



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

**TRABAJO DE TITULACIÓN EN OPCIÓN AL TÍTULO DE:
INGENIERO EN ELECTRÓNICA DIGITAL Y
TELECOMUNICACIONES**

TEMA:

**SISTEMA AUTOMÁTICO PARA EL CONTROL HIDRÁULICO E
ILUMINACIÓN DE LA CASCADA DECORATIVA DEL SANTUARIO
CORAZON DE JESÚS MEDIANTE UNA APLICACIÓN WEB**

ELABORADO POR EL AUTOR:

FABIO RAMIRO ENRÍQUEZ VILLARREAL.

TUTOR:

ING. RENE CORTIJO LEYVA, MG.

QUITO, ECUADOR

2019

DECLARACIÓN

Yo, Fabio Ramiro Enríquez Villarreal declaro bajo juramento que el presente proyecto aquí descrito es de mi Autoría, no ha sido previamente presentado para ningún grado o calificación profesional; y, que se a consultado referencias bibliográficas las cuales incluyen en este documento.

A través de la presente declaración, cedo los derechos de propiedad intelectual correspondiente a este trabajo a la Universidad Tecnológica Israel, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Fabio Ramiro Enríquez Villarreal

CERTIFICACIÓN DEL TUTOR

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “**SISTEMA AUTOMÁTICO PARA EL CONTROL HIDRÁULICO E ILUMINACIÓN DE LA CASCADA DECORATIVA DEL SANTUARIO CORAZÓN DE JESÚS MEDIANTE UNA APLICACIÓN WEB.**”, presentado por la Sr. Fabio Ramiro Enríquez Villarreal estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. Febrero del 2019

TUTOR

.....

Ing. Rene Cortijo Leyva, Mg

AGRADECIMIENTO

Al Ing. Rene Cortijo Leyva, Mg y en general a todos los profesores que a lo largo de mi vida estudiantil han sabido compartir sus conocimientos y consejos, sembrando cualidades que me permitan ser mejor profesional, para el bien de la sociedad.

DEDICATORIA

A Dios por la vida y por brindarme la oportunidad de terminar mi carrera, a mis padres que dejaron su bendición para siempre, a mis hermanos por su apoyo incondicional en todo momento para cumplir con mi objetivo.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO	1
PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN.....	1
OBJETIVO GENERAL	2
OBJETIVO ESPECÍFICO.....	2
ALCANCE	3
DESCRIPCIÓN DE LOS CAPÍTULOS	3
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Antecedentes.....	4
1.2 Fuentes de agua	4
1.3 Tipo de fuentes de agua.....	5
1.3.1 Fuentes de agua en interior.....	5
1.3.2 Fuentes de agua en exterior	7
1.4 Luminotecnia.....	7
1.4.1 Iluminación LED	8
1.4.2 Tipos de LED	9
1.5 Sistemas hidráulicos	10
1.5.1 Bomba hidráulica.....	10
1.5.2 Clasificación de las bombas	11
1.5.2.1 Bombas de desplazamiento positivo	11
1.5.2.2 Bombas de engranes	11
1.5.2.3 Bombas sumergibles.....	12
1.5.2.4 Bombas centrifugas	12
1.6 Control automático	13
1.6.1 Controlador.....	13
1.6.2 Actuador	13
1.7 Raspberry Pi3 B.....	14
1.8 Relé	15
1.9 Amplificadores operacionales	15
1.9.1 Filtro paso banda	16

1.9.2 Amplificador inversor/no inversor	16
1.9.3 Amplificador seguidor de tensión	17
1.9.4 Amplificadores integradores/diferenciadores.....	18
1.9.5 Transistor Darlington.....	19
1.10 Tipos de transistores	19
1.11 Aplicación web.....	20
CAPÍTULO II. PROPUESTA.....	21
2.1 Página web de usuario	22
2.2 Página web administrador	26
2.3 Desarrollo de hardware.....	28
Desarrollo del montaje del sistema.....	28
2.4. Raspberry Pi versión 3B	28
2.5. Etapa de filtrado y potencia para efecto audio rítmico.....	32
2.6. Etapa de aislamiento óptico y potencia para secuencia de LEDs.....	34
2.7 Iluminación RGB.....	34
2.8 Bomba.....	35
2.9 Etapa de censado de nivel y flujo de agua.....	36
2.10 Presupuesto y materiales	38
CAPÍTULO III. IMPLEMENTACIÓN	40
3.1. Desarrollo	40
3.2 Implementación	40
Circuito de potencia.....	40
Circuito de potencia.....	41
Fabricación de los circuitos impresos.....	43
3.3 Pruebas de funcionamiento y análisis de resultados.....	48
Anexo	55

LISTA DE FIGURAS

Figura 1 Fuente de mesa	5
Figura 2 Fuente ubicada en el piso	6
Figura 3 Fuente empotrada en la pared.....	6
Figura 4 Fuente cascada.....	7
Figura 5 Diodo LED	8
Figura 6 Tiras LED	9
Figura 7 Bombas de agua.....	10
Figura 8 Bombas de engranaje.....	11
Figura 9 Bomba sumergible.....	12
Figura 10 Bomba centrífuga	12
Figura 11 Placa de Raspberry Pi.....	14
Figura 12 Relé.....	15
Figura 13 Amplificador operacional.....	15
Figura 14 Filtro de paso banda real	16
Figura 15 Amplificador inversor/no inversor	17
Figura 16 Amplificador seguidor de tensión	17
Figura 17 Amplificador de suma y resta.....	18
Figura 18 Amplificador integrador/diferenciador.....	19
Figura 19 Simbología de los transistores.....	20
Figura 20 Diagrama general del sistema.....	21
Figura 21 Sublime Text editor de códigos.....	22
Figura 22 Archivos que componen la aplicación web.....	23
Figura 23 Archivo de HTML.....	23
Figura 24 Extensión de carpeta php.....	24
Figura 25 Conexión PDO	25
Figura 26 Librería de diseño y desarrollo	26
Figura 27 Archivo index	27
Figura 28 Carpeta PHP	27
Figura 29 Carpeta estilo.....	28
Figura 30 Archivo flujo.py	29
Figura 31 Archivo nivel.py.....	30
Figura 32 Archivo LEDs.py.....	31

Figura 33 Preamplificador de audio.....	32
Figura 34 Circuito filtro pasa banda angosta	33
Figura 35 Circuito de potencia para tira LED.....	33
Figura 36 Circuito de aislamiento y potencia para encendido de LED rojo.....	34
Figura 37 RGB.....	35
Figura 38 Bomba de agua	35
Figura 39 Sensor nivel de agua.....	36
Figura 40 Sensor nivel de agua.....	37
Figura 41 Sensor de flujo.....	37
Figura 42 Conexión del sensor de flujo	38
Figura 43 Vista 3D de la placa de filtros activos para luces audio rítmicas	41
Figura 44 Diseño del circuito impreso filtros activos.....	41
Figura 45 Vista 3D de la placa del circuito de potencia	42
Figura 46 Diseño del circuito impreso.....	42
Figura 47 Planchado de baquelita.....	43
Figura 48 Perforado de baquelita.....	43
Figura 49 Soldadura de baquelita	44
Figura 50 Circuito impreso ensamblado de los filtros	44
Figura 51 Montaje de potencia	45
Figura 52 Circuito de aislamiento y potencia	45
Figura 53 Tablero de control completo.....	46
Figura 54 Caja de control.....	46
Figura 55 proyecto Final.....	47

LISTA DE TABLAS

Tabla 1 Tabla de materiales y precios	38
Tabla 2 Lista de verificación de sistema de control.....	48
Tabla 3 Lista de verificación de sistema de control.....	48
Tabla 4 Datos ingreso a página web	49
Tabla 5 Datos de verificación de botones de página web.....	49
Tabla 6 Datos de verificación de botones de página web.....	50

RESUMEN

En la actualidad, la construcción y diseño de cascadas luminosas se han incrementado lo cual se evidencia en centros públicos, parques, conjuntos residenciales e iglesias; en general en lugares donde se desea promover e incrementar la afluencia de feligreses y turistas.

Este proyecto contiene el diseño e implementación del sistema automático para el control de una cascada decorativa con chorros de agua, iluminación LED RGB, que están controladas manualmente o automáticamente por el ritmo de la música.

Descartando cualquier estereotipo ya existente, se ha realizado un diseño propio, tanto en la parte de control electrónico, en donde se trabaja con una tarjeta Raspberry Pi3 modelo B de tamaño pequeño, características sencillas y de bajo costo. El software de programación está basado en PHP, JAVA, CSS y Paython. Agrupa comandos e interfaces de mejor manera, para el control de las tarjetas de audio, potencia, relés, temporizador, con el fin de ejecutar tres secuencias luminosas; así como, el riego de agua de forma rítmica, a través de una plataforma web.

Palabras Claves: Raspberry, Wifi, LED, RGB, Programación.

ABSTRACT

At present, the construction and design of luminous waterfalls have increased which is evident in public centers, parks, residential complexes and churches; in general in places where it is desired to promote and increase the influx of parishioners and tourists.

This project contains the design and implementation of the automatic system for the control of a decorative waterfall with water jets, LED RGB lighting, which are controlled manually or automatically by the rhythm of the music.

Discarding any existing stereotype, a design of its own has been made, both in the electronic control part, where it works with a Raspberry Pi3 model B of small size, simple features and low cost. The programming software is based on PHP, JAVA, CSS and Paython. Groups commands and interfaces in a better way, for the control of audio cards, power, relays, timer, in order to execute three light sequences; as well as, the irrigation of water rhythmically, through a web platform.

Keywords: Raspberry, Wifi, LED, RGB, Programming.

INTRODUCCIÓN

La sistematización de procesos hoy en día se ha convertido en la revolución comercial y existen varias técnicas de control automático de acuerdo a la evolución de la tecnología y necesidades que se presentan.

Las cascadas danzantes son un espectáculo a nivel mundial ofrecen movimientos de agua y luz al ritmo de la música.

ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO

En 1973, gracias a la labor de los padres Salesianos fue construido el Santuario Corazón de Jesús, ubicado en la parroquia Carcelén en la ciudad de Quito. La construcción del santuario tuvo en cuenta aspectos históricos y religiosos, considerando la devoción de los habitantes de la parroquia y sus visitantes.

Como mejoras en el lugar, el párroco del Santuario Corazón de Jesús con la ayuda de la comunidad tuvo la iniciativa de la construcción civil de una cascada, que impulse la evangelización educativa y el trabajo en equipo, además de incrementar feligreses y turistas.

PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

La cascada que se encuentra en construcción en el Santuario Corazón de Jesús, necesita la instalación y puesta en marcha un sistema de control automático para el agua y las luces, por ende, una vez terminado el trabajo de la cascada se implementará un sistema de control automatizado que permite coordinar el juego de luces y agua al ritmo de la música.

Dentro del control audio-rítmico se desarrollará la revisión del sistema mediante cualquier dispositivo sin la necesidad de estar presente.

Para que ocurra este juego de agua y luces de manera armónica es necesario realizar una programación de datos que serán enviados a una tarjeta conocida como Raspberry Pi3. Adicional la interfaz de la página web desarrollada será amigable con el usuario.

Teniendo en cuenta la necesidad de la parroquia, se propone realizar el proyecto:
SISTEMA AUTOMÁTICO PARA EL CONTROL HIDRÁULICO E ILUMINACIÓN DE LA CASCADA DECORATIVA DEL SANTUARIO CORAZÓN DE JESÚS MEDIANTE UNA APLICACIÓN WEB, dicho proyecto beneficiará al santuario y a la gente de la parroquia de Carcelén; ya que, servirá de un atractivo para fieles que asisten al Santuario, como para sus pobladores.

OBJETIVO GENERAL

Automatizar el sistema para el control hidráulico e iluminación de la cascada decorativa del santuario corazón de Jesús mediante una aplicación web.

OBJETIVO ESPECÍFICO

- ✓ Definir los dispositivos y elementos que se van a utilizar en sistema para el control hidráulico e iluminación de la cascada decorativa del santuario corazón de Jesús.
- ✓ Implementar un sistema hidráulico e iluminación en la cascada del Santuario Corazón de Jesús mediante una RASPBERRY PI3 el cual integra comunicación BLUETOOTH y WIFI.
- ✓ Monitorear desde una aplicación web el funcionamiento de las secuencias de luces, activación encendido apagado, flujo y nivel de la bomba sumergible, válvula de llenado electrónica con la ayuda de los datos generados por el sistema de control.
- ✓ Programar el sistema automático mediante un lenguaje de programación que permita el control y manejo del sistema
- ✓ Verificar el correcto funcionamiento del sistema del control hidráulico e iluminación.

ALCANCE

En el presente proyecto se pretende identificar los elementos necesarios para la elaboración del sistema de control que dará lugar a la automatización del sistema del control hidráulico e iluminación de la cascada decorativa del santuario corazón de Jesús mediante una aplicación web y así mejorar el aspecto de la parroquia e incentivar la fe católica a los feligreses.

DESCRIPCIÓN DE LOS CAPÍTULOS

El presente trabajo consta de los siguientes capítulos:

En el capítulo uno, presenta el fundamento teórico en la que se encuentra la información y diversos enfoques de la tecnología que se utiliza.

En el capítulo dos, especifica el diseño realizado, hardware y software especializado y su funcionamiento mediante diagramas de bloques y flujo.

En el capítulo tres, describe el proceso de construcción del hardware y software, describe la puesta a punto del proyecto desarrollado en la que involucra las pruebas de funcionamiento y análisis de los resultados obtenidos.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.1 Antecedentes

En la ciudad de Quito, específicamente en la parroquia de Carcelén, se encuentra ubicado el Santuario sagrado corazón de Jesús en el cual se realizó la construcción de una cascada por parte de los habitantes de la parroquia y como mejora a ese proyecto se decidió realizar un sistema automático para control hidráulico e iluminación que dará el realce respectivo al santuario.

Las cascadas de agua han sido un medio decorativo para plazas, iglesias, parques y lugares históricos. Hace muchos años las cascadas de agua eran colocadas para satisfacer la sed de los habitantes; así como también, para dar agua a los animales en tiempos calurosos. Actualmente, las cascadas son ocupadas como elementos decorativos, atractivos turísticos o como medios de diversión y entretenimiento.

En Ecuador estos atractivos son escasos, hay pocas ciudades en donde se encuentran este tipo de fuentes danzantes por ejemplo en la ciudad de Guayaquil se encuentra en el Malecón del Salado.

En nuestra ciudad este tipo de trabajo además de dar realce al lugar, incentiva el trabajo a nuevos proyectos enfocados a estudiantes y profesionales del área.

1.2 Fuentes de agua

El elemento principal es el agua, componente líquido que se puede usar sin ningún inconveniente, en ocasiones manipularlo en diferentes direcciones a manera decorativa. Una fuente de agua se encuentra en diferentes lugares, sirve de atracción para jardines, edificios, casas, iglesias, malecones, parques, entre otros

Una fuente de agua tiene un diseño que en la mayoría de los casos necesitan instalaciones eléctricas para que funcione, hay otras que son autónomas; las fuentes tienen un depósito de agua con una bomba enchufada a la red y el agua recircula; todo este proceso combinado con música y luz atrae a los habitantes y turistas aportando vida, armonía, emoción, paz y tranquilidad.

Debido a lo anteriormente mencionado, existen diversos tipos de fuentes de agua que poseen diversas características, así se pueden encontrar fuentes en áreas internas de menor tamaño y externas con gran volumen de agua.

1.3 Tipo de fuentes de agua

1.3.1 Fuentes de agua en interior.

Entre las fuentes de agua que se pueden utilizar en el interior, se encuentran las siguientes:

- ✓ **Fuentes de mesa.** Por su estructura utilizan volúmenes de agua pequeños y poseen bombas pequeñas, que ayudan a recircular el agua. (Figura 1)



Figura 1 Fuente de mesa

Fuente: arquitectura, 2019

- ✓ **Fuentes ubicadas en el piso.** Con respecto a las fuentes de mesa, se puede decir, que son de mayor tamaño, decorativas y ayudan a dividir espacios en interiores.(Figura 2)



Figura 2 Fuente ubicada en el piso

Fuente: Espacio hogar, 2015

- ✓ **Fuentes empotradas en la pared.** Por su estructura son difíciles de instalar ya que necesitan un sistema de tuberías especiales en la pared. (Figura 3)



Figura 3 Fuente empotrada en la pared

Fuente: Casa & Diseño, 2017

1.3.2 Fuentes de agua en exterior

Entre las fuentes de agua que se pueden utilizar en exteriores, se tiene las siguientes:

- ✓ **Cascada.** El diseño depende del espacio y la altura en la que el agua caiga este tipo de fuentes son muy atractivas y se las puede complementar con rocas de colores, estatuas, animales, puentes, entre otros. (Figura 4)

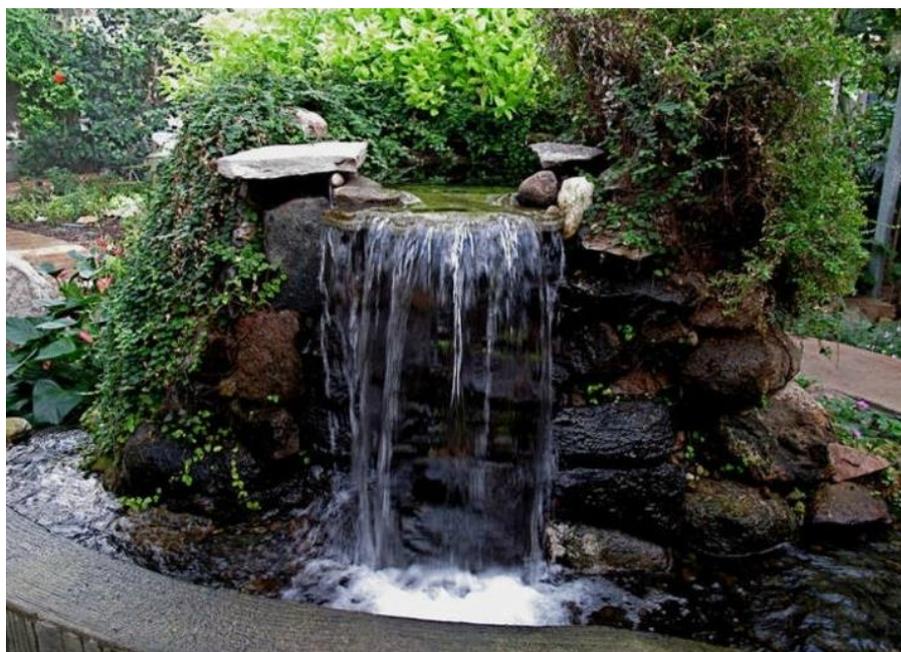


Figura 4 Fuente cascada

Fuente: García, 2018

1.4 Luminotecnia

Es la ciencia que estudia las diversas formas de producción de la luz, su control y aplicación. Actualmente se considera como el arte de la iluminación, que con la utilización de luz artificial, permite el desarrollo y diseño de diferentes proyectos arquitectónicos, que logran un gran impacto visual.

Gracias a esta técnica se puede hacer diversos diseños de iluminación, puesto que, designa los parámetros básicos y las principales magnitudes fotométricas, que guían al

iluminador a realiza un trabajo que logra cumplir con los objetivos de quien lo solicita, con la eficacia e intensidad luminosa requerida.

En el mercado se exhiben diferentes tipos de iluminación que se pueden utilizar en las fuentes de agua; existen luces halógenas, lámparas dicroicas, luces LEDs, entre otras. Las que se utilizan en este trabajo son tipo LEDs. A continuación una breve explicación de este tipo de iluminación:

1.4.1 Iluminación LED

“L.E.D, Light Emitting Diode o diodo emisor de luz, es un dispositivo semiconductor que transforma directamente la energía eléctrica en luz mediante electroluminiscencia. La robustez, una larga duración y una elevada eficacia luminosa con un potencial de incremento adicional son propiedades destacadas de los ledes” (LAGO, 2015, p. 9) (Figura 5)

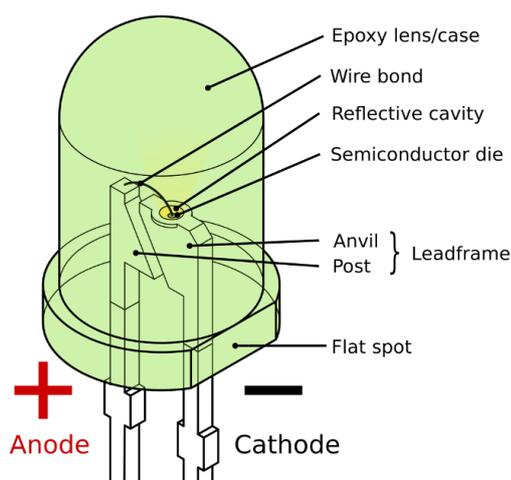


Figura 5 Diodo LED

Fuente: Cenaptec, 2019

Los diodos LED además de dar paso a la corriente en un solo sentido pueden emitir luz, cuando se conecta un diodo en el paso de la corriente esta polarizado directamente.

Los diodos LED en la actualidad tiene muchas aplicaciones y se extienden a casi todos los campos lumínicos, los sistemas que utilizan este tipo de iluminación tienen una gran ventaja con respecto a la iluminación tradicional debido a la durabilidad y poco consumo de voltaje, en su estructura no poseen gases o metales, por lo que,

contamina menos que el resto. Adicional a las ventajas mencionadas se destacan otras como, por ejemplo:

- ✓ Alta eficiencia
- ✓ Reducción de costos de mantenimiento
- ✓ Flexibilidad de instalación por el tamaño
- ✓ Sostenible para el ambiente debido a que la emisión de calor es reducida
- ✓ No contienen mercurio
- ✓ Encendido instantáneo al cien por ciento de su rendimiento
- ✓ Son insensible a vibraciones
- ✓ Ausencia de radiaciones
- ✓ Colores vivos y saturados in filtro

1.4.2 Tipos de LED

Hoy en día se conoce varios tipos de LED los más usados son SMD y tiras LED. A continuación se explica el tipo de LED a usarse en el presente proyecto. Una de las nuevas versiones de LED utilizadas son los LED flexibles. Estas tiras de LED flexibles están disponibles en muchas configuraciones, ya que pueden ser muy brillantes, para iluminar una habitación entera o pueden ser utilizadas cuando se requiere menos luz, en el caso, de que se necesite una iluminación más tenue.

Las tiras de LED flexibles también se pueden fabricar en diferentes longitudes y hay varios colores dependiendo de uso.

Algunas tiras LED también pueden cambiar colores en un amplio rango usando un controlador RGB. (Figura 6)



Figura 6 Tiras LED

Fuente: MundoLED, 2018

1.5 Sistemas hidráulicos

La hidráulica forma parte de la ciencia física, regula y comprende los movimientos y la fuerza de los líquidos; es decir es la transformación de la energía en forma mecánica o eléctrica en hidráulica. (Figura 7)

1.5.1 Bomba hidráulica



Figura 7 Bombas de agua

Fuente: Venebombas, 2015

Es un dispositivo que toma la energía mecánica y la convierte en energía hidráulica, permite incrementar la energía cinética de un caudal de agua.

La bomba se puede controlar mediante un procesador, en el caso del proyecto se utilizará un Raspberry PI3 mediante el cual se puede encender cualquier tipo de bomba de agua mediante salidas digitales o una salida por relé.

Al realizar un proyecto para la reestructuración de una fuente es necesario determinar si posible la ejecución antes de proceder a su arreglo por ello es importante conocer en qué estado se encuentra el sistema hidráulico.

1.5.2 Clasificación de las bombas

1.5.2.1 Bombas de desplazamiento positivo

El principio de operación se basa en hidrostática, de modo que el aumento de la presión es por el empuje de las paredes de las cámaras que varían su volumen. En este tipo de bombas, en cada ciclo genera positivamente un volumen dado o desplazamiento, también son llamados bombas volumétricas. En el caso de variar el volumen máximo del desplazamiento, se habla de bombas de volumen variable. Si este volumen no puede ser variado, entonces se dice que la bomba es de volumen fijo.

1.5.2.2 Bombas de engranes

Estas bombas se utilizan para aplicaciones de energía fluida y para distribuir lubricantes para componentes específicos de maquinaria. Consiste en dos engranajes que giran dentro de un alojamiento, en la dirección opuesta y una muy apretada con la otra. La periferia externa de los dientes del engranaje encaja muy bien con la superficie interna de la caja. (Figura 8) El fluido se transporta desde el almacén de suministro a la puerta de succión y se conduce en los espacios entre los dientes hasta la puerta de descarga, enviado a alta presión para el sistema. La presión con la que se envía depende de la resistencia del sistema. (Mott, 1996).

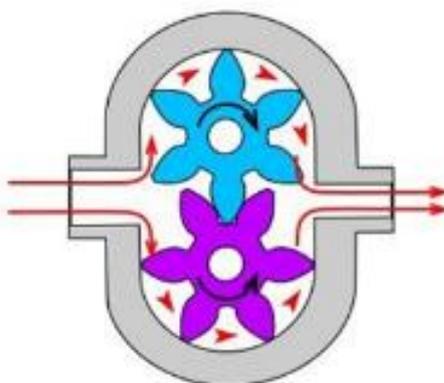


Figura 8 Bombas de engranaje

Fuente: Mott, 1996

1.5.2.3 Bombas sumergibles

Se utilizan para cascadas, estanques, jardines y reservas de agua. Debido a su pequeño tamaño son consideradas las más útiles en estanques.

Este tipo de bomba se sumerge en líquidos, como se aprecia en la (Figura 9), contienen un impulsor sellado que ayuda a bombear el líquido en el cual se encuentra sumergido y lo lleva hacia afuera.



Figura 9 Bomba sumergible

Fuente: Grainger, 2000

1.5.2.4 Bombas centrífugas

Son específicas en el campo industrial y agrícola, puede manejar grandes caudales por ello se puede utilizar en edificaciones o parques centrales. (Figura 10)

Este tipo de bombas puede trabajar con caudales pequeños o grandes.



Figura 10 Bomba centrífuga

Fuente: Inoxmim, 2018

1.6 Control automático

El control automático se basa en una serie de componentes conectados entre sí, actúan sin intervención del ser humano y ha tenido un crecimiento a gran velocidad en el ambiente industrial. Ayuda a controlar diversas variables como iluminación, temperatura, humedad, entre otros.

Un control automático ideal debe ser capaz de cumplir su objetivo cumpliendo los siguientes requisitos:

- ✓ Garantizar la estabilidad del proceso
- ✓ Ser tan eficiente como sea posible, según un criterio preestablecido.
- ✓ Ser fácilmente implementado y fácil de operar en tiempo real con ayuda de un computador

Los elementos básicos que forman parte de un control automático y permite su funcionamiento son los siguientes:

1.6.1 Controlador

Tarjetas electrónicas que poseen información de accesos permitidos, historial de entrada. Los controladores manejan diferentes variables a la misma vez, por ejemplo: de lectura, luces, sirenas, sensores, entre otros.

1.6.2 Actuador

Los actuadores son dispositivos utilizados para el control del sistema puede modificar el estado de los equipos o instalaciones.

Transforma diferentes tipos de energía para actuar sobre otro dispositivo y hacerlo mover, dependiendo de la fuerza el actuador se conoce como: neumático, hidráulico o eléctrico.

EL actuador electrónico solo requiere energía como fuente de poder, facilitan la vida del hombre debido a sus características son elementos fáciles de controlar.

1.7 Raspberry Pi3 B

Es un controlador de placa única de bajo costo, fue desarrollado en Reino Unido por la Fundación Raspberry Pi3 B, con el objetivo de incrementar los avances tecnológicos en el área de computación. (Figura 11)



Figura 11 Placa de Raspberry Pi

Fuente: Elaborador por el Autor

“Raspberry Pi ha demostrado ser un gadget de lo más útil, desde proyectos científicos con la robótica de por medio hasta automatizar pequeñas tareas del día a día o incluso ser un traductor, además de un sistema de entretenimiento”. (Contreras, 2019)

Esta placa trabaja con hardware y software de libre distribución, sistema operativo Linux y la licencia es open-source. El Raspberry Pi cuenta con varias funcionalidades y características que lo hacen más conveniente para trabajar, como:

- ✓ Dos puertos USB que son compatibles para conectar sus entradas y salidas en casi cualquier tipo de computadoras.
- ✓ Posee un puerto Ethernet para conexión a internet, sin embargo, puede usarse un adaptador USB para uso en una red inalámbrica.
- ✓ Facilidades de almacenamiento externo ya que posee puertos para tarjetas SD, SDXC almacenamiento de hasta 2 TB.
- ✓ Consumo de energía reducida a 3.5 W

1.8 Relé

Es un dispositivo electromecánico que trabaja como actuador, posee una bobina que al ser polarizado genera un campo electromagnético, que trabaja como un imán y tiene la función de interruptor, además puede ser abierto o cerrado. (Figura 12)

Este componente es importante para el proyecto ya que permite controlar el encendido y apagado del agua, mediante una entrada de datos.



Figura 12 Relé

Fuente: Elaborado por el Autor

1.9 Amplificadores operacionales

Los amplificadores operacionales son unos amplificadores diferenciales que se utilizan en circuitos electrónicos y dependiendo del tipo de amplificador que se use en el circuito se puede realizar diferentes funciones electrónicas, como: suma, resta, integración o diferenciación. (Figura 13)

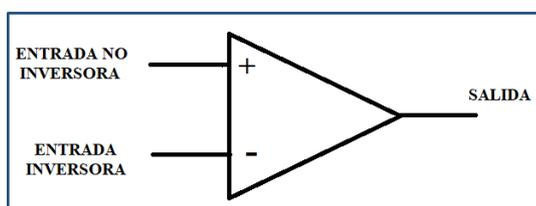


Figura 13 Amplificador operacional

Fuente: Elaborado por el Autor

1.9.1 Filtro paso banda

Son formados por resistencias, bobinas y condensadores, diseñados para dejar pasar solo aquellas frecuencias que están a un determinado valor y atenúan las frecuencias restantes.

En este filtro existe dos frecuencias una de corte inferior (f_1) y otra de corte superior (f_2), solo atenúan las señales donde la frecuencia sea menor que la frecuencia de corte inferior o solo atenúan las señales donde la frecuencia sea mayor que la frecuencia de corte superior. En conclusión, el filtro solo permite el paso de un rango o banda de frecuencia. (Figura 14)

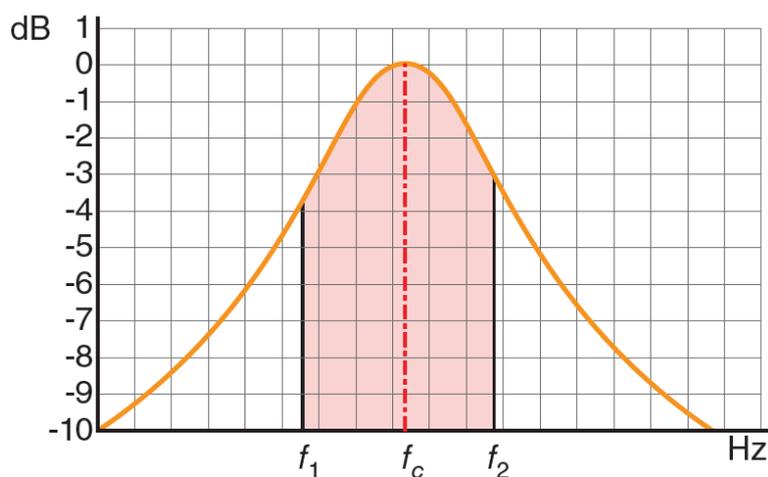


Figura 14 Filtro de paso banda real

Fuente: Rectificadores y filtros, 2013

1.9.2 Amplificador inversor/no inversor

El objetivo del amplificador es aumentar la señal de un circuito determinado. Lo que diferencia a los amplificadores operacionales de los demás es que realizan algunos procesos matemáticos adicionales en la señal mientras lo amplifican. Un amplificador operacional no inversor es esencialmente el tipo "base" que aumenta la ganancia de una señal electrónica sin ningún proceso adicional. Un amplificador operacional inversor aumentará la ganancia de la señal y también revertirá la polaridad de la señal de salida, de positivo a negativo o viceversa. (Figura 15)

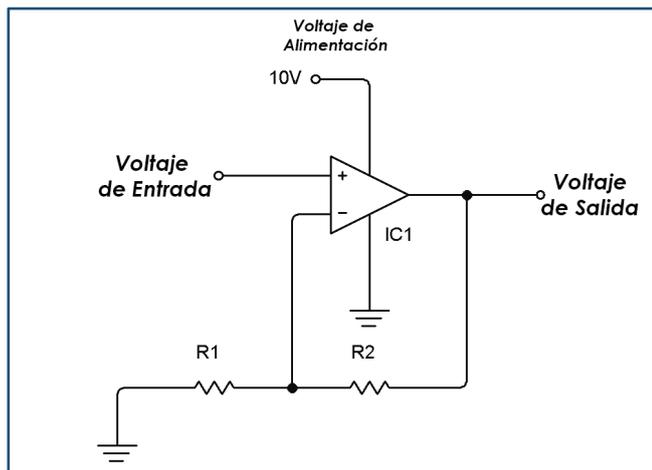


Figura 15 Amplificador inversor/no inversor

Fuente: Boylestad, 2009

1.9.3 Amplificador seguidor de tensión

Un seguidor de tensión se usa para aumentar la señal de circuitos con voltajes variables. Esto se aplica al mismo tipo de ganancia que el amplificador estándar, pero las variaciones en la ganancia de entrada son rastreadas y ajustadas por la salida. Este tipo de circuitos a menudo se usan antes que otros sistemas para evitar daños por cambios bruscos de tensión. (Figura 16)

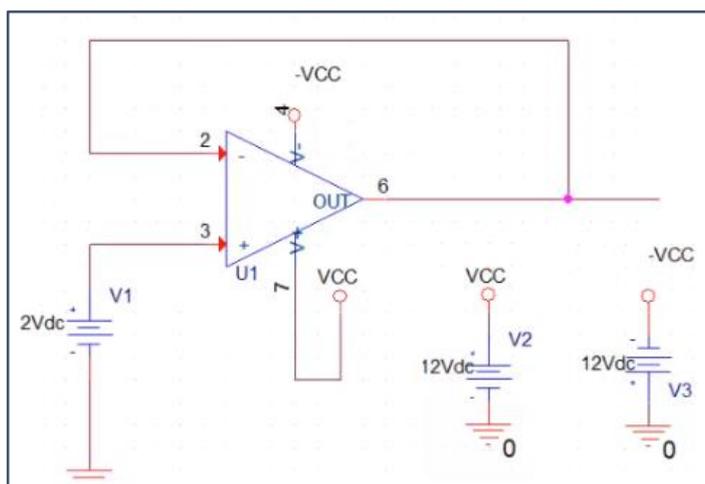


Figura 16 Amplificador seguidor de tensión

Fuente: Elaborado por el Autor

Estas dos variedades de amplificadores operacionales ejecutan un proceso numérico en la señal. Un amplificador de sustracción operativa extrae una señal que es

igual a la resta entre sus dos entradas. Un amplificador de suma combina diferentes tensiones de un número de entradas y hace una ganancia basada en las tensiones combinadas. Cualquiera de estos circuitos se puede configurar para operar como sistemas de inversión o no inversión. (Figura 17)

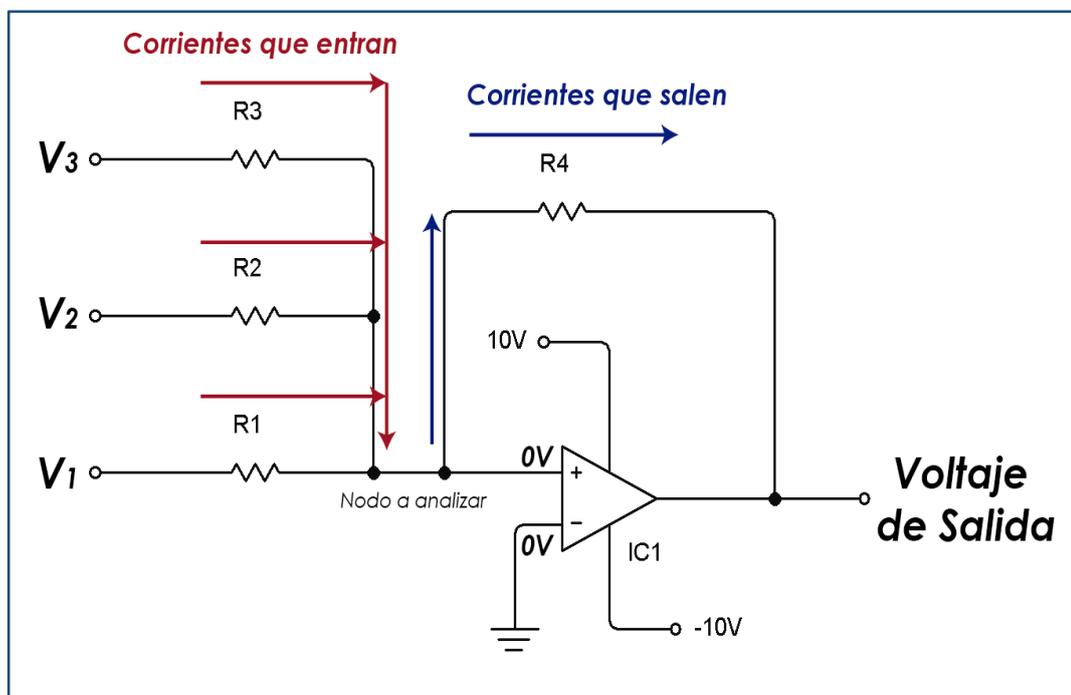


Figura 17 Amplificador de suma y resta

Fuente: Boylestad, 2009

1.9.4 Amplificadores integradores/diferenciadores

Los tipos más complejos de amplificadores operacionales son los integradores y los diferenciadores. La suma de un condensador al circuito significa que el integrador responde a los cambios en el voltaje a lo largo del tiempo. El tamaño de la tensión de salida cambia, según cuánto tiempo pasa una tensión como se muestra en la entrada. El diferencial es lo opuesto a esto. La tensión producida en el canal de salida es proporcional al de la entrada. Cambios más grandes y más rápidos en el voltaje de entrada darán voltajes de salida más altos. (Figura 18)

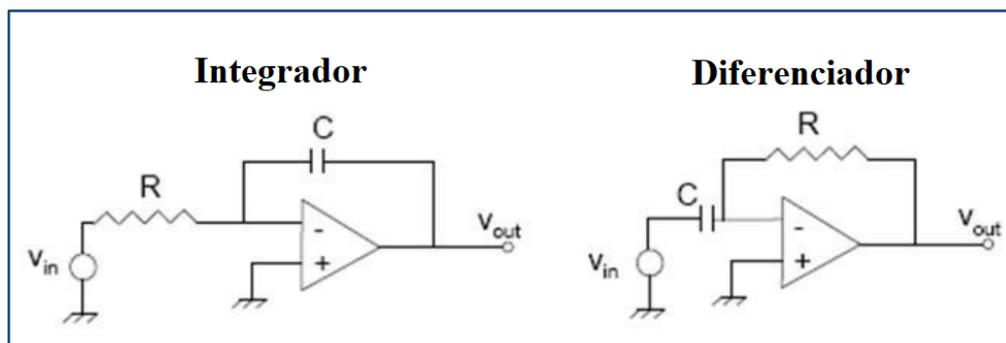


Figura 18 Amplificador integrador/diferenciador

Fuente: Amplificadores operacionales, 2012

1.9.5 Transistor Darlington

El transistor Darlington es un dispositivo semiconductor que combina dos transistores bipolares en un solo dispositivo. Esta configuración sirve para que el dispositivo sea capaz de proporcionar una gran potencia de refuerzo y para poder integrar requiere menos espacio que dos transistores normales en la misma configuración. La ganancia general de Darlington es el producto de la mejora de los transistores individuales. Un dispositivo típico tiene un aumento de potencia de 1000 o superior. También tiene un desplazamiento de fase más alto a altas frecuencias que un solo transistor, por lo que puede volverse inestable fácilmente. La tensión del emisor base también es mayor, ya que la suma de los voltajes base del emisor y de los transistores de silicio es superior a 1.2 V. La intensidad del colector se encuentra al multiplicar la intensidad de la base con la beta total

1.10 Tipos de transistores

Se conoce dos tipos de transistores estándar NPN y PNP, los más comunes son los NPN debido la facilidad para su fabricación a base de silicio. (Figura 19)

En donde, B es base, C es colector y E emisor, estos términos hacen referencia al funcionamiento interno del transistor.

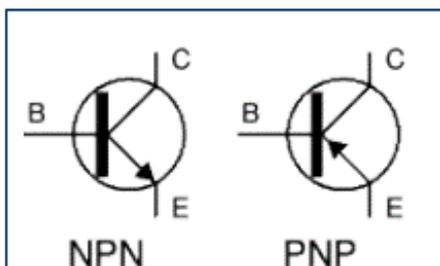


Figura 19 Simbología de los transistores

Fuente: simbología electrónica, 2016

Cuando se tiene un par de transistores ya sea NPN o PNP se lo denomina Darlington estos transistores se conectan juntos en un mismo encapsulado para incrementar la ganancia de corriente.

1.11 Aplicación web

Mediante la aplicación web el usuario podrá controlar las secuencias de luces y la bomba de agua remotamente, cargar de sonidos desde cualquier parte del mundo.

Se utilizarán las herramientas tales como:

- **PHP.** Es una herramienta simple y versátil pero que a la vez tiene cualidades avanzadas para uso de personas capacitadas de acuerdo con las últimas tendencias en el desarrollo web, debido a la velocidad, estabilidad, seguridad y facilidad de codificación.
- **HTML.** Es una herramienta que se va aplicar para dar forma a la página web que se va a crear, ése es el principal objetivo.
- **CSS.** Es una herramienta que se va aplicar para dar formato a la página web, es decir los colores, el diseño y las fuentes.
- **Java script:** Esta herramienta se usará para validar los parámetros cuando estos sean ingresados, modificados o actualizados en la página web.

CAPÍTULO II. PROPUESTA

Para realizar la automatización de la cascada, es necesario realizar el diseño del hardware y software del sistema hidráulico y de iluminación.

En este capítulo se detallan las características de los diferentes dispositivos utilizados para el diseño e implementación del sistema de control, así como también el desarrollo de una página web que permitirá controlar de manera inalámbrica todo el proceso.

A continuación, en la (Figura 20) se presenta un diagrama de bloques que ayudará a describir cada etapa.

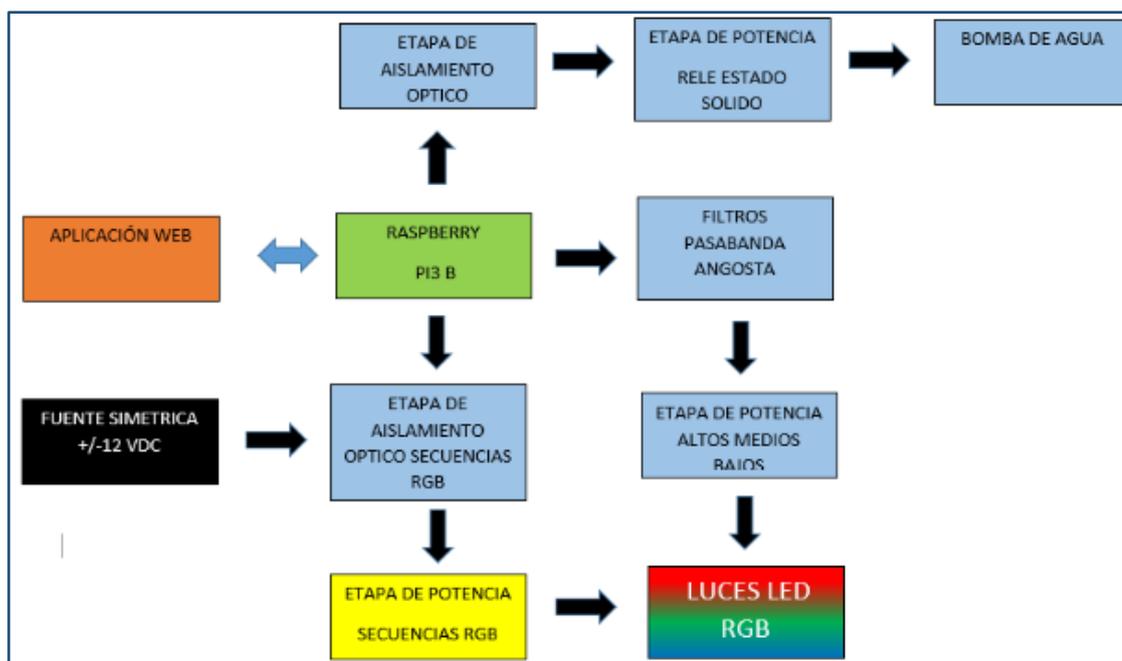


Figura 20 Diagrama general del sistema

Fuente: Elaborador por el Autor

2.1 Página web de usuario

Para el desarrollo de la aplicación web se utilizó el editor **SUBLIME TEXT**, el cual es un editor de código multiplataforma, ésta herramienta permite tener varios documentos y ventanas abiertas diseñada para programar sin distracciones, permite tener varios documentos y ventanas abiertas. Tiene un modo de pantalla completa, para aprovechar al máximo el espacio visual disponible.

El programa cuenta "de serie" con 22 combinaciones posibles de color, Además posee un panel llamado Minimap el cual, permite recorrer por el código rápidamente y soporta un gran número de lenguajes (C, C++, C#, CSS, D, Erlang, HTML, Groovy, Haskell, HTML, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, Matlab, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML).

Mediante Sublime Text se desarrolló los archivos HTML, PHP y JavaScript encargados del control de la página web. (Figura 21)

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Control Raspi</title>
6      <link rel="icon" href=" ../favicon.ico">
7      <link rel="stylesheet" href=" ../css/main.css">
8  </head>
9  <body>
10
11     <header>
12         <figure></figure>
13         <h1>Control Raspi</h1>
14     </header>
15
16     <ul class="leds">
17         <p id="secuencia">Secuencia</p>
18         <li id="led1">&nbsp;</li>
19         <li id="led2">&nbsp;</li>
20         <li id="led3">&nbsp;</li>
21         <li id="led4">&nbsp;</li>
22
23         <p id="texto_secuencia">Secuencia manual</p>
24     </ul>
25
26     <div id="texto" style="padding:0; margin: 0 auto;">
27         <p id="texto_enviado" style="padding:0;"></p>
28     </div>
29
30     <figure id="ver_imagen" style="width: 200px;"></figure>

```

Figura 21 Sublime Text editor de códigos

Fuente: Elaborado por el Autor

En la (Figura 22) se puede visualizar las carpetas y archivos que componen la aplicación web.

Nombre	Fecha de modificación	Tipo	Tamaño
admin	22/1/2019 04:59	Carpeta de archivos	
css	22/1/2019 04:59	Carpeta de archivos	
icomoon	22/1/2019 04:59	Carpeta de archivos	
img	22/1/2019 04:59	Carpeta de archivos	
js	22/1/2019 04:59	Carpeta de archivos	
php	22/1/2019 04:59	Carpeta de archivos	
como_instalar	22/1/2019 04:59	Documento de texto	1 KB
control_raspi.sql	22/1/2019 04:59	Archivo SQL	1 KB
favicon	22/1/2019 04:59	Icono	99 KB
flujo.py	22/1/2019 04:59	Archivo PY	1 KB
index	22/1/2019 04:59	Archivo HTML	6 KB
leds.py	22/1/2019 04:59	Archivo PY	2 KB
login	22/1/2019 04:59	Archivo HTML	1 KB
nivel.py	22/1/2019 04:59	Archivo PY	1 KB
README.md	22/1/2019 04:59	Archivo MD	1 KB

Figura 22 Archivos que componen la aplicación web

Fuente: Elaborado por el Autor

A continuación se detallará como se ha procedido a crear cada carpeta:

El primer archivo que se ejecuta cuando se abre la página web es el login.html, permite al usuario ingresar credenciales, es decir usuario y contraseña.

```

C:\Users\FABIO\Downloads\control_raspi-master\control_raspi-master\login.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

login.php x login.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Control Raspi</title>
6   <link rel="stylesheet" href="css/main.css">
7   <link rel="icon" href="favicon.ico">
8   <link rel="stylesheet" href="css/bootstrap.min.css">
9 </head>
10 <body>
11
12   <header id="login">
13     <figure></figure>
14     <h1>Control Raspi</h1>
15   </header>
16
17   <form id="inicio">
18     <input type="text" id="usuario" placeholder="Su nombre de usuario" required="true">
19     <input type="password" id="pass" placeholder="Su contrase&ntilde;a" required="true">
20     <input type="submit" value="Ingresar">
21   </form>
22
23

```

Figura 23 Archivo de HTML

Fuente: Elaborado por el Autor

Una vez validada el ingreso a la página web, el control del programa pasa al archivo index.html, este es el archivo que se muestra al usuario y permite realizar el control de las secuencias de luces (Figura 23), controla el encendido de LEDs y bomba de agua en modo manual, monitorea los sensores tanto de flujo como nivel de agua, también permite enviar al raspberry una imagen, un texto y un archivo de audio.

Dentro de la carpeta PHP se encuentra los archivos con extensión .php encargados de manejar la base de datos y realizar la conexión con el servidor.

A continuación en la (Figura 24) se puede visualizar la carpeta de la base de datos.

Nombre	Fecha de modificación	Tipo
tmp	22/1/2019 04:59	Carpeta de archivos
actualiza.php	22/1/2019 04:59	Archivo PHP
audio.php	22/1/2019 04:59	Archivo PHP
conexion.php	22/1/2019 04:59	Archivo PHP
config.php	22/1/2019 04:59	Archivo PHP
funciones.php	22/1/2019 04:59	Archivo PHP
leds	22/1/2019 04:59	Documento de texto
login.php	22/1/2019 04:59	Archivo PHP
logueado.php	22/1/2019 04:59	Archivo PHP
meta.php	22/1/2019 04:59	Archivo PHP
mostrar_imagen.php	22/1/2019 04:59	Archivo PHP
obtener_datos.php	22/1/2019 04:59	Archivo PHP
salir.php	22/1/2019 04:59	Archivo PHP

Figura 24 Extensión de carpeta php

Fuente: Elaborado por el Autor

El archivo conexion.php, obtiene las credenciales para la conexión a la BD y crea una nueva conexión PDO o como comúnmente se conoce base de datos. (Figura 25)

```
File Edit Selection Find View Goto Tools Project Preferences Help
conexion.php x
1 <?php
2 // Obtiene las credenciales para la conexión a la bd
3 require_once "config.php";
4
5 class Database{
6
7     /**
8      * Única instancia de la clase
9      */
10    private static $db = null;
11
12    /**
13     * Instancia de PDO
14     */
15    private static $pdo;
16
17    final private function __construct(){
18        try {
19            // Crear nueva conexión PDO
20            self::getDb();
21        } catch (PDOException $e) {
22            // Manejo de excepciones
23            print 'Error!: ' . $e->getMessage(). '<br/>';
24            die();
25        }
26    }
27 }
```

Figura 25 Conexión PDO

Fuente: Elaborado por el Autor

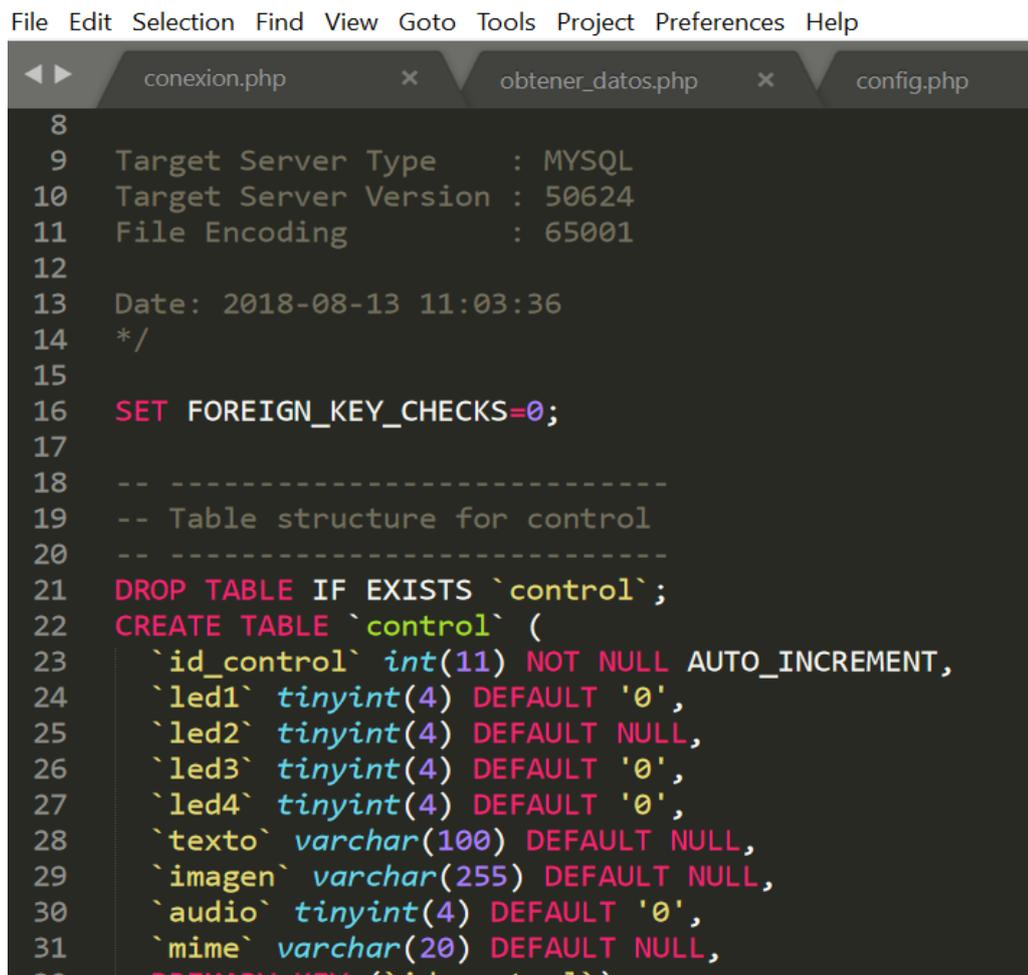
En el archivo actualiza.php se encuentra el control de las secuencias 1, 2, 3, es decir envía la orden al Raspberry para que pueda encender o apagar los puertos GPIO en donde están conectados los LEDs.

Posteriormente se trabajó en el estilo de la página que se visualiza en la carpeta CSS se, el estilo se trata de diseñar el tipo de letra, colores y espacios que va ocupar cada campo dentro de la página web.

Como se indicó anteriormente se trabajó con JavaScript que posee librerías con lenguaje de programación muy usados para el desarrollo web, son de código abierto y permiten tener una página interactiva. Los lenguajes usados en esta tesis fueron framework jQuery y Bootstrap.

jQuery debido a que posee una serie de funcionalidades permite modificar el contenido de la página web como es el texto, imágenes, enlaces, etc. Cambiar el diseño de la carpeta CSS utilizando códigos simples.

Bootstrap este framework permite crear interfaces web con CSS y JavaScript cuya singularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo donde se visualice la página web, es decir se acopla automáticamente al tamaño de un monitor de computadora, celular, entre otros. Esta librería realiza el diseño y desarrollo adaptivo. (Figura 26)



```
File Edit Selection Find View Goto Tools Project Preferences Help
conexion.php x obtener_datos.php x config.php
8
9 Target Server Type : MYSQL
10 Target Server Version : 50624
11 File Encoding : 65001
12
13 Date: 2018-08-13 11:03:36
14 */
15
16 SET FOREIGN_KEY_CHECKS=0;
17
18 -- -----
19 -- Table structure for control
20 -- -----
21 DROP TABLE IF EXISTS `control`;
22 CREATE TABLE `control` (
23   `id_control` int(11) NOT NULL AUTO_INCREMENT,
24   `led1` tinyint(4) DEFAULT '0',
25   `led2` tinyint(4) DEFAULT NULL,
26   `led3` tinyint(4) DEFAULT '0',
27   `led4` tinyint(4) DEFAULT '0',
28   `texto` varchar(100) DEFAULT NULL,
29   `imagen` varchar(255) DEFAULT NULL,
30   `audio` tinyint(4) DEFAULT '0',
31   `mime` varchar(20) DEFAULT NULL,
32   PRIMARY KEY (`id_control`)
```

Figura 26 Librería de diseño y desarrollo

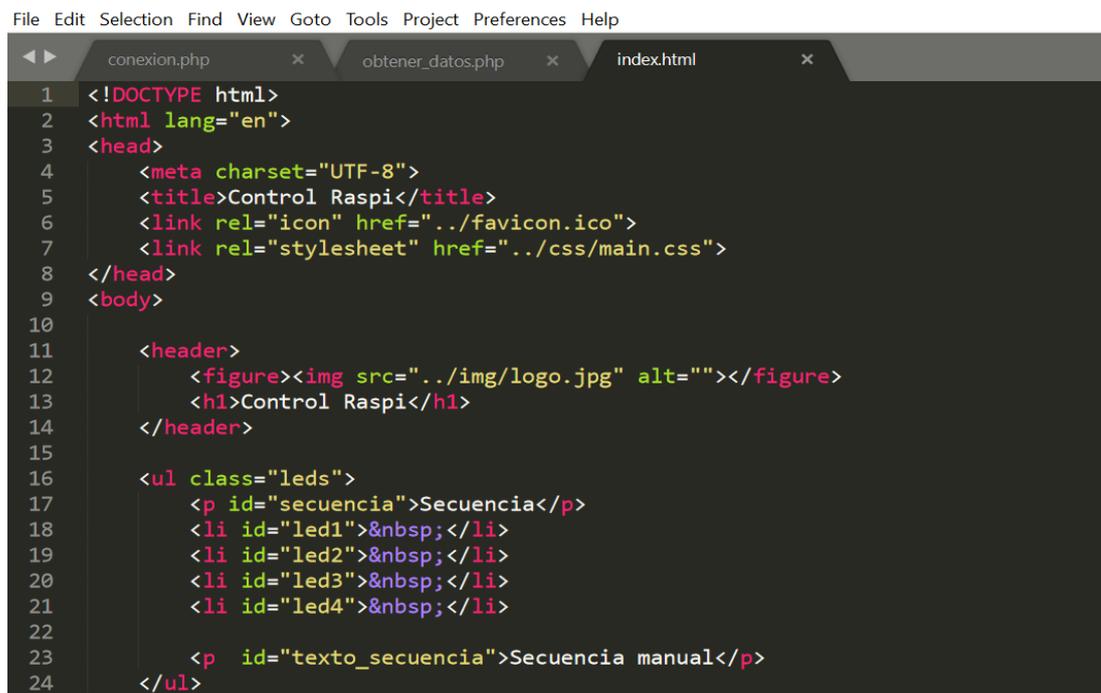
Fuente: Elaborado por el Autor

2.2 Página web administrador

Para la página del administrador es decir de la página que ejecuta la Raspberry se usan archivos similares como archivos de HTML, PHP, CSS y JavaScript. Esta página

permite que la Raspberry extraiga la información del servidor, para controlar el encendido de la secuencia de LEDs, la bomba, mostrar una imagen, reproducir un archivo de audio y mostrar un texto.

La (Figura 27), se muestra a continuación es el archivo index que es el encargado de crear la interfaz y mostrar la información del usuario.



```

File Edit Selection Find View Goto Tools Project Preferences Help
conexion.php x obtener_datos.php x index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Control Raspi</title>
6   <link rel="icon" href=" ../favicon.ico">
7   <link rel="stylesheet" href=" ../css/main.css">
8 </head>
9 <body>
10
11   <header>
12     <figure></figure>
13     <h1>Control Raspi</h1>
14   </header>
15
16   <ul class="leds">
17     <p id="secuencia">Secuencia</p>
18     <li id="led1">&nbsp;</li>
19     <li id="led2">&nbsp;</li>
20     <li id="led3">&nbsp;</li>
21     <li id="led4">&nbsp;</li>
22
23     <p id="texto_secuencia">Secuencia manual</p>
24 </ul>

```

Figura 27 Archivo index

Fuente: Elaborado por el Autor

Dentro de la carpeta PHP se tiene los archivos encargados del manejo, conexión y extracción de información de la base de datos. (Figura 28)

Nombre	Fecha de modificación	Tipo
actualiza.php	22/1/2019 04:59	Archivo PHP
conexion.php	22/1/2019 04:59	Archivo PHP
config.php	22/1/2019 04:59	Archivo PHP
obtener_datos.php	22/1/2019 04:59	Archivo PHP

Figura 28 Carpeta PHP

Fuente: Elaborado por el Autor

Para la parte dinámica de la página web se utilizó el framework de JavaScript jQuery. (Figura 29)

Nombre	Fecha de modificación	Tipo
 main	22/1/2019 04:59	Documento de hoja ...
 main.styl	22/1/2019 04:59	Archivo STYL

Figura 29 Carpeta estilo

Fuente: Elaborado por el Autor

Finalmente, en la carpeta CSS se encuentra los estilos de esta página web que se maneja igual como ya se explicó anteriormente.

2.3 Desarrollo de hardware

Desarrollo del montaje del sistema

Previo a la implementación del sistema se realizaron pruebas de funcionamiento de los elementos que consta el sistema.

Raspberry Pi 3B+ fue el primer componente en verificarse si está en buen funcionamiento antes de utilizarse, para lo cual se conectó a un computador a ver si existe alguna novedad.

2.4. Raspberry Pi versión 3B

Utilizar una Raspberry es utilizar una minicomputadora de reciente aparición en el mercado, que tiene capacidad de procesar datos, poco consumo de energía, comunicación wifi, manejo de entradas y salidas las cuales han permitido el desarrollo de esta tesis.

El sistema operativo de la tarjeta está bajo la versión de LINUX, la versión más conocida es la de los sistemas Debian, sistema que copia paquetes con formatos que facilitan la instalación.

La tarjeta tiene una memoria de 256 MB entre el CPU y el GPU, el procesador de la Raspberry pi, es el Broadcom BCM2835 SOC multimedia lo que marca la diferencia con las computadoras tradicionales. El procesador funciona con una fuente de poder de 5V que se alimenta por un puerto USB, he ahí de su consumo bajo de energía.

En sí la tarjeta contiene un procesador y un circuito integrado de gráficos, memoria RAM, varias interfaces y conectores para dispositivos externos, algunos de éstos son esenciales dependiendo de la necesidad otros son opcionales. Los que se consideran esenciales son: la tarjeta SD, teclado USB, monitor, fuente de poder y adaptador de conexión HDMI.

La programación se realiza en Python lenguaje de programación sencilla creado en los años 90 la cual tiene una sintaxis muy limpia, multiplataforma y que favorece un código legible para el control de los puertos GPIO tanto para entradas como para salidas, en este caso se realizó tres tipos de archivos:

- **Flujo.py.** El archivo flujo.py se utiliza para el control de los datos del sensor de flujo que posteriormente enviara los valores a la página web. (Figura 30)

```

File Edit Selection Find View Goto Tools Project Preferences Help
flujo.py x
1 import RPi.GPIO as GPIO
2 import time
3 import MySQLdb
4
5 GPIO.setmode(GPIO.BCM)
6 GPIO.setup(20, GPIO.IN) ## GPIO 20 como salida
7
8 def flujo():
9     estado_flujo = 0
10    estado_flujo = GPIO.input(20)
11
12    db = MySQLdb.connect(host="34.223.215.43", user="root", passwd="38540
13    cur = db.cursor()
14    cur.execute("UPDATE control SET nivel_agua="+str(estado_flujo)+"")
15    db.commit()
16
17 try:
18     while True:
19         flujo()
20         time.sleep(0.2)
21 except KeyboardInterrupt:
22     print "Script finalizado."
23     db.close ()
24     GPIO.cleanup()

```

Figura 30 Archivo flujo.py

Fuente: Elaborado por el Autor

- **Nivel.py.** El archivo nivel.py controla el sensor de nivel de agua, el Raspberry mediante este archivo detecta 1 lógico o 0 y envía los datos a la página web el estado del sensor indicando si contiene o no agua. (Figura 31)

```

File Edit Selection Find View Goto Tools Project Preferences Help
conexion.php x flujo.py x nivel.py x
1 import RPi.GPIO as GPIO
2 import time
3 import MySQLdb
4
5 GPIO.setmode(GPIO.BCM)
6 #GPIO.setup(22, GPIO.IN, pull_up_down = GPIO.PUD_UP) ## GPIO 22 como
7 GPIO.setup(22, GPIO.IN) ## GPIO 22 como entrada
8
9 global count
10 count = 0
11
12 def guardar_dato(nivel):
13     db = MySQLdb.connect(host="34.223.215.43", user="root", passwd='
14     cur = db.cursor()
15     cur.execute("UPDATE control SET nivel_flujo='"+str(nivel)+"'")
16     db.commit()
17
18 def contador_pulso():
19     global count
20     count = 0
21     for x in range(0, 60):
22         if GPIO.input(22) == 0:
23             count = count + 1
24             #time.sleep(0.001)
25     print(count)
26     flow = (count * 60 * 2.25 / 1000)
27     print "Flujo: %.3f L/m" % (flow)
28     guardar_dato(flow)
29

```

Figura 31 Archivo nivel.py

Fuente: Elaborado por el Autor

- **LEDs.py**. Se encarga del encendido de la secuencia de los LEDs y la bomba de igual forma fue hecho en PYTHON es un programa que sirve para el control de los puertos. (Figura 32)

```
File Edit Selection Find View Goto Tools Project Preferences Help
leds.py x
1 import RPi.GPIO as GPIO
2 import time
3 import MySQLdb
4
5 GPIO.setmode(GPIO.BCM)
6 GPIO.setup(26, GPIO.OUT) ## GPIO 26 como salida
7 GPIO.setup(19, GPIO.OUT) ## GPIO 19 como salida
8 GPIO.setup(13, GPIO.OUT) ## GPIO 13 como salida
9 GPIO.setup(6, GPIO.OUT) ## GPIO 6 como salida
10
11 def lectura():
12     db = MySQLdb.connect(host="34.223.215.43", user="")
13     cur = db.cursor()
14     cur.execute("SELECT * FROM control LIMIT 1")
15     for row in cur.fetchall():
16         estados = (str(row[8])).split('*')
17         audio_activo = row[10]
18
19         if (audio_activo==1):
20             for linea in estados:
21                 # obtiene el estado de los pines
22                 valor = linea.split('-')
23
24                 # led 1
25                 if (valor[0]=='1'):
26                     GPIO.output(26, True)
27                 else:
28                     GPIO.output(26, False)
29                 # led 2
30                 if (valor[1]=='1'):
31                     GPIO.output(19, True)
32                 else:
33                     GPIO.output(19, False)
34                 # led 3
35                 if (valor[2]=='1'):
```

Figura 32 Archivo LEDs.py

Fuente: Elaborado por el Autor

2.5. Etapa de filtrado y potencia para efecto audio rítmico

En la etapa del circuito de filtrado se obtiene la señal de audio de la tarjeta Raspberry, esta señal es pre amplificada y posteriormente pasa a filtros activos de banda angosta mediante el uso de amplificadores operacionales.

Para poder obtener el efecto de luces audio-rítmicas, se usará tres filtros activos el primer filtro para dejar pasar las señales de frecuencia baja, el segundo filtro para señales de frecuencia media y por último un tercer filtro para señales de frecuencias altas, cada frecuencia de los filtros envía una señal al circuito de potencia para que esto a su vez active el grupo de luces led, para la etapa de filtrado se utilizó una fuente simétrica de +/- 12Vdc.

En la (Figura 33), se observa el circuito preamplificador, en el que se usará el operacional LM358

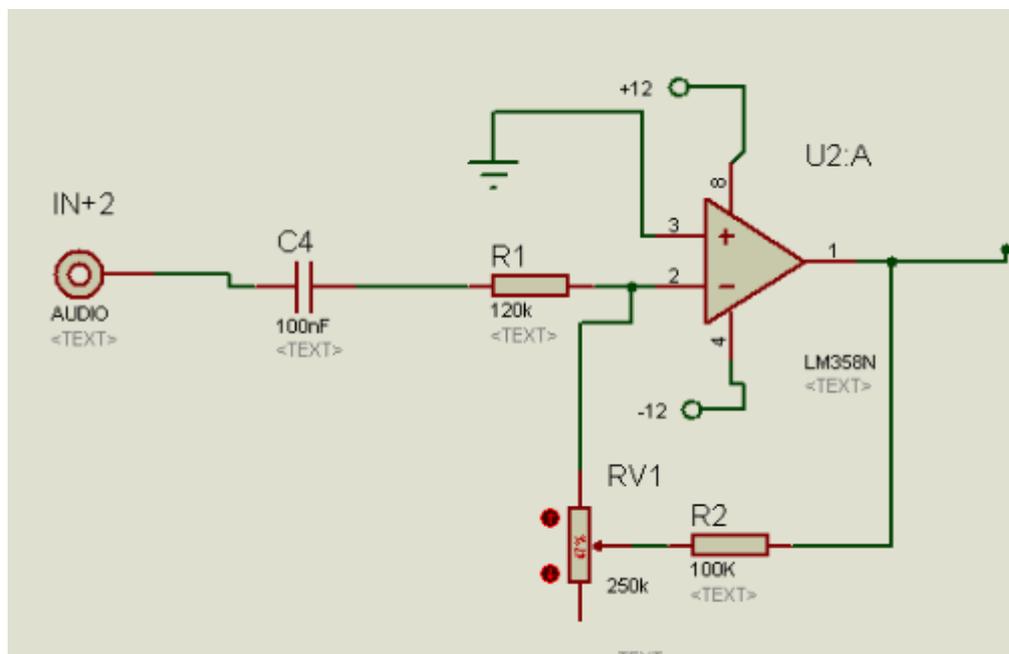


Figura 33 Preamplificador de audio

Fuente: Elaborado por el Autor

En la (Figura 34) se observa el circuito del filtro pasa banda para la frecuencia baja, para este circuito se usó el amplificador operacional LM324 puesto que integra 4 amplificadores operacionales en un solo chip.

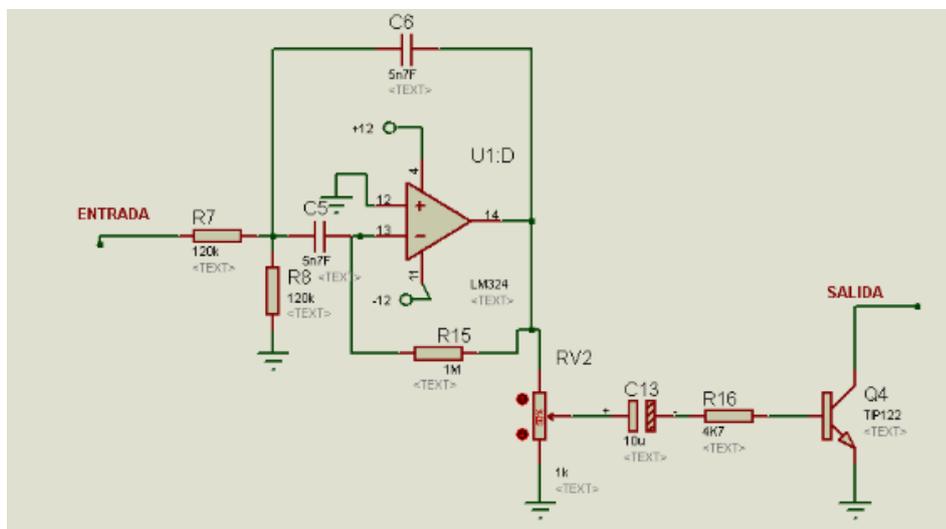


Figura 34 Circuito filtro pasa banda angosta

Fuente: Elaborado por el Autor

El circuito de potencia está compuesto por transistores Darlington, el transistor utilizado es el TIP 127 capaz de manejar una carga de hasta 4 A y 80 V según las especificaciones del fabricante, básicamente amplifica la señal que proviene de los filtros audio rítmicos, para poder conectar los leds RGB de la cascada.

En la (Figura 35) se muestra el circuito de potencia encargado de enviar el voltaje +12V hacia las tiras LED

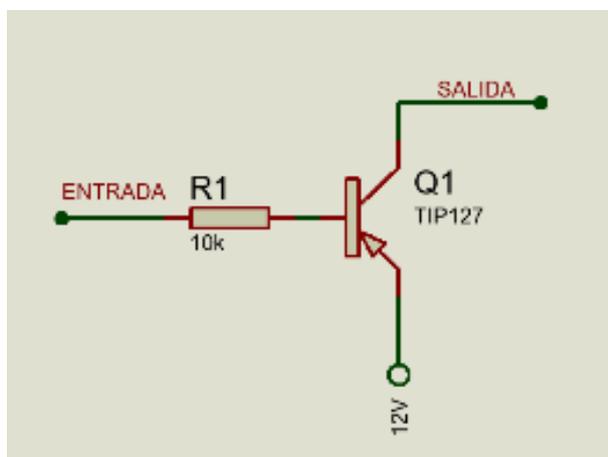


Figura 35 Circuito de potencia para tira LED

Fuente: Elaborado por el Autor

2.6. Etapa de aislamiento óptico y potencia para secuencia de LEDs

Mediante la aplicación web, el usuario puede realizar el control del encendido de las luces, mediante la selección de secuencias 1, 2, 3 o mediante el control manual de las mismas. Para el manejo de los LED se utilizó los puertos GPIO 26, 19 y 13, estos pines controlan el encendido de los colores rojo, verde y azul.

En la (Figura 36) se muestra el circuito de aislamiento óptico, este circuito aísla la parte de control de la parte de potencia, en este circuito se utilizó el optoacoplador 4N25, adicional se muestra el circuito de potencia compuesto por el transistor TIP122 el mismo que es capaz de manejar una carga de hasta 4 A y 80V. El transistor TIP 122 se encuentra configura en corte y saturación,

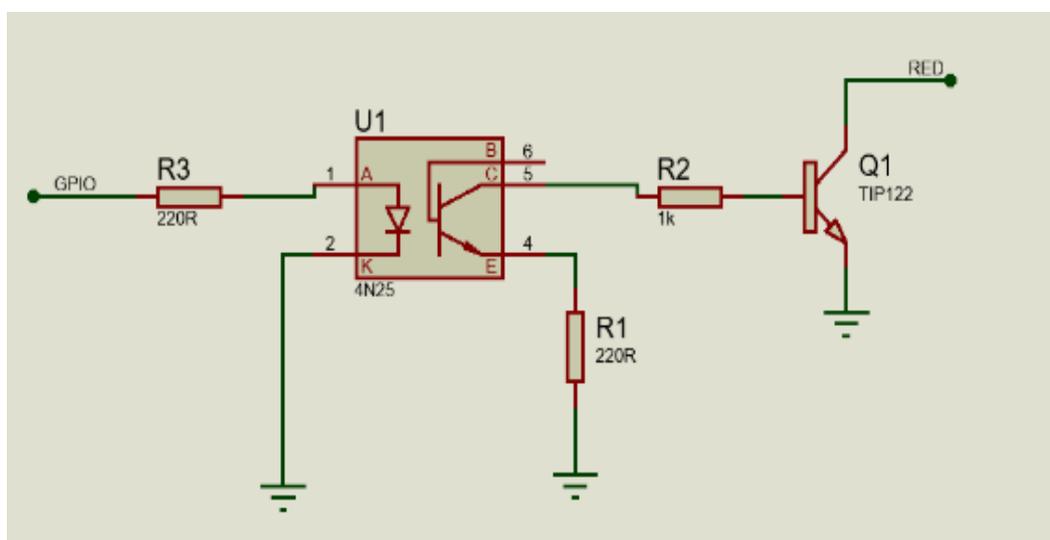


Figura 36 Circuito de aislamiento y potencia para encendido de LED rojo

Fuente: Elaborado por el Autor

2.7 Iluminación RGB

Para la iluminación se utilizó manguera LED RGB de 12V, la cual se polariza en el ánodo con voltaje positivo proveniente del circuito de filtrado audio rítmico y las terminales negativas RGB se obtienen del encendido de las secuencias o control manual. (Figura 37)



Figura 37 RGB

Fuente: Luxled, 2017

2.8 Bomba

Para crear los efectos de cascada se utilizará una bomba sumergible de agua de 110V. Para el funcionamiento de la bomba se usará un relé de estado sólido que se controlará mediante la tarjeta Raspberry desde los puertos GPIO. (Figura 38)



Figura 38 Bomba de agua

Fuente: Lawn & Garden Mexicana S.A, 2010

La bomba sumergible es ideal para uso residencial y comercial con bajo flujo de agua y con proceso de recirculación.

Las características de la bomba se detallan a continuación:

- Motor de 0.16 hp (1/6 hp) / 1 fase / 115 v
- Diseñada para uso continuo
- Descarga de 1" con adaptador para manguera de jardín
- Protector térmico
- Capacidad de manejar sólidos de hasta 1/8"
- Temperatura máxima del líquido 49°C
- Impulsor fabricado en nylon y diseño semi abierto
- Carcasa del motor en aluminio fundido

2.9 Etapa de censado de nivel y flujo de agua

En la aplicación web se puede monitorear el estado del nivel de agua y el flujo mediante sensores. El circuito del sensor de nivel de agua está compuesto por un sensor flotador de ángulo.

Para este trabajo fue necesario buscar una acción de mejora que fue usar un sensor con mayores características, el mismo que se puede usar para una operación de llenado y vaciado de un tanque de agua. El sensor actúa como interruptor de nivel de agua en ángulo. El cual está basado en un interruptor magnético tipo Reed Switch que se encuentra herméticamente sellado dentro del vástago del sensor y que se acciona al mover el flotador de espuma el cual contiene un imán permanente. (Figura 39)



Figura 39 Sensor nivel de agua

Fuente: Elaborado por el Autor

En la (Figura 40) se muestra el circuito utilizado, mediante este circuito conectado con una resistencia pull up de 1k, el Raspberry puede leer 3v o 0V según el nivel del agua sea bajo o alto, el Raspberry lee el puerto GPIO 20 y actualiza los valores en la base de datos de la página web.

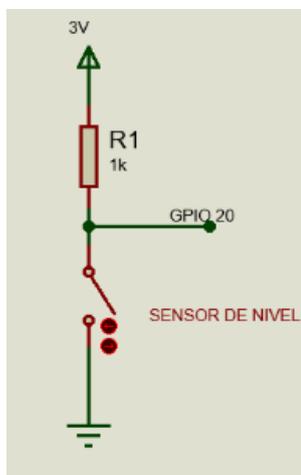


Figura 40 Sensor nivel de agua

Fuente: Elaborado por el Autor

El sensor de flujo utilizado es YF-S201, El sensor de flujo de agua de 1/2" YF-S201 sirve para medir caudal de agua en tuberías de 1/2" de diámetro. También puede ser empleado con otros líquidos de baja viscosidad, como: bebidas gasificadas, bebidas alcohólicas, combustible, entre otros. Es un caudalímetro electrónico de tipo turbina. Compatible con sistemas digitales como Arduino, Pic, Raspberry Pi, PLCs. El sensor posee tres cables: rojo (VCC: 5VDC), negro (tierra) y amarillo (salida de pulsos del sensor de efecto Hall). (Figura 41)



Figura 41 Sensor de flujo

Fuente: Elaborado por el Autor

En la (Figura 42) se muestra la conexión del sensor de flujo, el Raspberry realiza la lectura del puerto GPIO22 y actualiza el valor en la base de datos de la aplicación web.

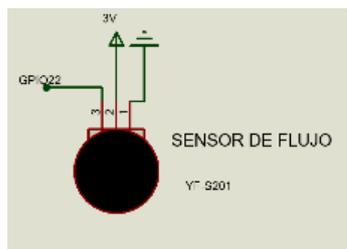


Figura 42 Conexión del sensor de flujo

Fuente: Elaborado por el Autor

2.10 Presupuesto y materiales

A continuación, se describirá los elementos que son necesarios para la implementación de dispositivo propuesto.

Tabla 1 Tabla de materiales y precios

N°	DESCRIPCIÓN	CANT.	V.UNIT	V. TOTAL
1	PLACA PCB PARA CONEXIÓN	1	\$11,20	\$11,20
2	TL074	1	\$0,80	\$0,80
3	ZÓCALO DE 14 PINES	1	\$0,16	\$0,16
4	LT082	1	\$0,75	\$0,75
5	ZÓCALO DE 8 PINES	1	\$0,10	\$0,10
6	TIP 110V	6	\$0,60	\$3,60
7	POTENCIÓMETRO 10K	4	\$0,38	\$1,52
8	CAPACITORES 10UF-25V	3	\$0,15	\$0,45
10	CONDENSADORES CERÁMICOS	7	\$0,50	\$3,50
11	LED AUTOBRILLO	1	\$0,25	\$0,25
12	RESISTENCIAS 1/4W	27	\$0,03	\$0,81
13	BORNERA 3 PINES	5	\$0,35	\$1,75
14	JACK DE AUDIO	2	\$0,65	\$1,30
15	PLACA PCB 9,5 CM*6 CM	1	\$8,00	\$8,00
16	4N25	4	\$0,55	\$2,20
17	2N3906	1	\$0,10	\$0,10
18	RELAY 12V	1	\$1,00	\$1,00
19	BORNERA 2 PINES	6	\$0,28	\$1,68
20	1N4007	1	\$0,10	\$0,10
23	TIP 127	1	\$0,60	\$0,60
24	BARRA RGB LED	15	\$2,25	\$33,75
25	MÓDULO AMPLIFICADOR	1	\$15,00	\$15,00

N°	DESCRIPCIÓN	CANT.	V.UNIT	V. TOTAL
27	CABLE AUXILIAR AUDIO	2	\$4,00	\$8,00
28	CONECTOR AUDIO	1	\$0,85	\$0,85
29	BORNERA JACK DC	1	\$1,00	\$1,00
30	FUENTE 12V	1	\$12,90	\$12,90
31	CABLE DE PODER	1	\$3,50	\$3,50
32	CABLES PARA ARDUINO	15	\$0,15	\$2,25
33	BOMBA DE AGUA	1	\$174,00	\$174,00
34	MÓDULO TEMPORIZADOR CON EL 555	1	\$18,00	\$18,00
35	CINTA LED METROS	2	\$3,50	\$7,00
36	INTERRUPTOR ON/OFF	1	\$2,20	\$2,20
37	GABINETE 60CM*40CM	1	\$21,00	\$21,00
38	EXTENSIÓN DE CABLE FLEXIBLE PARA LUCES	1	\$19,90	\$19,90
39	SENSOR DE FLUJO	1	\$12,50	\$12,50
40	SENSOR DE NIVEL	1	\$18,00	\$18,00
41	MÓDULO RELÉ DE 4 CANALES	1	\$8,50	\$8,50
43	IMPREVISTOS Y VARIOS	1	\$50,00	\$50,00
			TOTAL	\$854,52

CAPÍTULO III. IMPLEMENTACIÓN

3.1. Desarrollo

El sistema está controlado por medio de una Raspberry pi, que es una placa que permite la comunicación entre módulos de potencia, relees, filtros, audio e iluminación. Las características que se destacan de esta tarjeta son la capacidad de memoria interna, la cual permite el almacenamiento de datos del sistema; el número de puertos GPIO requeridos, los cuales se programan para la activación de los dispositivos tales como el motor, luces LEDs, y sensores.

3.2 Implementación

Previo a la implementación de los dispositivos en las tarjetas electrónicas, se realizó el diseño de las placas de: potencia, relees y amplificador en la herramienta Proteus. Este software permitió hacer pruebas de conexión con los diferentes dispositivos para verificar su correcto funcionamiento.

Verificado el correcto funcionamiento de los dispositivos electrónicos se procede a la implementación de las tres placas. Teniendo en cuenta que estas se integran, se realizan pruebas independientes con cada una de ellas y al obtener un resultado favorable se procede a integrarlas para poder constatar su funcionamiento en conjunto con el fin de obtener el control del motor, luces y sonido.

Una vez hechas las pruebas y verificado el correcto funcionamiento de los dispositivos electrónicos se procede a la implementación de las tres placas. Teniendo en cuenta que estas se unen, se realizan pruebas independientes con cada una de ellas y al obtener un resultado favorable se procede a integrarlas para poder constatar su funcionamiento en conjunto, con el fin de obtener el control del motor, luces y sonido.

Circuito de potencia

En la (Figura 43) se muestra el diagrama de la placa del circuito de filtros activos con visualización en 3D.

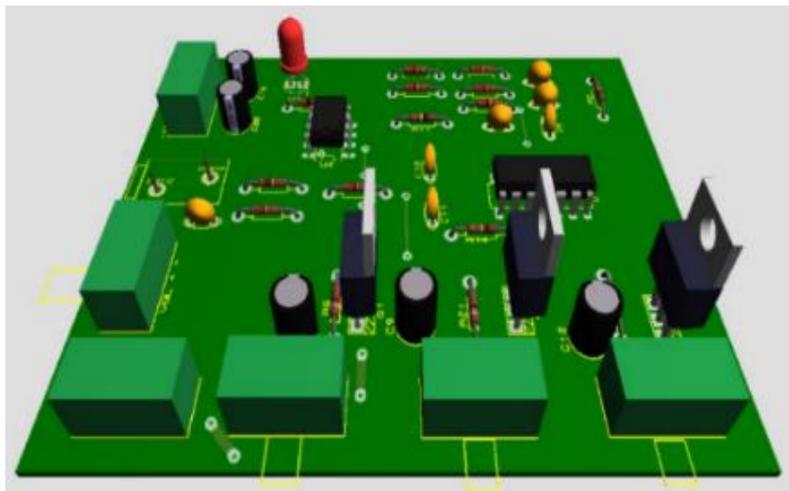


Figura 43 Vista 3D de la placa de filtros activos para luces audio rítmicas

Fuente: Elaborado por el Autor

En la (Figura 44) se muestra el diseño del circuito impreso de la placa de filtros activos para luces audio rítmico.

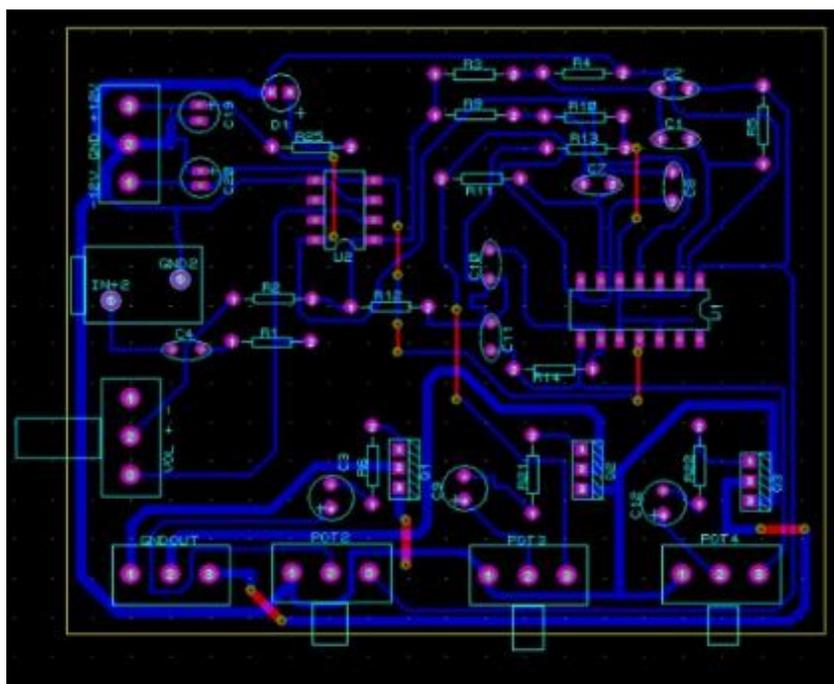


Figura 44 Diseño del circuito impreso filtros activos

Fuente: Elaborado por el Autor

Circuito de potencia

El circuito de potencia está conformado por transistores Darlington, encargados de encender las luces LED correspondientes a cada frecuencia. (Figura 45)

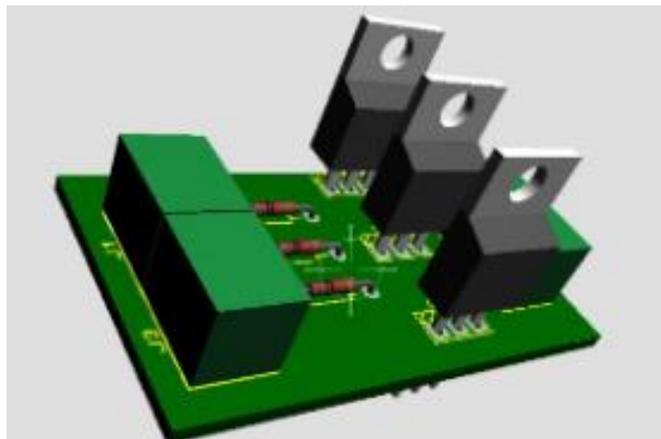


Figura 45 Vista 3D de la placa del circuito de potencia

Fuente: Elaborado por el Autor

En la (Figura 46) se muestra el diseño del circuito impreso de la placa de potencia para la activación luces audio rítmicas.

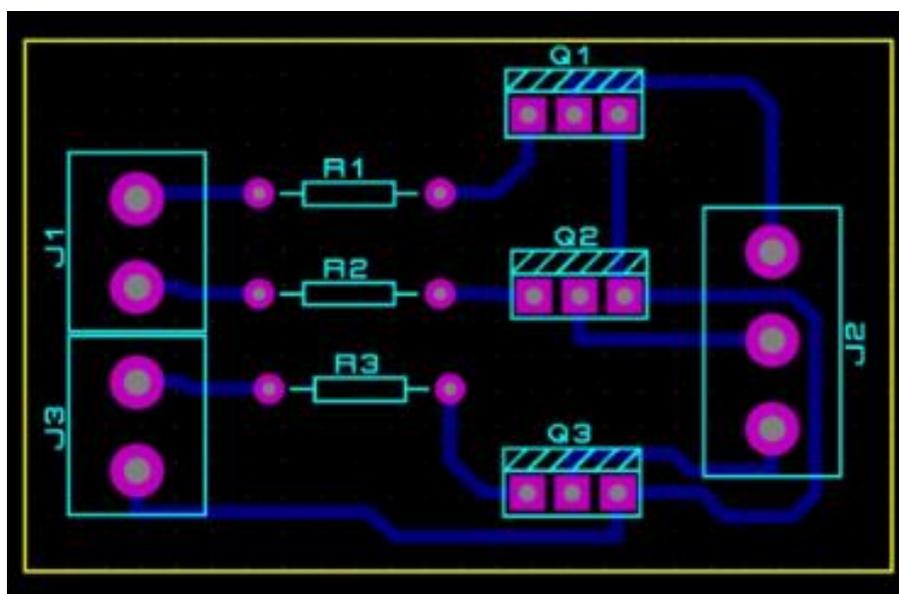


Figura 46 Diseño del circuito impreso

Fuente: Elaborado por el Autor

Fabricación de los circuitos impresos

Para la fabricación del circuito impresión se utilizó el proceso de transferencia térmica y posterior atacado con ácido cloruro férrico. (Figura 47, 48)

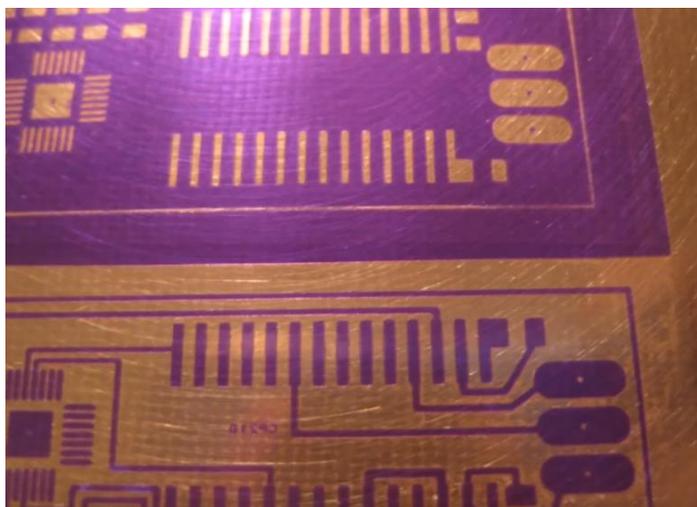


Figura 47 Planchado de baquelita

Fuente: Elaborado por el Autor

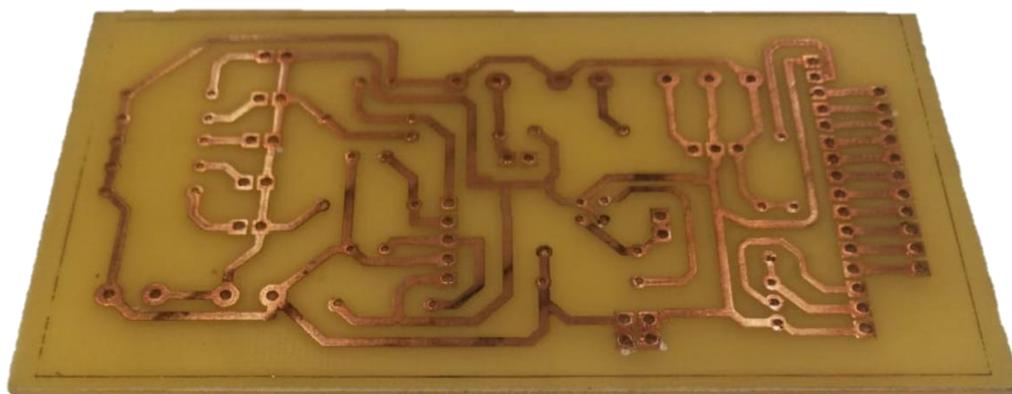


Figura 48 Perforado de baquelita

Fuente: Elaborado por el Autor

Posteriormente al diseño de la placa, se procede a soldar los elementos. Ver (Figuras: 49, 50, 51 y 52)

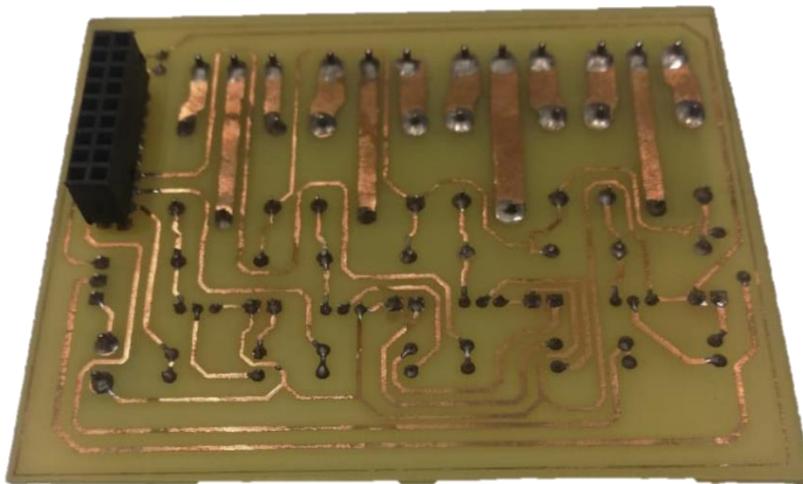


Figura 49 Soldadura de baquelita

Fuente: Elaborado por el Autor

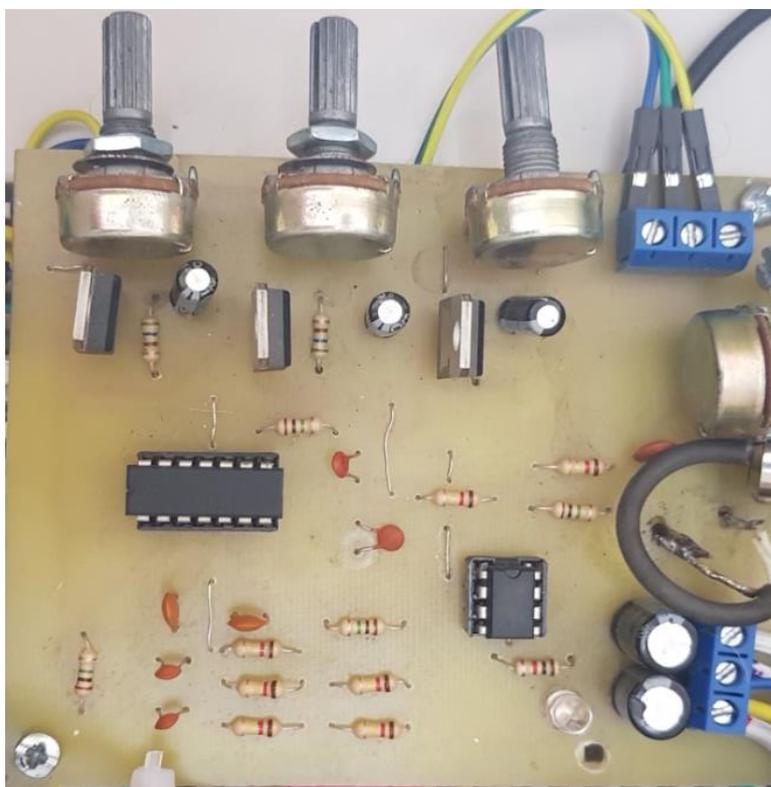


Figura 50 Circuito impreso ensamblado de los filtros

Fuente: Elaborado por el Autor



Figura 51 Montaje de potencia

Fuente: Elaborado por el Autor



Figura 52 Circuito de aislamiento y potencia

Fuente: Elaborado por el Autor

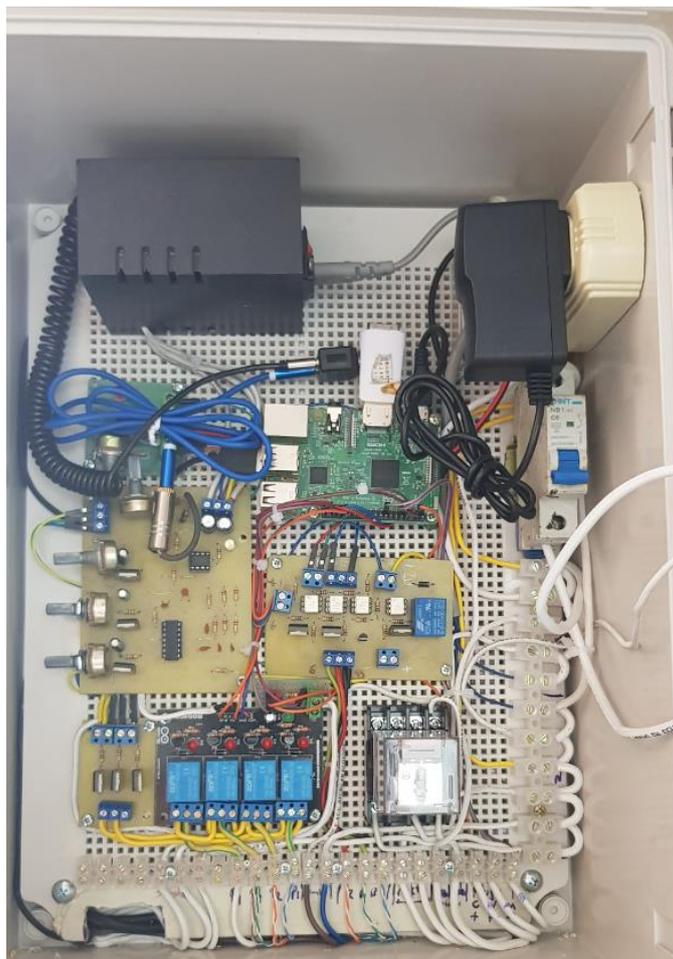


Figura 53 Tablero de control completo

Fuente: Elaborado por el Autor



Figura 54 Caja de control

Fuente: Elaborado por el Autor



Figura 55 proyecto Final

Fuente: Elaborado por el Autor

3.3 Pruebas de funcionamiento y análisis de resultados.

Se realizó la comprobación de los dispositivos del sistema antes de colocarlo definitivamente en la cascada. A continuación, en las tablas 2 y 3 se pueden constatar que todos los componentes se encuentran funcionando.

Tabla 2 Lista de verificación de sistema de control

PRUEBA 1 DE ARRANQUE DE DISPOSITIVOS		
ELEMENTOS	ARRANCA	
	SI	NO
FUENTE DE VOLTAJE +12V,-12V	√	
ENCENDIDO DE RASPBERRY	√	
SENSOR DE NIVEL	√	
SENSOR DE FLUJO	√	
BOMBA DE AGUA	√	
LUCES RGB	√	

Tabla 3 Lista de verificación de sistema de control

PRUEBA 2 DE ARRANQUE DE DISPOSITIVOS		
ELEMENTOS	ARRANCA	
	SI	NO
FUENTE DE VOLTAJE +12V,-12V	√	
ENCENDIDO DE RASPBERRY	√	
SENSOR DE NIVEL	√	
SENSOR DE FLUJO	√	
BOMBA DE AGUA	√	
LUCES RGB	√	

Se realizó la comprobación de la entrada a la página web, ingresando los datos de usuario y contraseña requeridos, cuyos resultados se muestra en tabla 4, en la que se puede evidenciar el acceso sin inconvenientes.

Tabla 4 Datos ingreso a página web

PRUEBA DE INGRESO A LA PAGINA WEB			
DESCRIPCIÓN		INGRESO	
		SI	NO
PRUEBA #1	USUARIO INCORRECTO Y CONTRASEÑA CORRECTA		√
PRUEBA #2	USUARIO CORRECTO Y CONTRASEÑA INCORRECTA		√
PRUEBA #3	USUARIO CORRECTO Y CONTRASEÑA CORRECTA	√	

Se realizó la comprobación, de los botones de luces automático-manual, secuencias, audio, nivel de flujo, nivel de agua, luz cálida, de la página web, tal como se muestra en tablas 5 y 6 en las que se puede constatar, que una vez hechas las pruebas, los botones funcionan de acuerdo a lo requerido.

Tabla 5 Datos de verificación de botones de página web

PRUEBA 1 BOTONES DE ACCESOS DE PAG WEB		
BOTONES	ANTIVACIÓN	
	SI	NO
AUTOMÁTICA Y MANUAL	√	
LUZ AZUL	√	
LUZ VERDE	√	
LUZ ROJA	√	
LUZ CÁLIDA	√	
SECUENCIA 1	√	
SECUENCIA 2	√	
SECUENCIA 3	√	
AUDIO	√	
NIVEL DE FLUJO	√	
NIVEL DE AGUA	√	

Tabla 6 Datos de verificación de botones de página web

PRUEBA 1 BOTONES DE ACCESOS DE PAG WEB		
BOTONES	ANTIVACIÓN	
	SI	NO
AUTOMÁTICA Y MANUAL	√	
LUZ AZUL	√	
LUZ VERDE	√	
LUZ ROJA	√	
LUZ CÁLIDA	√	
SECUENCIA 1	√	
SECUENCIA 2	√	
SECUENCIA 3	√	
AUDIO	√	
NIVEL DE FLUJO	√	
NIVEL DE AGUA	√	

CONCLUSIONES

- ✓ Se implementó un sistema hidráulico e iluminación en la cascada del Santuario Corazón de Jesús mediante un dispositivo RASPBERRY PI3, el cual permite la actualización de datos cada segundo e integra comunicación WIFI, para establecer el acceso remoto desde cualquier lugar.
- ✓ Para el desarrollo del presente proyecto se definió los dispositivos electrónicos a utilizarse, teniendo en cuenta su fácil adquisición en el mercado, su bajo costo y factibilidad de reemplazo; en el caso de daño, desgaste o mala manipulación del usuario
- ✓ El sistema permite monitorear el correcto funcionamiento de las secuencias de luces, la activación de encendido-apagado, el flujo del agua y el nivel de la bomba sumergible; mediante una aplicación web y la ayuda de los datos generados por el sistema de control.
- ✓ El presente proyecto utiliza un sistema automático-manual que integra lenguajes de programación sencilla como PHP, JAVA, CSS, PYTHON, que permitieron la creación de una interface amigable para el manejo y control por parte del usuario.
- ✓ El sensor de flujo que se implementó está programado de tal manera que pueda llegar a una precisión bastante aceptable, teniendo en cuenta, las calibraciones de acuerdo a la posición y distribución de los elementos electrónicos.
- ✓ Se utilizaron varias librerías de arduino para facilitar el diseño e implementación de los elementos en las placas del sistema, obteniendo mayor entendimiento y orden en los módulos electrónicos.
- ✓ Con la automatización del sistema para el control hidráulico e iluminación de la cascada decorativa del santuario corazón de Jesús, se fomenta el trabajo conjunto de parte de su párroco y los habitantes de la parroquia; con el fin de mantener la fe católica, y el bienestar de la comunidad.

RECOMENDACIONES

- ✓ Para que los sensores de lectura flujo y nivel funcionen correctamente hay que revisar su posición con frecuencia, ya que puede dar una lectura errónea al momento de subir los datos a la página web.

- ✓ Antes de hacer cualquier mantenimiento, se debe desconectar todas las fuentes de alimentación; además, se debe tener en cuenta que la persona que haga esta tarea debe tener conocimientos de electrónica para evitar daños del sistema.

- ✓ Debido a la circulación de agua se recomienda, la limpieza que ayude a evitar que la bomba pueda tener a futuro un deterioro en su funcionamiento.

- ✓ Se recomienda realizar el mantenimiento y limpieza cada seis meses, a la caja de control, con elementos que no dañen, ni afecten el correcto funcionamiento de los módulos electrónicos.

BIBLIOGRAFÍA

- Argüello, J. (2016). Interfaz máquina-máquina para el control de presión que ejerce las bombas de distribución de agua en unidades habitacionales. *Ingeniero en Computación*. Universidad Autónoma del estado de México, México.
- Contreras. (2019). <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>. Obtenido de <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>.
- England. (2012). *www.tecnologiayeducacion.com*. Obtenido de [www.tecnologiayeducacion.com](http://www.tecnologiayeducacion.com/%C2%BFques-luz-led/): <http://www.tecnologiayeducacion.com/%C2%BFques-luz-led/>
- Guillermo, C. (23 de Noviembre de 2014). *Las 10 fuentes más bellas y espectaculares del mundo*. Obtenido de Las 10 fuentes más bellas y espectaculares del mundo: <https://www.infobae.com/2014/11/23/1610069-las-10-fuentes-mas-bellas-y-espectaculares-del-mundo/>
- LAGO, A. S. (2015). *APLICACIONES DEL LED EN DISEÑO DE ILUMINACIÓN*. marcombo. Obtenido de www.tecnologiayeducacion.com: www.marcombo.com
- Mott, L. (1996). *Mecánica de Fluidos aplicada*. México: Pearson Educación.
- OrangeSmile. (2002). <http://www.orangesmile.com>. Obtenido de <http://www.orangesmile.com>: <http://www.orangesmile.com/extreme/es/fuentes-espectaculares/fuentes-en-dubai.htm>
- QUIMINET. (2019). <https://www.quiminet.com/articulos/como-se-logra-el-efecto-de-las-fuentes-danzantes-61746.htm>. Obtenido de <https://www.quiminet.com/articulos/como-se-logra-el-efecto-de-las-fuentes-danzantes-61746.htm>: <https://www.quiminet.com/articulos/como-se-logra-el-efecto-de-las-fuentes-danzantes-61746.htm>

ANEXOS

Anexo 1.

Programación del sistema automático para el control hidráulico e iluminación de la cascada decorativa del Santuario Corazón de Jesús mediante una aplicación web.

- PROGRAMACION DEL RASPBERRY

NIVEL.PY

```
import RPi.GPIO as GPIO
import time
import MySQLdb

GPIO.setmode(GPIO.BCM)
#GPIO.setup(22, GPIO.IN, pull_up_down = GPIO.PUD_UP) ## GPIO 22
como entrada
GPIO.setup(22, GPIO.IN) ## GPIO 22 como entrada

global count
count = 0

def guardar_dato(nivel):
    db = MySQLdb.connect(host="34.223.215.43", user="root",
passwd="385402292Mica_02", db="control_raspi")
    cur = db.cursor()
    cur.execute("UPDATE control SET
nivel_flujo='"+str(nivel)+"'")
    db.commit()

def contador_pulso():
    global count
    count = 0
    for x in range(0, 60):
        if GPIO.input(22) == 0:
            count = count + 1
            #time.sleep(0.001)
    print(count)
    flow = (count * 60 * 2.25 / 1000)
    print "Flujo: %.3f L/m" % (flow)
    guardar_dato(flow)

#GPIO.add_event_detect(22, GPIO.FALLING,
callback=contador_pulso)

try:
    while True:
        time.sleep(1)
        contador_pulso()
        time.sleep(1)
```

```
except KeyboardInterrupt:
    print "Script finalizado."
    GPIO.cleanup()
```

FLUJO.PY

```
import RPi.GPIO as GPIO
import time
import MySQLdb

GPIO.setmode(GPIO.BCM)
GPIO.setup(20, GPIO.IN) ## GPIO 20 como salida

def flujo():
    estado_flujo = 0
    estado_flujo = GPIO.input(20)

    db = MySQLdb.connect(host="34.223.215.43", user="root",
passwd="385402292Mica_02", db="control_raspi")
    cur = db.cursor()
    cur.execute("UPDATE control SET
nivel_agua='"+str(estado_flujo)+"'")
    db.commit()

try:
    while True:
        flujo()
        time.sleep(0.2)
except KeyboardInterrupt:
    print "Script finalizado."
    db.close ()
    GPIO.cleanup()
```

LEDS.PY

```
import RPi.GPIO as GPIO

import time
import MySQLdb

GPIO.setmode(GPIO.BCM)
GPIO.setup(26, GPIO.OUT) ## GPIO 26 como salida
GPIO.setup(19, GPIO.OUT) ## GPIO 19 como salida
GPIO.setup(13, GPIO.OUT) ## GPIO 13 como salida
GPIO.setup(6, GPIO.OUT) ## GPIO 6 como salida

def lectura():
    db = MySQLdb.connect(host="34.223.215.43", user="root",
passwd="385402292Mica_02", db="control_raspi")
    cur = db.cursor()
```

```

cur.execute("SELECT * FROM control LIMIT 1")
for row in cur.fetchall() :
    estados = (str(row[8])).split('*')
    audio_activo = row[10]

    if (audio_activo==1):
        for linea in estados:
            # obtiene el estado de los pines
            valor = linea.split('-')

            # LED 1
            if (valor[0]=='1'):
                GPIO.output(26, True)
            else:
                GPIO.output(26, False)
            # LED 2
            if (valor[1]=='1'):
                GPIO.output(19, True)
            else:
                GPIO.output(19, False)
            # LED 3
            if (valor[2]=='1'):
                GPIO.output(13, True)
            else:
                GPIO.output(13, False)
            # LED 4
            if (valor[3]=='1'):
                GPIO.output(6, True)
            else:
                GPIO.output(6, False)

            time.sleep(0.2)
    else:
        # LED 1
        if (row[1]==1):
            GPIO.output(26, True)
        else:
            GPIO.output(26, False)
        # LED 2
        if (row[2]==1):
            GPIO.output(19, True)
        else:
            GPIO.output(19, False)
        # LED 3
        if (row[3]==1):
            GPIO.output(13, True)
        else:
            GPIO.output(13, False)
        # LED 4
        if (row[4]==1):
            GPIO.output(6, True)
        else:
            GPIO.output(6, False)
        time.sleep(0.5)
cur.close()
db.close ()

```

```

try:
    while True:
        lectura()
        time.sleep(0.2)
except KeyboardInterrupt:
    print "Script finalizado."
    GPIO.cleanup()

```

- **PROGRAMACIÓN DE PÁGINA WEB**

LOGIN

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Control Raspi</title>
    <link rel="stylesheet" href="css/main.css">
    <link rel="icon" href="favicon.ico">
    <link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>

    <header id="login">
        <figure></figure>
        <h1>Control Raspi</h1>
    </header>

    <form id="inicio">
        <input type="text" id="usuario" placeholder="Su
nombre de usuario" required="true">
        <input type="password" id="pass" placeholder="Su
contrase&ntilde;a" required="true">
        <input type="submit" value="Ingresar">
    </form>

    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/main.js"></script>
</body>
</html>

```

INDEX

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Control Raspi</title>
  <link rel="stylesheet" href="css/main.css">
  <link rel="icon" href="favicon.ico">
  <link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>

  <header>
    <figure></figure>
    <h1>Control Raspi</h1>
    <a href="#" onclick="Salir(); return
false;">Salir</a>
  </header>

  <div class="componentes">
    <div class="lado">
      <ul class="LEDs">
        <p>Control de Luces</p>

        <li id="LED1">&nbsp;</li>
        <!-- <li id="LED2">&nbsp;</li>
        <li id="LED3">&nbsp;</li>
        <li id="LED4">&nbsp;</li> -->
        <input type="hidden" id="val_LED1"
value="0">
        <!-- <input type="hidden" id="val_LED2"
value="0">
        <input type="hidden" id="val_LED3"
value="0">
        <input type="hidden" id="val_LED4"
value="0"> -->
      </ul>

      <ul class="botones_secuencia">
        <li><a href="#" id="btn_LED1" class="btn
btn-lg btn-outline-secondary" onclick="Acciona_LED('LED1');
return false;">HABILITAR</a></li>
        <!-- <li><a href="#" id="btn_LED2"
class="btn btn-lg btn-warning" onclick="Acciona_LED('LED2');
return false;">ROJO</a></li>
        <li><a href="#" id="btn_LED3" class="btn
btn-lg btn-warning" onclick="Acciona_LED('LED3'); return
false;">VERDE</a></li>
        <li><a href="#" id="btn_LED4" class="btn
btn-lg btn-warning" onclick="Acciona_LED('LED4'); return
false;">AZUL</a></li> -->
      </ul>
    </div>
    <div class="lado">
```

```

        <ul class="LEDs">
            <p>Control Manual</p>

            <!-- <li id="LED1">&nbsp;</li> -->
            <li id="LED2">&nbsp;</li>
            <li id="LED3">&nbsp;</li>
            <li id="LED4">&nbsp;</li>
            <input type="hidden" id="val_LED1"
value="0">
            <input type="hidden" id="val_LED2"
value="0">
            <input type="hidden" id="val_LED3"
value="0">
            <input type="hidden" id="val_LED4"
value="0">
        </ul>

        <ul class="botones_secuencia">
            <!-- <li><a href="#" id="btn_LED1"
class="btn btn-lg btn-outline-secondary"
onclick="Acciona_LED('LED1'); return false;">AUX</a></li> -->
            <li><a href="#" id="btn_LED2" class="btn
btn-lg btn-warning" onclick="Acciona_LED('LED2'); return
false;">ROJO</a></li>
            <li><a href="#" id="btn_LED3" class="btn
btn-lg btn-warning" onclick="Acciona_LED('LED3'); return
false;">VERDE</a></li>
            <li><a href="#" id="btn_LED4" class="btn
btn-lg btn-warning" onclick="Acciona_LED('LED4'); return
false;">AZUL</a></li>
        </ul>
    </div>
</div>

    <ul class="botones_secuencia">
        <p>Control Secuencia</p>
        <li><a href="#" id="btn_sec_1" class="btn btn-lg btn-
outline-secondary" onclick="Reproducir_Secuencia(1); return
false;">SECUENCIA 1</a></li>
        <li><a href="#" id="btn_sec_2" class="btn btn-lg btn-
outline-secondary" onclick="Reproducir_Secuencia(2); return
false;">SECUENCIA 2</a></li>
        <li><a href="#" id="btn_sec_3" class="btn btn-lg btn-
outline-secondary" onclick="Reproducir_Secuencia(3); return
false;">SECUENCIA 3</a></li>
    </ul>

    <div class="division">
        <div class="columna">
            <form class="form-inline" id="texto">

                <input type="text" class="form-
control mb-2 mr-sm-2" id="texto_envio" placeholder="Texto a
enviar" required="true">

```

```

        <button type="submit" class="btn btn-
primary mb-2">Enviar</button>
        <span class="mensaje">&nbsp;</span>
    </form>

    <form class="form-inline" id="imagen"
accept-charset="multipart/form-data" method="post">
        <label for="">Archivo de
Imagen</label>
        <input type="file" class="form-
control mb-2 mr-sm-2" id="imagen_envio" name="imagen_envio"
placeholder="Escoger imagen" accept="image/jpeg"
required="true">
        <button type="submit" class="btn btn-
primary mb-2">Subir</button>
        <span
class="mensaje_imagen">&nbsp;</span>
    </form>

    <form class="form-inline" id="audio"
accept-charset="multipart/form-data" method="post">
        <label for="">Archivo de
Audio</label>
        <input type="file" class="form-
control mb-2 mr-sm-2" id="audio_envio" name="audio_envio"
placeholder="Escoger Audio" accept="audio/mp3" required="true">
        <button type="submit" class="btn btn-
primary mb-2">Subir</button>
        <span
class="mensaje_audio">&nbsp;</span>
    </form>

    <ul id="reproductor">
        <li class="icon-play"
id="audio_play"></li>
        <li class="icon-stop activo"
id="audio_stop"></li>
        <span class="mensaje">Audio
Detenido</span>
    </ul>
</div>

<div class="columna">
    <div class="botones_secuencia">
        <p>Nivel de Flujo</p>
        <div class="cuadro">
            <input type="text" class="form-
control mb-2 mr-sm-2" id="nivel_flujo" readonly>
        </div>

        <div class="LED_flujo"
id="LED_nivel_flujo"></div>
    </div>

    <div class="botones_secuencia">
        <p>Nivel de Agua</p>

```

```

        <div class="cuadro">
            <input type="text" class="form-
control mb-2 mr-sm-2" id="nivel_agua" readonly value="NIVEL
BAJO">
        </div>
        <div class="LED_flujo"
id="LED_nivel_agua"></div>
    </div>
</div>

    <!-- <form class="form-inline" id="secuencias">
        <select name="seleccion_secuencia"
id="seleccion_secuencia" class="form-control mb-2 mr-sm-2"
required="true">
            <option value="0"
selected="true">&nbsp;</option>
            <option value="1">Secuencia 1</option>
            <option value="2">Secuencia 2</option>
            <option value="3">Secuencia 3</option>
        </select>
        <input type="button" onclick="Reproducir_Secuencia();
return false;" id="rep_sec" value="Reproducir Secuencia"
class="btn btn-primary">
    </form> -->

<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/main.js"></script>
<script>
    $(document).ready(function(){
        var OpSys = checkOS();
        if (OpSys=="Linux"){
            window.location.href = 'admin/index.html';
        }else{
            Verifica_Inicio();
        }
    })
</script>
</body>
</html>

```

ACTUALIZA.PHP

```
<?php

$respuesta['estado'] = false;

try{
    include 'funciones.php';
    require 'meta.php';

    $respuesta['datos'] = $_POST;
    $respuesta['imagen'] = $_FILES;

    if (isset($_POST['texto'])) {
        $actualiza = Meta::Actualizar_Campo('control',
        'texto', $_POST['texto'], 'id_control', 1);
    }elseif (isset($_POST['audio'])) {
        $actualiza = Meta::Actualizar_Campo('control',
        'audio', $_POST['audio'], 'id_control', 1);
        $nombre_audio = Meta::Consulta_Unico("SELECT
        nombre_audio FROM control LIMIT 1");
        $respuesta['nombre_audio'] =
        $nombre_audio['nombre_audio'];

        if (isset($_POST['secuencia'])) {
            switch ($_POST['secuencia']) {
                case 0:
                    # sin secuencia
                    $actualiza =
                    Meta::Actualizar_Campo('control', 'mime', '0-0-0-0',
                    'id_control', 1);
                    break;
                case 1:
                    # secuencia 1
                    $actualiza =
                    Meta::Actualizar_Campo('control', 'mime', '1-1-0-0*1-0-1-0*1-0-
                    0-1*0-1-0-0*0-0-1-0*0-0-0-1', 'id_control', 1);
                    break;
                case 2:
                    # secuencia 2
                    $actualiza =
                    Meta::Actualizar_Campo('control', 'mime', '1-1-0-0*1-1-1-0*1-0-
                    1-0*1-0-1-1*1-0-0-1*1-1-0-1*0-1-0-0*0-1-1-0*0-0-1-0*0-0-1-1*0-0-
                    0-1*0-1-0-1', 'id_control', 1);
                    break;
                case 3:
                    # secuencia 3
                    $actualiza =
                    Meta::Actualizar_Campo('control', 'mime', '1-1-0-0*1-1-1-1*1-0-
                    0-1*1-1-1-1*1-0-1-0*1-1-1-1*0-1-0-0*0-1-1-1*0-0-0-1*0-1-1-1*0-0-
                    1-0*0-1-1-1', 'id_control', 1);
                    break;
            }
        }
    }elseif (isset($_POST['control_secuencia'])) {
```

```

        $actualiza = Meta::Actualizar_Campo('control',
'secuencia', $_POST['control_secuencia'], 'id_control', 1);

        if (isset($_POST['secuencia'])) {
            switch ($_POST['secuencia']) {
                case 0:
                    # sin secuencia
                    $actualiza =
Meta::Actualizar_Campo('control', 'mime', '0-0-0-0',
'id_control', 1);
                    break;
                case 1:
                    # secuencia 1
                    $actualiza =
Meta::Actualizar_Campo('control', 'mime', '1-1-0-0*1-0-1-0*1-0-
0-1*0-1-0-0*0-0-1-0*0-0-0-1', 'id_control', 1);
                    break;
                case 2:
                    # secuencia 2
                    $actualiza =
Meta::Actualizar_Campo('control', 'mime', '1-1-0-0*1-1-1-0*1-0-
1-0*1-0-1-1*1-0-0-1*1-1-0-1*0-1-0-0*0-1-1-0*0-0-1-0*0-0-1-1*0-0-
0-1*0-1-0-1', 'id_control', 1);
                    break;
                case 3:
                    # secuencia 3
                    $actualiza =
Meta::Actualizar_Campo('control', 'mime', '1-1-0-0*1-1-1-1*1-0-
0-1*1-1-1-1*1-0-1-0*1-1-1-1*0-1-0-0*0-1-1-1*0-0-0-1*0-1-1-1*0-0-
1-0*0-1-1-1', 'id_control', 1);
                    break;
            }
        }
    }elseif (isset($_POST['LED'])) {
        $LED = $_POST['LED'];
        $valor = $_POST['valor'];
        $valor = abs($valor-1);

        $actualiza = Meta::Actualizar_Campo('control', $LED,
$valor, 'id_control', 1);
    }elseif (isset($_FILES['imagen_envio']['name'])) {

        Imagen($_FILES, 500, 500);
        $actualiza = Meta::Actualizar_Campo('control',
'imagen', $_FILES['imagen_envio']['name'], 'id_control', 1);
    }

    $respuesta['estado'] = true;
} catch (Exception $e) {
    $respuesta['error'] = $e->getMessage();
}

echo json_encode($respuesta);

```

AUDIO.PHP

```
<?php

$respuesta['estado'] = false;

try{
    require 'meta.php';

    $nombre = $_FILES['audio_envio']['name'];
    $tipo = $_FILES['audio_envio']['type'];
    $tamano = $_FILES['audio_envio']['size'];
    $ruta = $_FILES['audio_envio']['tmp_name'];
    $destino = "tmp/".$nombre;

    $actualiza = Meta::Actualizar_Campo("control",
"nombre_audio", $nombre, "id_control", 1);

    move_uploaded_file($_FILES['audio_envio']['tmp_name'],
$destino);

    $respuesta['estado'] = true;
} catch(Exception $e){
    $respuesta['error'] = $e->getMessage();
}

echo json_encode($respuesta);
```

CONEXIÓN.PHP

```
<?php
// Obtiene las credenciales para la conexión a la bd
require_once "config.php";

class Database{

    /**
     * Ñnica instancia de la clase
     */
    private static $db = null;

    /**
     * Instancia de PDO
     */
    private static $pdo;

    final private function __construct(){
        try {
            // Crear nueva conexión PDO
            self::getDb();
        } catch (PDOException $e) {
            // Manejo de excepciones
            print 'Error!: '. $e->getMessage(). '<br/>';
        }
    }
}
```

```

        die();
    }
}

/**
 * Retorna en la Única instancia de la clase
 * @return Database|null
 */
public static function getInstance()
{
    if (self::$db === null) {
        self::$db = new self();
    }
    return self::$db;
}

/**
 * Crear una nueva conexi3n PDO basada
 * en los datos de conexi3n
 * @return PDO Objeto PDO
 */
public function getDb()
{
    if (self::$pdo == null) {
        self::$pdo = new PDO(
            'mysql:dbname=' . DATABASE .
            ';host=' . HOSTNAME .
            ';port:63343;',
            USERNAME,
            PASSWORD,
            array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET
NAMES utf8")
        );

        // Habilitar excepciones
        self::$pdo->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    }

    return self::$pdo;
}

/**
 * Evita la clonaci3n del objeto
 */
final protected function __clone()
{
}

function _destructor()
{
    self::$pdo = null;
}
}
?>

```

CONFIG.PHP

```
<?php

define("HOSTNAME", "localhost");// Nombre del host (localhost
aws instance)
define("USERNAME", "root"); // Nombre del usuario
define("PASSWORD", "385402292Mica_02"); // Contraseña
define("DATABASE", "control_raspi"); // Nombre de la base de
datos
```

LOGIN.PHP

```
<?php

$respuesta['estado'] = false;

try{
    session_start();
    $_SESSION['usuario'] = 'admin';

    $respuesta['estado'] = true;
} catch(Exception $e){
    $respuesta['error'] = $e->getMessage();
}

echo json_encode($respuesta);
```

LOGUEADO.PHP

```
<?php

$respuesta['estado'] = false;

try{
    session_start();
    $_SESSION['usuario'] = 'admin';

    $respuesta['estado'] = true;
} catch(Exception $e){
    $respuesta['error'] = $e->getMessage();
}

echo json_encode($respuesta);
```

META.PHP

```
<?php

/**
 * Representa el la estructura de las metas
 * almacenadas en la base de datos
 */
require 'conexion.php';

class Meta
{
    function __construct()
    {
    }

    /**
     * Retorna en la fila especificada de la tabla 'meta'
     *
     * @param $idMeta Identificador del registro
     * @return array Datos del registro
     */
    public static function Consulta($sql)
    {
        $consulta = $sql;
        try {
            // Preparar sentencia
            $comando = Database::getInstance()->getDb()-
>prepare($consulta);
            // Ejecutar sentencia preparada
            $comando->execute();

            return $comando->fetchAll(PDO::FETCH_ASSOC);

        } catch (PDOException $e) {
            return false;
        }
    }

    /**
     * Obtiene los campos de una meta con un identificador
     * determinado
     *
     * @param $id_test Identificador de la meta
     * @return mixed
     */
    public static function Ejecutar($sql) {
        // Consulta de la meta
        $consulta = $sql;

        try {
            // Preparar sentencia
            $comando = Database::getInstance()->getDb()-
>prepare($consulta);
            // Ejecutar sentencia preparada
```

```

        $comando->execute();

        return 1;

    } catch (PDOException $e) {
        // AquÃ puedes clasificar el error dependiendo de la
excepciÃ³n
        // para presentarlo en la respuesta Json
        return -1;
    }
}

/**
 * Obtiene los campos de una meta con un identificador
 * determinado
 *
 * @param $id_test Identificador de la meta
 * @return mixed
 */
public static function Consulta_Unico($sql) {
    // Consulta de la meta
    $consulta = $sql;

    try {
        // Preparar sentencia
        $comando = Database::getInstance()->getDb()-
>prepare($consulta);
        // Ejecutar sentencia preparada
        $comando->execute();
        // Capturar primera fila del resultado
        $row = $comando->fetch(PDO::FETCH_ASSOC);
        return $row;

    } catch (PDOException $e) {
        // AquÃ puedes clasificar el error dependiendo de la
excepciÃ³n
        // para presentarlo en la respuesta Json
        return -1;
    }
}

/**
 * Eliminar el registro con el identificador especificado
 *
 * @param $idMeta identificador de la meta
 * @return bool Respuesta de la eliminaciÃ³n
 */
public static function Actualizar_Campo($tabla, $campo,
$valor, $id, $id_valida)
{
    // Sentencia DELETE
    $comando = "UPDATE ".$tabla." SET ".$campo."=? WHERE
".$id."=?";

    // Preparar la sentencia

```

```

        $sentencia = Database::getInstance()->getDb()-
>prepare($comando);

        return $sentencia->execute(array($valor, $id_valida));
    }

/**
 * Insertar una nueva meta
 *
 * @param $id_test      identificador
 * @param $nombre      nuevo titulo
 * @param $edad        nueva descripcion
 * @return PDOStatement
 */
    public static function Nuevo_Control($LED1, $LED2, $LED3,
$LED4, $texto, $imagen, $audio, $nombre_audio)
    {
        // Sentencia INSERT
        $comando = "INSERT INTO control (LED1, LED2, LED3, LED4,
texto, imagen, audio, nombre_audio) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

        // Preparar la sentencia
        $sentencia = Database::getInstance()->getDb()-
>prepare($comando);

        return $sentencia->execute(array($LED1, $LED2, $LED3,
$LED4, $texto, $imagen, $audio, $nombre_audio));

    }

}

?>

```

MOSTRAR_IMAGEN.PHP

```

<?php

require 'meta.php';

$con consulta = Meta::Consulta_Unico("SELECT * FROM control WHERE
id_control=1");
echo $con consulta['mime'];
if ($con consulta['imagen']!= ''){
    header("Content-Type: ".$con consulta['mime']);
    echo $con consulta['imagen'];
}

```

OBTENER_DATOS.PHP

```
<?php

$respuesta['estado'] = false;

try{
    require 'meta.php';

    $aux = Meta::Consulta_Unico("SELECT * FROM control ORDER BY
id_control DESC LIMIT 1");

    if ($aux['id_control']==''){
        // no existe ningun registro, entonces crea uno
        $nuevo = Meta::Nuevo_Control(0, 0, 0, 0, '', '', 0);

        $aux = Meta::Consulta_Unico("SELECT * FROM control
ORDER BY id_control DESC LIMIT 1");
        $respuesta['datos'] = $aux;
    }else{
        $respuesta['datos'] = $aux;
    }

    $respuesta['estado'] = true;

}catch(Exception $e){
    $respuesta['error'] = $e->getMessage();
}

echo json_encode($respuesta);
```

SALIR.PHP

```
<?php

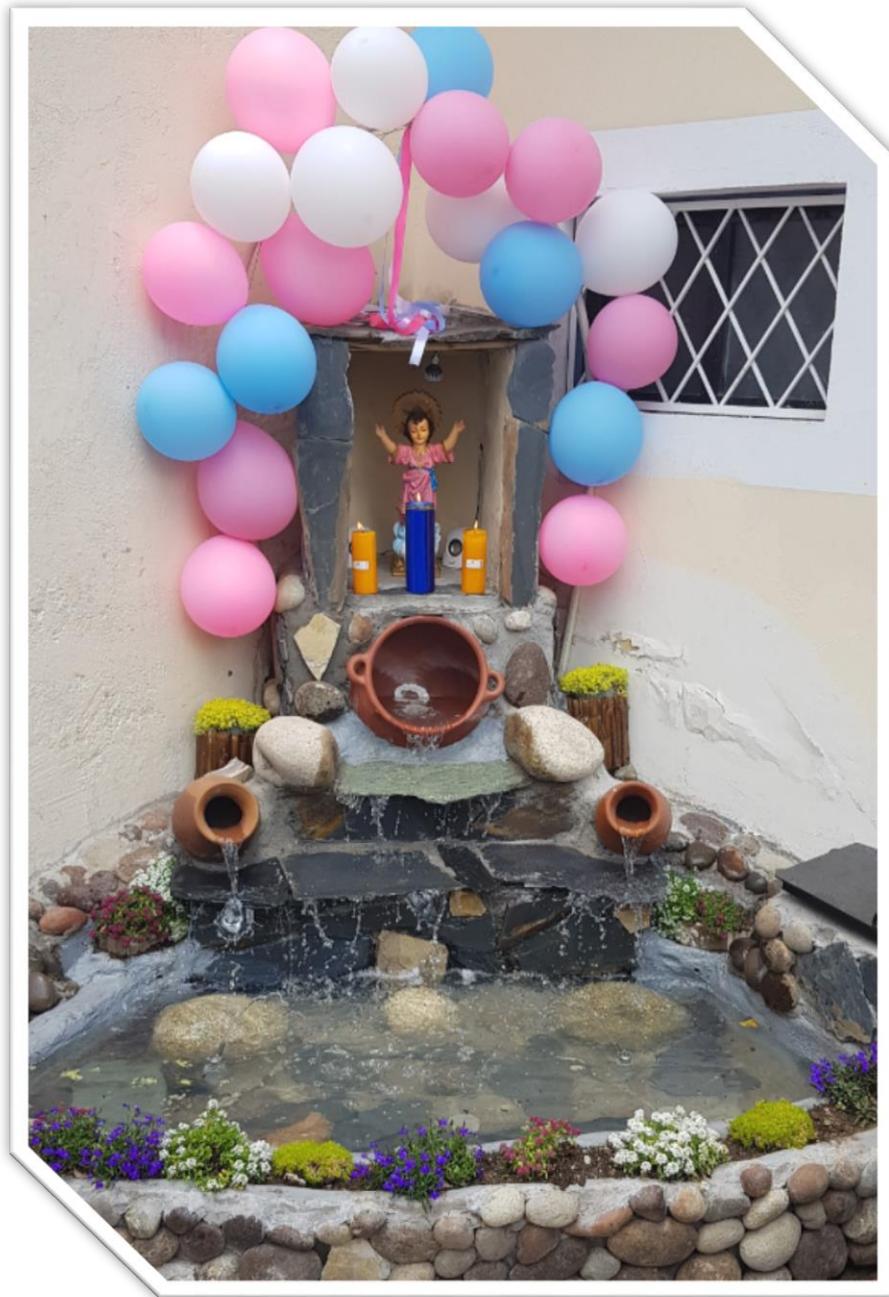
$respuesta['estado']=false;

try{
    session_start();
    session_unset();
    session_destroy();

    $respuesta['estado'] = true;
}catch(Exception $e){
    $respuesta['error'] = $e->getMessage();
}

echo json_encode($respuesta)
```

MANUAL DE USUARIO



**SISTEMA AUTOMÁTICO PARA EL CONTROL
HIDRÁULICO E ILUMINACIÓN DE LA CASCADA
DECORATIVA DEL SANTUARIO CORAZÓN DE JESÚS
MEDIANTE UNA APLICACIÓN WEB**

CONTENIDO

Introducción	3
Instrucciones de seguridad.....	4
Funcionamiento del sistema	5
Componentes de la caja de control.....	9
Solución de problemas	10
Contacto.....	11

INTRODUCCIÓN

El presente manual está elaborado con la finalidad de brindarle:

- Información adecuada acerca del funcionamiento y forma de operar el sistema.
- Ayuda a identificar a los usuarios que pueden utilizar el sistema, y qué conocimientos mínimos deben tener para lograr una comprensión de casa una de las funciones del mismo.

Es de mucha importancia leer este manual antes y/o durante la utilización del sistema, ya que lo guiará paso a paso en el manejo de todas las funciones.

INSTRUCCIONES DE SEGURIDAD

PRECAUCIONES

 <p>A yellow triangular warning sign with a black lightning bolt symbol and the text "RIESGO ELECTRICO" below it.</p>	<p>Mantenga los cables de alimentación de energía en buenas condiciones, ya que un cable eléctrico sin su recubrimiento puede provocar descargas eléctricas, corto circuito e incluso incendios de las instalaciones.</p>
 <p>A blue square icon containing a white electrical plug.</p>	<p>Garantice que la toma corriente sea de 110V, para evitar una sobrecarga eléctrica al dispositivo.</p>
 <p>A blue square icon containing three white water drops.</p>	<p>Por ningún motivo permita que el dispositivo tenga contacto con líquidos, ya que esto puede causar daños al dispositivo, personas e instalaciones por causa de un incendio.</p>
 <p>A red circle icon containing a white exclamation mark.</p>	<p>Manténgase alerta, ponga atención a lo que hace y use el sentido común al operar el sistema. Lea cuidadosamente antes de operar el equipo.</p>
 <p>A yellow triangular warning sign with a black hand reaching towards a lightning bolt symbol.</p>	<p>Evite el encendido accidental. Asegúrese de que el interruptor esté en la posición de apagado antes de conectar a la alimentación eléctrica.</p>
 <p>A blue square icon containing a white wrench.</p>	<p>Para poder realizar mantenimiento preventivo o correctivo del sistema es recomendable desconectar del suministro de energía eléctrica y acudir a un técnico capacitado.</p>

FUNCIONAMIENTO DEL SISTEMA

Para poner el equipo en funcionamiento se realiza varias conexiones:

1. Conecta el cable a la red eléctrica de 110 VAC, 60 Hz.
2. Posteriormente colocar el interruptor en estado "ON", la luz piloto debe encenderse para verificar el ingreso de voltaje de la red eléctrica al equipo.
3. Luego de activar el interruptor, el Raspberry se enciende automáticamente, a través del mouse se ingresa al navegador chromium y se digita la URL o <http://controlraspi.ddns.net/login.html>.
4. Esta opción del menú principal tiene como objetivo ingresar y actualizar las secuencias de luces, carga de música e imágenes. Aparece la siguiente pantalla.



The image shows a login page for 'Universidad Israel Control Raspi'. At the top left is the university's logo, which includes a Star of David and the text 'UNIVERSIDAD ISRAEL'. To the right of the logo, the text 'Universidad Israel' is displayed in a large, blue, sans-serif font. Below this, the title 'Control Raspi' is centered. The page features three input fields: the first is labeled 'Su nombre de usuario', the second is labeled 'Su contraseña', and the third is a blue button labeled 'INGRESAR'.

5. La primera pantalla que se visualiza es la página de Logeo, las credenciales para el ingreso son:
Contraseña: admin
Usuario: 1234
6. Luego de ingresar las credenciales se muestra la página principal en la que se puede visualizar el control manual de la iluminación.



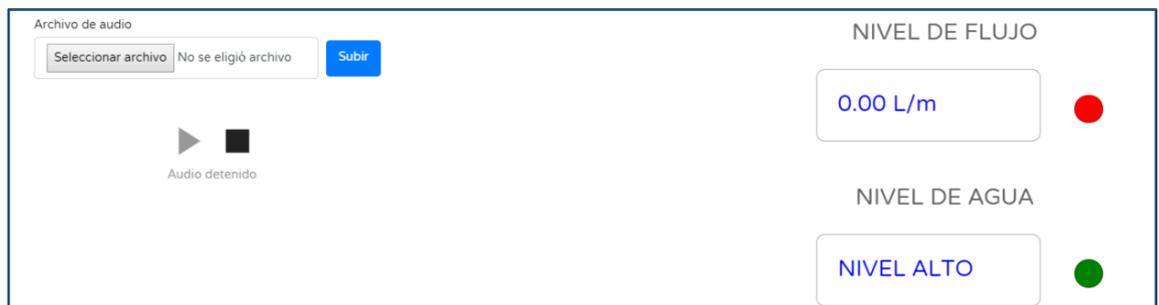
7. En la parte superior derecha se tiene el control manual de iluminación, basta con dar clic en el color del LED que se desea encender, se tiene los 3 colores primarios digitales RGB rojo verde y azul, pero se puede combinar el encendido de estos LEDs para generar más colores.

Mientras la pista de audio se encuentra reproduciendo estos colores van a tener un efecto audio rítmico, pero si la pista de audio está detenida las luces van a estar encendidas sin realizar ningún efecto.

8. En la parte superior izquierda se muestra el botón Habilitar/Deshabilitar luces, como su nombre lo indica permite habilitar el encendido y apagado de las luces ya sea para que funcione en modo audio rítmico o siempre encendidas sin efectos.
9. En la parte central de la página se tiene los botones de control manual de las secuencias 1, secuencia 2 y secuencia 3, cada secuencia realiza un juego de luces y combinación de colores sobre la iluminación LED, también controla el encendido de la bomba.



10. Cuando una pista de audio se está reproduciendo el sistema enciende las luces con una combinación de efectos de colores (secuencias) y un efecto audio rítmico.
11. Cuando una pista de audio no se está reproduciendo, las luces solo se encienden mediante los efectos de colores (secuencias).
12. En la parte izquierda de la página se muestra las opciones para enviar al servidor un texto, una imagen y un archivo de audio, estos 3 archivos se suben al servidor y el Raspberry mediante la página web las descarga y los muestra en pantalla.
13. En el lado derecho de la página web se tienen los campos para el monitoreo del estado de los sensores.



14. El sensor de nivel de agua nos muestra mediante texto "NIVEL BAJO" cuando existe poca agua y un indicador en color rojo, cuando el nivel de agua se encuentra normal este LED cambia a color verde.

15. El segundo sensor usado permite realizar las lecturas del flujo de agua en L/m, esto nos permite saber el correcto funcionamiento de la bomba de agua o si tenemos fugas de agua en alguna parte de las instalaciones, también tenemos un indicador que se encuentra en color rojo cuando no existe flujo de agua, y cuando detecta que si hay flujo de agua este indicador cambia a color verde.
16. Una vez ya realizado la carga de música, revisión de la iluminación y la bomba el Raspberry se enciende automáticamente, y en el monitor se observa el arranque del sistema operativo raspbian, a través del mouse se ingresa al navegador y se digita la URL de la página web del Raspberry.
<http://controlraspi.ddns.net/admin/index.html>



COMPONENTES DE LA CAJA DE CONTROL

El sistema está compuesto por la siguiente caja de control.

Sistema de control



N°	ELEMENTOS
1	LUZ PILOTO INDICADOR
2	INTERRUPTOR GENERAL DE ENCENDIDO

SOLUCIÓN DE PROBLEMAS

P: ¿Por qué el sistema no se enciende?

R: Asegúrese de que la fuente de alimentación estén correctamente conectada a la toma de 110V.

P: ¿Por qué el sistema no sensa los datos de nivel predeterminados?

R: Asegúrese que sensor de nivel este en una posición intermedia y no esté fuera del rango de la cantidad de agua.

P: ¿Por qué los datos no se actualizan?

R: Apague y encienda nuevamente el switch de encendido para reinicializar el sistema.

P: ¿Por qué en el monitor serie muestra datos erróneos al encender?

R: Apague y encienda nuevamente el switch de encendido para reinicializar el sistema.

P: ¿Por qué el motor sumergible no funciona?

R: Asegúrese de que los cables de alimentación del sistema estén conectado a la toma de 110V.

CONTACTO

Para más información sobre garantía, mantenimiento o consejos técnicos contactarse con:

Ing. Fabio Enríquez.

Celular: 09 95070581

Email: fabioenriquezv@hotmail.com

MANUAL TÉCNICO



**SISTEMA AUTOMÁTICO PARA EL CONTROL
HIDRÁULICO E ILUMINACIÓN DE LA CASCADA
DECORATIVA DEL SANTUARIO CORAZÓN DE JESÚS
MEDIANTE UNA APLICACIÓN WEB**

CONTENIDO

Objetivo general	3
Objetivo específicos	3
Partes del sistema	4
Listado de materiales	6
Diagramas esquemáticos	7
Diagramas PCB	10
Solución de problemas	11
Contacto	12

OBJETIVO GENERAL

Dar la información necesaria al técnico encargado del mantenimiento y modificaciones en la programación del sistema.

OBJETIVOS ESPECÍFICOS

- ✓ Proporcionar las características de cada uno de los componentes del dispositivo.

- ✓ Brindar información clara sobre las conexiones del dispositivo

- ✓ Proporcionar diseño de PCB del circuito.

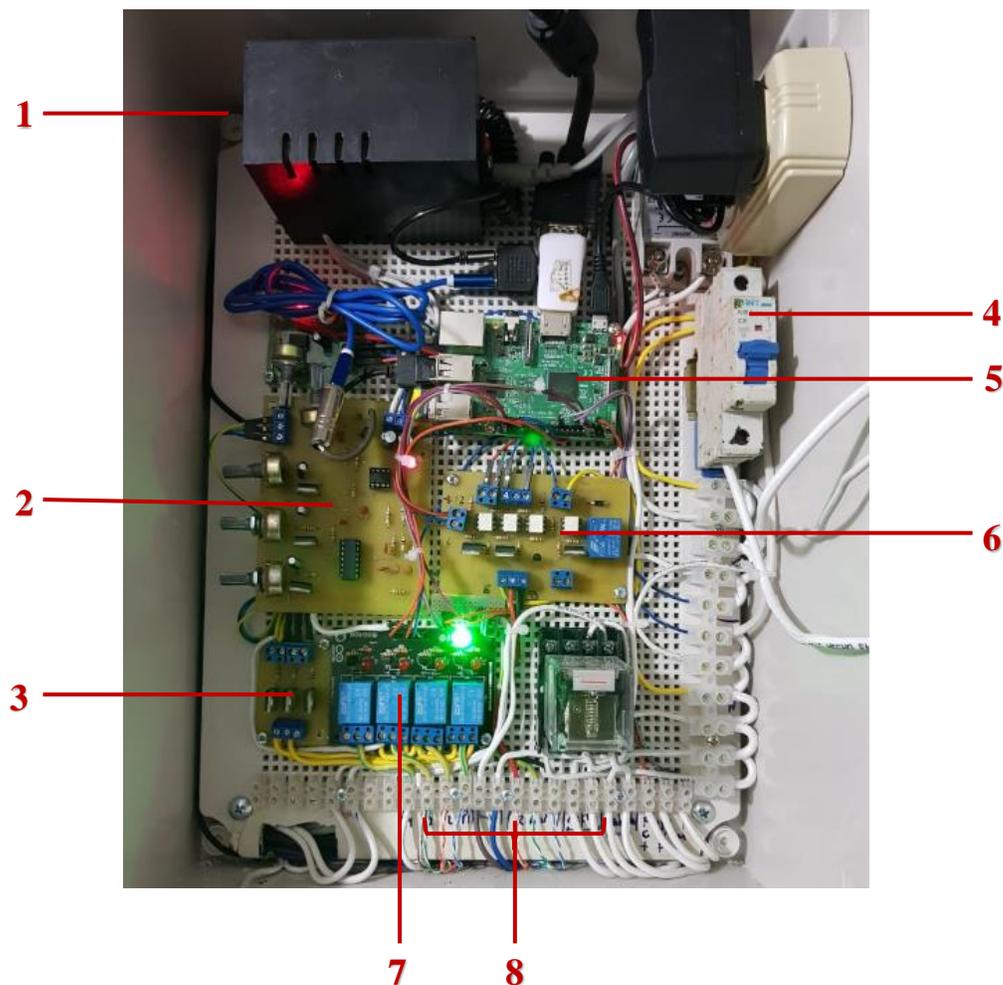
PARTES DEL SISTEMA

Externos.



Nº	ELEMENTOS
1	LUZ PILOTO INDICADOR
2	INTERRUPTOR GENERAL DE ENCENDIDO

Internos.



N°	ELEMENTOS
1	FUENTE DE +12V, -12V
2	TARJETA DE FILTRO PASO BANDA
3	TARJETA DE POTENCIA
4	BREAKER
5	TARJETA ARDUINO
6	TARJETA DE AISLAMIENTO ÓPTICO
7	TARJETA DE RELES
8	SALIDAS DEL SISTEMA

LISTADO DE MATERIALES

MATERIALES
TL074
LT082
TIP 110V
POTENCIÓMETRO 10K
CAPACITORES 10UF-25V
RESISTENCIAS 1/4W
4N25
2N3906
RELAY 12V
1N4007
TIP 127
BARRA RGB LED
BOMBA DE AGUA
INTERRUPTOR ON/OFF
SENSOR DE FLUJO
SENSOR DE NIVEL
MÓDULO RELÉ DE 4 CANALES

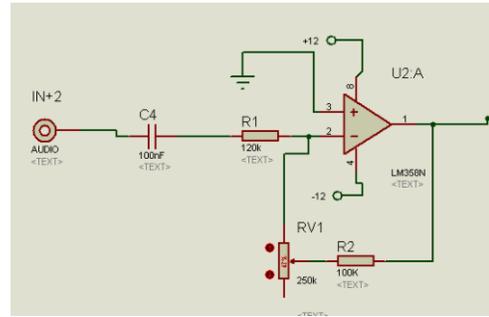
DIAGRAMAS ESQUEMÁTICOS

Diagrama de bloques.

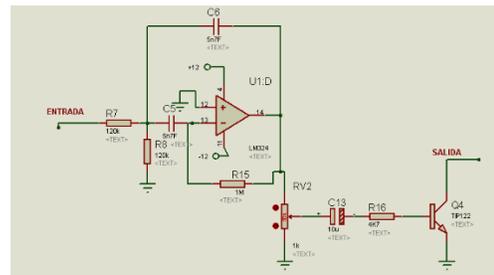


Circuitos esquemáticos.

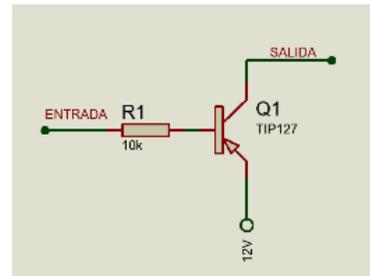
- Circuito de filtrado y potencia para efecto audio rítmico



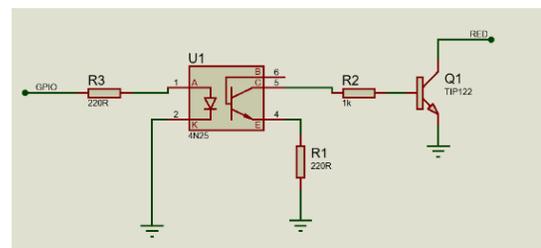
- Circuito filtro pasa banda angosta



- Circuito de potencia para tira LED



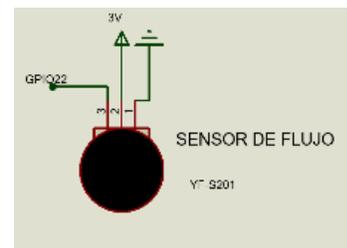
- Circuito de aislamiento y potencia para encendido de LED, RGB



- Circuito del Sensor nivel

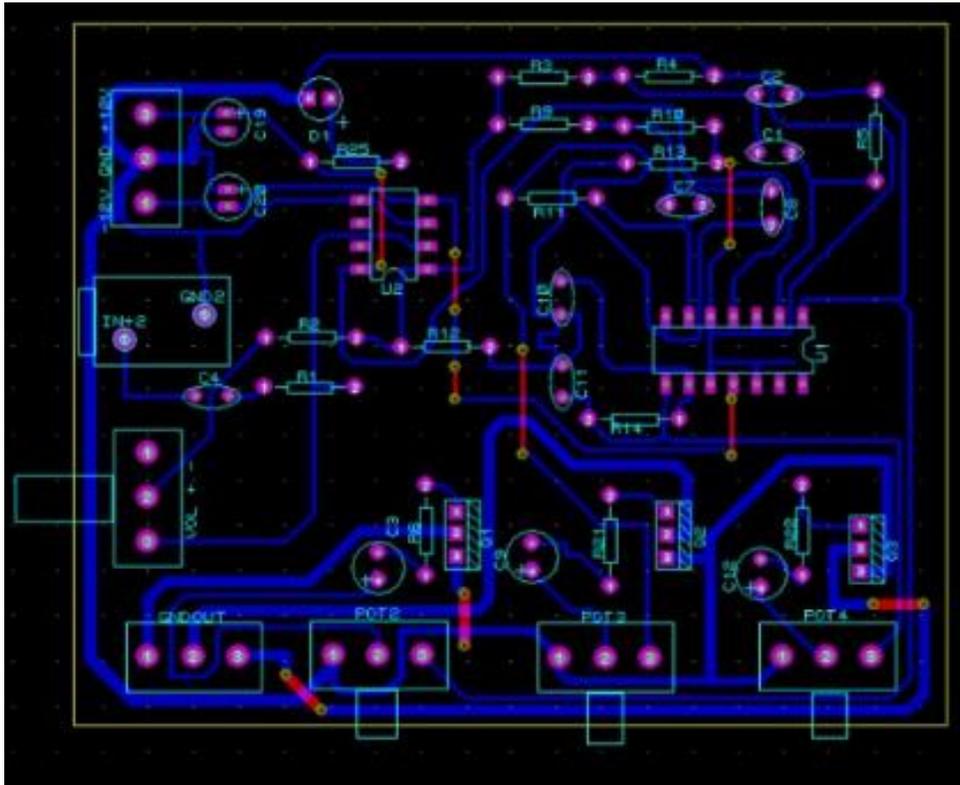


- Circuito del Sensor flujo

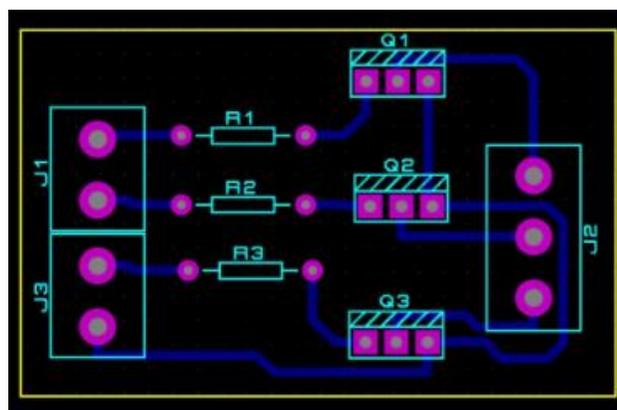


DIAGRAMAS PCB

- Circuito impreso filtros activos



- Circuito impreso de potencia



SOLUCIÓN DE PROBLEMAS

P: ¿Por qué el sistema no se enciende?

R: Asegúrese de que la fuente de alimentación estén correctamente conectada a la toma de 110V.

P: ¿Por qué el sistema no sensa los datos de nivel predeterminados?

R: Asegúrese que sensor de nivel este en una posición intermedia y no esté fuera del rango de la cantidad de agua.

P: ¿Por qué los datos no se actualizan?

R: Apague y encienda nuevamente el switch de encendido para reinicializar el sistema.

P: ¿Por qué en el monitor serie muestra datos erróneos al encender?

R: Apague y encienda nuevamente el switch de encendido para reinicializar el sistema.

P: ¿Por qué el motor sumergible no funciona?

R: Asegúrese de que los cables de alimentación del sistema estén conectado a la toma de 110V.

CONTACTO

Para más información sobre garantía, mantenimiento o consejos técnicos contactarse con:

Ing. Fabio Enríquez.

Celular: 09 95070581

Email: fabioenriquezv@hotmail.com

CRONOGRAMA DE ACTIVIDADES
SISTEMA AUTOMATICO PARA EL CONTROL HIDRÁULICO E ILUMINACIÓN DE LA CASCADA DECORATIVA DEL SANTUARIO CORAZON DE JESÚS MEDIANTE
UNA APLICACIÓN WEB

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo												
					tri 3, 2018				tri 4, 2018				tri 1, 2019			
					jun	jul	ago	sep	oct	nov	dic	ene	feb			
1		Diseño e implementacion del sistema	180 días	vie 1/6/18												
2		Aprobacion de plan pic	30 días	vie 1/6/18												
3		Analisis de herramientas para la aplicación web	15 días	dom 15/7/18												
4		Analisis de componentes electronicos y hadware	3 días	jue 2/8/18												
5		Desarrollo de la aplicación y base de datos	60 días	mar 7/8/18												
6		Implementacion de la aplicación en un servidor local	25 días	lun 29/10/18												
7		Desarrollo de software para Raspberry en Payton	15 días	vie 30/11/18												
8		Desarrollo de circuitos de filtros analógicos	10 días	vie 30/11/18												
9		Implementación de la aplicación en un servidor web	5 días	jue 20/12/18												
10		Pruebas de prototipo completo	3 días	jue 27/12/18												
11		Análisis y resultados de prototipo	8 días	mié 2/1/19												
12		Pruebas de campo	5 días	vie 11/1/19												
13		Implementacion definitiva	15 días	vie 18/1/19												

DATASHEET

LM258, LM358, LM358A, LM358E, LM2904, LM2904A, LM2904E, LM2904V, NCV2904

Single Supply Dual Operational Amplifiers

Utilizing the circuit designs perfected for Quad Operational Amplifiers, these dual operational amplifiers feature low power drain, a common mode input voltage range extending to ground/ V_{EE} , and single supply or split supply operation. The LM358 series is equivalent to one-half of an LM324.

These amplifiers have several distinct advantages over standard operational amplifier types in single supply applications. They can operate at supply voltages as low as 3.0 V or as high as 32 V, with quiescent currents about one-fifth of those associated with the MC1741 (on a per amplifier basis). The common mode input range includes the negative supply, thereby eliminating the necessity for external biasing components in many applications. The output voltage range also includes the negative power supply voltage.

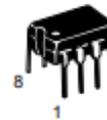
Features

- Short Circuit Protected Outputs
- True Differential Input Stage
- Single Supply Operation: 3.0 V to 32 V
- Low Input Bias Currents
- Internally Compensated
- Common Mode Range Extends to Negative Supply
- Single and Split Supply Operation
- ESD Clamps on the Inputs Increase Ruggedness of the Device without Affecting Operation
- NCV Prefix for Automotive and Other Applications Requiring Unique Site and Control Change Requirements; AEC-Q100 Qualified and PPAP Capable
- These Devices are Pb-Free, Halogen Free/BFR Free and are RoHS Compliant



ON Semiconductor®

www.onsemi.com



PDIP-8
N, AN, VN SUFFIX
CASE 626

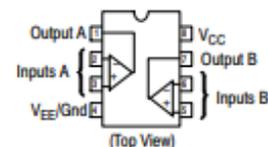


SOIC-8
D, VD SUFFIX
CASE 751



Micro8™
DMR2 SUFFIX
CASE 846A

PIN CONNECTIONS



ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 10 of this data sheet.

LM258, LM358, LM358A, LM358E, LM2904, LM2904A, LM2904E, LM2904V, NCV2904



Figure 1.

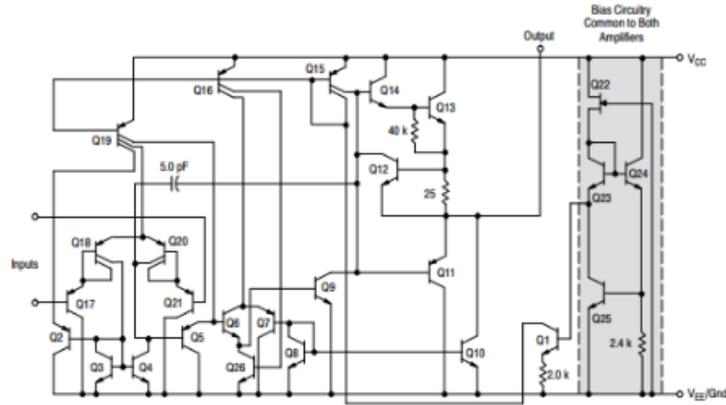


Figure 2. Representative Schematic Diagram (One-Half of Circuit Shown)

LM258, LM358, LM358A, LM358E, LM2904, LM2904A, LM2904E, LM2904V, NCV2904

MAXIMUM RATINGS (T_A = +25°C, unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltages Single Supply Split Supplies	V _{CC} V _{CC} , V _{EE}	32 ±16	Vdc
Input Differential Voltage Range (Note 1)	V _{IDR}	±32	Vdc
Input Common Mode Voltage Range	V _{ICR}	-0.3 to 32	Vdc
Output Short Circuit Duration	I _{SC}	Continuous	
Junction Temperature	T _J	150	°C
Thermal Resistance, Junction-to-Air (Note 2)	R _{θJA}	Case 846A 238 Case 751 212 Case 626 161	°C/W
Storage Temperature Range	T _{stg}	-65 to +150	°C
Operating Ambient Temperature Range	T _A	LM258 -25 to +85 LM358, LM358A, LM358E 0 to +70 LM2904, LM2904A, LM2904E -40 to +105 LM2904V, NCV2904 (Note 3) -40 to +125	°C

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

- Split Power Supplies.
- All R_{θJA} measurements made on evaluation board with 1 oz. copper traces of minimum pad size. All device outputs were active.
- NCV2904 is qualified for automotive use.

ESD RATINGS

Rating	HBM	MM	Unit
ESD Protection at any Pin (Human Body Model - HBM, Machine Model - MM)			
NCV2904 (Note 3)	2000	200	V
LM358E, LM2904E	2000	200	V
LM358DG/DR2G, LM2904DG/DR2G	250	100	V
All Other Devices	2000	200	V

LMx24-N, LM2902-N Low-Power, Quad-Operational Amplifiers

1 Features

- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain 100 dB
- Wide Bandwidth (Unity Gain) 1 MHz (Temperature Compensated)
- Wide Power Supply Range:
 - Single Supply 3 V to 32 V
 - or Dual Supplies ± 1.5 V to ± 16 V
- Very Low Supply Current Drain (700 μ A) —Essentially Independent of Supply Voltage
- Low Input Biasing Current 45 nA (Temperature Compensated)
- Low Input Offset Voltage 2 mV and Offset Current: 5 nA
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Large Output Voltage Swing 0 V to $V^+ - 1.5$ V
- **Advantages:**
 - Eliminates Need for Dual Supplies
 - Four Internally Compensated Op Amps in a Single Package
 - Allows Direct Sensing Near GND and V_{OUT} also Goes to GND
 - Compatible With All Forms of Logic
 - Power Drain Suitable for Battery Operation
 - In the Linear Mode the Input Common-Mode Voltage Range Includes Ground and the Output Voltage
 - Can Swing to Ground, Even Though Operated from Only a Single Power Supply Voltage
 - Unity Gain Cross Frequency is Temperature Compensated
 - Input Bias Current is Also Temperature Compensated

2 Applications

- Transducer Amplifiers
- DC Gain Blocks
- Conventional Op Amp Circuits

3 Description

The LM124-N series consists of four independent, high-gain, internally frequency compensated operational amplifiers designed to operate from a single power supply over a wide range of voltages. Operation from split-power supplies is also possible and the low-power supply current drain is independent of the magnitude of the power supply voltage.

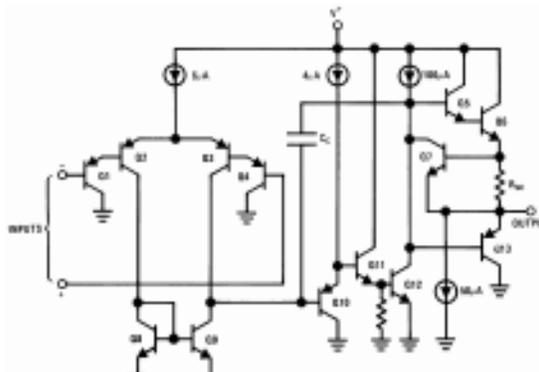
Application areas include transducer amplifiers, DC gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM124-N series can directly operate off of the standard 5-V power supply voltage which is used in digital systems and easily provides the required interface electronics without requiring the additional ± 15 V power supplies.

Device Information⁽¹⁾

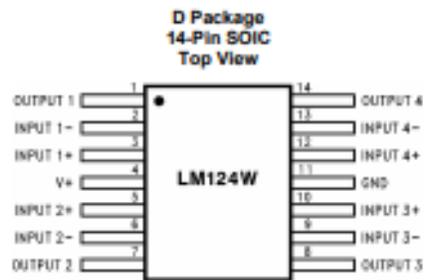
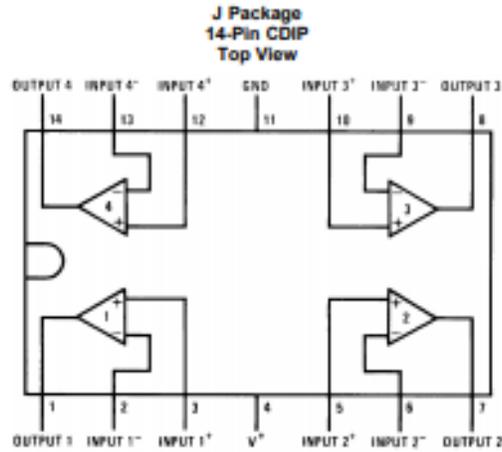
PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM124-N	CDIP (14)	19.56 mm × 6.67 mm
LM224-N		
LM324-N	CDIP (14)	19.56 mm × 6.67 mm
	PDIP (14)	19.177 mm × 6.35 mm
	SOIC (14)	8.65 mm × 3.91 mm
	TSSOP (14)	5.00 mm × 4.40 mm
LM2902-N	PDIP (14)	19.177 mm × 6.35 mm
	SOIC (14)	8.65 mm × 3.91 mm
	TSSOP (14)	5.00 mm × 4.40 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Schematic Diagram



5 Pin Configuration and Functions



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
OUTPUT1	1	O	Output, Channel 1
INPUT1-	2	I	Inverting Input, Channel 1
INPUT1+	3	I	Noninverting Input, Channel 1
V+	4	P	Positive Supply Voltage
INPUT2+	5	I	Nonverting Input, Channel 2
INPUT2-	6	I	Inverting Input, Channel 2
OUTPUT2	7	O	Output, Channel 2
OUTPUT3	8	O	Output, Channel 3
INPUT3-	9	I	Inverting Input, Channel 3
INPUT3+	10	I	Noninverting Input, Channel 3
GND	11	P	Ground or Negative Supply Voltage
INPUT4+	12	I	Noninverting Input, Channel 4

TIRA LED RGB

Integrating sphere test report

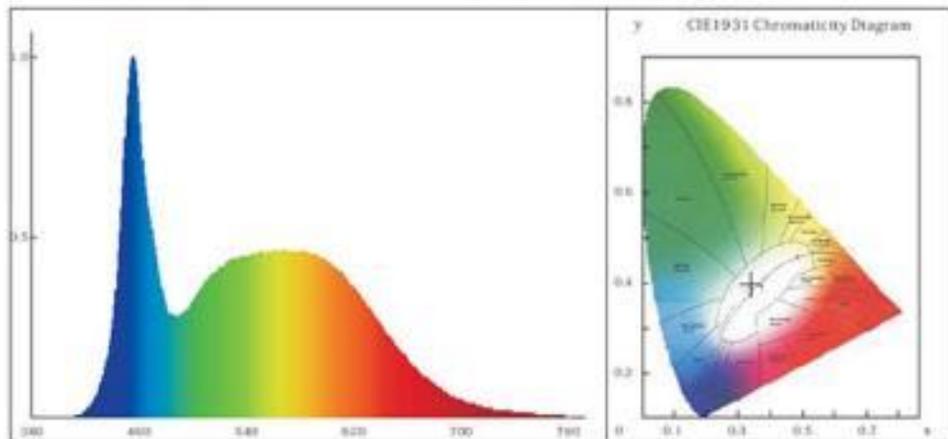
HPLED Optical&Electrical Fast Meters Test Report

LED Test Report

Product Mark

Product Type: LC-BB5050W60
Temperature: Wang℃
Operator: Wang

Manufacturer: LED Color
Humidity: 65%
Test Date: 2015-06-08



Chroma Parameters

Chro. Coor.: $x = 0.3159$ $y = 0.3292$ $u = 0.2000$ $v = 0.3126$ $duv = -0.0017$
CCT: $T_c = 6331K$ Dominant Wave.: 489.7nm Purity: 6.2%
R ratio: $R = 14.2$ Peak Wave.: 452.6nm Half Width: 25.3nm

Rending Index $R_a = 85.4$

$R1 = 84$	$R2 = 92$	$R3 = 94$	$R4 = 82$	$R5 = 84$
$R6 = 87$	$R7 = 88$	$R8 = 72$	$R9 = 20$	$R10 = 80$
$R11 = 63$	$R12 = 41$	$R13 = 69$	$R14 = 91$	$R15 = 63$

Photo Parameters

Flux: 1280.0lm Effi.: 97.15lm/W Radiant: 3467.3mW Iv: 0.0mcd

Ele. Parameters

Forward Current: $I_f = 1098mA$ Forward Voltage: $V_f = 12V$
Reverse Voltage: $V_r = 5.00V$ Reverse Current: $I_r = 0.00\mu A$

Instrument state

IntgeTime: 40.287ms VPeak: 3475 VDark: 559
Scan Range: 380-780nm

TIP122 Y TIP127

TIP120, TIP121, TIP122 (NPN); TIP125, TIP126, TIP127 (PNP)

Plastic Medium-Power Complementary Silicon Transistors

Designed for general-purpose amplifier and low-speed switching applications.

Features

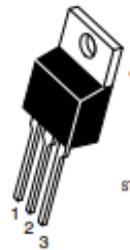
- High DC Current Gain –
 $h_{FE} = 2500$ (Typ) @ $I_C = 4.0$ Adc
- Collector–Emitter Sustaining Voltage – @ 100 mAdc
 $V_{CEO(sus)} = 60$ Vdc (Min) – TIP120, TIP125
 $= 80$ Vdc (Min) – TIP121, TIP126
 $= 100$ Vdc (Min) – TIP122, TIP127
- Low Collector–Emitter Saturation Voltage –
 $V_{CE(sat)} = 2.0$ Vdc (Max) @ $I_C = 3.0$ Adc
 $= 4.0$ Vdc (Max) @ $I_C = 5.0$ Adc
- Monolithic Construction with Built-In Base–Emitter Shunt Resistors
- Pb–Free Packages are Available*



ON Semiconductor®

www.onsemi.com

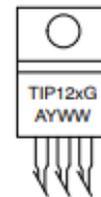
**DARLINGTON
5 AMPERE
COMPLEMENTARY SILICON
POWER TRANSISTORS
60–80–100 VOLTS, 65 WATTS**



TO-220AB
CASE 221A
STYLE 1

STYLE 1:
PIN 1. BASE
2. COLLECTOR
3. EMITTER
4. COLLECTOR

MARKING DIAGRAM



TIP12x = Device Code
x = 0, 1, 2, 5, 6, or 7
A = Assembly Location
Y = Year
WW = Work Week
G = Pb–Free Package

ORDERING INFORMATION

See detailed ordering and shipping information on page 3 of this data sheet.

TIP120, TIP121, TIP122 (NPN); TIP125, TIP126, TIP127 (PNP)

MAXIMUM RATINGS

Rating	Symbol	TIP120, TIP125	TIP121, TIP126	TIP122, TIP127	Unit
Collector-Emitter Voltage	V_{CEO}	60	80	100	Vdc
Collector-Base Voltage	V_{CB}	60	80	100	Vdc
Emitter-Base Voltage	V_{EB}	5.0			Vdc
Collector Current - Continuous - Peak	I_C	5.0 8.0			Adc
Base Current	I_B	120			mAdc
Total Power Dissipation @ $T_C = 25^\circ\text{C}$ Derate above 25°C	P_D	65 0.52			W W/ $^\circ\text{C}$
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	2.0 0.016			W W/ $^\circ\text{C}$
Unclamped Inductive Load Energy (Note 1)	E	50			mJ
Operating and Storage Junction, Temperature Range	T_J, T_{stg}	-65 to +150			$^\circ\text{C}$

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction-to-Case	$R_{\theta JC}$	1.92	$^\circ\text{C}/\text{W}$
Thermal Resistance, Junction-to-Ambient	$R_{\theta JA}$	62.5	$^\circ\text{C}/\text{W}$

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

1. $I_C = 1\text{ A}$, $L = 100\text{ mH}$, P.R.F. = 10 Hz, $V_{CC} = 20\text{ V}$, $R_{BE} = 100\ \Omega$

ELECTRICAL CHARACTERISTICS ($T_C = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
----------------	--------	-----	-----	------

OFF CHARACTERISTICS

Collector-Emitter Sustaining Voltage (Note 2) ($I_C = 100\text{ mAdc}$, $I_B = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	$V_{CEO(sus)}$	60 80 100	- - -	Vdc
Collector Cutoff Current ($V_{CE} = 30\text{ Vdc}$, $I_B = 0$) ($V_{CE} = 40\text{ Vdc}$, $I_B = 0$) ($V_{CE} = 50\text{ Vdc}$, $I_B = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	I_{CEO}	- - -	0.5 0.5 0.5	mAdc
Collector Cutoff Current ($V_{CB} = 60\text{ Vdc}$, $I_E = 0$) ($V_{CB} = 80\text{ Vdc}$, $I_E = 0$) ($V_{CB} = 100\text{ Vdc}$, $I_E = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	I_{CBO}	- - -	0.2 0.2 0.2	mAdc
Emitter Cutoff Current ($V_{BE} = 5.0\text{ Vdc}$, $I_C = 0$)		I_{EBO}	-	2.0	mAdc

TIP120, TIP121, TIP122 (NPN); TIP125, TIP126, TIP127 (PNP)

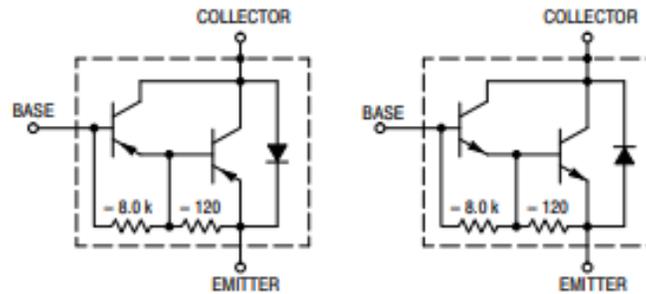


Figure 1. Darlington Circuit Schematic

ORDERING INFORMATION

Device	Package	Shipping
TIP120	TO-220	50 Units / Rail
TIP120G	TO-220 (Pb-Free)	50 Units / Rail
TIP121	TO-220	50 Units / Rail
TIP121G	TO-220 (Pb-Free)	50 Units / Rail
TIP122	TO-220	50 Units / Rail
TIP122G	TO-220 (Pb-Free)	50 Units / Rail
TIP125	TO-220	50 Units / Rail
TIP125G	TO-220 (Pb-Free)	50 Units / Rail
TIP126	TO-220	50 Units / Rail
TIP126G	TO-220 (Pb-Free)	50 Units / Rail
TIP127	TO-220	50 Units / Rail
TIP127G	TO-220 (Pb-Free)	50 Units / Rail

SENSOR DE FLUJO

Flow-Pulse	Flow-Curve	Connection method
Flow Range: 100L/H/1800H-L/H		
Flow (L/H)	Freqz (HZ)	Error range ± 10
120L/H	16	
240L/H	32.5	
360L/H	49.3	
480L/H	65.5	
600L/H	82	
720L/H	90.2	

Connection method:

YIFA the plastics Ltd
Product Introduction

- Model: YF-21
- Product Name: Hall sensor
- Flow Range: 1-30L/MIN
- (1) Connection Method

- (2) Voltage Range: 3.5-24VDC, Pulse Characteristic: $F=7Q(L/MIN)$
- (3) Extent of error: $\pm 5\%$
- (4) Flow-Pulse
 - 2L/MIN=16HZ
 - 4L/MIN=32.5HZ
 - 6L/MIN=49.3HZ
 - 8L/MIN=65.5HZ
 - 10L/MIN=82HZ

5. Bom

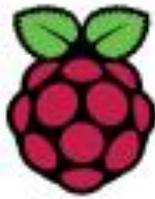
No.	Item	Material	Qty
1	Connection wire		1
2	Bonnet	PA	1
3	Screw		4
4	Valve body	PA	1
5	Leak press valve		1
6	Magnet		1
7	Hall		1
8	Impeller	POM	1
9	Stainless steel axis	SUS304	1
10			
11			

BOMBA DE AGUA



Technical Specs

Item	Utility Pump	GPM of Water @ 25 Ft. of Head	4
Pump HP	1/6	Height	7-3/4"
Voltage - Pumps	120VAC	Diameter	6-1/8"
Switch Type - Pumps	No Switch Included	Base Material	Nylon
Cord Length - Pumps	18 ft.	Impeller Material - Pumps	Nylon
Discharge NPT	1" F	Length	6-1/16"
Max. Head - Pumps	26 ft.	Width	4-5/16"
Pump Body Material	Aluminum	Max. Liquid Temp.	120 Degrees F
GPM of Water @ 5 Ft. of Head	18	Shaft Seal Material - Pumps	Viton
GPM of Water @ 10 Ft. of Head	17	Includes	3/4" Garden Hose Adapter
GPM of Water @ 15 Ft. of Head	14	Item - Dewatering Pumps	Utility/Dewatering Pump
GPM of Water @ 20 Ft. of Head	9		



Raspberry Pi 3 Model B+



Overview



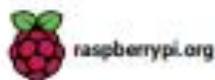
The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT

The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.

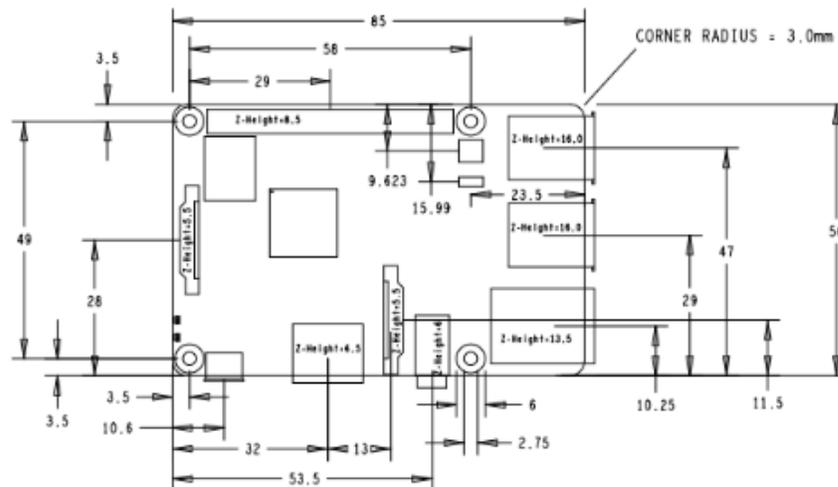
The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

Specifications

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	<ul style="list-style-type: none">■ 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE■ Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	<ul style="list-style-type: none">■ 1 × full size HDMI■ MIPI DSI display port■ MIPI CSI camera port■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	<ul style="list-style-type: none">■ 5V/2.5A DC via micro USB connector■ 5V DC via GPIO header■ Power over Ethernet (PoE)-enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50 °C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.



Physical specifications



Warnings

- This product should only be connected to an external power supply rated at 5V/2.5A DC. Any external power supply used with the Raspberry Pi 3 Model B+ shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.
- Whilst in use, this product should be placed on a stable, flat, non-conductive surface and should not be contacted by conductive items.
- The connection of incompatible devices to the GPIO connection may affect compliance, result in damage to the unit, and invalidate the warranty.
- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors, and mice when used in conjunction with the Raspberry Pi.
- The cables and connectors of all peripherals used with this product must have adequate insulation so that relevant safety requirements are met.

Safety instructions

To avoid malfunction of or damage to this product, please observe the following:

- Do not expose to water or moisture, or place on a conductive surface whilst in operation.
- Do not expose to heat from any source; the Raspberry Pi 3 Model B+ is designed for reliable operation at normal ambient temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Whilst it is powered, avoid handling the printed circuit board, or only handle it by the edges to minimise the risk of electrostatic discharge damage.

FOTOS DE ENTREGA DEL PROYECTO



