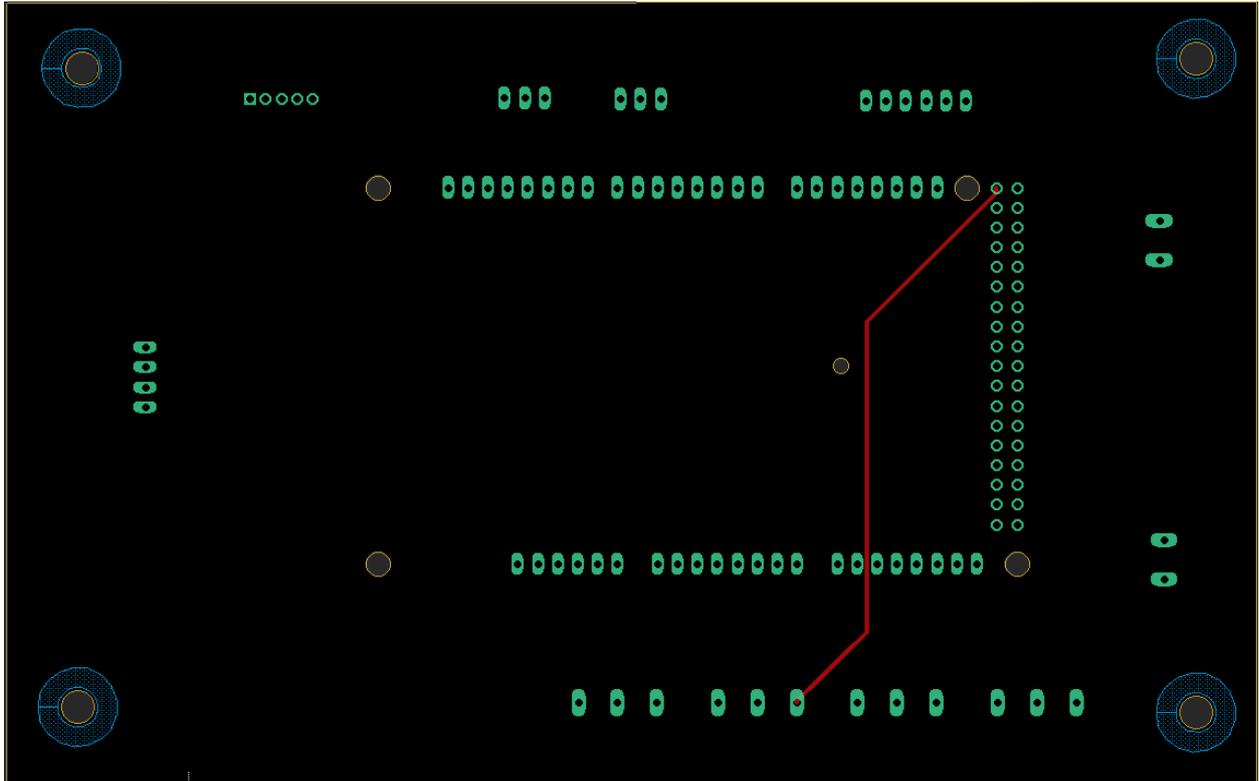
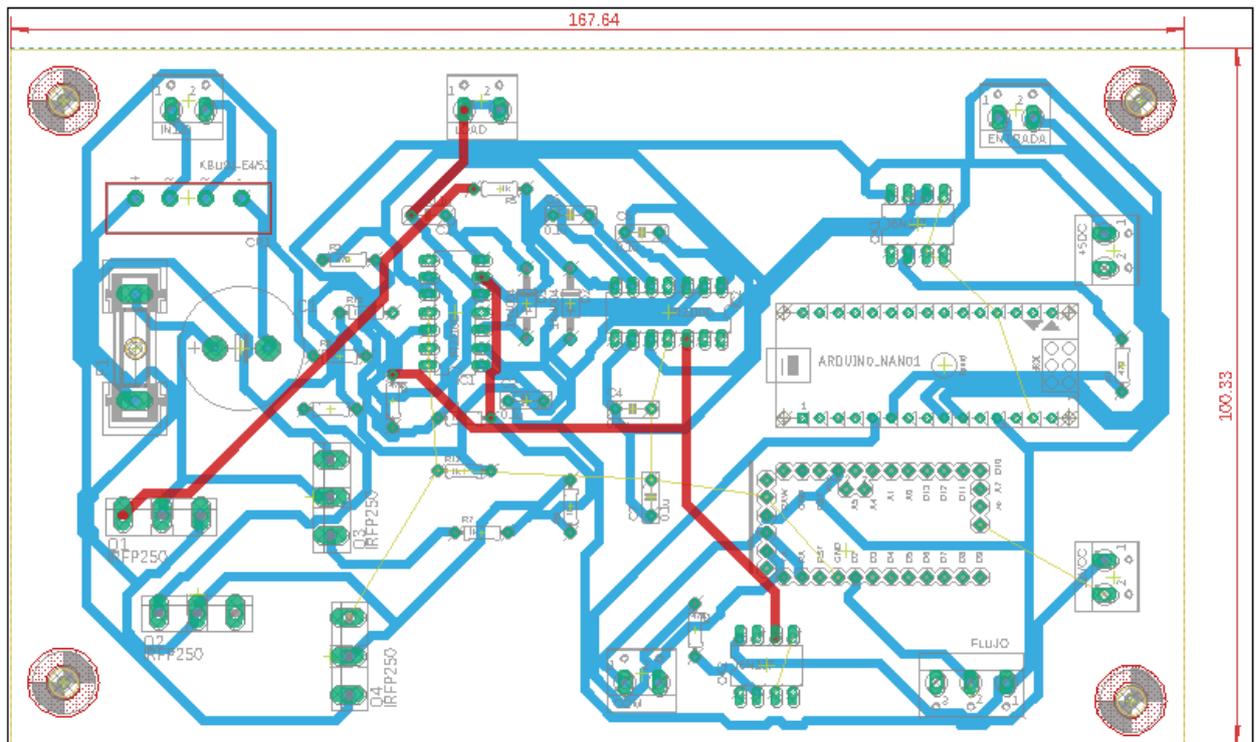


Ilustración 25. Vista Superior Arduino Maestro



Size: 6982 x 4176 mil

Ilustración 26. Dimensión de la placa PBC para los Arduino Esclavo y sistema rectificador de potencia

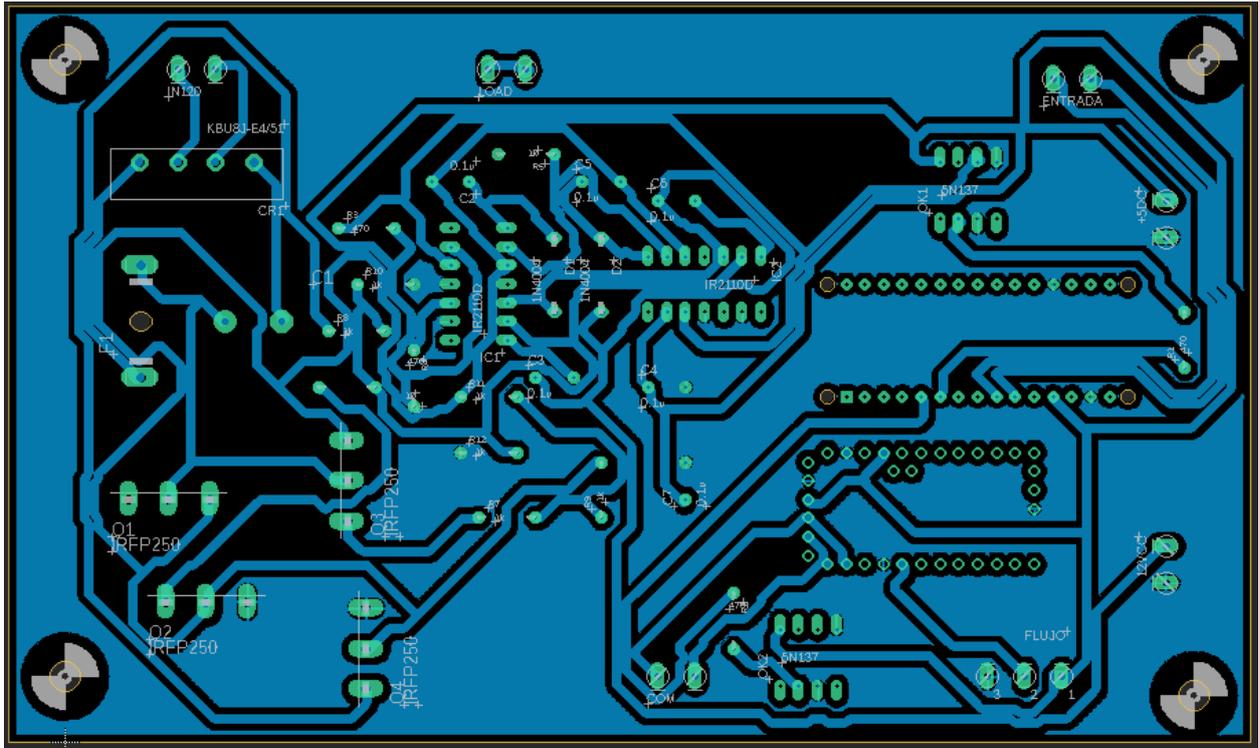


Ilustración 27. Vista Posterior Arduino Esclavos

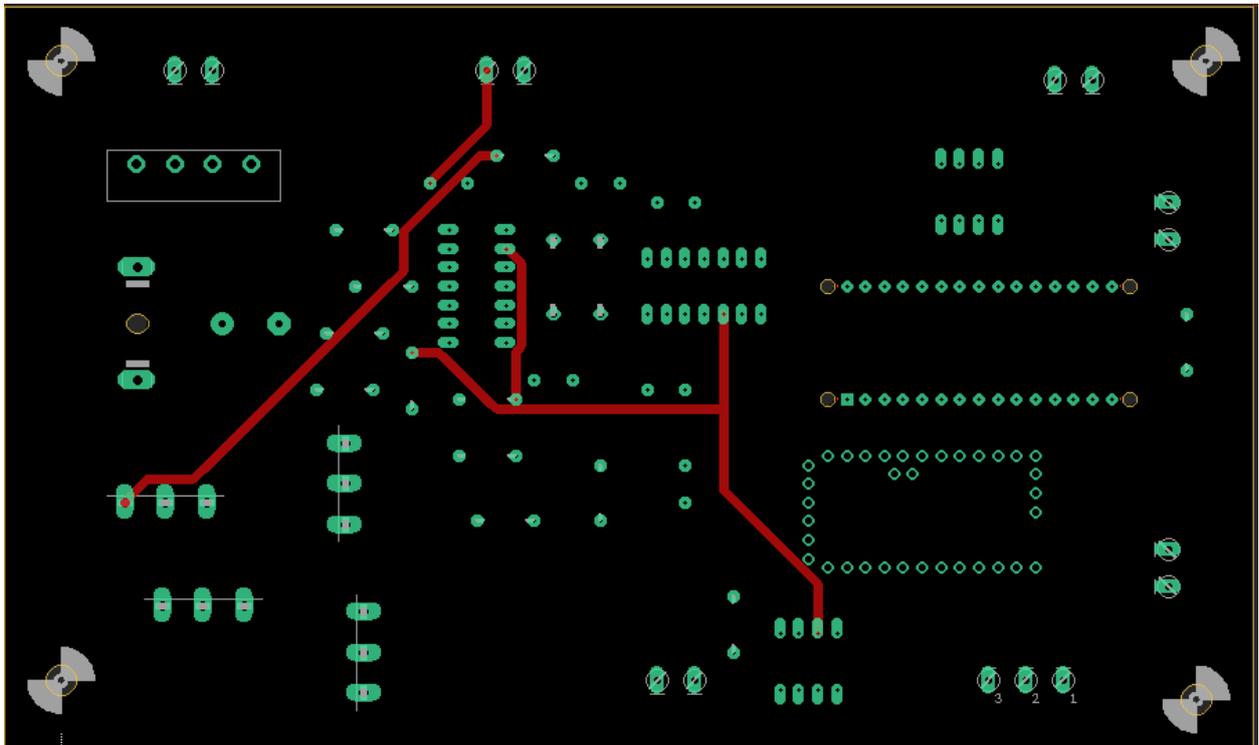


Ilustración 28. Vista Superior Arduino Esclavos

Para el circuito de la ilustración 22 es necesario la unión de todas las tierras y la conexión del pin 5 del IR2110 a la unión entre emisor del MOSFET alto y el colector del

MOSFET bajo, debido que esta misma permite la adecuada conmutación de la etapa de potencia.

2.5.7. Selección del sistema UV

Para el sistema de depuración a diseñar se necesitará implementar un sistema UV con el cual se logra la purificación del agua de los agentes microbiano, en base al requerimiento de la longitud de onda y caudal se seleccionó un sistema UV marca AquaUV STL12 con las siguientes características.

Longitud de onda: 258mm

Potencia: 39W

Dosis: 30mj/cm²

Caudal: 3 m³

Tiempo de vida: 10000 Horas.

Es importante destacar que a un mayor caudal la capacidad del sistema UV para filtrar el agua de agentes microbianos es menor.



Ilustración 29. Sistema Filtro UV

2.5.8. Dimensionamiento de bomba para recirculación

Para la selección de las tuberías, se usó como criterio principal la velocidad de circulación del agua, una de las recomendaciones de la FAO para sistemas a pequeña escala la velocidad mínima sea 1.2 m³/h, por lo cual recordando la selección de nuestro sistema UV se necesitará una bomba que provea de 2.14m de cabezal para 2.4m³/h. Por tanto, se selecciona una bomba para acuicultura PM21 de la marca IWAKI debido a

que coincide con un punto de operación cercano y dentro de nuestros límites establecidos

$$L_{\min}=1.2\text{m}^3/\text{h}$$

$$L_{\max}=3\text{ m}^3/\text{h}$$

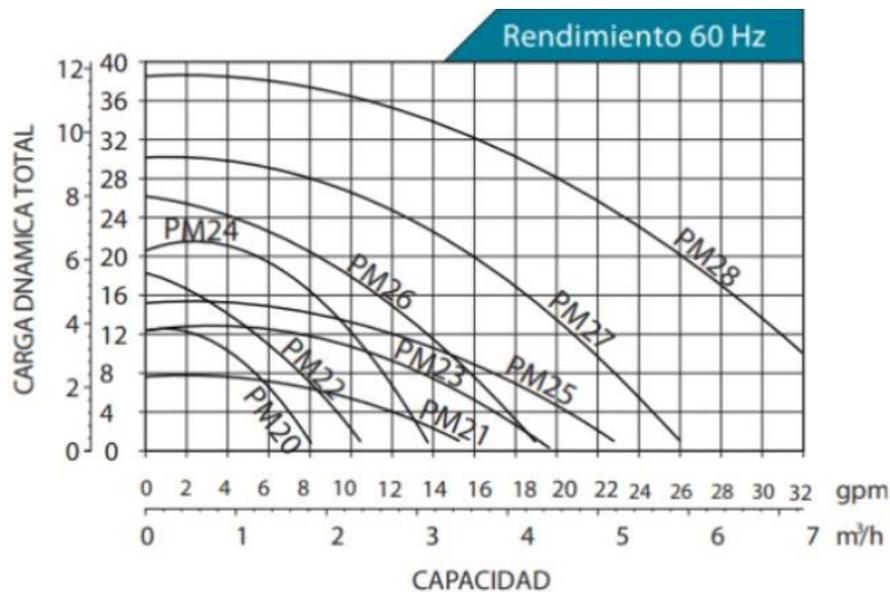


Ilustración 30. Curvas de operación Bomba Pentair PM21



Ilustración 31. Bomba Pentair, modelo PM21

2.5.9. Dimensionamiento de bomba para tanque de alimentación

El sistema completo de depuración funciona en 3 configuraciones, en la cual el controlador del sistema manipula las distintas válvulas de modo que se cumpla con los requerimientos de cada trabajo. En la primera configuración el tanque de almacenamiento bombea agua salada a los tres tanques de recirculación por lo cual se justifica la implementación de una bomba para el tanque de alimentación.

Por dicha razón es necesario determinar el caudal necesario para la bomba y se encontró que una bomba de 10m llena los tanques de recirculación por medio de sus ramificaciones con caudales de 2.52, 2.23 y 2.55 m³/h respectivamente. Además, de pasar un caudal de 7.3 m³/h que posea resistencia a agua salada.

Por las características enunciadas anteriormente se seleccionó la bomba IWAKI PM42, por cómo se puede observar en la curva de funcionamiento el punto de operación se encuentra muy cercano al punto deseado permitiendo obtener un caudal capaz de llenar los tanques de forma rápida sin reducir la capacidad de filtrado por el sistema UV.

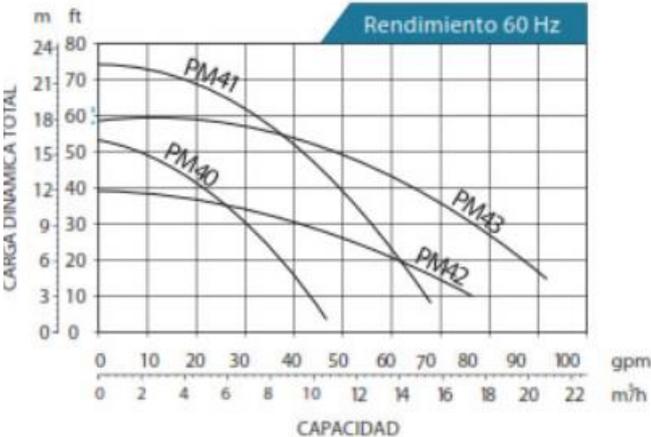


Ilustración 32. Curva de operación de las bombas Pentair modelo PM42



Ilustración 33. Modelo seleccionado Acuamain Pentair PM42

2.5.10. Diseño físico del tablero eléctrico (Control y Potencia)

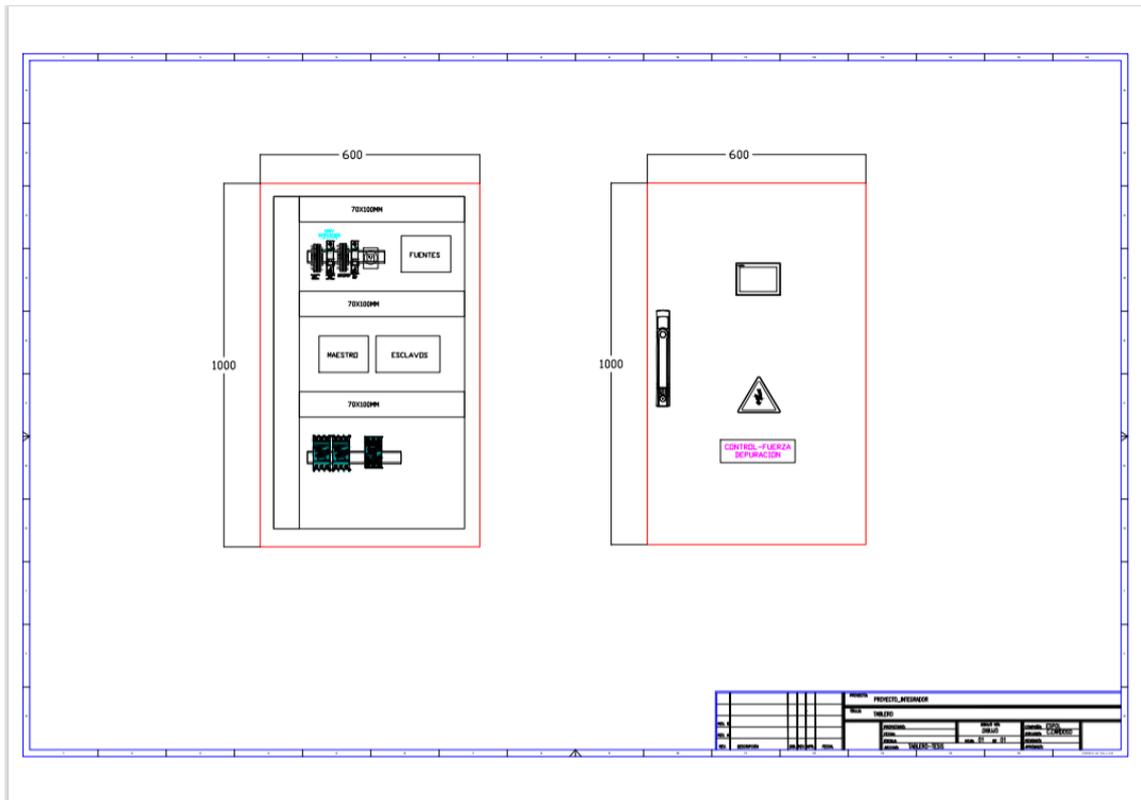


Ilustración 34. Diseño del tablero físico de Control y Potencia

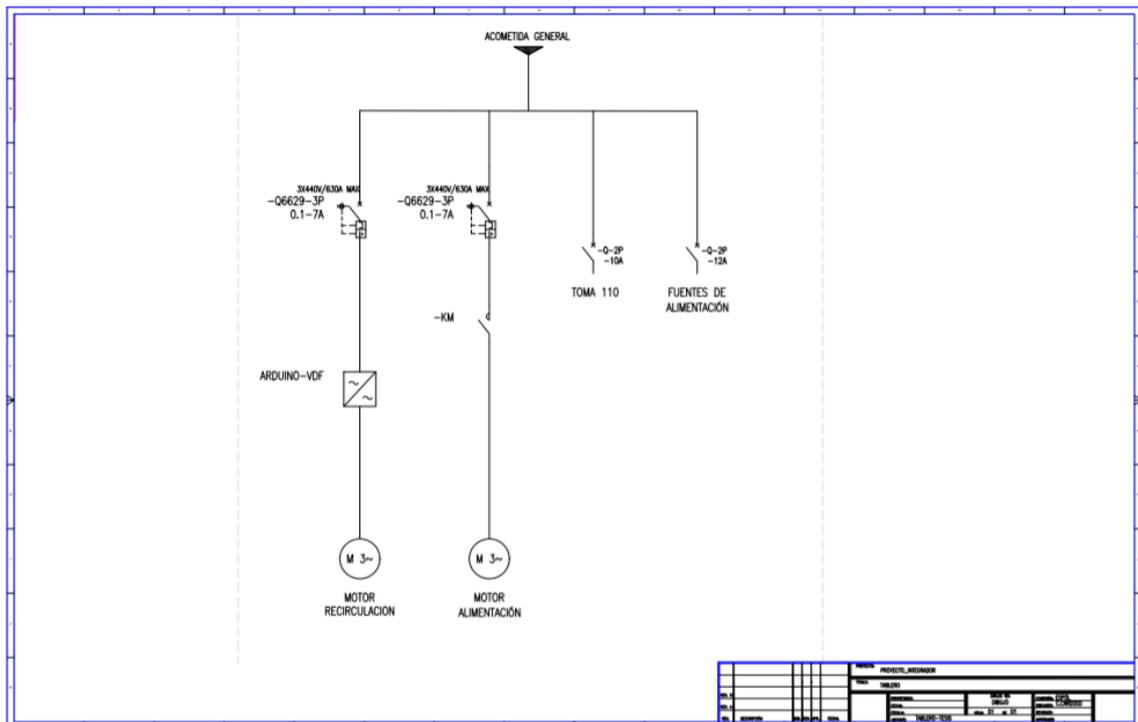


Ilustración 35. Diagrama unifilar de Potencia y Control

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

3.1. Resultados de la simulación

Los resultados que se observaron mediante la simulación son satisfactorios y confirman el diseño realizado por lo que puede pasar a la construcción en físico del sistema automatizado de depuración de moluscos bivalvos, este conlleva en obtener los valores de los parámetros de funcionamiento mediante sensores, control de motores y comunicación I2c entre microcontroladores.

A continuación, se puede observar la señal SPWM generada mediante la programación del Arduino, a través del osciloscopio que ofrece la plataforma Proteus. Se visualiza que la señal generada es más continua en su parte central, esto se debe a que durante este intervalo la señal sinusoidal llega a su valor máximo.

En base a las posibles funciones del sistema, se ha configurado 2 funcionalidades una a través de 30Hz y la otra a frecuencia nominal de 60Hz.

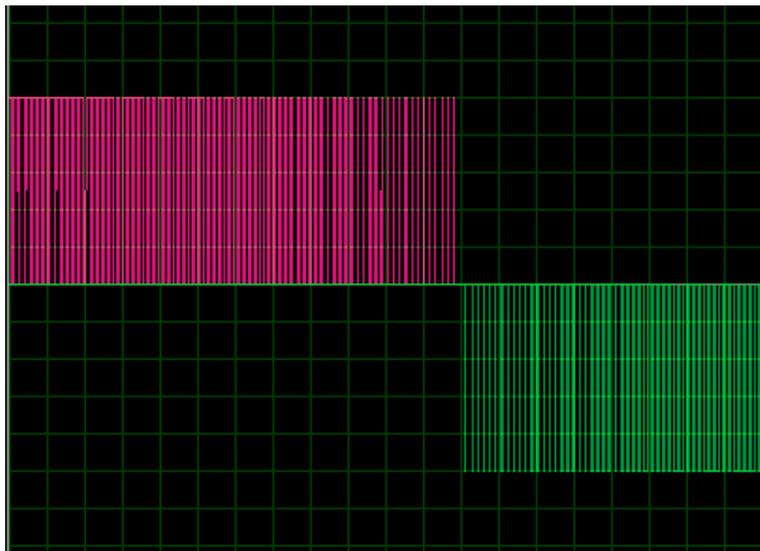


Ilustración 36. Señal SPWM generado por Arduino en la plataforma Proteus

Además, se realizó una simulación por separado realizando una pequeña modificación con la finalidad de eliminar el armónico más influyente en la señal SPWM generando de esta manera una senoidal pura.

Por otro lado, observamos que el comportamiento de la pantalla Nextion cumple con los parámetros establecidos, pues cada uno de los botones envían sus correspondientes tramas modificando de esta manera el estado de funcionamiento para el microcontrolador.



Ilustración 37. Simulador de la Pantalla Nextion comunicada con la simulación de Arduino

Considerando que es una fase de prueba y error la regulación de ellos parámetros del sistema se recomienda implementar la programación del controlador de velocidad de la bomba mediante el Software de Matlab esto se debe a que dicho programa permite realizar un ajuste de los parámetros del programa en tiempo real, además de proveer de una interfaz más intuitiva y exacta sin límite de procesamiento como es Simulink. A continuación, se puede observar las señales obtenidas mediante la programación en Matlab.

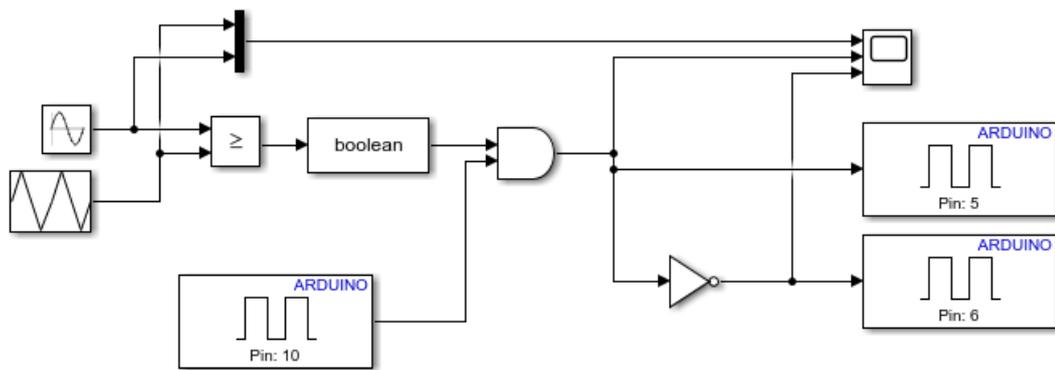


Ilustración 38. Programación de Arduino mediante Simulink-Matlab

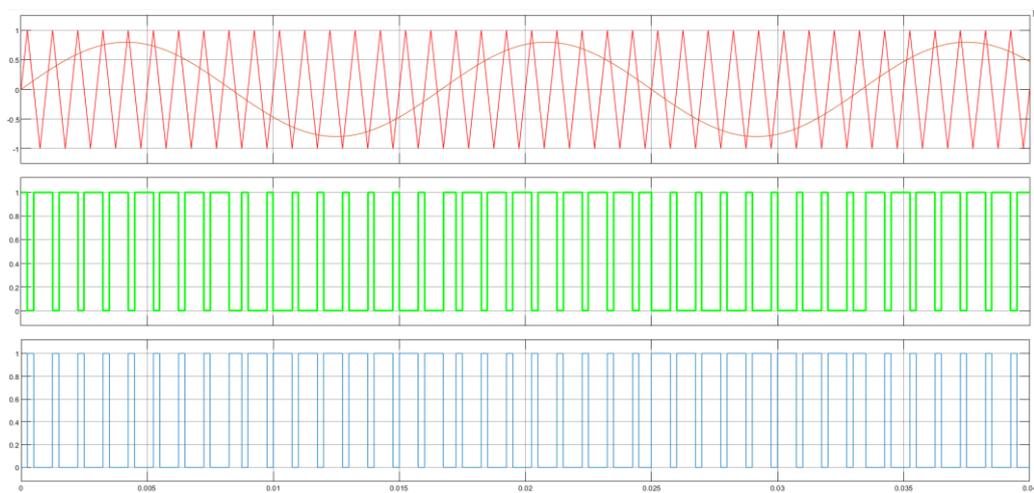


Ilustración 39. Señales de Control SPWM obtenidas por medio de la plataforma de Simulink

3.2. Costos de Ingeniería

En la tabla 1 se muestran los costos generados por el personal que intervendrá en el proyecto.

Tabla 2. Costos del Personal que Intervino en el proyecto

Personal	Cantidad	Horas	Costos p/h\$	Importe \$
Ingeniero en control y automatización	1	80	15	1,200
Asesor	1	4	40	160
Capturista	1	15	8	120
Dibujante	1	8	12	96
Técnico en electrónica	1	16	9	144
			Total	1,720

3.3. Catálogo de conceptos

En la tabla 2 se muestra la relación de los materiales y maquinaria utilizados para el desarrollo de nuestro proyecto.

Tabla 3. Catálogo de conceptos

Partida	Concepto	Marca o similar	Cantidad	Precio Unitario	Importe
1	Arduino Mega 2560	Arduino	1	20.00	20.00
2	Arduino Nano v3	Arduino	1	8.00	8.00
3	Arduino Micro	Arduino	1	7.00	7.00
4	Electroválvulas	VYRSA	3	35.85	107.55
5	Sensor de Nivel	GENERICO	3	6.00	18.00
6	Sensor de Temperatura	GENERICO	1	7.00	7.00
7	Sensor PH + Acondicionador	DfRobot	1	50.00	50.00
8	Sensor de Conductividad	DfRobot	1	79.90	79.90
9	Sensor de Oxígeno Disuelto	DfRobot	1	169.00	169.00
10	Pantalla Nextion 7´	Nextion	1	100.00	100.00
11	Módulo de relé 4 canales	GENERICO	2	6.50	13.00
12	Barniz de Tropicalización	GENERICO	2	25.00	50.00
13	Luz de emergencia estroboscópica	GENERICO	1	13.00	13.00
14	Sistema UV	AquaUV STL12	1	750.19	750.19
15	Bomba de recirculación	IWAKI PM21	1	350.00	350.00
16	Bomba de llenado	IWAKI PM42	1	350.00	350.00
17	Optoacoplador	6N137	2	0.50	1.00
18	Resistencia 470	Genérico	4	0.20	0.80
19	Driver Disparador	IR2110	2	0.35	0.70
20	Capacitor 1uF 16V Y5V 0805	KEMET	6	2.60	15.6
21	Diodo Zener	GENERICO	2	1.10	2.20
22	MosFet	IRFP250	4	0.75	3.00
23	Resistencia 1K 1/4W	Genérico	8	0.54	4.32
24	Base Fusible y fusible	Genérico	1	0.38	0.38
25	Puente rectificador de diodos	KBU8J-E4/51	1	3.80	3.80
26	Capacitor 2700uF	EPCOS	1	33.29	33.29
27	Baquelita 20*30cm	Fibra de Vidrio	3	7.99	23.97
28	Acido Cloruro Ferrico	Genérico	3	1.00	3.00
				Total	2,184.70

Finalmente, de acuerdo con las tablas 2 y 3 obtenemos que el costo de maquinaria y automatización del proyecto es de \$3,904.70.

CAPÍTULO 4

4. Conclusiones Y Recomendaciones

4.1. Conclusiones

- Durante la simulación del sistema se ha observado que la comunicación mediante puertos virtuales es un poco más lenta que disponer de los elementos físicamente, pero se debe indicar que los tiempos de respuesta de las alarmas es casi al instante, lo cual permite elaborar una simulación confiable del sistema.
- Implementar este prototipo permite obtener un sistema de alta fidelidad a un costo muy económico en relación a la tecnología disponible actualmente en el mercado, con el cual se podrán realizar los diferentes estudios determinando las condiciones ambientales idóneas para una correcta depuración de los moluscos bivalvos.
- La generación de una señal SPWM mediante Arduino y MatLab permite implementar un variador de frecuencia de muy bajo costo, permitiendo que el proyecto sea más interactivo y poder estipular las limitaciones del caudal con respecto a los distintos moluscos.
- Al disponer de una interfaz gráfica fácil de entender y de manejar, esto genera una ventaja muy grande para el operador puesto que en caso de no disponer de los conocimientos necesarios del funcionamiento del proceso es suficiente con poder arrancar el sistema, visualizar los valores y alarma de las distintas variables censadas.
- Al momento de realizar la polarización de varios Mosfet que cumplen el propósito de interruptor (apertura y cierre) de señales, se debe implementar CI (Circuitos integrados) la separación de las referencias de cada transistor, pues al no implementarse se observarían errores de amplitud en la señal de salida del inversor.

4.2. Recomendaciones

- Implementar el sistema de programación mediante Matlab puesto que dicho sistema nos permite un control de parámetros en tiempo real, esto se debe a que aún se debe determinar el parámetro ideal de funcionamiento para nuestro producto final.

- Se recomienda realizar esta placa de control de velocidad debido que este diseño nos permite reducir costos en relación a implementar un variador de velocidad modular.
- Se recomienda una vez implementado el prototipo realizar pruebas con la finalidad de poder determinar ciertos parámetros de programación, como son las ganancias de los sensores analógicos al momento de ser enviados a la pantalla Nextion.

Bibliografía

Fao.org. 2020. [online] Available at: <<http://www.fao.org/3/i0201s/i0201s.pdf>> [Accessed 10 September 2020].

óptica., M. (2020). [PDF] Métodos de medida del ph para estudios invasivos prolongados: ISFET y fibra óptica. - Free Download PDF. Retrieved 10 September 2020, from <https://silo.tips/download/metodos-de-medida-del-ph-para-estudios-invasivos-prolongados-isfet-y-fibra-optic>

Luna, E. (2020). Instrumentacion Industrial 8 edicion Antonio Cruz Sole. Retrieved 10 September 2020, from https://www.academia.edu/31478892/Instrumentacion_Industrial_8_edicion_Antonio_Cruz_Sole

Andriach, Juan Pablo; Diaz, Ariel Iván; Mariani, Cristian; Silva Bustos, Matías; Iparraguirre, Javier Generación de ondas SPWM con Arduino para la excitación de inversores trifásicos. Universidad Tecnológica Nacional Facultad Regional Bahía Blanca Argentina. 2012.

Chávez, Ibarra Samuel; Jiménez, Herrera Humberto. Diseño de un control de velocidad de un motor de inducción a 220 V CA.

Rashid Muhammad H. Electrónica de potencia: circuitos, dispositivos y aplicaciones, Tercera edición. S.A

Maquinas Eléctricas y Sistemas de Potencia, Sexta Edición

Savage, J. A. (2020, 1 julio). Arduino y Dynamixel AX-12. savageelectronics. <https://savageelectronics.blogspot.com/2011/01/arduino-y-dynamixel-ax-12.html>

Arduino Mega 2560 Rev3 | Arduino Official Store. (s. f.). store arduino. Recuperado 9 de junio de 2020, de <https://store.arduino.cc/arduino-mega-2560-rev3>

How to generate a sine wave from Arduino or atmega328, Disponible en: <http://www.eprojectszone.com/how-to-generate-a-sine-wave-from-arduino-or-atmega-328/>

Teaching the Operation of Full Bridge Converters: Power Interchange Analysis between Inductors and Capacitors, Nicolas Muñoz-Galeno, Disponible en: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-50062016000100013

Desarrollo de un inversor trifásico con modulación SPWM e Interfaz gráfica para el laboratorio de electrónica de potencia, Reyes y Moya, Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/15079/1/UPS%20-%20ST003403.pdf>

APÉNDICES

Apéndice A

Fuentes de alimentación 5VDC, 15VDC.

Se deberá implementar dos transformadores con salida de 6VCA a 1A y 24VCA a 1^a, ambos con una entrada de 120 VAC. Ambos alimentan a los puentes de diodos.

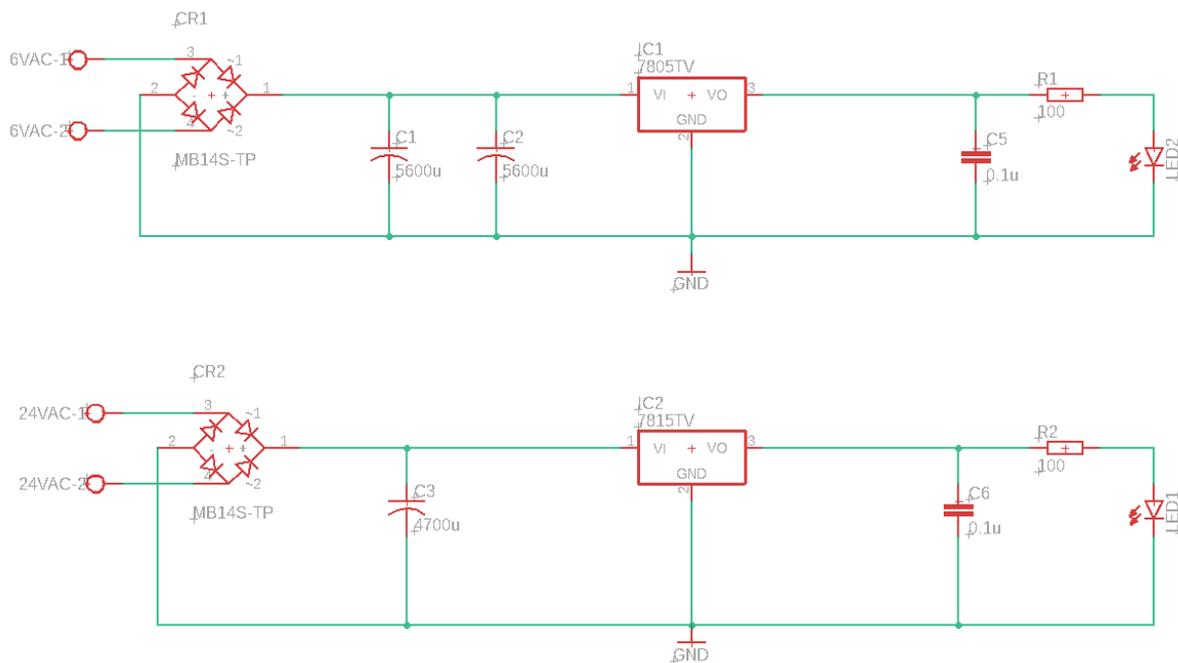


Ilustración 40. Fuentes de alimentación DC

Cálculos para la obtención de los valores de C1, C2 y C4

Para C1 y C2:

$$V_{max} = V_i * 1.41422$$

$$C = \frac{5 * I}{f * V_{max}}$$

Valor de la salida del transformador $V_i = 6V$ y $I = 1A$

Por lo cual, obtenemos:

$$V_{max} = 6 * 1.41422 = 8.4852$$

$$C = \frac{5 * 1}{60 * 8.4852} = 9821.02 \mu F$$

Como el valor obtenido no hace referencia a los valores comerciales disponibles se implementaron dos capacitores de 56000uF.

Para C4:

$$V_{max} = V_i * 1.41422$$

$$C = \frac{5 * I}{f * V_{max}}$$

Valor de la salida del transformador $V_i = 15V$ y $I = 1A$

Por lo cual, obtenemos:

$$V_{max} = 15 * 1.41422 = 21.2113$$

$$C = \frac{5 * 1}{60 * 21.2113} = 3928.4 \mu F$$

Como el valor obtenido no hace referencia a los valores comerciales disponibles se implementó un capacitor de 4700uF.

Apéndice B

TIMER/CONTADOR de 8bits con PWM

El timer/contador es un módulo para temporizadores y conteo de 8 bits de propósito general, con 2 unidades comparadoras de salida y soporte a PWM con varios modos de operación. Por este motivo se decidió implementar este contador con la finalidad de poder escribir los ciclos de trabajo es decir los semiciclos, pues este módulo permite temporizar con una precisión muy alta la ejecución de la duración en alto y bajo de nuestra señal hecha vector. En la cual el 250 representa un ciclo en la configuración Phase Correct PWM.

A continuación, se enlistan las características más relevantes del TCO:

- Dispone de 2 unidades independientes con comparadores de salida
- Registros en dichos comparadores y además de 2 búfer
- Periodo de PWM Variable
- Generador de Frecuencia

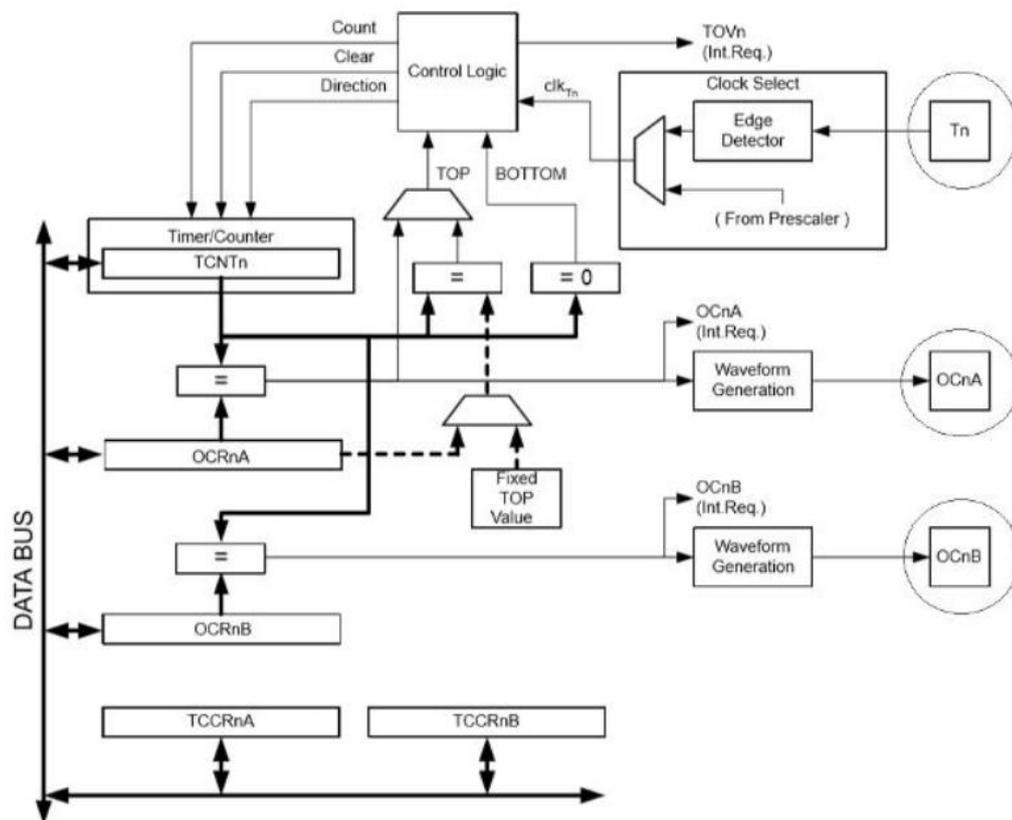


Ilustración 41. Diagrama en Bloques de Timer/Counter 0

ANEXOS

Código de programa Arduino Maestro

```
#include <avr/io.h>
#include<avr/interrupt.h>
#define DEBUG(a) Serial1.println(a);
unsigned int velocidad_grafica;
String vectorCaracteres;
char vectorChar[10]; //No creo que se coloquen mas de 9 digitos mas el signo
boolean TransmisionCompleta = false;
int velocidad = 0;
float flow_Lmin = 0;

//variables
unsigned int codigo;
unsigned int temp;
unsigned int ph;
unsigned int sali;
unsigned int oxi;
char arranque=0;
char actual=0;
unsigned long tiempo1 = 0;
unsigned long tiempo2 = 0;
unsigned long tiempominutos = 0;

//comunicacion
char envia[30]; //trama a enviar
char trama[30]; //trama recibida
unsigned char dato;
char cuenta=0;
boolean flag_rx=0;

//digitales IO
char valvulaA=30;
char valvulaR=31;
char valvulaDe=37;
char MotorIn=36;
char MotorR=35;
char nbajo=32;
char nalto=33;
char ndesague=34;
char FUV=38;

//analógicas I
char Temperatura=3;
char pH=4;
char Salinidad=5;
char OxigenoD=6;
```

```

//constantes de alarmas
unsigned int Mintemp=10;
unsigned int Maxtemp=20;
unsigned int Minph=5;
unsigned int Maxph=10;
unsigned int Minsali=40;
unsigned int Maxsali=70;
unsigned int Minox=70;
unsigned int Maxox=90;
unsigned int alarmat;
unsigned int alarmap;
unsigned int alarmas;
unsigned int alarmao;

const int ledPin = 29; // the number of the LED pin Alarm
int ledState = LOW; // ledState used to set the LED
long previousMillis = 0; // will store last time LED was updated
long intervalOn = 4000; // medio segundo ON
long intervalOff = 2000; // cinco segundos OFF

void serialEvent()
{
  while (Serial.available())
  {
    dato =(unsigned char)Serial.read();
    trama[cuenta]=dato;

    if (( cuenta>3) && ((unsigned char)trama[cuenta]==0xff) && ((unsigned char)trama[cuenta-
1]==0xff) && ((unsigned char)trama[cuenta-2]==0xff) )
    {
      flag_rx=1;
    }
    cuenta++;
    if (cuenta>=30)
    {
      memset (trama,0,30);
      cuenta=0;
    }
  }
}

void procesa_rx()
{
  if (flag_rx==1)
  {
    if ( strstr(trama,"led1=on" )) {arranque=1;}
    else if ( strstr(trama,"led2=on" )) {arranque=5;}
    else if ( strstr(trama,"del1=off" )) {arranque=7;}
    else if ( strstr(trama,"del2=off" )) {arranque=4;}
    else if ( strstr(trama,"del3=off" )) {arranque=6;}
  }
}

```

```

    memset(trama,0,30);
    cuenta=0;
    flag_rx=0;//vuelve a limpiar rx
  }
}

void setup() {
  // put your setup code here, to run once:
  pinMode(ledPin, OUTPUT); //Alarma del sisetma
  pinMode(FUV, OUTPUT); // Filtro UV
  pinMode(MotorIn,OUTPUT); //Motor alimentacion no controlado
  pinMode(MotorR,OUTPUT); //Motor recirculacion controlado
  pinMode(valvulaA,OUTPUT); //Valvula de alimentacion
  pinMode(valvulaR,OUTPUT); //Valvula de recirculacion
  pinMode(valvulaDe,OUTPUT); //valvula de desague
  pinMode(3,OUTPUT); //salida PWM
  pinMode(nbajo,INPUT);
  pinMode(nalto,INPUT);
  Serial.begin(9600);
  Serial1.begin(9600);
}

void loop() {

  //Falta calibrar los valores de las señales
  codigo=analogRead(Temperatura);
  temp=codigo*50.0*10.0/1023.0;

  codigo=analogRead(pH);
  ph=(codigo*50.0*10.0/1023.0)*0.0657+1.8571;

  codigo=analogRead(Salinidad);
  // sali=(1000*(codigo/1024*5000)/820/200)/(1+0.0185*(temp-25));
  sali=(codigo*50.0*10.0/1023.0)*0.6667;

  codigo=analogRead(OxigenoD);
  oxi=(codigo*50.0*10.0/1023.0)*0.6667;

  if(arranque==2 or arranque==3){
    if(temp<Mintemp or temp> Maxtemp){alarmat=1;
    sprintf(envia,"Detalles.n0.bco=63488");
    Serial.print(envia);
    Serial.print("\xFF\xFF\xFF");}
    else{alarmat=0;
    sprintf(envia,"Detalles.n0.bco=50712");
    Serial.print(envia);
    Serial.print("\xFF\xFF\xFF");}
    if(ph<Minph or ph> Maxph){alarnap=1;
    sprintf(envia,"Detalles.n1.bco=63488");
    Serial.print(envia);

```

```

Serial.print("\xFF\xFF\xFF");
else{alarmap=0;
sprintf(envia,"Detalles.n1.bco=50712");
Serial.print(envia);
Serial.print("\xFF\xFF\xFF");}
if(sali<Minsali or sali> Maxsali){alarmas=1;
sprintf(envia,"Detalles.x2.bco=63488");
Serial.print(envia);
Serial.print("\xFF\xFF\xFF");}
else{alarmas=0;
sprintf(envia,"Detalles.x2.bco=50712");
Serial.print(envia);
Serial.print("\xFF\xFF\xFF");}
if(oxi<Minox or oxi> Maxoxi){alarmao=1;
sprintf(envia,"Detalles.x3.bco=63488");
Serial.print(envia);
Serial.print("\xFF\xFF\xFF");}
else{alarmao=0;
sprintf(envia,"Detalles.x3.bco=50712");
Serial.print(envia);
Serial.print("\xFF\xFF\xFF");}

if(alarmat==1 or alarmap==1 or alarmas==1 or alarmao==1){
  unsigned long currentMillis = millis();
  if (ledState == LOW) {
    if(currentMillis - previousMillis > intervalOff) {
      previousMillis = currentMillis;
      ledState = HIGH;
    }
  } else {
    if(currentMillis - previousMillis > intervalOn) {
      previousMillis = currentMillis;
      ledState = LOW;
    }
  }
  }
digitalWrite(ledPin, ledState);}
}

switch (arranque) {
case 1:
  digitalWrite(valvulaA,HIGH);
  digitalWrite(valvulaR,LOW);
  digitalWrite(valvulaDe,LOW);//desague
  digitalWrite(FUV,HIGH);
  digitalWrite(MotorR,LOW);
  digitalWrite(MotorIn,HIGH);
  Serial.print("Sistema.p0.pic=3");
  Serial.print("\xFF\xFF\xFF");
  Serial.print("Sistema.p1.pic=4");
  Serial.print("\xFF\xFF\xFF");
  Serial.print("Sistema.p3.pic=4");

```

```

Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p5.pic=9");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p4.pic=5");
Serial.print("\xFF\xFF\xFF");
actual=arranque;
if(digitalRead(nbajo)==1 and digitalRead(nalto)==0){
  arranque=2;
}
if(digitalRead(nbajo)==1 and digitalRead(nalto)==1){
  arranque=3;
}
procesa_rx();
delay(100);
break;
case 2:
digitalWrite(valvulaA,HIGH);
digitalWrite(valvulaR,HIGH);
digitalWrite(valvulaDe,LOW);//desague
digitalWrite(FUV,HIGH);
digitalWrite(MotorR,HIGH);
digitalWrite(MotorIn,HIGH);
Serial.print("Sistema.p0.pic=3");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p1.pic=3");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p3.pic=3");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p5.pic=8");
Serial.print("\xFF\xFF\xFF");
actual=arranque;
if(digitalRead(nbajo)==0 and digitalRead(nalto)==0){
  arranque=1;
}
if(digitalRead(nbajo)==1 and digitalRead(nalto)==1){
  arranque=3;
}
procesa_rx();
delay(100);
break;
case 3:
digitalWrite(valvulaA,LOW);
digitalWrite(valvulaR,HIGH);
digitalWrite(valvulaDe,LOW);//desague
digitalWrite(FUV,HIGH);
digitalWrite(MotorR,HIGH);
digitalWrite(MotorIn,LOW);
Serial.print("Sistema.p0.pic=4");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p1.pic=3");
Serial.print("\xFF\xFF\xFF");

```

```

Serial.print("Sistema.p3.pic=3");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p5.pic=7");
Serial.print("\xFF\xFF\xFF");
actual=arranque;
if(digitalRead(nbajo)==1 and digitalRead(nalto)==0){
  arranque=2;
}
procesa_rx();
delay(100);
break;
case 4:
  sprintf(envia,"Detalles.n0.val=%d",(unsigned int)temp);
  Serial.print(envia);
  Serial.print("\xFF\xFF\xFF");

  sprintf(envia,"Detalles.n1.val=%d",(unsigned int)ph);
  Serial.print(envia);
  Serial.print("\xFF\xFF\xFF");

  sprintf(envia,"Detalles.n2.val=%d",(unsigned int)sali);
  Serial.print(envia);
  Serial.print("\xFF\xFF\xFF");

  sprintf(envia,"Detalles.n3.val=%d",(unsigned int)oxi);
  Serial.print(envia);
  Serial.print("\xFF\xFF\xFF");

  procesa_rx();
  delay(100);
  break;
case 5:
  arranque=actual;
  procesa_rx();
  delay(100);
  break;
case 6:
  procesa_rx();
  delay(100);
  break;
case 7:
  digitalWrite(valvulaA,LOW);//alimentacion
  digitalWrite(valvulaR,LOW);//recirculacion
  digitalWrite(valvulaR,LOW);
  digitalWrite(valvulaDe,HIGH);//desague
  if(digitalRead(ndesague)==1){
    arranque=0;
  }
  delay(100);
  break;
default:

```

```

digitalWrite(valvulaA,LOW);
digitalWrite(FUV,LOW);
digitalWrite(valvulaR,LOW);
digitalWrite(valvulaDe,LOW);//desague
digitalWrite(MotorR,LOW);
digitalWrite(MotorIn,LOW);
Serial.print("Sistema.p0.pic=4");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p1.pic=4");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p3.pic=4");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p5.pic=9");
Serial.print("\xFF\xFF\xFF");
Serial.print("Sistema.p4.pic=5");
Serial.print("\xFF\xFF\xFF");
procesa_rx();
delay(100);
break;
}
}

```

Código de programación Arduino Esclavo SPWM (NANO)

```

int i=0;
int x=0;
int OK=0;
int
sinPWM60[]={1,3,6,9,12,15,18,21,24,26,29,32,35,38,41,44,47,50,53,56,59,62,65,68,71,73,76,79,82,85,8
8,91,93,96,99,102,104,107,110,112,115,118,120,123,
126,128,131,133,136,138,141,143,146,148,151,153,155,158,160,162,165,167,169,171,173,176,178,180,
182,184,186,188,190,192,194,196,197,199,201,203,205,206,208,
210,211,213,214,216,217,219,220,222,223,224,226,227,228,229,230,232,233,234,235,236,237,238,239,
239,240,241,242,243,243,244,244,245,246,246,247,247,247,248,248,
248,249,249,249,249,249,249,249,249,249,249,249,249,248,248,248,247,247,247,246,246,245,244,244,
243,243,242,241,240,239,239,238,237,236,235,234,233,232,230,229,
228,27,226,224,223,222,220,219,217,216,214,213,211,210,208,206,205,203,201,199,197,196,194,192,1
90,188,186,184,182,180,178,176,173,171,169,167,165,162,160,158,155,
153,151,148,146,143,141,138,136,133,131,128,126,123,120,118,115,112,110,107,104,102,99,96,93,91,
88,85,82,79,76,73,71,68,65,62,59,56,53,50,47,44,41,38,35,32,29,26,24,21,18,15,12,9,6,3,1};
int
sinPWM30[]={0,2,3,5,6,8,9,11,12,14,15,17,18,20,21,22,24,25,27,28,30,31,33,34,36,37,39,40,42,43,45,46
,48,49,51,52,54,55,57,58,59,61,62,64,65,67,68,70,71,73,74,75,77,78,80,
81,83,84,85,87,88,90,91,92,94,95,97,98,99,101,102,103,105,106,108,109,110,112,113,114,116,117,118,
120,121,122,123,125,126,127,129,130,131,133,134,135,136,138,139,140,141,143,144,
145,146,147,149,150,151,152,153,155,156,157,158,159,160,162,163,164,165,166,167,168,169,171,172,
173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,
194,195,196,197,198,199,200,201,201,202,203,204,205,206,207,208,208,209,210,211,212,212,213,214,
215,215,216,217,218,218,219,220,221,221,222,223,223,224,225,225,226,227,227,228,229,229,
230,230,231,231,232,233,233,234,234,235,235,236,236,237,237,238,238,239,239,239,240,240,241,241,
242,242,242,243,243,243,244,244,244,245,245,245,246,246,246,246,247,247,247,247,248,248,248,
248,248,248,249,249,249,249,249,249,249,249,250,250,250,250,250,250,250,250,250,250,250,250,
250,250,250,250,250,250,249,249,249,249,249,249,249,249,248,248,248,248,248,248,247,247,247,247,

```

```

246,246,246,246,245,245,245,244,244,244,243,243,243,242,242,242,241,241,240,240,239,239,239,238,
238,237,237,236,236,235,235,234,234,233,233,232,231,231,230,230,229,229,228,227,227,226,225,
225,224,223,223,222,221,221,220,219,218,218,217,216,215,215,214,213,212,212,211,210,209,208,208,
207,206,205,204,203,202,201,201,200,199,198,197,196,195,194,193,192,191,190,189,188,187,186,185,
184,183,182,181,180,179,178,177,176,175,174,173,172,171,169,168,167,166,165,164,163,162,160,159,
158,157,156,155,153,152,151,150,149,147,146,145,144,143,141,140,139,138,136,135,134,133,131,130,
129,127,126,125,123,122,121,120,118,117,116,114,113,112,110,109,108,106,105,103,102,101,99,98,97
,95,94,92,91,90,88,87,85,84,83,81,80,78,77,75,74,73,71,70,68,67,65,64,62,61,59,58,57,55,54,52,51,
49,48,46,45,43,42,40,39,37,36,34,33,31,30,28,27,25,24,22,21,20,18,17,15,14,12,11,9,8,6,5,3,2,0};
void setup() {
Serial.begin(9600);
pinMode(10, INPUT);
pinMode(9, INPUT);
pinMode(5, OUTPUT);
pinMode(6,OUTPUT);
if (digitalRead(10)==1) {
cli();// stop interrupts
TCCR0A=0;//reset the value
TCCR0B=0;//reset the value
TCNT0=0;//reset the value
//0b allow me to write bits in binary
TCCR0A=0b10100001;//phase correct pwm mode
TCCR0B=0b00000001; //no prescaler
TCCR1A=0;//reset the value
TCCR1B=0;//reset the value
TCNT1=0;//reset the value
OCR1A=509;// compare match value
TCCR1B=0b00001001; //WGM12 bit is 1 and no prescaler

TIMSK1 |=(1 << OCIE1A);

sei();// enable interrupts
}
}
ISR(TIMER1_COMPA_vect){// interrupt when timer 1 match with OCR1A value
if(i>260 && OK==0){// final value from vector for pin 6
i=0;// go to first value of vector
OK=1;//enable pin 5
}
if(i>260 && OK==1){// final value from vector for pin 5
i=0;//go to firs value of vector
OK=0;//enable pin 6
}
if (digitalRead(9)==1) {
x=sinPWM30[i];
}
else {
x=sinPWM60[i];
}
i=i+1;// go to the next position
if(OK==0){

```

```

OCR0B=0;//make pin 5 0
OCR0A=x;//enable pin 6 to corresponding duty cycle
}
if(OK==1){
OCR0A=0;//make pin 6 0
OCR0B=x;//enable pin 5 to corresponding duty cycle
}
}
void loop() {

}

```

Código de programación Matlab para la generación de los vectores SPWM a diferentes frecuencias

```

clc
clear all
x=0;
y=0;
z=0;
v=0;
i=1;
s="";
frecuencia_pwm=31372;%%Esta frecuencia se debe a la fase
correcta del PWM
T_pwm= 1/frecuencia_pwm;
frecuencia_fase= 45
T_fase= 1/frecuencia_fase;
n_pulsos= round((T_fase/2)/T_pwm)
pasos= 180/ n_pulsos
val= zeros([1 158]);
while x<90
    v = x*pi/180;
    y = sin(v);
    z = round (y*250);
    x = x+ pasos;
    val(i)=z;
    i=i+1;
end
nuevo = val(1,1:i-2);
B = sort(nuevo,'descend');
semiciclo = [nuevo,B];
str = string(semiciclo)
arui="";
for j=1:1:length (semiciclo)
    arui= arui + "," + str(j);
end

```

```

ardui
nega= -1*semiciclo;
onda= [semiciclo,nega];
plot(semiciclo)
hold on

```

Código de programación Arduino Esclavo Flujo (Micro)

```

const int sensorPin = 2;
const int measureInterval = 2500;
float frequency = 0;
float flow_Lmin = 0;
volatile int pulseConter;
// FS400A
const float factorK = 3.5;

void ISRCountPulse()
{
  pulseConter++;
}

float GetFrequency()
{
  pulseConter = 0;
  interrupts();
  delay(measureInterval);
  noInterrupts();
  return (float)pulseConter * 1000 / measureInterval;
}

void setup() {
  // put your setup code here, to run once:
  attachInterrupt(digitalPinToInterrupt(sensorPin), ISRCountPulse, RISING);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  // obtener frecuencia en Hz
  frequency = GetFrequency();
  // calcular caudal L/min
  flow_Lmin = frequency / factorK;
  char dato= Serial.read();//Guardamos en la variable dato el valor leído
  switch(dato){ //Comprobamos el dato
    case '!': //Si recibimos una '!' ...
      Serial.write("%d/n", flow_Lmin);
      break;
  }
}

```

Programación en Simulink Matlab para un control continuo en el variador de frecuencia de la bomba de recirculación

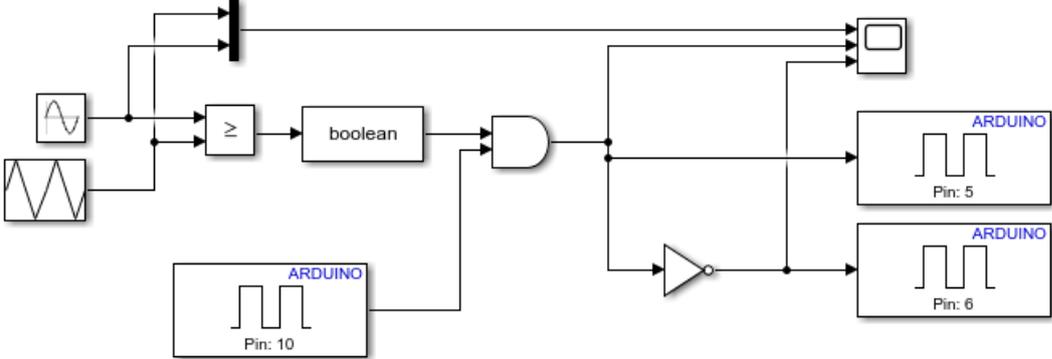


Ilustración 42. Control de Arduino mediante la plataforma Simulink