Optimización del abastecimiento de agua potable por bombeo eléctrico mediante una aplicación IoT.

Pazmiño Pérez, José Andrés

Departamento de Eléctrica y Electrónica

Carrera de Tecnología en Electrónica mención Instrumentación & Aviónica

Monografía, previo a la obtención del título de Tecnólogo en Electrónica mención Instrumentación y Aviónica

Ing. Calvopiña Osorio, Jenny Paola

Latacunga, 22 de Junio del 2021

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE TECNOLOGÍA EN ELECTRÓNICA MENCIÓN

INSTRUMENTACIÓN Y AVIÓNICA

Certificación

Certifico que la monografía, **"Optimización del abastecimiento de agua potable por bombeo eléctrico mediante una aplicación IoT"** fue realizado por el señor **Pazmiño Pérez, José Andrés**, la cual ha sido revisada y analizada en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 22 de junio del 2021

Firmado electrónicamente por:
JENNY PAOLA CALVOPINA OSORIO

Ing. Calvopiña Osorio, Jenny Paola
C.C.: 0503390239

Reporte de verificación

Curiginal

Document Information

Analyzed document	Proyecto_Pazmiño.pdf (D109319148)
Submitted	6/20/2021 12:19:00 AM
Submitted by	
Submitter email	japazmino10@espe.edu.ec
Similarity	1%
Analysis address	jpcalvopina1.espe@analysis.urkund.com

Sources included in the report

W	URL: https://www.mcielectronics.cl/shop/product/arduino-yun-rev2-25677 Fetched: 6/20/2021 12:20:00 AM	1
W	URL: https://diotronic.com/placas/15491-arduino-yun-2-atmega32u4 Fetched: 6/20/2021 12:20:00 AM	1
W	URL: https://sites.google.com/site/rafaelprogramacion Fetched: 11/21/2019 4:27:05 AM	1

Firmado electrónicamente por:
JENNY PAOLA CALVOPINA OSORIO

Ing. Calvopiña Osorio, Jenny Paola
C.C.: 0503390239

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE TECNOLOGÍA EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

Responsabilidad de autoría

Yo, **Pazmiño Pérez, José Andrés,** con cédula de ciudanía **N° 1600466799**, declaro que este trabajo de titulación **"Optimización del abastecimiento de agua potable por bombeo eléctrico mediante una aplicación IoT"** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 22 de Junio del 2021

Pazmiño Pérez, José Andrés

C.C.: 1600466799

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE TECNOLOGÍA EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

Autorización de publicación

Yo, **Pazmiño Pérez, José Andrés**, con cédula de ciudadanía **N° 1600466799,** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el presente trabajo de titulación **"Optimización del abastecimiento de agua potable por bombeo eléctrico mediante una aplicación IoT"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad

Latacunga, 22 de Junio del 2021

Pazmiño Pérez, José Andrés

C.C.: 1600466799

Dedicatoria

A mi madre, hermanas y familiares que con esfuerzo me apoyaron y guiaron durante todo el transcurso de mi carrera universitaria.

Pazmiño Pérez, José Andrés

Agradecimiento

A mi madre María Pérez, hermanas Andrea y Ángela, y a mi cuñado Alejandro quienes de corazón puedo expresar mi gratitud y con quienes de por vida estaré en deuda por apoyarme y guiarme en todo momento, persistiendo en mi formación universitaria.

Además, agradezco a todas las personas que fueron parte de mi formación académica pertenecientes a la Universidad de las Fuerzas Armadas y que impartieron sus conocimientos logrando ser asertivos con ellos.

Gracias a la Ingeniera Paola Calvopiña por estar siempre presta a las necesidades manifestadas durante el desarrollo del proyecto y que con su guía fue posible su culminación.

Pazmiño Pérez, José Andrés

Tabla de contenidos

Índice de tablas

Índice de figuras

Resumen

En el presente escrito se detalla el proceso de optimización del abastecimiento de agua potable por bombeo eléctrico. Para lo cual se evaluaron los elementos necesarios para el correcto funcionamiento del sistema, simplificándolo en los bloques de interfaz, control y potencia. Determinando el bloque de interfaz, como la creación de la aplicación para el smartphone en Blynk, el cual tiene las opciones manual/automático necesarias para la prueba y funcionamiento del sistema. En el bloque de control, se puede encontrar la parte de configuración y programación, en el cual se detallan las líneas más importantes utilizadas para el correcto funcionamiento del sistema, además de los pasos para establecer la conexión a internet de la placa de Arduino YUN. Y finalmente, en el bloque de potencia se ubican los circuitos de alimentación y acople, donde se detalla el diseño de la fuente de alimentación para el microcontrolador el cual trabaja a un voltaje línea regulado de 5VDC, mientras que para el acople entre la parte de control y potencia se establece el uso de un circuito optoacoplador. Por otra parte, en el bloque se determinan también las conexiones de los elementos de campo en conjunto con la placa de Arduino.

Palabras clave:

- **ARDUINO**
- **BLYNK**
- **OPTOACOPLADOR**

Abstract

This document details the optimization process of the potable water supply by electric pumping. Which the necessary elements for the correct operation of the system were evaluated, simplifying it in the interface, control and power blocks. The interface block is determined such as the smartphone application development on Blynk, which has the manual/automatic options necessary for the testing and operation of the system. On the control block, the configuration and programming are found, which the used most important lines for the correct operation of the system are detailed, as well as the steps to establish the Arduino YUN board internet connection. Finally, power block details the power supply for the microcontroller which works at 5VDC regulated voltage, while for the coupling between the control and power part an opto-isolator circuit is used. Besides, the block determines the connections of the field elements in conjunction with the Arduino board.

Keywords:

- **ARDUINO**
- **BLYNK**
- **OPTO-ISOLATOR**

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

Tema: Optimización del abastecimiento de agua potable por bombeo eléctrico mediante una aplicación IoT.

Capítulo I

1. Generalidades

1.1. Antecedentes

En tiempos remotos, el control y monitoreo de una variable física, ocurrirían de manera empírica mediante procesos que además de conllevar mucho tiempo y energía, suponían un riesgo en la vida del ser humano debido a la práctica y al contacto directo con el proceso mismo, proceso que significaba la utilización de elementos o máquinas puramente mecánicas. Posteriormente, con el avance de la tecnología, se abrió la oportunidad de unir la electricidad con la mecánica, lo que representó en el control y monitoreo de procesos, seguridad y facilidad de operación. Desde entonces, la electrónica ha llegado a formar parte de este grupo de componentes industriales gracias a elementos de potencia que permiten su acoplamiento, lo cual simbolizó una gran oportunidad de digitalización y manejo del mismo como cualquier dato informático.

Actualmente, dicho procesamiento de datos ha dado origen en conjunto con el servicio más utilizado hoy en día llamado internet a las famosas aplicaciones IoT, las cuales hacen uso de un servicio de red, denominada nube, para funcionar de manera remota e inalámbrica, siendo operadas por usuarios desde cualquier dispositivo con acceso a internet.

Por tal razón, el ser humano al sentir que existe un déficit de algún servicio o con el fin de facilitar la utilización de alguno de ellos, se propone utilizar sistemas

inteligentes, tal como lo menciona Edward Stiven Rodríguez Moreno y Víctor Felipe López Ordoñez de la Universidad Francisco José de Caldas, en su proyecto de graduación del año 2017, titulado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE PARA UN EDIFICIO MEDIANTE IOT UTILIZANDO EL PROTOCOLO DE COMUNICACIÓN LORAWAN”, en el mismo que además determinaron que los sistemas inteligentes son mucho más efectivos, y garantizan mayor eficiencia al permitir ser monitoreados y controlados remotamente (Rodríguez & López, 2017).

Mientras que por otro lado, en el proyecto de titulación denominado “El Internet de las Cosas y las consideraciones de seguridad” presentado en la Pontificia Universidad Católica del Ecuador en el año del 2015 por parte de Fabián Geovanny Cuzme Rodríguez, se evaluó y determinó el impacto positivo que conlleva la utilización de la tecnología IoT, debido a su versatilidad de aplicación, haciendo hincapié en la seguridad de los datos, como un factor primordial para implementar dicha tecnología (Cuzme Rodríguez, 2015).

Como se ha demostrado, la investigación e implementación de la tecnología IoT está en auge debido al rendimiento del mismo para mejorar la calidad de vida.

1.2. Planteamiento del problema

Toda civilización se sustenta en torno a los servicios básicos, donde el agua potable es uno de ellos, significando su ausencia o escasez una complicación en cuestión de salud e higiene. Lo que, debido a la gran cantidad de viviendas a la cual se somete el sistema de suministro de agua potable en conjunto con la particularidad geográfica de la zona Sierra, afecta directamente el abastecimiento del servicio, mismo que se puede ver reflejado en el desequilibrio de presión de agua existente en las viviendas de las diferentes localidades.

En la ciudad de Ambato, específicamente en la calle E casa 107 del conjunto habitacional Valle Hermoso III se ha manifestado la necesidad de mejorar el abastecimiento de dicho servicio, ya que diariamente esta complicación afecta en las actividades que se realizan de higiene y alimentación de las personas residentes, limitando su consumo en ciertos periodos de tiempo y obstaculizando el transcurso normal de las demás labores.

De no implementarse este proyecto se mantendrá el malestar y falta de confort durante la práctica de las actividades domésticas mencionadas anteriormente, las mismas que son esenciales para el ser humano, y que en los tiempos de pandemia en los que vivimos actualmente son de vital importancia.

Es por tal razón, que se plantea el tema “Optimización del abastecimiento de agua potable por bombeo eléctrico mediante una aplicación IoT”, que además de mejorar el suministro de agua, se pueda hacer uso y monitoreo del proceso de manera segura mediante una aplicación sin la necesidad de que el usuario se encuentre en contacto directo con el proceso. Además de ello, existe la posibilidad de programar el funcionamiento por intervalos de tiempo, ya que durante ciertos periodos es irrelevante la utilización de la bomba, esto con la finalidad de ahorrar el consumo de energía eléctrica y alargar la vida útil de la bomba la cual es el elemento más susceptible a daños.

1.3. Justificación

El propósito del presente proyecto se basa principalmente en la comodidad y eficiencia que se generará durante el curso de las actividades domésticas y de higiene personal, las cuales son fundamentales especialmente para la salud y el desarrollo de los residentes, y posibles familiares o visitantes que ingresen en la propiedad.

En la actualidad, la tecnología IoT, es la forma más viable y simple para controlar dispositivos o equipos del hogar de manera segura y efectiva desde la facilidad de un Smart Phone, Tablet o un ordenador. Es por tal razón que para solucionar el déficit del abastecimiento de agua de manera factible, se plantea utilizar una bomba eléctrica y una tarjeta electrónica de Arduino YUN, la cual al permitir la conexión a una red Wi-Fi, brinda la oportunidad de agregar funcionalidades conforme surgen necesidades en la residencia a futuro, lo cual se resume en un precio accesible y/o inversión económica para los residentes como familia.

1.4. Objetivos

1.4.1. Objetivo General

- Optimizar el abastecimiento de agua potable por bombeo eléctrico mediante una aplicación IoT.

1.4.2. Objetivos Específicos

- Determinar el modo de funcionamiento para el sistema de abastecimiento de agua potable mediante la investigación de las demandas en el domicilio.
- Controlar la activación de la bomba para el abastecimiento de agua mediante Arduino.
- Desarrollar la interfaz mediante el gestor de aplicaciones IoT Blynk para la interacción entre el usuario y la electrobomba.
- Comprobar la correcta operación del sistema implementado tanto en software como en hardware mediante pruebas de funcionamiento.

1.5. Alcance

El proyecto se realizará en la residencia 104 de la calle D del conjunto habitacional Valle Hermoso III, localizado en la parroquia de Macasto, Ambato –

Ecuador.

Mismo que será desarrollado mediante una placa electrónica de Arduino YUN, que controlará el accionamiento de la bomba. Por lo que, el sketch será realizado mediante el software de Arduino IDE. Además, para facilidad de los usuarios se realizará una aplicación IoT mediante un gestor llamado Blynk, el cual presentará visualmente el estado de la bomba con el respectivo switch.

Capítulo II

2. Marco Teórico

2.1. Microcontroladores

2.1.1. Introducción

A continuación se expondrán generalidades que determinan a un microcontrolador, como son la definición, composición y arquitecturas de diseño.

Definición

Se define a microcontrolador como un microchip compacto que incluye en su interior los mismos elementos que un microcomputador, creado principalmente para el trabajo de comunicación y control. (Mandado Pérez, Menéndez Fuertes, Fernandez Ferreira, & López Matos, 2007)

Composición

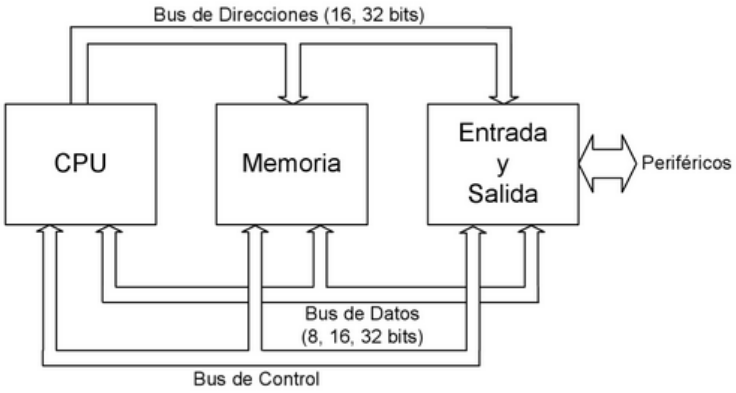
Los elementos componentes de un microcontrolador se los puede generalizar en tres grandes bloques, tales como:

- CPU
- Memoria
- Entradas/Salidas

Los mismos que mediante buses interactúan para realizar las instrucciones o tareas que han sido programadas. A continuación en la figura 1, se puede visualizar representados los tres grandes bloques que definen al microcontrolador.

Figura 1.

Esquema básico de un microcontrolador



Nota. Recuperado de (Mandado Pérez, Menéndez Fuertes, Fernandez Ferreira, & López Matos, 2007).

En donde el CPU es el componente encargado de hacer funcionar los periféricos de salida en base al programa almacenado y a las entradas, ya que en dicho programa se definen las instrucciones que deben ser realizadas por el microcontrolador dependiendo de los datos ingresados.

Por otra parte, en el bloque de memoria podemos encontrar, la memoria de programa y la memoria de instrucciones; definiendo así que, la memoria de programa es aquella encargada de almacenar las instrucciones a las cuales va acceder el CPU dependiendo de lo que el usuario haya escrito en ella. Y, la memoria de instrucciones, es el almacenamiento en donde se encuentran todas las instrucciones posibles, capaz de realizar el microcontrolador.

Finalmente, en el bloque de entradas/salidas, se definen a los componentes con los cuales se interactúa con el exterior, ya que son los que permiten percibir o evaluar que dicho microcontrolador esté funcionando de acuerdo a lo establecido por el usuario, englobando en este bloque a componentes de comunicación (serie/paralelo), temporizadores, entradas/salidas tanto digitales como analógicas.

Arquitecturas

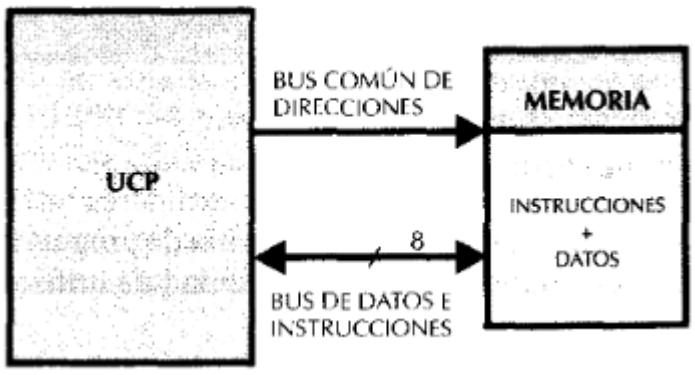
Para la fabricación de los microcontroladores, se han utilizado los dos diferentes modelos, Von Neumann y Harvard.

Estos modelos de arquitectura detallan la interacción o comunicación de los componentes de memoria de programa e instrucciones en conjunto con el CPU, definiendo así su velocidad y eficiencia.

Así pues, en la arquitectura de Von Neumann se puede determinar que los buses de datos, y dirección o instrucciones comparten el mismo medio físico, mientras que en la arquitectura de Harvard se puede determinar que dichos buses utilizan medios diferentes, permitiendo el funcionamiento en paralelismo, tal y como se muestran en las figuras 2 y 3 respectivamente. Lo cual se refleja en el rendimiento ya mencionado (Angulo Usategui & Angulo Martinez, 2003)

Figura 2.

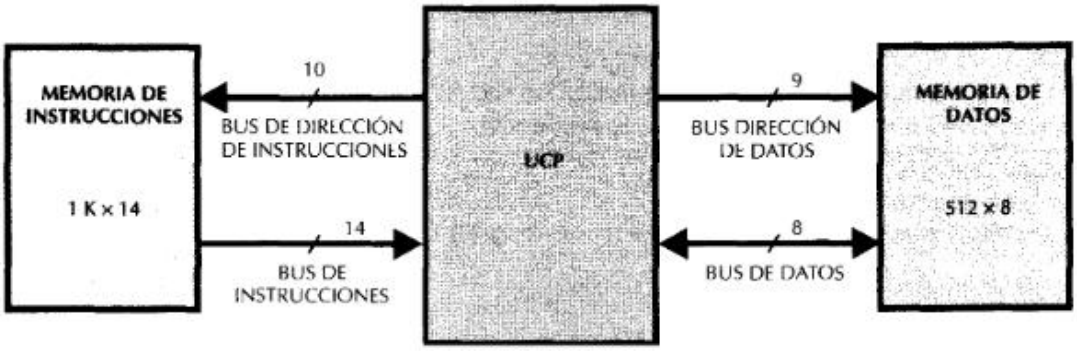
Esquema básico de arquitectura Von Neumann



Nota. Recuperado de (Angulo Usategui & Angulo Martinez, 2003).

Figura 3.

Esquema básico de arquitectura Harvard



Nota. Recuperado de (Angulo Usategui & Angulo Martinez, 2003).

En la actualidad, podemos encontrar microcontroladores con microprocesadores de tipo CISC o RISC, lo cual mejora el rendimiento del microprocesador.

2.1.2. Principio de funcionamiento de microcontroladores

Como se mencionó en el punto anterior, las arquitecturas son quienes definen como se encuentran físicamente diseñados los componentes, principalmente de memoria y como estos interactúan con el CPU.

Pero lo que determina la diferencia entre ellas es la finalidad con la cual fueron diseñadas, y es esa finalidad la que también define el tipo de memoria que será utilizada.

Para lo cual a continuación se definirán ambas memorias, además del tipo que se utilizan comúnmente.

Memoria de instrucciones

Esta memoria solo se lee al inicio de la ejecución del programa, y es por esta razón, que utilizan memorias de acceso aleatorio no volátil que son denominadas o

más conocidas por el inglés como ROM (Read Only Memory), ya que en su interior contiene la esencia o lenguaje ensamblador capaz de interpretar las instrucciones programadas por el usuario, información que es proporcionada por el fabricante y el mismo que no debe ser modificado. Además al especializarse en solo lectura, el tiempo con el que accede a sus datos es mucho menor que el otro tipo de memoria.

Memoria de datos

Por otra parte esta memoria, es la encargada de funcionar mientras se esté ejecutando el programa, lo cual significa que durante su actuación se realizará el proceso de escritura/lectura simultáneamente. Por lo que, a este tipo de memorias se les denomina memorias de acceso aleatorio volátiles o activas, o conocidas también como RAM (Random Access Memory), ya que la información que retienen es sólo mientras el microcontrolador se encuentra funcionando.

2.1.3. Aplicaciones

Los microcontroladores pueden existir de todo tipo, desde la más básica hasta una muy compleja, lo cual se ve reflejado en la capacidad de realizar una mayor cantidad de tareas o aplicaciones, en mayor o menor cantidad de tiempo, siendo una más efectiva que otra.

Pero entre las aplicaciones que se les puede emplear estos microcontroladores son automoción, equipos médicos, industriales, de comunicación, instrumentos electrónicos, entre otros.

Y es la variedad de microcontroladores existente en el mercado, que ha permitido que los consumidores podamos elegir dependiendo de la necesidad, ya que así como existen microcontroladores muy complejos, o creados específicamente para una tarea en concreto, el costo es el que refleja dicho detalle.

2.2. Arduino

2.2.1. Hardware & Software

Se puede definir al conjunto de hardware y software de Arduino como una plataforma open-source, determinando así que cualquier persona puede acceder y hacer uso de la documentación y diagramas esquemáticos de dichas placas de desarrollo. Además, en lo referente a software, permite hacer uso de herramientas para programar el microcontrolador de dichas placas (Arduino, 2018)

Hardware

En lo referente al hardware de Arduino, toda placa de desarrollo está compuesto fundamentalmente por un microcontrolador, y pines de entrada y salida digitales y analógicas. Los cuales además de ello tienen circuitos de protección, y algunos incluso reguladores de voltaje para su utilización mediante baterías, pilas, u otras fuentes portátiles.

El microcontrolador encargado de procesar la información, por lo general es de la familia AVR perteneciente a Atmel, que en la actualidad forma parte de Microchip Technology.

Dentro del gran conjunto que abarca Arduino se pueden diferenciar las placas de otras categorías como shields, kits, accesorios y módulos, que en sí son un conjunto de sensores y actuadores, y/o placas capaces de realizar el acople con un entorno de Ethernet, Wifi, Bluetooth, Radiofrecuencia, entre otras tecnologías que permiten la comunicación, con otros componentes tecnológicos comunes en el diario vivir, todo ello con el respectivo acondicionados para la utilización en conjunto con una placa de desarrollo basada en un microcontrolador (Peña, Arduino - De cero a experto, 2017)

A continuación se enlistan algunos de los ejemplares de placas de desarrollo más comunes en el mercado.

- Arduino UNO
- Arduino NANO
- Arduino LEONARDO
- Arduino MEGA

Software

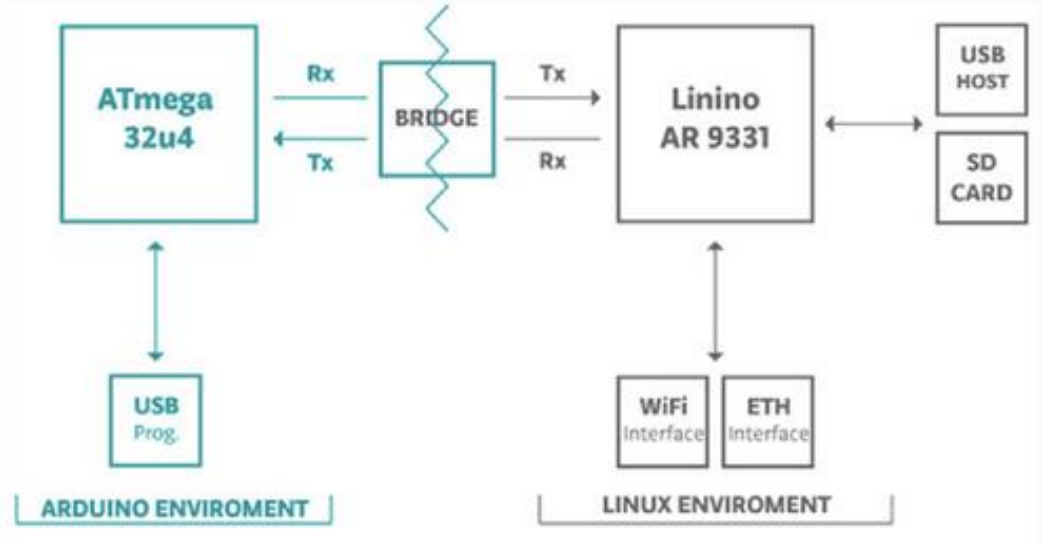
El software al igual que el hardware es una de las partes importantes que abarca la plataforma de Arduino, ya que este establece las instrucciones y parámetros para que la placa funcione de acuerdo a lo deseado. Para lo cual se utiliza un conjunto de herramientas llamado IDE, el cual se encuentra desarrollado en una sola aplicación distribuida por los creadores en la página oficial de Arduino. (Peña, Arduino - De cero a experto, 2017)

2.2.2. Características de Arduino YUN

Arduino YUN, es una de la gran cantidad de placas desarrolladas por Arduino, pero de la gran minoría que permite el acceso a internet, y con ella el entorno que brinda para su utilización para aplicaciones IoT. Esta característica cualidad, se debe a que la placa además de funcionar con el microcontrolador común, ATMega32U4, contiene un microprocesador adicional, Atheros AR9331, el cual se basa en Linino/Linux, destinado específicamente para la interfaz de red, que al igual que Arduino es una plataforma Open-Source. Y la comunicación de dichos elementos se lo realiza mediante la herramienta BRIDGE, la misma que se puede encontrar en el gestor de librerías de Arduino IDE. Así pues, en la figura 4 se representa dicha interacción mediante un diagrama de bloques (Kuhmel, 2015)

Figura 4

Representación gráfica de comunicación entre microcontrolador y microprocesador.



***Nota*:** BRIDGE es una librería basada en API REST (Application Programming Interface) la cual es una herramienta que habilita interconectar diferentes servicios, y basa su funcionamiento en *request/response*. Recuperado de (Kuhmel, 2015).

Además, a continuación se presentan las especificaciones técnicas del microcontrolador y microprocesador que componen Arduino YUN, en la Tabla N° 1 y 2 respectivamente.

Arduino Microcontrolador AVR

Tabla 1

Especificaciones de Microcontrolador ATmega32U4

Microcontrolador	**ATmega32U4**
Voltaje de Operación	5V
Voltaje de Entrada	5V
Pines digitales I/O	20

Microcontrolador	**ATmega32U4**
Salidas PWM	7
Pines analógicos I/O	12
Corriente DC por pines I/O	40mA en pines I/O; 50mA en pin 3,3V
Memoria Flash	32KB (4KB utilizados para arranque)
SRAM	2,5KB
EEPROM	1KB
Frecuencia	16MHz

Nota. Información recuperada de la hoja de especificaciones de Arduino YUN REV 2

Microprocesador Linux

Tabla 2

Especificaciones de Microprocesador Atheros AR9331

Microprocesador	**Atheros AR9331**
Arquitectura	MIPS
Voltaje de Operación	3,3V
Ethernet	802,3 10/100Mbit/s
WiFi	802,11b/g/n 2,4GHz
Tipo USB	2.0 Host
Lector de tarjeta	Micro-SD
RAM	64MB DDR2
Memoria Flash	16MB
Frecuencia	400MB

Nota. Información recuperada de la hoja de datos de Arduino YUN REV 2

2.2.3. Arquitectura del microcontrolador ATMEGA32U4-AUR

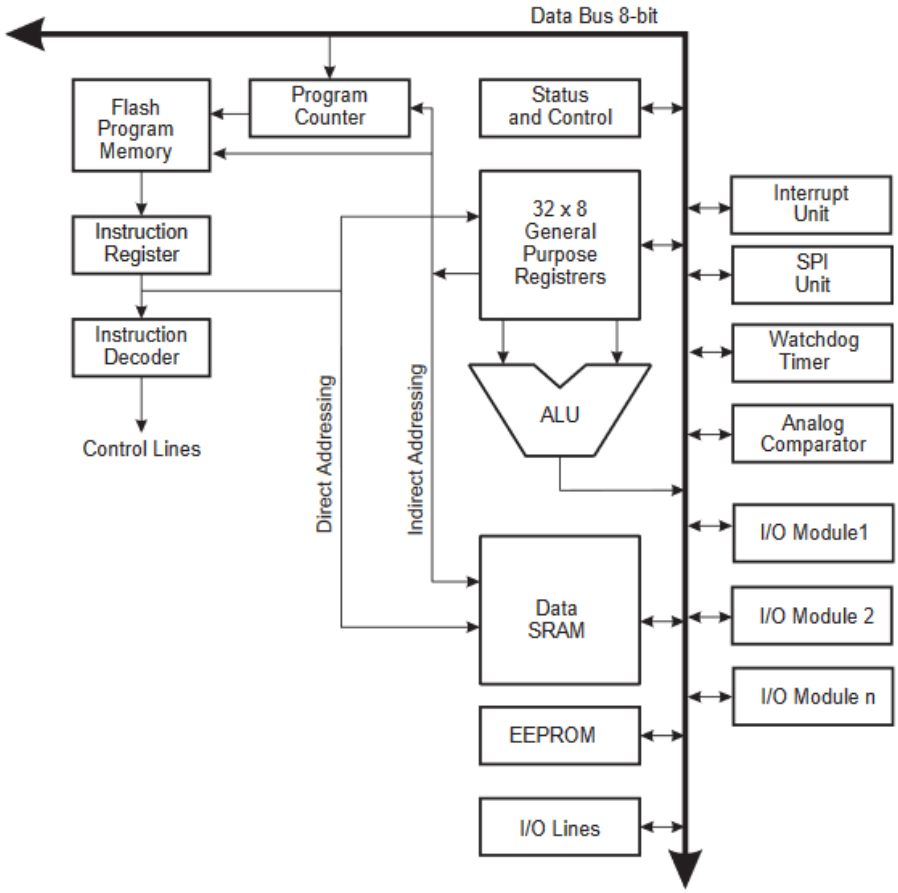
Como se ha explicado anteriormente la mayoría de microcontroladores modernos basan su funcionamiento en la arquitectura Harvard, y esta no es la excepción, que para maximizar el rendimiento y permitir el paralelismo, el microcontrolador utiliza memorias y buses separados para instrucciones y datos, como se puede visualizar en la figura 5. Así, las instrucciones del programa se ejecutan con una canalización de un solo nivel, lo que significa que línea tras línea de instrucción del programa se va efectuando en cada ciclo de reloj.

Cuando se refiere a ciclo de reloj, se está expresando que el microcontrolador funciona de acuerdo al tiempo establecido por los componentes que lo conforman, ya que en la placa de desarrollo de Arduino, existe un oscilador el mismo que brinda la frecuencia a la que funciona eléctricamente dicho componente, y eso por tal razón que el ciclo de reloj establecido, no es la unidad de tiempo comúnmente conocida como el segundo, sino el tiempo que determina los 16MHz marcados en el oscilador.

Además, el espacio de la memoria Flash se divide en dos secciones, una de arranque y otra de aplicación. Ambas secciones con bits de bloqueo dedicado para la protección de escritura y escritura/lectura. Explicando así el consumo de memoria en el arranque como se detalla en la tabla de especificaciones.

Figura 5.

Diagrama de bloques de la arquitectura de ATMega32U4.



***Nota*.** Recuperado de ATMega32U4 Datasheet, ubicado en la página oficial de Arduino.

2.3. Introducción de Arduino IDE

Se define Arduino IDE por sus siglas que significan Entorno de Desarrollo Integrado, el mismo que en esencia es una aplicación que permite escribir y cargar instrucciones mediante líneas de programación en un microcontrolador integrado en una placa de desarrollo de Arduino e incluso otras tarjetas de proveedores externos los cuales sean compatibles (Peña, Arduino IDE: configuración y uso, 2020)

Esta herramienta en conjunto con la placa permite desarrollar un sinnúmero de aplicaciones de manera sencilla y rápida para los usuarios.

El procedimiento que realiza el software al presionar “Subir" en su interfaz, es el de transformar de lenguaje C al lenguaje que es entendido por el microcontrolador, lo cual por lo general es lo más complicado de lograr al programar un microcontrolador, y esto se puede realizar fácilmente gracias al compilador AVR-GCC que tiene incorporado en su estructura (Massimo, 2008)

Por otro lado, como se mencionó al inicio del punto 2 del presente capítulo, este software es Open-Source, lo que significa que es libre en licencia y distribución gratuita. Además, otra ventaja del software es que es multiplataforma, es decir, se lo puede utilizar en diferentes sistemas operativos, tales como Windows, Linux, Mac, entre otros.

El software de Arduino IDE, utiliza una programación basada en C++, el cual es un lenguaje de programación estructurado y de alto nivel, es decir, “fácil de programar”, en comparación con el lenguaje de máquina, el cual es un lenguaje de bajo nivel y es el mismo lenguaje en el que desarrolla el componente microprogramable.

2.3.1. Introducción Lenguaje C

La ambición del ser humano por manipular una gran cantidad de información al tratar de entender el funcionamiento del mundo en general, ha formado en su pensamiento la necesidad de buscar y crear métodos para dicho objetivo que en conjunto con el desarrollo de la tecnología ha dado origen al estudio de lo que en la actualidad conocemos como informática o computación, y con ella el canal de comunicación para que el ser humano sea capaz de entenderse con su creación.

Como se describió en puntos anteriores, Arduino en conjunto con su herramienta de programación Arduino IDE basan su funcionamiento en programas embebidos los que mismos que tienen como propósito el de leer información, procesarla, y operar las salidas como el usuario lo requiera. Siendo los pilares fundamentales del procedimiento en general al iniciar a programar, ya que de la misma manera se procede a realizar la estructura de un programa basado en lenguaje C o su versión mejorada C++

Definimos lenguaje C o C++, como el lenguaje entendido por ordenador, tarea que es posible gracias a los entorno de desarrollo que han creado con la finalidad de facilitar a programadores el desarrollo de software y, siendo nombrado así debido a su predecesor Lenguaje B.

Estos entornos de desarrollo son los encargados de compilar, depurar y cargar el programa, diferenciando que se refiere a programación orientada a programación de dispositivos electrónicos. A diferencia de software creado para ordenador, es que en estos programas realizados en los nombrados ‘entornos de desarrollo’ se nos permite realizar un ejecutable capaz de instalar en cualquier ordenador como un archivo instalable o portable (Didact, 2005)

Algoritmo

Todo programa se basa en la resolución de un problema que por lo general se encuentra en la vida real. Y es el ser humano, que ha creado la tecnología que a través de la misma se solucionen, pero está en la capacidad de la persona programadora la de realizar o mentalizar los pasos necesarios para realizar la tarea de la manera más eficiente y con los mínimos recursos posibles.

A estos pasos o procedimientos, es lo que se debe estudiar o desarrollar para comprender el funcionamiento de un programa de computadora, ya que esta es la esencia que lo hace trabajar, convirtiéndose así este término como la esencia misma de la informática (Sedgewick, 1992)

Es a partir de algoritmos simples que se originan procesos completamente más complejos, capaces de utilizar operaciones matemáticos muy complejos, para solucionar problemas que en la vida real con el esfuerzo humano se lo realizaría de manera muy sencilla, pero que para que una maquina lo asimile y realice, necesita de un proceso mucho más complejo.

Esto se debe a que en el mundo real todo dato que se debe tomar en consideración para su resolución, es analógico, lo cual transformarlo y utilizarlo en un sistema digital es complicado, transportando a un nivel superior el razonamiento del ser humano.

Lenguaje de bajo nivel

Podemos definir en esta categoría a los lenguajes que por sus características se acercan más a la arquitectura del ordenador.

Entre los lenguajes que podemos encontrar en este nivel son: el **Lenguaje Máquina** y el **Lenguaje Ensamblador**.

Para la resolución de un problema, en el mundo real, se sigue un conjunto de pasos el cual lo entendemos y plantearnos en nuestro lenguaje natural. De la misma manera, en un ordenador o un microcomputador, para poder solucionar un problema debemos darle la secuencia ordenada de pasos a realizar, la misma que debe ser entendida y planteada en un lenguaje que entienda dicho elemento electrónico. Así, podemos determinar que el **Lenguaje Máquina** es el que entiende directamente

nuestros ordenadores, y en lo que concierne y puede entender una persona común y corriente, es un conjunto o una combinación de ‘ceros y unos’ denominados bits. Además, esta combinación de bits puede ser leída exclusivamente por un solo tipo de procesador, es decir que si su composición física varía, dicha combinación de ‘ceros y unos’ significará otra instrucción o tarea (Peña Basurto & Cela Espín, 2000)

Mientras que, **Lenguaje Ensamblador** se define como la evolución del Lenguaje Máquina el cual se basa en mnemotécnicos, refiriéndose así a las abreviaturas de las palabras que indican nombres de instrucciones.

Para la utilización de este lenguaje, es necesario de un vasto conocimiento de la estructura y funcionamiento interno del ordenador, además del dominio de diferentes sistemas de numeración, como el binario, hexadecimal, octal, etc.

Y la única ventaja que tiene frente al lenguaje de alto nivel es que utiliza menos memoria, pero que como se ha dicho anteriormente, se utilizaría solo para ordenadores con el mismo procesador, refiriéndose a nivel de ordenador, la cual es una característica un poco irrelevante en la utilización de microcontroladores, en donde el entorno de desarrollo se encarga de ‘subir’ el programa en **Lenguaje Máquina** las instrucciones escritas.

Lenguaje de alto nivel

Este tipo de lenguaje de programación es mucho más avanzado que el anteriormente descrito, ya que se asemeja más al lenguaje natural del programador, y entre los lenguajes más conocidos podemos encontrar:

- FORTRAN
- BASIC
- Pascal

- Modula
- C
- C++
- Ada
- Java

Pero lo que más caracteriza a este lenguaje es la desventaja que tienen los lenguajes de bajo nivel, el cual es su facilidad de portar y utilización genérica frente a ordenadores de diferente procesador.

Características

En la actualidad, en el mercado existe una numerosa cantidad de software utilizado para programar dispositivos electrónicos que utilizan por lo general lenguajes de alto nivel, tal es el caso de Labview manejado en dispositivos de National Instrument, que utiliza un lenguaje orientado a objetos, el cual es un poco más avanzado, pero que basa sus tareas en un conjunto de códigos de C.

Pero a pesar de esto, la mayoría de programadores se inclina más por la utilización de C++, ya que entre sus virtudes podemos encontrar:

- Potencia
- Flexibilidad
- Popularidad
- Portabilidad
- Sencillez
- Estructura

2.3.2. Variables

La concepción de variable define a toda la información posible dentro de un

sistema lógico, tal como es el caso de su manejo en procesos matemáticos, el cual de la misma manera puede compararse a procesos informáticos.

En del campo de la informática, como se conoce y define a este concepto; variable es el contenedor de un dato en específico, el mismo que puede estar sujeto a cambios, o simplemente el nombre que recibe una constante, pero que dentro de un programa informático, es utilizado para su manipulación y beneficio del funcionamiento de dicho programa.

Como se ha expresado anteriormente, todo dato pertenece a una variable, las mismas que son definidas por el programador, y que de la misma manera podemos discernir que no todos los datos encontrados en un sistema informático son del mismo tipo. Dicho esto se puede determinar que los tipos de datos fundamentales recaen en los grandes grupos de carácter y numeral, siendo el de tipo numeral el que contiene una variedad de tipos de variables. Esto se debe a que existen rangos en su límite infinito de posibilidades, lo cual en un sistema digital no se ha podido fabricar un procesador y memoria capaz de soportar dicha infinidad (Liberty & Horvath, 2001)

Es por tal razón que a continuación se mostrarán todos los tipos posibles de variables, con su tamaño y rango de valores, que se pueden encontrar en el mundo de la informática.

Tabla 3

Tipos de variable soportado por Arduino IDE

TIPO	TAMAÑO	VALORES
Bool	1 byte	Verdadero o falso
Unsigned short int	2 bytes	0 a 65535
Short int	2 bytes	-32768 a 32767

TIPO	TAMAÑO	VALORES
Unsigned long int	4 bytes	0 a 4294967295
Long int	4 bytes	-2147483648 a 2147483647
Int	2 bytes	Enteros de -32768 a 32767
Unsigned int	2 bytes	Enteros de 0 a 65535
Char	1 byte	256 valores, -128 a 127, de tipo caracter
Float	4 bytes	-1,2e-38 a 3,4e38
Double	8 bytes	-2,2e-308 a 1,8e308
Word	2 bytes	Mismos valores que '*unigned int*'
Unsigned char	1 byte	256 valores, 0 a 255, de tipo caracter
Byte	1 byte	Enteros de 0 a 255

Nota. Recuperado de (Liberty & Horvath, 2001)

Asignar un tipo y nombre a una variable se conceptualiza en lo que se conoce como declaración de variable. Dicha acción dependiendo del lugar en el que se lo realice determina otro tipo de variable, en otro tipo de clasificación que no recae en la naturaleza del dato, más bien en la estructuración del programa, además de su utilización. Por lo que a continuación se definen los dos tipos posibles y su posicionamiento.

Globales

Las variables globales se definen y ubican después de declarar las librerías a las que tendrá acceso y podrá hacer uso el programa, y se declaran las variables en dicha posición ya que todo el programa podrá hacer uso de dichas variables.

Locales

Por otro lado, se definen variables locales a las que son específicas para una subrutina, o también denominado subprograma, ubicándolas dentro de dichas partes del programa en general. Y la diferencia de las globales, ya que si el programa se está ejecutando fuera de la subrutina, no se puede tener acceso y hacer uso de dicha variable. Por lo general se definen de tipo local para evitar la creación de diversas variables, además de los recursos de memoria que utilizará el programa durante su ejecución.

Operadores

Cuando nos referimos a sistemas lógicos, en realidad de lo que estamos hablando es de la utilización de las matemáticas, las mismas que son aplicadas y dan origen a dichos sistemas, por lo que la esencia es la manera en la que utilizamos expresiones matemáticas para que nuestro programa realice las tareas deseadas. (Didact, 2005)

Por tal razón, las expresiones matemáticas que significan el pilar fundamental de un programa necesitan de:

Operadores Lógicos

Estos operadores tienen como objetivo unir condiciones simples que mediante operadores relacionales se crean expresiones más complejas, para determinar si dicha condición establecida cumple los resultados booleano, es decir verdadero o falso.

Por tal razón en la siguiente tabla se expresan algunos operadores con su significado y su equivalente en lenguaje C++

Tabla 4

Operadores Lógicos.

OPERADOR	SIGNIFICADO	C++
Y	Conjunción	'AND' o '&&'
O	Disyunción	'OR' o '||'
NO	Negación	'NOT' o '!'

Nota. Recuperado de (Didact, 2005)

Operadores Aritméticos

Como su nombre lo expresa, los operadores aritméticos permiten la formación de operaciones aritméticas, y que se puede aplicar a constantes, variables y expresiones y, de la misma manera a continuación se presentan los operadores aritméticos enlistados en una tabla.

Tabla 5

Operadores Aritméticos

OPERADOR/C++	SIGNIFICADO
+	Suma
-	Resta
*****	Multiplicación
/	División
%	Residuo de división

Nota. Recuperado de (Didact, 2005)

Operadores Relacionales

Mencionado anteriormente en los Operadores Lógicos, los operadores relacionales son utilizados para crear expresiones lógicos simples, que permiten comparar entre dos variables, que por lo general se utilizan entre variables tipo int.

Tabla 6

Operadores Aritméticos

OPERADOR/ C++	SIGNIFICADO
>	Mayor que
>=	Mayor igual que
<	Menor que
<=	Menor igual que
== o =	Igual que
!= o <>	Distinto que

Nota. Los dos últimos operadores relacionales encontrados en la tabla 6, se pueden expresar de dos maneras diferentes. Esto se puede explicar debido a que, dependiendo de la utilización varía su forma, es decir si se utiliza en una condición (?) se utiliza la primera forma, mientras que cuando es una proposición se utiliza la segunda forma. Recuperado de (Didact, 2005)

2.3.3. Estructura de un sketch

Todo programa, que en este caso a los programas desarrollados en el software de Arduino IDE se los denomina sketch, cumplen con una estructura establecida, que permite que un programa se desarrolle y funcione correctamente de acuerdo a la arquitectura del procesador del ordenador o microcontrolador.

Por otra parte, al basarse un sketch en lenguaje C++, podemos definir que el funcionamiento de los programas, se basa en la lectura y procesamiento línea a línea, es decir, cada instrucción descrita en el sketch, se ejecuta una tras otra, desde el inicio al final.

Es por tal razón, que por lo general para estructurar un programa se lo debe realizar basándose primeramente en lo que conocemos como diagrama de flujo, el cual se lo debe realizar posteriormente al algoritmo correspondiente.

Así tenemos que las partes que componen el sketch básicamente son:

1. **Instrucciones directivas.-** Establecido de tal manera para el conjunto de librerías que tendrá acceso y hará uso el programa o sketch durante su ejecución.
2. **Comentarios.-** Al final de cada línea o entre líneas de programación se pueden redactar comentarios que sirvan de guía o que brinde información útil a la persona que programa. Sin que intervenga en la ejecución o funcionamiento del programa.
3. **Declaración de variables.-** como se definió en el punto anterior, se determina así a la acción de especificar el nombre y tipo de datos que ocuparan un espacio de almacenamiento en la memoria del ordenador o microcontrolador, que se utilizará en la ejecución o funcionamiento del programa.
4. **Funciones o subrutinas.-** son los subprogramas que se ejecutarán, dentro del programa principal, que por lo general llevan tareas más simples, y que pueden ayudar a la simplificación de sistemas mayores.
5. **Programa principal.-** es el programa que se está ejecutando constantemente por el ordenador, y que enfoca toda la atención de los componentes del mismo, para que su funcionamiento cumpla con el objetivo establecido por el usuario.

2.3.4. Funciones

El funcionamiento de los programas desarrollados en lenguaje C++, utilizan instrucciones que se encuentran definidas previamente en la estructura básica del ordenador o microcontrolador, del cual podemos hacer uso, a estas instrucciones también se las conoce como palabras reservadas, y que por lo general al escribirlas en el entorno de desarrollo, cambian de color o se ponen en negrillas, dependiendo del software utilizado, son herramientas que tienen como objetivo simplificar líneas de programación que aumentarían la utilización de la memoria de programa (Didact, 2005)

En el caso de Arduino IDE, las palabras reservadas determinadas para lo que conocemos como funciones cambian a un color verde, y que por su importancia, a continuación se describirán las funciones que podemos encontrar en un programa.

IF-ELSE

Función utilizada para realizar toma de decisiones, la cual en todo programa basado en C++ se expresa su sintaxis de la siguiente manera:

```
If (Condición)
{
// Expresión si cumple la función;
}
Else
{
// Expresión si no cumple la función;
}
```

Para realizar decisiones dentro de decisiones se puede escribir ElseIf y posteriormente realizar la condición siguiente, hasta terminar con el proceso de toma de decisión.

FOR

Función utilizada principalmente para realizar un número específico de bucles definiendo el inicio y fin del proceso, la cual en todo programa basado en C++ se expresa su sintaxis de la siguiente manera:

```
For (i=inicio; i<=fin; i++)
{
//Sentencia a repetirse definidamente
}
```

En el ejemplo planteado 'i' es una variable de tipo int declarado previamente, que indica el inicio y fin del bucle. Bucle que repetirá la sentencia definida en su interior. Además se puede apreciar la sentencia 'i++' que define un contador ascendente en unidad.

***Nota*:** Generalmente al programar en lenguaje C++, se utilizan las letras i, j, k para la realización de contadores.

WHILE

Función utilizada principalmente para realizar bucles indefinidos, mientras se cumpla la condición establecida.

```
While (Condición)
{
//Sentencia a repetirse indefinidamente
}
```

SWITCH: CASE: BREAK

Función utilizada principalmente para realizar menús, en el que existe una variable de selección, y que cambia de acuerdo al operador del programa, la cual en todo programa basado en C++ se expresa su sintaxis de la siguiente manera:

Switch (*Variable de selección*)

{

Case *'dato de variable de selección'***:**

//Sentencia o subprograma a ejecutarse al ser seleccionado

Break;

...

Default:

//Sentencia o subprograma a ejecutarse en el caso de no seleccionar opción

Break;

}

***Nota*:** En la mayoría de programas en los que se utiliza la función *'switch-case'* el dato de variable de selección es un dato de tipo *'int'*, que empieza de 0 hasta el número de límite de casos a programar, pero también se puede utilizar datos de diferente tipo, el cual se debe definir en la interfaz para el operador.

2.3.5. Compilador

Un entorno de desarrollo de alto nivel es un software basado en un conjunto de herramientas que permiten el proceso de programación, y es por esta razón que una de ellas y que primordialmente utiliza este software es el compilador, que en el caso de

la mayoría de software utilizado para la programación de microcontroladores se basa en GCC. Lo cual reitera que es un software libre y distribución gratuita.

Y, como se ha expresado en puntos anteriores, el objetivo de un compilador es la de transformar un lenguaje de alto nivel a su equivalente en lenguaje máquina o también llamado código objeto. Así podemos definir que las actividades que realiza un compilador y determina dicho proceso son las de leer, analizar y traducir en fases separadas, cada una de las instrucciones descritas en las líneas de programación (Peña Basurto & Cela Espín, 2000)

Por lo general, al proceso de compilar se lo acopla en conjunto con un depurador, el cual se encarga de recopilar dicho programa compilado, ejecutarlo virtualmente, y encontrar fallos o errores en la sintaxis del programa.

2.4. Internet of Things

2.4.1. Introducción

Traducido al español como Internet de las Cosas; en otros casos se lo puede encontrar también como IoE (Internet of Everythig) o inclusive en el campo industrial como IIoE (Industrial Internet of Thing), es una tecnología que con el tiempo ha evolucionado desde la conexión más simple de objetos digitales de la vida cotidiana, hasta grandes sistemas automatizados que toman decisiones para beneficio del usuario sin la intervención directa del mismo. (Quiñonez Muñoz, 2019)

Esto ha sido posible gracias a los avances tecnológicos electrónicos e informáticos, que en conjunto con el conocimiento sobre el procesamiento de datos y la comunicación, dio a luz al internet en primera instancia, el cual permitió la interconexión de las personas alrededor del mundo, y que próximamente con el ingenio y trabajo de algunas personas, se crearon tecnologías capaces de controlar

dispositivos aprovechando este servicio que hoy en día la mayoría de personas consume llamado internet.

Así, se puede definir este término como una red de dispositivos a través de internet, los cuales interactúan mediante sistemas embebidos, redes de comunicación, bases de datos, y/o aplicaciones en la nube. Lo cual en conjunto con otros sistemas, generan una aplicación más inteligente para facilitar el trabajo de los usuarios (Quiñonez Muñoz, 2019)

Tecnología que llevado al campo de la industria forma parte de las bases de la denominada Cuarta Revolución Industrial o también conocida como Industria 4.0, la misma que corresponde a la organización de los medios de producción, y que se fundamenta esencialmente en digitalizar todos los campos que realiza el operador, lo cual en unos pocos años a futuro tendrá un gran impacto en procesos y en la vida misma de todo el mundo.

2.4.2. Importancia

Como se ha podido expresar anteriormente, el Internet de las Cosas se ha producido gracias al avance tecnológico hasta la actualidad, que con el paso del tiempo muchas personas se suman a la utilización de esta tecnología por lo que, a futuro expertos afirman que no tienen claro los cambios que logrará en la sociedad además de la importancia que implicará en el campo de la comunicación, y esto se debe a que esta tecnología incluso se puede aplicar desde objetos con dimensiones nanométricas, que podemos tener en nuestros cuerpos hasta grandes ciudades a miles de distancia con dispositivos de mayor tamaño, los cuales a fin de cuentas son sensores y actuadores que estarán interconectados por la red de internet (Quiñonez Muñoz, 2019)

2.4.3. Aplicaciones

En la actualidad es una de las tecnologías más novedosas que se pueden encontrar en el mercado, dirigido principalmente para personas que conforme a sus necesidades se permiten adquirirlas, ya que como se ha propuesto anteriormente, la mayoría de personas tiene acceso a una red de internet, lo cual es requisito fundamental para el funcionamiento de aplicaciones IoT. Y es por tal razón que hoy en día podemos encontrar este servicio principalmente en el campo de la domótica, siendo muy demandado en el mercado y cumpliendo su objetivo.

Mientras que por otra parte podemos encontrar este servicio también en el campo de la hostelería, que como se puede observar en la distribución organizada en los patios de comida, entregan dispositivos electrónicos que señalan al cliente cuando la orden de comida se encuentre lista.

Además de las grandes expectativas que crea en el campo de la industria, ya sea en el control directo de la maquinaria como en gestión logística de una empresa, ya que con la adquisición de datos constante se puede realizar un mejor seguimiento de la producción en general de las diferentes plantas y crear una planificación capaz de mejorar el rendimiento.

Otros de los campos en los que se puede encontrar son en la agricultura y ganadería, que permiten al operador monitorear los procesos correspondientes a un espacio controlado para la cría de plantas y animales, ya sea en regadíos o temperatura en galpones o invernaderos respectivamente, los cuales mediante software puede controlar dichas variables físicas.

2.5. Electrobombas

2.5.1. Generalidades

En la actualidad se pueden encontrar motores de todo tamaño en cualquier lugar cumpliendo cualquier tipo de aplicación posible, desde los más pequeños que por lo general utilizan licuadoras en el hogar hasta los más grandes apreciados en la industria principalmente.

De la misma manera en aplicaciones para el traslado y/o movimiento de líquidos, sea con o sin sólidos, se utilizan motores a los que se les ha denominado con el nombre de electrobomba o motobomba dependiendo del tipo de motor.

A los motores se los clasifica primordialmente en dos grandes grupos, dependiendo del principio de funcionamiento, teniendo así al grupo que utiliza electricidad y al grupo de motores que utiliza combustible fósil para su funcionamiento, definiendo su diferencia en la transformación de energía, pero ambos con el mismo objetivo; crear movimiento.

En el presente trabajo investigativo, se enfocará en los motores que utilizan electricidad para generar movimiento, y que específicamente se utiliza para el movimiento o transferencia de agua sin sólidos, lo que a su vez define su nombre como electrobomba.

En el mercado, se pueden encontrar diferentes tipos, cada uno diseñado para aplicaciones específicas y denominadas generalmente de la siguiente manera:

Electrobombas Centrifugas.

Son aquellas que pueden encontrarse de tipo vertical u horizontal, y que en conjunto con sensores de nivel, evitan que funcionen en vacío debido a posibles daños del mismo.

En el mercado además de su clasificación por aplicación, se pueden encontrar por alimentación eléctrica, tales como monofásico y trifásico que por lo general pueden llegar hasta 1.1kW y 22kW de potencia respectivamente.

Electrobombas Sumergibles.

Se pueden encontrar de este tipo para aplicaciones generalmente de drenaje y residuos, y otras para pozos profundos.

Este tipo de electrobombas por lo general tiene incorporado consigo un sensor de nivel tipo flotador que se lo debe conectar en serie para que realice su correcto funcionamiento y evite que la electrobomba funcione en vacío y así evitar daños en la misma.

2.5.2. Arranques

Para el accionamiento de motores existen diferentes tipos, ya que dependiendo de la aplicación y del tipo de motor, se deben realizar configuraciones de conexión eléctrica diferentes, lo cual permita que el motor al salir de su estado de reposo y a movimiento no genere daños o sobrecarga en el circuito interior del motor, así podemos encontrar:

Arranque Directo o A plena tensión

Generalmente se utiliza esta configuración en motores monofásicos y para los que su carga previamente estudiada, no sobrecargue al dispositivo inductivo, siendo el más simple, y que no necesita de circuitería externa, más la que de su conexión a la línea de alimentación, el mismo que es accionado por su respectivo circuito de control (Lladonosa, 1996)

Arranque A tensión reducida

Dentro de este tipo de arranque podemos encontrar diferentes configuraciones, las mismas que se exponen a continuación:

- Estrella - triángulo.
- Por resistencias estatóricas.
- Por autotransformador.

Siendo “estrella - triángulo” el método más común encontrado hasta la actualidad y que podemos encontrar en la mayoría de paneles de control en la industria, debido a su fiabilidad, eficiencia y eficacia durante el arranque de motores trifásicos.

Con el tiempo, la utilización de contactores eléctricos para realizar dicha conexión, va siendo reemplazada por “arrancadores suaves”, los cuales realizan la misma función pero con menor cantidad de elementos eléctricos, además de la facilidad que brinda su instalación.

Otra técnica que se utiliza para determinar si realizar o no la conexión mediante estrella - triángulo, es que en la placa de especificaciones del motor, existen los voltajes nominales, y si en dicha placa especifica que el voltaje máximo nominal, coincide con la alimentación de donde se vaya a realizar la conexión del motor, se recomienda este tipo de método.

2.5.3. Circuitos de potencia

Para el diseño e implementación de motores es necesario diferenciar los circuitos de control y de potencia, los mismos que permiten el manejo de estos elementos de forma segura, y que por lo general funcionan en conjunto con diversos actuadores mayores formando así sistemas más grandes.

Para lo cual, la parte de control existen diversos tipos, se por medio de lógica de contactores, como se lo ha utilizado hasta la actualidad mediante pulsadores, o que conforme avanza la tecnología se van acoplando la utilización de microcontroladores, como se ha especificado en puntos anteriores y a lo cual va enfocado este proyecto.

Así, podemos definir a circuitos de control como los elementos que están en contacto directo con el actuador, y que comprenden el mayor consumo de energía eléctrica. Y es por tal razón, que existen acoples para microcontroladores, que por lo general se definen como relés y/o optoacopladores, que en conjunto de elementos de potencia como triacs, SCRs, además de elementos de protección permiten el control de este tipo de cargas.

Elementos de protección

Son determinados de esta forma a los elementos que permiten la desconexión de la carga de la red eléctrica, para evitar que sufran daños, y que de la misma manera evitar accidentes. A continuación se enlistan algunos de los elementos de protección que se pueden encontrar en el mercado.

- Fusibles
- Relés térmicos
- Relés electromagnéticos
- Interruptores magnetotérmicos
- Interruptores diferenciales

Capítulo III

3. Introducción de desarrollo

En el capítulo a continuación se detallará el diseño e implementación del proyecto que, como se puede apreciar en la figura 6, la bomba es la encargada de dar la presión necesaria en el conducto, para que de esa forma el consumo en la residencia no se vea afectado por la falta de la misma. Además de ello, podemos encontrar sensores de campo, los cuales están señalados en la figura 24, y que son utilizados para el control y protección del sistema.

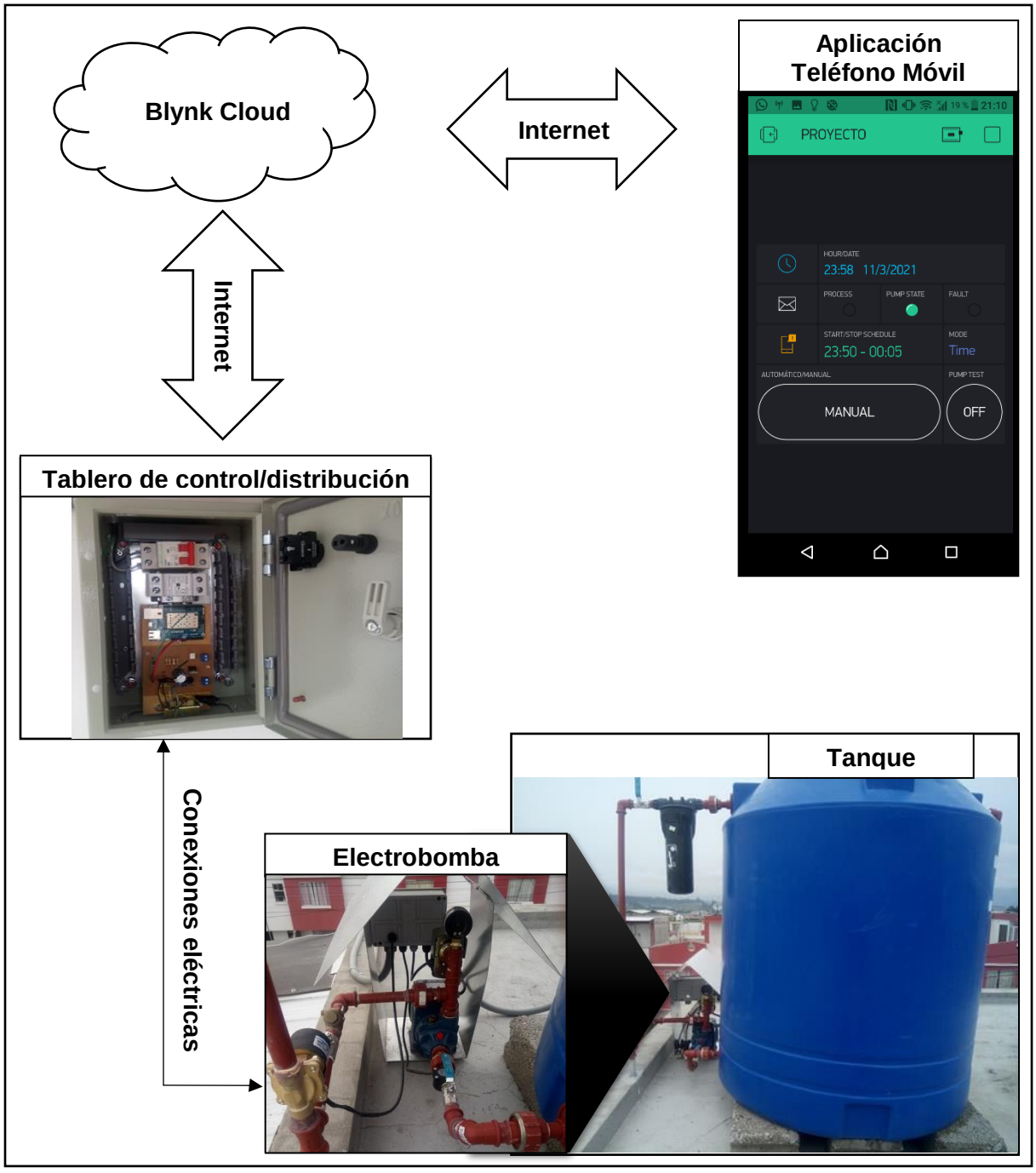
Por otra parte, el control de la bomba se la realiza mediante la placa de Arduino que fue ubicada en la caja de control/distribución, en conjunto con los elementos eléctricos esenciales para la protección del proceso, además de la placa realizada en PCB, la misma que contiene los circuitos necesarios para alimentar la placa de Arduino y su acople con la etapa de potencia.

Finalmente, en la figura 6 mostrada a continuación se puede apreciar un teléfono móvil, el mismo que contiene la interfaz de usuario y permite la manipulación de la bomba mediante un servicio de internet de manera remota, servicio que comúnmente se conoce como IoT.

Para más detalles, en los siguientes puntos a continuación se especificarán los diferentes circuitos y componentes mencionados anteriormente.

Figura 6

Diagrama del proceso.



***Nota*.** La caja ubicada junto a la bomba es una caja utilizada para facilidad de uso y conexión de los elementos de campo, no es equivalente a caja de control/distribución.

3.1. Control

Como se indicó al inicio del capítulo, el control basa su funcionamiento en la placa de desarrollo Arduino YUN Rev 2, que debido a sus características de conexión a internet sin módulos externos fue seleccionado, para lo cual se presentan las líneas de programación y configuración del mismo posteriormente.

3.2. Programación y configuración de Arduino YUN

Programación

A continuación, se exponen figuras que muestran las líneas de programación más relevantes y que forman parte de la estructura del sketch de Arduino, además de una breve descripción de cada una de ellas.

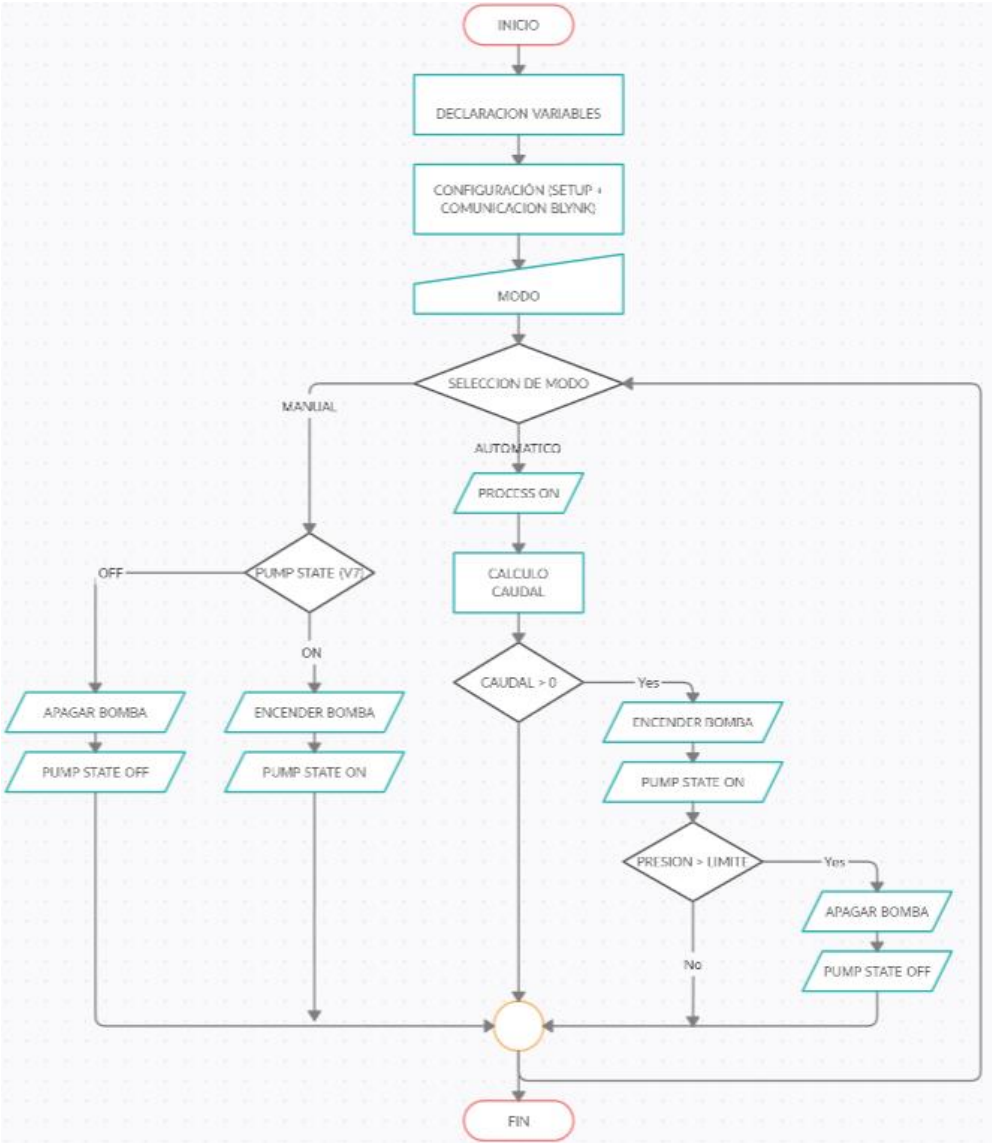
Pero primeramente para iniciar con esta sección, se ha realizado un diagrama de flujo el mismo que lo puede observar en la figura 7.

Este diagrama tiene como finalidad explicar de manera simplificada el proceso de ejecución del sketch de Arduino, el mismo que para cumplir con las funcionalidades que se presentan en la aplicación del teléfono móvil, realiza un sinnúmero de subprocesos que se basan en operaciones lógicas y aritméticas.

Para referenciar y guiar en conjunto con la aplicación, en el diagrama de flujo se han utilizado las etiquetas de identificación de los ‘objetos’ presentes en la interfaz de usuario, tal es el caso de los indicadores que demuestran el funcionamiento de los elementos físicos.

Figura 7.

Diagrama de flujo.



La conexión a una red local propia o a una con acceso a internet, es gracias al microprocesador y tarjeta de red que forman parte de la placa de desarrollo de Arduino YUN, que para hacer uso de este servicio en conjunto con el microcontrolador se deben definir las librerías esenciales para su funcionamiento, como se puede apreciar en la figura 8, las mismas que además se puede adquirir en el gestor de librerías de Arduino IDE.

Figura 8.

Librerías utilizadas en el Sketch.

```
#include <SPI.h>
#include <Ethernet.h>
#include <Bridge.h>
#include <BlynkSimpleYun.h>
#include <TimeLib.h>
#include <WidgetRTC.h>
```

***Nota*.** La tecnología que permite esta relación entre microprocesador y microcontrolador se denomina bridge, la cual desarrolladores han sintetizado y formado paquetes de librerías que en su beneficio les permite mediante plataformas digitales facilitar la creación de aplicaciones/proyectos IoT, tal es el caso de la plataforma utilizada denominada Blynk

Por otra parte además de las librerías necesarias para la conexión de Arduino con la red de internet, encontramos definidas librerías de tiempo, las cuales son “TimeLib.h” y “WidgetRTC.h”, y que son utilizadas durante la ejecución del programa para objetivos afines.

Además de ello, otro punto importante es el enlazamiento de la aplicación en la plataforma con el dispositivo de Arduino, para lo cual se define el identificativo de la conexión como se muestra en la figura 9.

Figura 9.

Declaración de Auth - Token.

```
char auth[] = "Le2XCQZlyDqUnBuyI6IZ3QiK3iT9Yd4z";
```

***Nota*:** Esta declaración de la variable auth[] de tipo char es el identificativo que genera la aplicación de Blynk al conectar un dispositivo compatible en su nube, y que se entrega vía email una vez registrado.

Durante la ejecución del programa es necesario la adquisición y envío de datos desde y hasta la aplicación. Por lo que, para que la información pueda ser receptada por el dispositivo de Arduino se utiliza las líneas de programación mostradas en la figura 10.

Figura 10.

Subrutinas de lectura del estado de widgets en aplicación.

```
BLYNK_WRITE(V0)
{
  //Manual/Automatico
  pinValueV0 = param.asInt();
BLYNK_WRITE(V6) {
 TimeInputParam t(param);
BLYNK_WRITE(V7)
{
  //On/Off Manual
  pinValueV7 = param.asInt();
}
```

***Nota*:** Cada subrutina o función permite la lectura del estado de los switches y widget de tiempo utilizado en la aplicación de Blynk respectivamente, el mismo que cambia por la interacción con el usuario. Los valores que varían son entre “0” y “1” para switches (V0 y V7), mientras que para el widget de tiempo (V6), se adquieren los datos numéricos de hora y minutos, ingresados por el usuario.

Una vez manifestado como se realiza la adquisición de datos, el envío para el cambio de estado de indicadores que simulan Leds, se lo realiza tal y como se demuestra en la figura 11.

Figura 11.

Definición de variables correspondientes a los indicadores ubicados en aplicación.

```
WidgetLED electrobomba(V4);
WidgetLED proceso(V8);
WidgetLED fallo(V3);
proceso.on();
proceso.off();
```

***Nota*.** Cada línea iniciada con la instrucción WidgetLED indica el nombre utilizado en el sketch del pin virtual (V4, V8 y V3) correspondiente a los Leds ubicados en la aplicación, mientras que las líneas siguientes son ejemplos de instrucciones que efectúan el cambio de estado respectivamente.

Como se mencionó en el capítulo anterior, en la sección de estructura de un sketch de Arduino, dos de las partes primordiales del mismo son *void setup() y void loop*, por lo que a continuación en la figura 12 y 13 se puede observar cómo fueron realizados respectivamente.

Figura 12.

Función Setup.

```
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth);

  setSyncInterval(10 * 60);
  // Sync interval in seconds (10 minutes)
  // Display digital clock every 10 seconds
  timer.setInterval(10000L, clockDisplay);

  Blynk.virtualWrite(V5, "Select");

  pinMode(btnPin, INPUT);
  pinMode(outMoc, OUTPUT);
}
```

***Nota*.** En la función *"Setup"* se ubican los parámetros de inicialización del programa, los mismos que definen entradas y salidas del Arduino, para lo cual, como podemos observar en la figura, la variable "*outMoc*" es la salida que está en contacto directo con el optoacoplador y que define el estado de la bomba.

Figura 13.

Función Loop.

```
void loop()
{
  switch (m)
  {
   case 1:
   Blynk.virtualWrite(V5, "Auto");
   automatico();
   proceso.on();
   break;

   case 2:
   Blynk.virtualWrite(V5, "Manual");
   proceso.off();
     if (pinValueV7 == 1)
     {
     digitalWrite(outMoc,HIGH);
     electrobomba.on();
     }
     else
     {
     digitalWrite(outMoc,LOW);
     electrobomba.off();
     }
   break;

   case 3:
   Blynk.virtualWrite(V5, "Time");
     if (n == 1)
     {
     automatico();
     proceso.on();
     }
     else
     {
     proceso.off();
     }
   break;

   default:
   Blynk.virtualWrite(V5, "____");
   electrobomba.off();
   proceso.off();
   break;
  }
  Blynk.run();
  timer.run();
}
```

***Nota*.** En la función Loop se puede encontrar el menú realizado con el redireccionamiento a las funciones pertinentes de cada situación posible en la aplicación.

Para más detalle, visualizar el código completo ubicado en el Anexo A.

Configuración de Arduino YUN

1. Se conecta la placa mediante el puerto micro USB al ordenador,
2. En el ordenador, aparecerá una red con el nombre de “Arduino YUN” con su respectivo MAC a la cual se conectará, como se muestra en la figura 14.

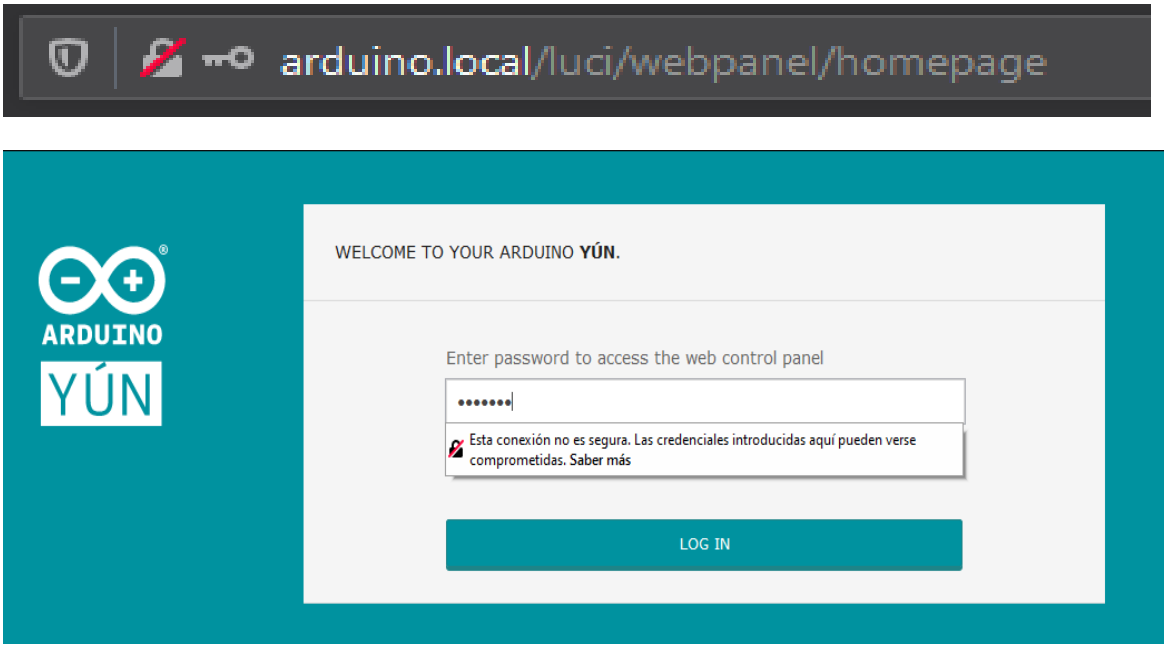
Figura 14.

Conexión de red de Arduino YUN.

3. En la barra del navegador web, escribimos “*arduino.local”*, en el cual nos aparecerá una página en la que solicita una contraseña, la misma que por defecto es *“arduino”,* tal y como se observar en la figura 15.

Figura 15.

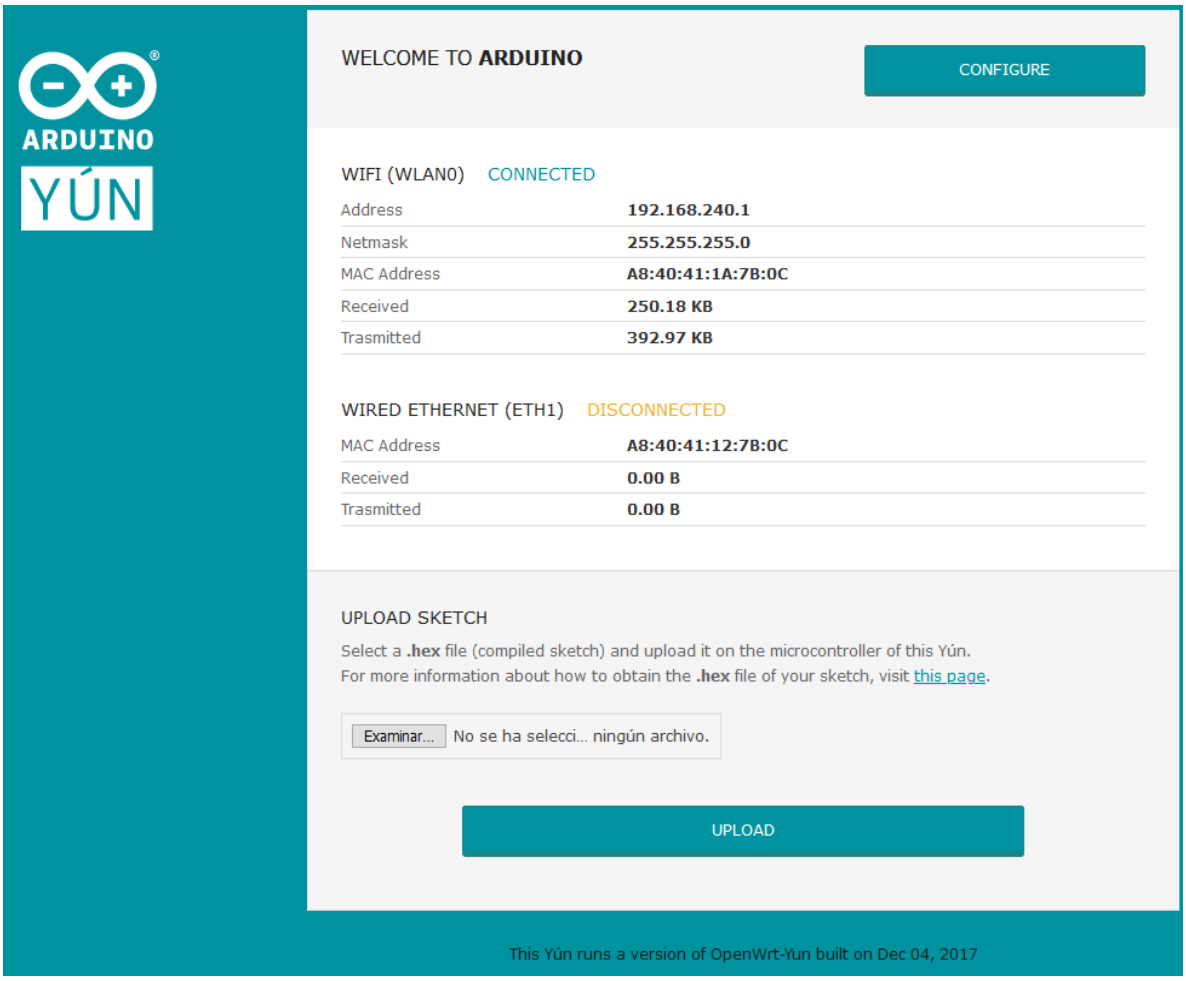
Link y página de inicio de Arduino.



4. Al ingresar y aceptar la contraseña; se actualizará la página con información de nuestro Arduino, como se expone en la figura 16. En la misma que al presionar en *“Configure”* se abrirán los parámetros editables de red de la placa de Arduino, para lo cual en nuestro caso se realizó la conexión de Arduino a una red con acceso a internet.

Figura 16.

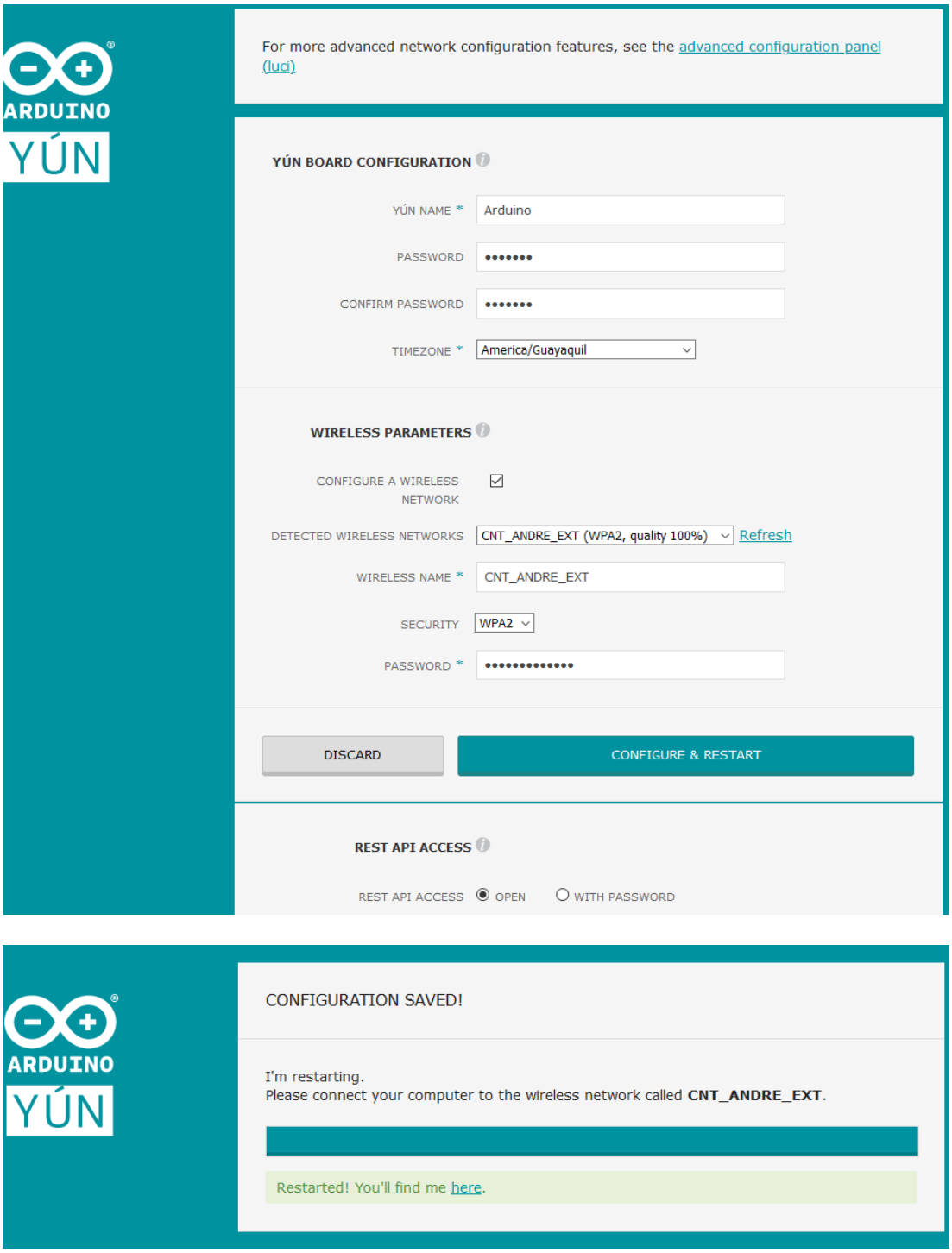
Página de información de la placa Arduino.



5. Finalmente, se hizo click en el botón *"Configure and Restart"*, y tras la carga del frame presentado en la figura 17, se reinició la placa y ahora se encuentra disponible en la red local con acceso a internet.

Figura 17.

Página y confirmación de configuración de Arduino.



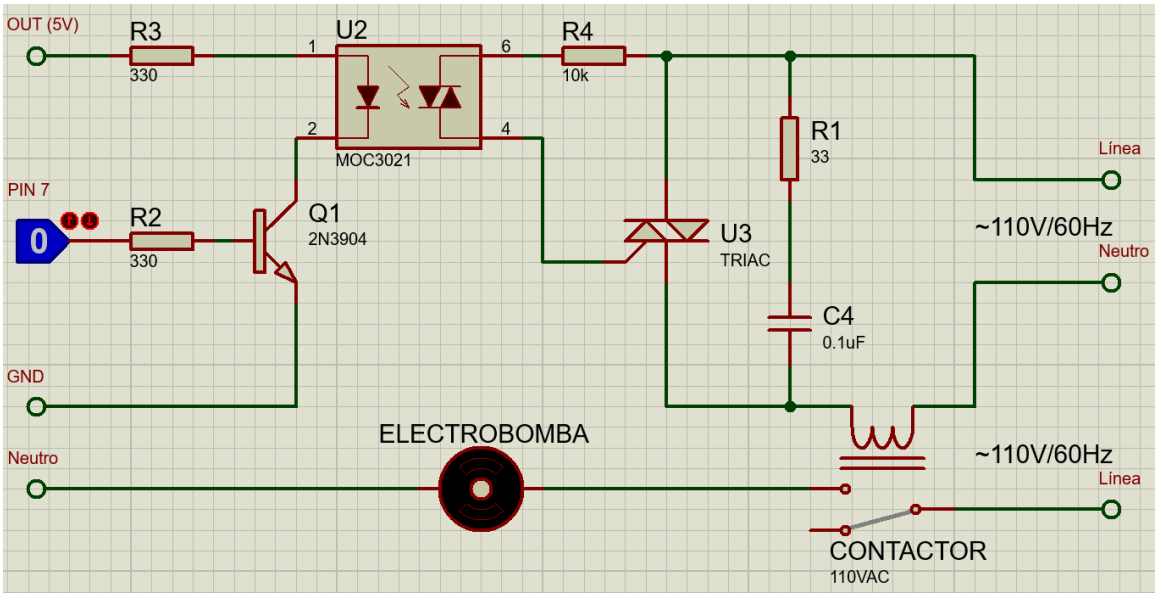
***Nota*:** Es posible realizar la programación y cargar el sketch mediante el puerto de red, ya que el Arduino y el ordenador se encuentran conectados en la misma red.

3.3. Etapa de potencia

Para acoplar la parte de control y de potencia se utilizó un circuito basado en un optoacoplador MOC3021 y un TRIAC BT136, que fueron utilizados como interfaz de comunicación entre el microcontrolador y la electrobomba, como se puede observar en la figura 18 a continuación.

Figura 18.

Diagrama eléctrico de acomplamiento.



***Nota*.** Para simulación se utilizó un “Logicstate”, “Relay” y “Motor” capaces de representar la función de la salida digital de la placa de Arduino y la función de un Contactor con la Electrobomba respectivamente.

Se seleccionó ambos elementos debido a sus especificaciones técnicas, las cuales se ajustan mejor a las necesidades, teniendo así el triac entre sus terminales A1 y A2 una capacidad de hasta 12A, lo cual es suficiente para la instalación y correcto funcionamiento del contactor en conjunto con la electrobomba seleccionada.

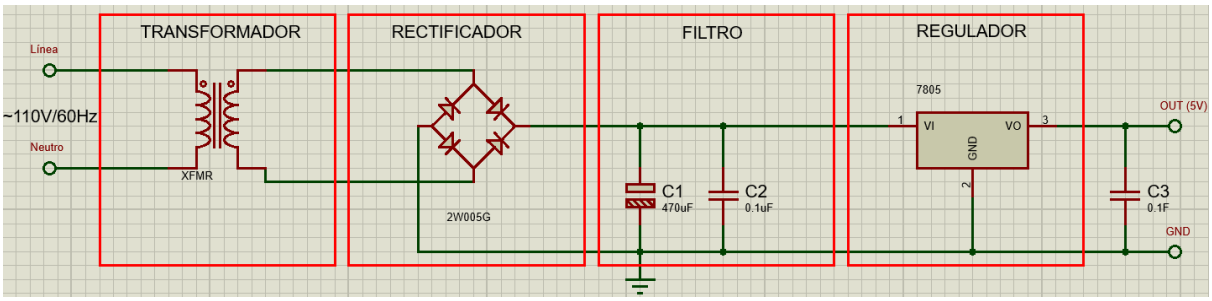
De la tarjeta de Arduino se tomó el pin 7 como salida, que conectado al circuito del optoacoplador, permite el control ON-OFF de la electrobomba mediante el TRIAC.

Fuente de alimentación

Para alimentar la placa de Arduino Yun, que se encuentra ubicado en el tablero de distribución/control, se realizó una fuente de alimentación regulada de 5V, para lo cual a continuación en la figura 19, se puede observar el diagrama eléctrico de la misma, con una identificación de las diferentes etapas que la compone, y una descripción de los elementos utilizados en cada una de ellas.

Figura 19.

Diagrama eléctrico de fuente de alimentación de 5V.

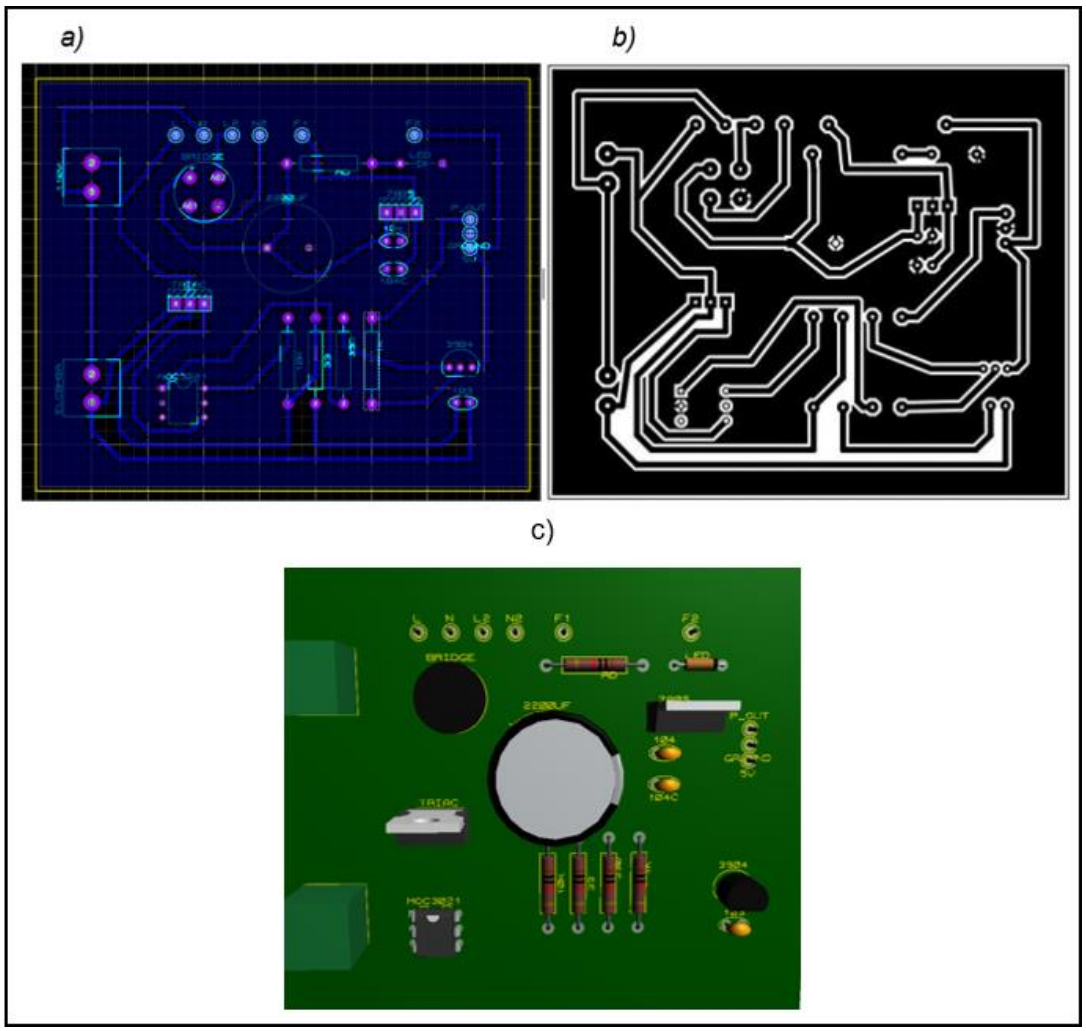


Nota. Para lo cual en la etapa de transformación se utilizó un transformador reductor 1:10, obteniendo en el secundario un voltaje de 12VAC, para lo cual seguidamente se realizó la etapa de rectificación, la misma que en su salida permitió obtener una salida de 12VDC, posteriormente para colocar el capacitor electrolítico se realizaron los cálculos pertinentes para una carga de 500mA, lo mismo que significó un valor aproximado a los 2200uF, y que el voltaje en la salida de este es de aproximadamente unos 17VDC máximo, para lo cual finalmente se colocó el regulador lineal, el cual es el encargado de entregar en la salida los 5VDC necesarios para alimentar la tarjeta de Arduino.

Una vez realizado y comprobado la simulación en el software Proteus, se procedió a realizar el diseño de la placa en PCB teniendo como resultado los gráficos de las pistas expuestos en las imágenes a y b de la figura 20, y el mismo que una vez terminado permite tener una vista física aproximada con los componentes colocados, tal como se puede observar en la imagen c de la figura 20.

Figura 20.

Diseño de PCB en Proteus.



***Nota*.** a) Vista de PCB Layout en Proteus del diseño realizado, b) Representación del diseño a imprimir y c) Aproximación física estimada del diseño realizado por la herramienta 3D Visualizer de Proteus.

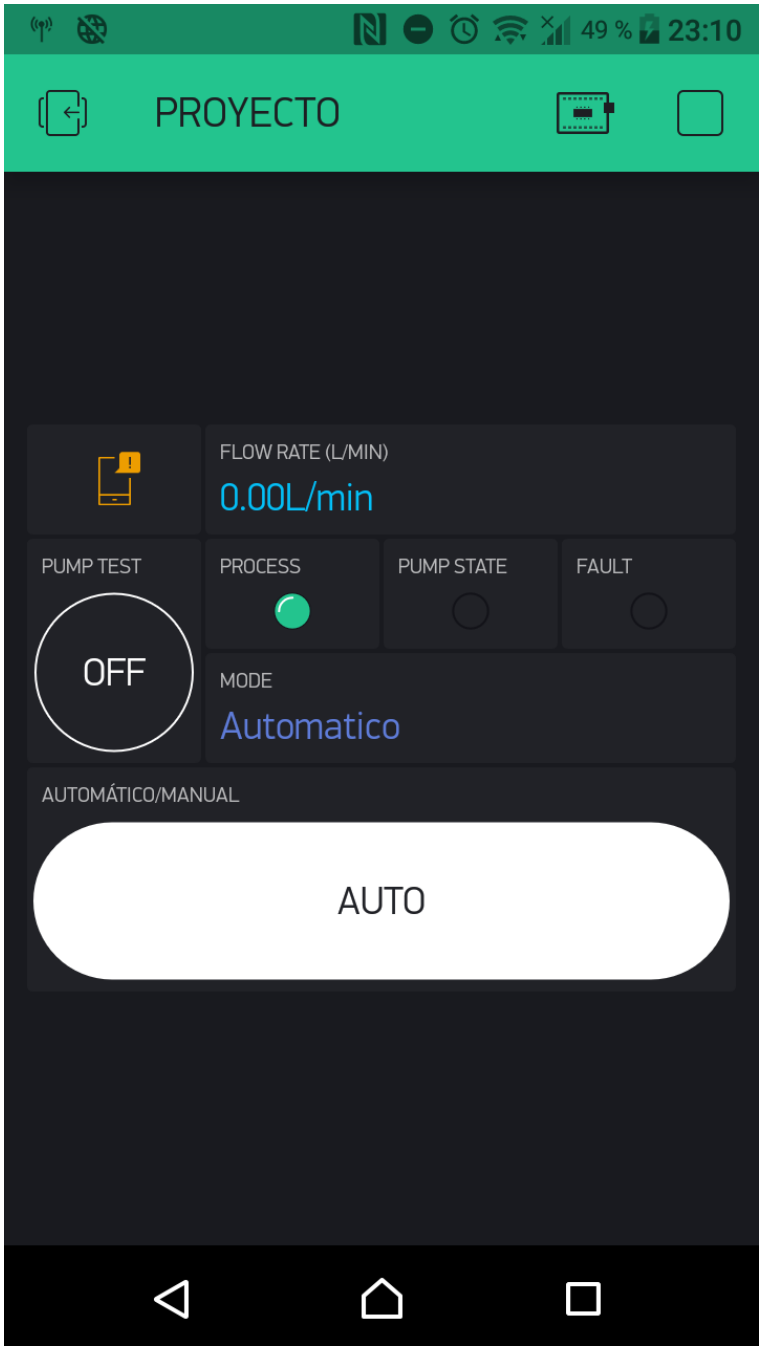
3.4. Diseño y desarrollo de la aplicación IoT

Para el diseño y desarrollo de la aplicación IoT se utilizó el servicio que brinda Blynk, el cual mediante un apartado en la nube y su gestor de aplicaciones en Android, permite realizar el ingreso de pulsadores, switches, indicadores, sliders, entre otros en un espacio de trabajo, los cuales en conjunto son componentes útiles para el control, monitoreo y/o supervisión de sistemas a larga distancia mediante internet; servicio que se encuentra especializado para su utilización en conjunto con dispositivos como Arduino, y afines que sean compatibles.

En este caso, para el presente proyecto como se muestra en la figura 21, se utilizaron dos switches que realizan un control ON-OFF manual y automático de la electrobomba respectivamente, cada uno con una variable virtual propia que será de utilidad en la elaboración del sketch de Arduino.

Figura 21.

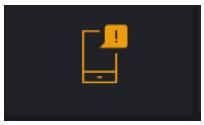
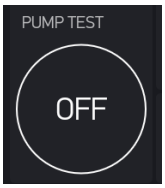
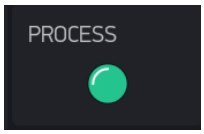
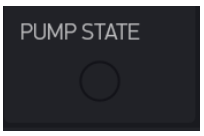
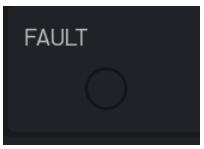
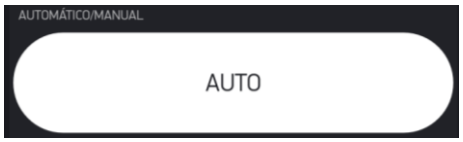
Aplicación realizada en Blynk.



A continuación en la tabla 7 se detallan los componentes de la aplicación presentes en la figura anterior.

Tabla 7.

Listado de elementos utilizados en aplicación.

WIDGET	SIGNIFICADO
	Habilita realizar notificaciones al teléfono
	Switch definido para la prueba de la electrobomba en modo manual
	Indicador del estado del sistema en modo automático
	Indicador del estado de la electrobomba
	Indicador en caso de fallo.
	Switch de selección de modo automático/manual.

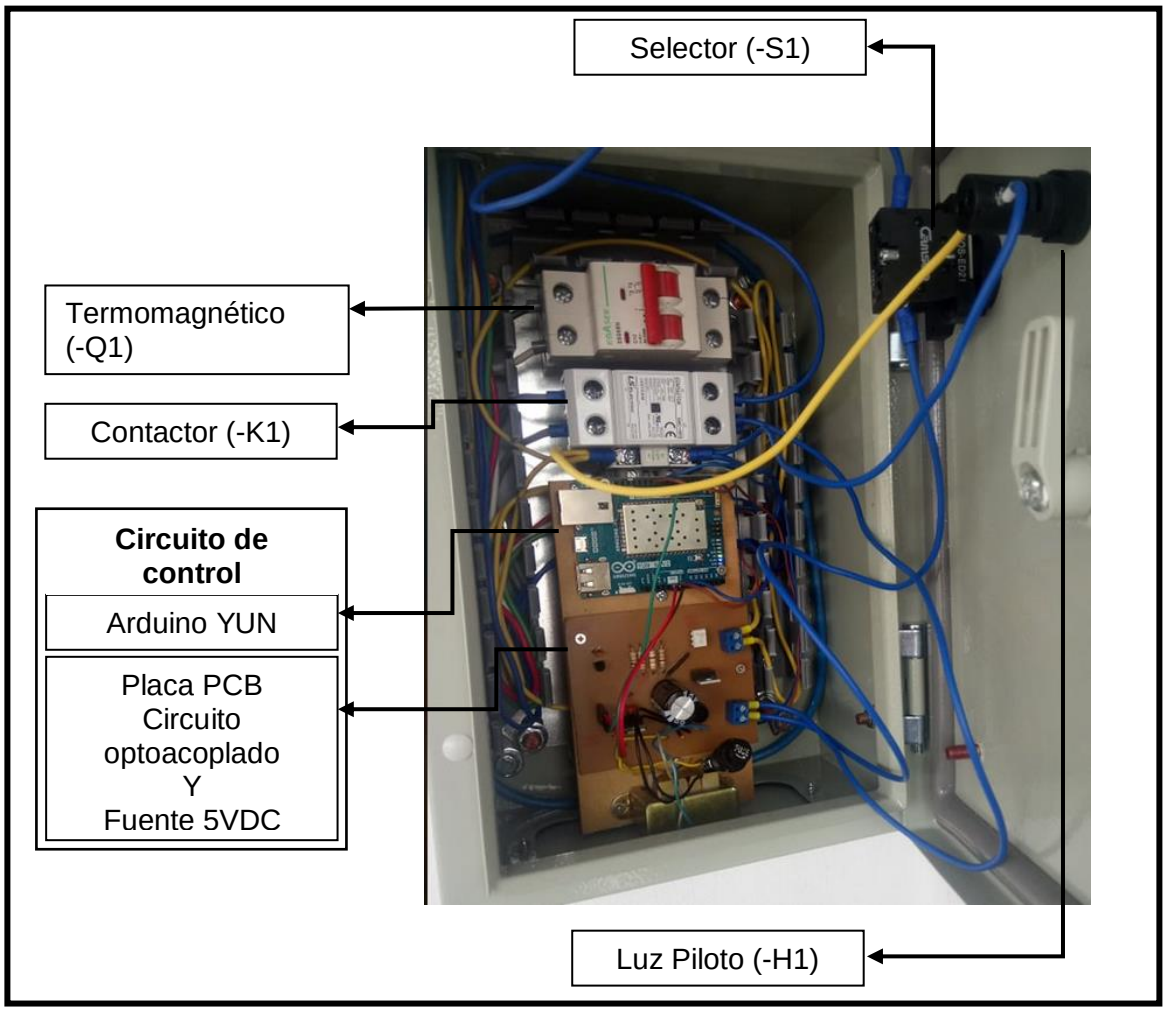
3.5. Instalación de la bomba eléctrica

En este punto se expondrán fotografías de los distintos puntos de conexión y una breve descripción con la finalidad de cada uno.

Para lo cual, se empieza con el tablero de control/distribución, el cual como se puede observar en la figura 22 contiene en su interior la placa de desarrollo de Arduino junto a la placa de PCB.

Figura 22.

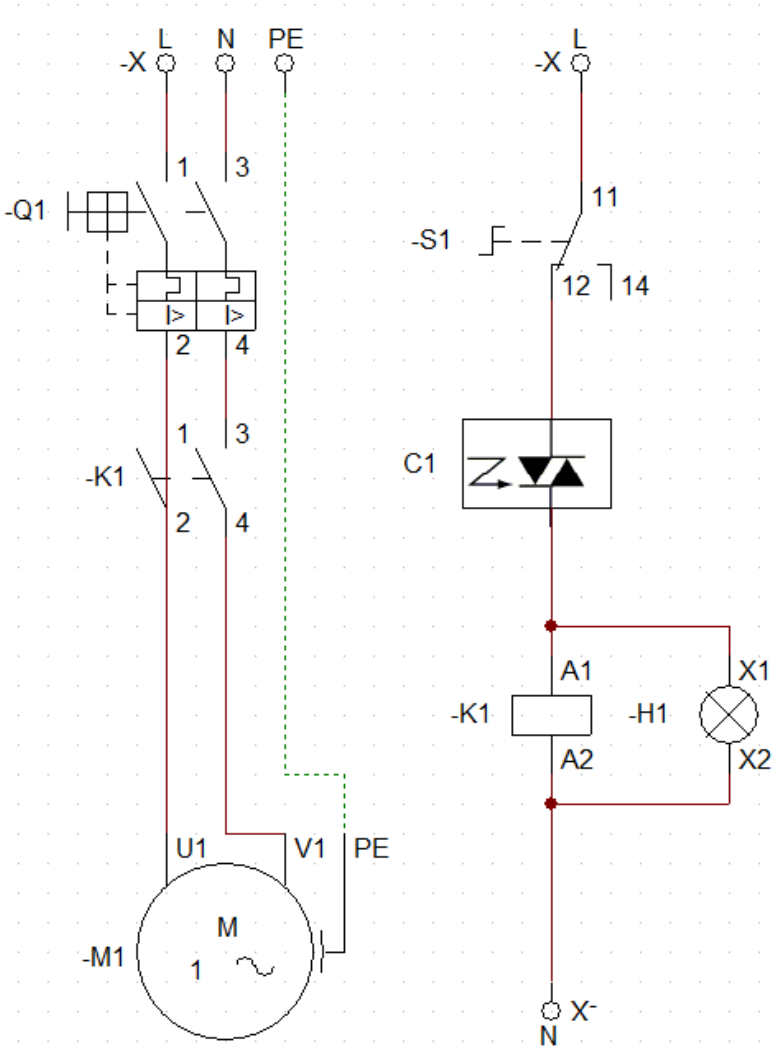
Tablero de control/distribución.



Además de ello, se ubica un termomagnético en la parte superior para protección de todo el proceso en general, seguido del contactor correspondiente a la bomba. Para lo cual, la conexión eléctrica se la puede representar con el diagrama unifilar mostrado en la figura 23, en el cual además se pueden reconocer los diferentes componentes con los mismos identificativos señalados en la figura anterior.

Figura 23.

Esquema unifilar del circuito de control y potencia.



***Nota*.** C1 representa el conjunto del circuito lógico digital (Arduino y placa de optotriac).

Por otra parte en el área de campo, se pueden ubicar los diferentes sensores (presóstato y caudal-YF-S201), la electrobomba y una caja estanca utilizada para la distribución de los diferentes componentes. Lo cual, a continuación en la siguiente imagen, figura 24, se puede evidenciar los elementos mencionados.

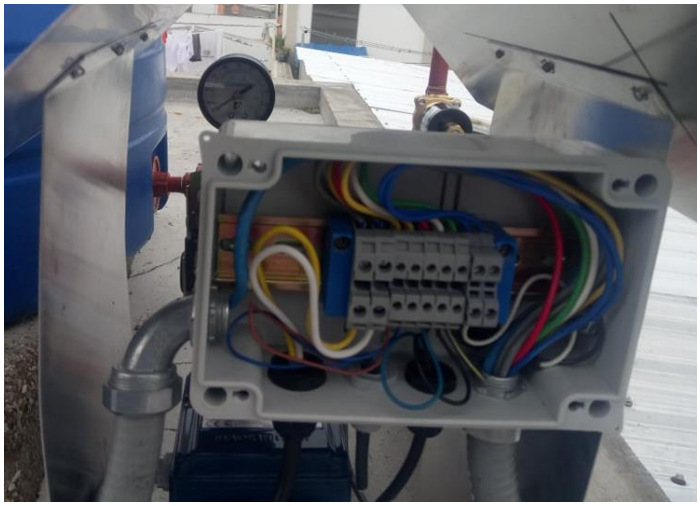
Figura 24

Implementación de sensores y electrobomba.

Y como se puede observar la caja estanca ubicada, fue utilizada para distribuir de mejor manera las conexiones eléctricas de los componentes, tal como se puede observar en la figura 25, el cual muestra el interior de la caja estanca.

Figura 25

Caja estanca de distribución de campo.



Finalmente, para terminar con la instalación del sistema, se realizó la calibración del presóstato, el cual fue fijado en el límite de los 40Psi máximos, ya que durante las pruebas realizadas se comprobó que dicho valor es el de uso normal, mientras que en caso de exceder dicho valor el Arduino actuaría conforme a lo programado.

3.6. Pruebas y resultados

En el transcurso de cuatro días se realizaron pruebas para diferenciar el flujo de agua con y sin sistema ante las actividades más comunes en el hogar.

Estos datos fueron tomados durante los periodos en los que la red de agua potable no abastecía el caudal necesario para un desarrollo normal de dichas actividades, y los cuales se pueden evidenciar en el resumen realizado a continuación en la tabla 8 que para más detalle puede dirigirse al Anexo B.

Tabla 8.

Comparación del caudal con/sin el sistema.

Actividades	**Sin Sistema**	**Con Sistema**
Uso de lavadora	3.83 L/min ~ 4.01 L/min	18.41 L/min ~ 18.71 L/min
Uso de ducha	10.31 L/min ~ 10.73 L/min	15.59 L/min
Uso de inodoro	9.71 L/min ~ 10.29 L/min	18.96 L/min ~ 19.59 L/min
Uso de grifos	8.97 L/min ~ 10.09 L/min	16.45 L/min ~ 17.01 L/min
Caudal Promedio	8.22 L/min ~ 8.78 L/min	17,35 L/min ~ 17,72 L/min

***Nota*.** Los datos expuestos en la tabla representan una mejora de aproximadamente el doble del caudal nominal de acuerdo a las muestras adquiridas, lo cual se resume en la conformidad de los usuarios frente a los problemas que existían anteriormente, y que impedían en cuestión de eficiencia la disponibilidad del servicio de agua potable en el hogar.

Capítulo IV

4. Conclusiones y Recomendaciones

4.1. Conclusiones

- Para el análisis y diseño del proyecto se ha englobado la demanda de agua potable en el conjunto de grifos, lavadora, duchas e inodoros, los cuales para un correcto funcionamiento requieren de un flujo aproximado de 15 litros por minuto, por lo que se realizó la colocación de un tanque reservorio de 500 litros en la parte más alta de la residencia el cual durante los periodos de abastecimiento normal se mantendrá lleno evitando su desborde gracias a la válvula de flotador que lo conforma, mientras que en los momentos de necesidad será el encargado de brindar el abastecimiento en conjunto con la electrobomba, la misma que para su elección y compra se lo realizó en base al costo y necesidad, resumiéndose en una electrobomba monofásica de 1/2 HP, debido a que es la potencia mínima encontrada en el mercado y que al no utilizarse para elevar fluidos no fue necesario de requerimientos mayores.
- La placa de Arduino YUN para controlar la parte de potencia, dejando de lado la interacción del usuario, basa su funcionamiento en la acción del sensor de caudal y presión. Para lo cual las señales de dichos sensores son procesados por el microcontrolador que simplifica el funcionamiento del sistema, de tal manera que el sensor de caudal activa el contactor al detectar flujo en la tubería, mientras que el presóstato desactiva el contactor al sobrepasar el límite calibrado.
- La operación, comprobación y monitoreo del sistema es posible gracias a las funciones que comprende la aplicación realizada en Blynk, las cuales están conformadas por el botón de selección de modo manual/automático, botón

ON/OFF de la bomba considerada solo para pruebas de la misma en modo manual, y alerta por notificación en caso de fallo del sistema o bomba. Además de ello en la aplicación se puede visualizar información del caudal en litros por minuto, estado del sistema, estado de la bomba, y fallo mediante los indicadores PROCESS, PUMP STATE y FAULT.

- Se realizaron pruebas del abastecimiento durante los periodos de falta de presión en la red de agua potable, con y sin el sistema en funcionamiento durante el desarrollo de las mismas demandas en ambos casos. Brindando confort a los residentes frente a las necesidades, ya que se estableció el valor nominal de caudal a un promedio de 17.5 litros por minuto mediante el sistema, lo cual significó el doble aproximadamente en comparación a la demanda sin este.

4.2. Recomendaciones

- En caso de fallo esperar a que el indicador en la aplicación se apague para continuar con el uso del sistema, enfocado al modo manual.
- Para el uso del estado de prueba de la bomba en modo manual, es preferible que algún elemento de la red de agua este abierto previamente a su activación, para que no cause fallos.
- La ubicación de la placa de desarrollo de Arduino debe ser en un lugar en el que la cobertura a internet no sea tan baja, ya que podría detenerse la comunicación y con ello el sistema dejaría de funcionar.

Bibliografía

Angulo Usategui, J. M., & Angulo Martinez, I. (2003). *Microcontroladores PIC: Diseño práctico de aplicaciones.* Mexico: McGraw-Hill Interamericana.

Arduino. (5 de Febrero de 2018). *Getting Started*. What is Arduino?: Recuperado el 2 de Marzo de 2021 de https://www.arduino.cc/en/Guide/Introduction

Didact, S. L. (2005). *Manual de Programación Lenguaje C++.* Sevilla: MAD.

Kuhmel, C. (2015). *Arduino for the Cloud: Arduino Yun and Dragino Yun Shield.* Florida: Universal Publishers.

Liberty, J., & Horvath, D. (2001). *Aprendiendo C++ Para Linux en 21 Días.* México: Pearson Educación.

Mandado Pérez, E., Menéndez Fuertes, L. M., Fernandez Ferreira, L., & López Matos, E. (2007). *Microcontroladores PIC. Sistema integrado para el autoaprendizaje.* Barcelona: MARCOMBO, S.A.

Massimo, B. (2008). *Getting Started with Arduino.* Sebastopol: O'Reilly Media.

Peña Basurto, M., & Cela Espín, J. (2000). *Introducción a la programacion en C.* Barcelona: Edicions UPC.

Peña, C. (2017). *Arduino - De cero a experto.* Buenos Aires: Six Ediciones.

Peña, C. (2020). *Arduino IDE: configuración y uso.* Buenos Aires: Plandos S.A.

Quiñonez Muñoz, O. (2019). *Internet de las Cosas (IoT).* Ibukku LLC.

Sedgewick, R. (1992). *Algorithms in C++.* Reading, Massachusetts: Addison-Wesley Publish Company.

Anexos