

**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE QUITO**

**CARRERA:**

**INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:**

**INGENIERO ELECTRÓNICO**

**TEMA:**

**“CONTROL DE VELOCIDAD DE UN MOTOR DC A TRAVÉS DE LA  
NUBE, MEDIANTE LA TARJETA STM32F4 DISCOVERY UTILIZANDO  
HERRAMIENTAS IOT”**

**AUTOR:**

**JUAN DIEGO GALARZA ZAMBRANO**

**TUTOR:**

**CARLOS GÉRMAN PILLAJO ANGOS**

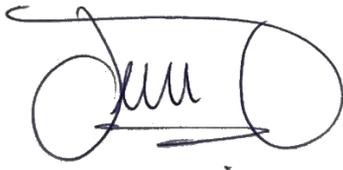
**Quito, enero del 2021**

## CESIÓN DE DERECHOS DE AUTOR

Yo Juan Diego Galarza Zambrano con documento de identificación N° 1720530821, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado/titulación intitulado: “Control de velocidad de un motor DC a través de la nube, mediante la tarjeta STM32F4 Discovery utilizando herramientas IoT.”, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservó los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, enero del 2021

A handwritten signature in black ink, appearing to read 'Juan', enclosed within a large, stylized circular flourish.

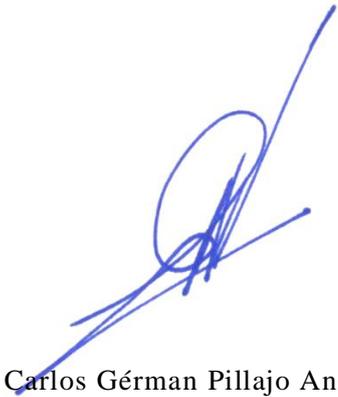
Juan Diego Galarza Zambrano

CI: 172053082-1

## **DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR**

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto técnico “Control de velocidad de un motor DC a través de la nube, mediante la tarjeta STM32F4 Discovery utilizando herramientas IoT.” Realizado por el Sr. Juan Diego Galarza Zambrano, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, enero de 2021

A handwritten signature in blue ink, consisting of several overlapping loops and lines, positioned above the printed name.

Carlos Gérman Pillajo Angos

CI: 1709255119

## **DEDICATORIA**

Dedico este trabajo a mis padres por haberme forjado como la persona que soy en la actualidad ya que han sido pilar fundamental en la culminación de mi carrera como Ingeniero, gracias a ellos no he desvanecido en el camino de toda la carrera que con esfuerzo mutuo me encuentro hoy en día finalizando una meta más para mi vida. Siempre estaré agradecido con su apoyo, ayuda e infinita perseverancia que hare de todos sus esfuerzos fruto de ellos, para brindarles un futuro maravilloso.

A mi novia y a mis hermanos, que con altos y bajos siempre estuvieron para apoyar en los momentos difíciles, que en las malas y en las buenas nunca dieron la mano a torcer e hicieron de ello una enseñanza más, un valor, una virtud nueva para mi vida.

## **AGRADECIMIENTO**

Primero a Dios por darme la salud y la vida, la virtud y la fortaleza, la sabiduría y perseverancia, para levantarme de cada caída y tropiezo que tuve en este arduo camino de la ingeniería agradeciendo su completa bendición de otorgarme este logro en mi vida.

A mi padre que siempre tuvo el sueño de tener sus hijos profesionales y con estudios superiores. Hoy al cielo va directamente embarcado en una estrella para que brille junto a él y sea el orgullo de quien en vida fue Don Juan Paulino Galarza Taco “El Gran Don Juanito”.

A mi madre y mis hermanos que a pesar de las dificultades recientes están firmes en las decisiones de mi vida apoyando así todo emprendimiento y lucha constante para sobresalir adelante ante la adversidad “ya que una familia unida es una gran familia”.

A mi novia que siempre ha sido el pedestal que nunca dejo caer el sueño de mis padres y el mío, que me apoyo en los momentos más difíciles de mi vida con su energía, su alegría y su sencillez.

A la familia Galarza y a la familia Zambrano que han sido testigos del esfuerzo realizado y del tiempo empleado para culminar mis estudios superiores.

A mi Tutor el Ingeniero Carlos Gérman Pillajo Angos por la paciencia, la acogida y el trato recibido en el desarrollo de trabajo de titulación y por sus conocimientos recibidos que eh adquirido mediante su tutela hasta llegar a la culminación de mis metas como ingeniero.

## TABLA DE CONTENIDO

<b>CESIÓN DE DERECHOS DE AUTOR</b> .....	<b>i</b>
<b>DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR</b> .....	<b>ii</b>
<b>DEDICATORIA</b> .....	<b>iii</b>
<b>AGRADECIMIENTO</b> .....	<b>iv</b>
<b>INDICE DE FIGURAS</b> .....	<b>viii</b>
<b>INDICE DE TABLAS</b> .....	<b>x</b>
<b>RESUMEN</b> .....	<b>xi</b>
<b>ABSTRACT</b> .....	<b>xii</b>
<b>CAPÍTULO 1</b> .....	<b>1</b>
<b>ANTECEDENTES</b> .....	<b>1</b>
<b>PROBLEMA DE ESTUDIO</b> .....	<b>1</b>
<b>JUSTIFICACIÓN</b> .....	<b>1</b>
<b>1.3 OBJETIVOS</b> .....	<b>2</b>
1.3.1 Objetivo General.....	2
1.3.2 Objetivos Específicos .....	2
<b>CÁPITULO 2</b> .....	<b>3</b>
<b>FUNDAMENTACIÓN TEÓRICA</b> .....	<b>3</b>
<b>2.1 IOT (INTERNET DE LAS COSAS)</b> .....	<b>3</b>
2.1.1 Aplicaciones del internet de las cosas .....	4
<b>2.3 PROTOCOLO DE CONEXION MQTT</b> .....	<b>5</b>
2.3.1 Paquetes de control MQTT .....	5
2.3.2 Tipos de paquetes de control MQTT .....	5
2.3.3 Bróker MQTT .....	7
<b>2.4 ESTABLECIMIENTO DE LA CONEXIÓN</b> .....	<b>7</b>

2.5 THINGSBOARD .....	8
2.6 INTERACCIÓN GATEWAY-BROKER MQTT .....	9
2.7 NODE RED.....	9
2.7.1 Configuración en Node-RED .....	9
2.8 LENGUAJE DE PROGRAMACIÓN PHP.....	11
2.9 Tarjeta ESP32 .....	11
2.10 Telemetría .....	14
2.11 SISTEMAS DE CONTROL PARA REDES INALÁMBRICA WNC S .....	14
2.12 CONTROL EN LAZO CERRAZO .....	15
2.12.1 Controlador PID.....	15
<b>CAPÍTULO 3.....</b>	<b>16</b>
<b>IMPLEMENTACION DEL SISTEMA .....</b>	<b>16</b>
3.1 ESTACIÓN DE CONTROL .....	16
3.1.1 Tarjeta STM32F4.....	17
3.1.2 Sensor tacómetro.....	19
3.1.3 Fuente de energía y alimentación .....	21
3.1.4 Implementación de medición de revoluciones del motor .....	22
3.1.5 Implementación de control motor por PWM .....	24
3.1.6 Conexión de alimentación .....	25
3.1.7 Sistemas de control en tiempo discreto .....	26
3.1.9 Programación de la Tarjeta ESP32 .....	29
3.2 CONECTIVIDAD AL BRÓKER MQTT .....	32
3.2.1 Salida de datos de la planta por comunicación serial para monitoreo .....	32
3.2.3 Recepción de las constantes para el control PID .....	33
3.2.4 Diseño de la tarjeta de control .....	34
3.2.6 Integración de MQTT .....	36
3.2.7 Nodo de recepción y representación de Node-red .....	36
3.2.8 Nodo de transmisión desde Node-Red .....	38
3.2.9 Telemetría para el control de constantes .....	38
3.2.10 Control manual.....	39
3.2.11 Control automático .....	39

<b>CAPÍTULO 4</b> .....	<b>40</b>
<b>ANÁLISIS Y RESULTADOS</b> .....	<b>40</b>
4.1 CARACTERÍSTICAS DEL MOTOR DC .....	40
4.2 PRUEBAS DE ESCALÓN UNITARIO Y ESTABILIZACIÓN DEL MOTOR	41
4.3 MODELAMIENTO .....	45
CONCLUSIONES .....	50
RECOMENDACIONES .....	51
<b>Referencias</b> .....	<b>52</b>

## INDICE DE FIGURAS

Figura 2. 1 Internet de las cosas .....	3
Figura 2. 2 Protocolo MQTT .....	5
Figura 2. 3 etapas de los paquetes de control MQTT .....	6
Figura 2. 4 Modelamiento del bróker .....	7
Figura 2. 5 Modelo OSI mediante el protocolo MQTT .....	8
Figura 2. 6 Arquitectura Thingsboard .....	8
Figura 2. 7 Interacción gateway-broker MQTT .....	9
Figura 2. 8 Estructura Node red .....	10
Figura 2. 9 Ejemplo de tablero configurado en NODE RED .....	10
Figura 2. 10 Diseño de una petición de navegador en PHP .....	11
Figura 2. 11 Módulo ESP-01 y NodeMCU. Ambos con el SoC ESP8266 .....	12
Figura 2. 12 Diagrama de bloques de la ESP32 de su interior.....	12
Figura 2. 13 Pinout del ESP32. ....	13
Figura 2. 14 Diagrama de bloques PID .....	15
Figura 3. 1 Diagrama de bloques general. ....	16
Figura 3. 2 Tarjeta STM32F4 DISCOVERY. ....	17
Figura 3. 3 Diagrama de bloques de la funcionalidad del STM32F4 Discovery.....	18
Figura 3. 4 Pines de entradas y salidas de los periféricos de la tarjeta STM32F4 Discovery.. .....	19
Figura 3. 5 Diagrama de bloques de un sensor tacómetro. ....	20
Figura 3. 6 Conexión del sensor tacómetro al microcontrolador.....	20
Figura 3. 7 Esquema del diseño de la fuente de energía de la planta. ....	21
Figura 3. 8 Algoritmo para medir las revoluciones del motor en función del tiempo. .....	22
Figura 3. 9 Segmento de código de la interrupción en STM32cube IDE. ....	23
Figura 3. 10 STM32cube MX usado para la configuración de periféricos imagen....	23
Figura 3. 11 Ciclo de trabajo de la modulación de ancho de pulso.....	24
Figura 3. 12 Segmento de código para el control PWM. ....	24
Figura 3. 13 Modulo L298n en una placa lista para el uso. ....	25
Figura 3. 14 L298N conexión a un motor con las líneas de control. ....	26
Figura 3. 15 Esquema básico PID continuo. ....	26
Figura 3. 16 Esquema básico PID discreto.....	27

Figura 3. 17 Segmento de código para el control PID en el algoritmo. ....	28
Figura 3. 18 IDE Arduino para programar el ESP32. ....	29
Figura 3. 19 Gestor de librerías de Arduino. ....	30
Figura 3. 20 Algoritmo del funcionamiento de la ESP32. ....	30
Figura 3. 21 Segmento de código de la ESP32 programado en el IDE de Arduino...	31
Figura 3. 22 Pines en la ESP32 para comunicarse con la STM32F4. ....	31
Figura 3. 23 Segmento de código del algoritmo para conectividad al bróker MQTT. .....	32
Figura 3. 24 Módulo USB a TTL. ....	32
Figura 3. 25 Módulo USB a TTL. ....	33
Figura 3. 26 Segmento de código de la salida de datos de las variables de control. ..	33
Figura 3. 27 Segmento de código de la recepción de los datos para el control PID.	34
Figura 3. 28 Circuito lógico de la planta. ....	35
Figura 3. 29 Diagrama UML de la interacción del cliente y bróker MQTT. ....	36
Figura 3. 30 Node-Red con los nodos para recibir los datos de la planta. ....	37
Figura 3. 31 Diagrama de flujo para recepción de datos de la planta. ....	37
Figura 3. 32 Control de los Slider para el cambio de datos de la planta. ....	38
Figura 3. 33 Pantalla para el control de telemetría de la planta. ....	39
Figura 3. 34 Slider para el control manual. ....	39
Figura 4. 1 Fuente de poder para el motor DC y medición de voltaje y corriente. ....	40
Figura 4. 2 Gráfico de las señales de la planta ante un escalón unitario. ....	42
Figura 4. 3 Interfaz de Node-Red para el operador. ....	42
Figura 4. 4 Tarjeta electrónica para el control de la planta. ....	43
Figura 4. 5 Motor que es el actuador de la planta. ....	44
Figura 4. 6 Vista superior de la planta. ....	44
Figura 4. 7 Segmento de código para la identificación de planta. ....	46
Figura 4. 8 Herramienta de MATLAB para el modelamiento de la planta System Identificación. ....	46
Figura 4. 9 Adquisición de datos de la planta para el modelamiento. ....	47
Figura 4. 10 Resultados del modelamiento con una estimación de 94.91% .....	47
Figura 4. 11 Resultados del modelamiento con una estimación de 94.91% .....	48
Figura 4. 12 Resultado de la sintonización PID. ....	48
Figura 4. 13 Función Transferencia de la sintonización PID. ....	49
Figura 4. 14 Resultado de la sintonización PID en simulink. ....	49

## INDICE DE TABLAS

Tabla 2 1 Aplicaciones con el internet de las cosas .....	4
Tabla 4. 1 Características del motor DC Transmotec 775-5520F-CC.....	40
Tabla 4. 2 Valores para las constantes del control PID usadas.....	41
Tabla 4. 3 Descripción de las señales del gráfico 3.2.....	41

## RESUMEN

En el presente proyecto se desarrolló una planta de control emulando un proceso industrial controlado por el algoritmo-PID. El proceso industrial es el control la variable de velocidad de un motor DC con uso un tacómetro con encoder de 100 pulsos por revolución para obtener la más alta resolución y con un puente H que incorpora el chip L298. El modelamiento y sintonización de la planta se la resolvió por medio de la herramienta MATLAB para lograr una aproximación más realista. El algoritmo del control PID se lo implementó en lenguaje C/C++ en el IDE de ST denominado STM32cubeIDE, además de ser el lenguaje nativo y formal del fabricante de la tarjeta STM32F4 Discovery que tiene una gran robustez para este tipo de procesos industriales. Se usó un bróker de mensajes MQTT para enviar y recibir los datos del cliente hasta la planta y viceversa logrando una gran estabilidad y una latencia aceptable para un proceso de control supervisado de manera remota. Para la interfaz gráfica para el operador se usó el dashboard de Node-Red donde se implementó dos tipos de indicadores de las variables de la planta como son el valor del sensor, la potencia al motor y el error calculado en ese instante de tiempo y para el control del lazo PID se incorporó slider para que el operador pueda cambiar los valores de control que se ejecutan tan pronto deje fijo la slider de la variable que esté manipulando.

## **ABSTRACT**

In this project, a control plant was developed emulating an industrial process under the supervision of a Proportional, Integral and Derivative control loop better known as PID control. The industrial process is based on the speed control of a direct current motor using a tachometer with an encoder of 100 pulses per revolution to obtain the highest resolution and with an H bridge that incorporates the L298 chip. The modeling and tuning of the plant was solved by means of the MATLAB tool to achieve a more realistic approach. The PID control algorithm was implemented in C / C ++ language in the ST IDE called STM32cubeIDE, in addition to being the native and formal language of the manufacturer of the STM32F4 Discovery card, which is highly robust for this type of industrial process. An MQTT message broker was used to send and receive customer data to the plant and vice versa achieving great stability and acceptable latency for a remotely supervised control process. For the graphical interface for the operator, the Node-Red dashboard was used where two types of indicators of the plant variables were implemented, such as the sensor value, the motor power and the error calculated at that instant of time and for the control of the PID loop has been incorporated slider so that the operator can change the control values that are executed as soon as the slider of the variable that he is manipulating is fixed.

# **CAPÍTULO 1**

## **ANTECEDENTES**

### **1.1 PROBLEMA DE ESTUDIO**

En el Ecuador la introducción de sistemas que implementen el uso de nuevas tecnologías de desarrollo es demasiado inestable debido a que no se genera confianza. En este momento en las industrias el mantenimiento preventivo y correctivo de los actuadores se los hace presencialmente, lo que ocasiona que si el operador no se encuentra dentro de la planta en el intervalo de algún fallo todo el proceso se detenga, lo que implica pérdidas y costos a la manufactura de la industria. (Cedeño Nuñez, P & Sanchez Paredes T, 2019)

La introducción de dispositivos con tecnología IoT que permiten la interconexión entre equipos presentando desafíos en el ámbito de estudio IoT. Desafortunadamente para lograr todo el potencial de IoT, no es suficiente que los equipos estén activos en la red, también necesitan establecer conexión e interactuar entre sí. (Yacchirema Vargas, 2019). Las empresas tienen temor a invertir en estos campos tecnológicos, además la introducción de dispositivos y sistemas embebidos como la tarjeta STM32f407

### **1.2 JUSTIFICACIÓN**

En un futuro, la mayoría de objetos serán “Inteligentes” y estarán interconectados a la red para explotar los paradigmas del Internet de las cosas, presentando muchos beneficios que ofrece la comunicación de dispositivos mediante IoT, como realizar un seguimiento en tiempo real y/o controlar desde diferentes puntos fuera de la central donde se sitúan los actuadores, ya que son manipulados mediante la nube de esta forma permite dar paso al desarrollo y aplicación de nuevos servicios e investigaciones. (Yacchirema Vargas, 2019)

Con el desarrollo de un sistema de control a través de la nube permitirá monitorear y controlar la velocidad del motor DC utilizando un sistema embebido en donde estará el algoritmo de control, evitando que la velocidad del motor se exceda a las velocidades programadas por lo que se logrará obtener un mejor rendimiento y consumo energético del motor.

## **1.3 OBJETIVOS**

### **1.3.1 Objetivo General**

Desarrollar un sistema de control automático para la variación de velocidad de un motor DC a través de la nube mediante la tarjeta STM32F4Discovery y utilizando herramientas IoT.

### **1.3.2 Objetivos Específicos**

- Diseñar un sistema de control en lazo cerrado para la variación de velocidad de un motor.
- Implementar una arquitectura mediante protocolos de conexión para que permita la interconexión de equipos físicos con la integración de plataformas IoT estándar.
- Realizar el monitoreo del sistema implementado mediante el envío de datos por la nube para el monitoreo del funcionamiento de su respectivo proceso.
- Validar las pruebas del funcionamiento del sistema de control diseñado para su análisis e implementación en futuras investigaciones.

## CÁPITULO 2

### FUNDAMENTACIÓN TEÓRICA

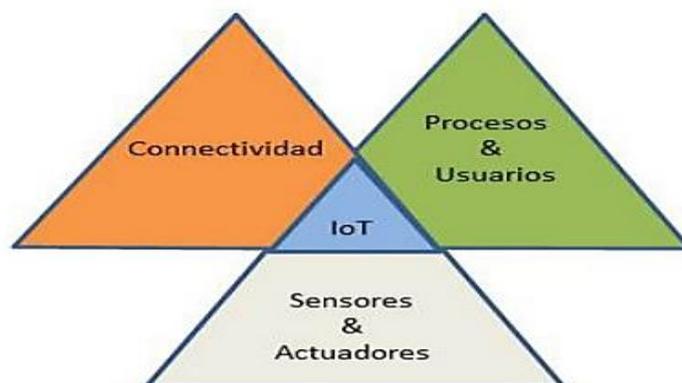
En este capítulo se detalla la parte teórica de la que se fundamenta el proyecto de titulación, se toma aspectos en relación al internet de las cosas y sus aplicaciones, también se explica detalladamente el protocolo MQTT y la función del bróker MQTT con el Node-Red y la plataforma Thingsboard para aplicaciones IoT. Para finalizar se hace una breve introducción del algoritmo PID para procesos de control.

#### 2.1 IOT (INTERNET DE LAS COSAS)

IoT es una arquitectura que facilita el intercambio de servicios entre redes de enlace de suministros, genera un impacto significativo de privacidad y defensa entre actores comprendidos. Ofrece un gran cambio en la calidad de vida y oportunidad de acceso a diversos servicios y datos específicos como seguridad, salud o transporte público entre diversos campos donde se puede aplicar.

Ofrece oportunidades que mejora en su gestión y seguimiento de activos en productos incrementando la variedad de datos almacenados por lo que permite aumentar el uso de recursos con lo que se traduce una reducción de costes. Otro beneficio es crear nuevos dispositivos interconectados e inteligentes con el fin de abrir espacio a nuevos modelos de negocio. (Salazar, 2016).

Figura 2. 1 Diagrama de funciones IoT



Fuente: (Salazar, 2016)

### 2.1.1 Aplicaciones del internet de las cosas

El IoT ofrece un sinnúmero de aplicaciones en industrias y específicos para el usuario de la IoT. Mientras que los dispositivos y redes proveen conectividad física, acceden aplicaciones de IoT de dispositivo a dispositivo de manera fiable y robusta. Aplicaciones de la IoT necesitan cerciorar de que los datos / mensajes han sido admitidos y ejecutados de manera conveniente y oportuna algunas aplicaciones que se puede presentar en diferentes campo de producción podemos verlos en siguiente tabla (Bonilla-Fabela, I., Tavizon-Salazar, A., Morales-Escobar, M., Guajardo-Muñoz, L. T., & Laines-Alamina, C. I., 2016)

Tabla 2 1 Aplicaciones con el internet de las cosas

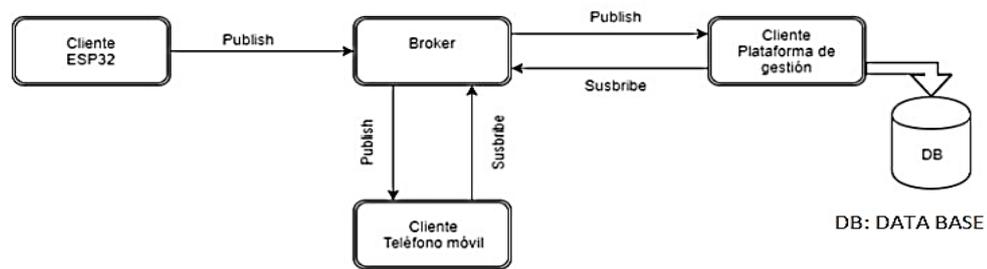
Concepto	Aplicación	Descripción
Ciudades inteligentes	Iluminación inteligente	Iluminación de la vía pública de manera inteligente y adaptativa en función del tiempo.
Contadores inteligentes	Red eléctrica inteligente	Control y gestión del consumo de energía.
CONTROL INDUSTRIAL	Aplicaciones M2M	Auto-diagnóstico y control de activos.
	Monitorización de la temperatura	Control de la temperatura en el interior de refrigeradores industriales y médicos con mercancía sensible.
DOMÓTICA Y AUTOMATIZACIÓN DEL HOGAR	Uso de energía y agua	Monitorización y seguimiento del consumo de agua y energía

Fuente: (Bonilla-Fabela, I., Tavizon-Salazar, A., Morales-Escobar, M., Guajardo-Muñoz, L. T., & Laines-Alamina, C. I., 2016)

## 2.3 PROTOCOLO DE CONEXION MQTT

Se define como un protocolo de comunicación asíncrona M2M es un modelo de muy simple y ligero de publicación y suscripción, idóneo para dispositivos de pocos recursos, permite conectarse a redes de banda ancha limitada o latencia alta, por su flexibilidad se lo puede implementar en escenarios diferentes o demanda de recursos. Para enlazar los dispositivos mediante el protocolo MQTT se lo hace mediante un bróker. En la figura 2.3 se visualiza en protocolo de conexión. (Moreno Cerdà, 2018)

Figura 2. 2 Protocolo MQTT



Fuente: (Moreno Cerdà, 2018)

### 2.3.1 Paquetes de control MQTT

Las estructuras de los paquetes de control se dividen en tres elementos los cuales vienen en el siguiente orden:

- Encabezado Fijo (este encabezado siempre está fijo y tiene una memoria de 2 bytes)
- Encabezado Variable (este encabezado no es fijo puede ser usado dependiendo de la aplicación)
- Payload (Posee una capacidad de 256MB de información limitada). (Yi, D., Binwen, F., Xiaoming, K., & Qianqian, M., 2016)

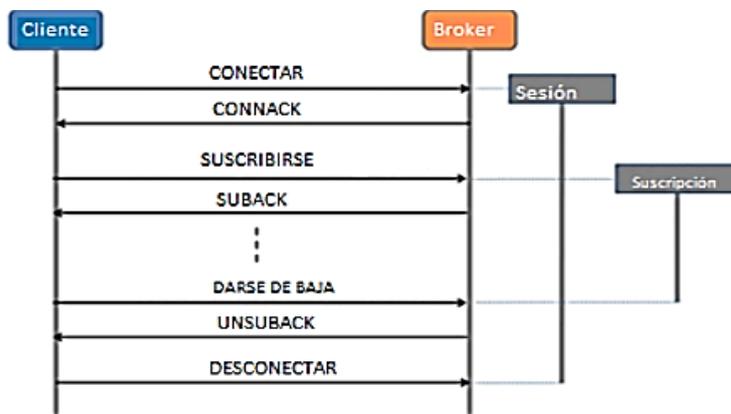
### 2.3.2 Tipos de paquetes de control MQTT

MQTT intercambia diferentes paquetes los cuales se enuncian a continuación. (Houimli, M., Kahloul, L., & Benaoun, S., 2017)

- CONNECT es el encargado de enviar el primer paquete desde cliente al servidor después de haber realizado el levantamiento de una red solicitando una conexión.
- CONNACK es el primer paquete que envía una solicitud pidiendo confirmación a la conexión que se transporta desde el servidor hacia el cliente. Si el paquete CONNACK se demora una cantidad tiempo razonable, al cliente le corresponde cerrar la conexión de red.
- PUBLISH este paquete permite enviar un mensaje de aplicación. Que puede ser transportado por el servidor o el cliente
- SUBSCRIBE este paquete debe poseer un nivel de calidad de servicio (QoS) para cada suscriptor, para que el servidor proceda emitir mensajes de aplicación hacia el cliente.
- SUBACK este paquete confirma el procesamiento y la recepción de un paquete SUBSCRIBE mediante un acuse de recibo.
- UNSUBSCRIBE sirve para dar de baja a los tópicos del paquete SUBSCRIBE.
- UNSUBACK confirma la aceptación de un paquete UNSUBSCRIBE
- PINGREQ señala al servidor que está activo en la red el cliente
- PINGRESP este paquete señala al cliente que el servidor está activo.
- DISCONNECT este paquete es enviado desde el cliente al servidor para informar su desconexión.

En la figura 2.4 se visualizan las etapas de los diferentes paquetes de control

Figura 2. 3 etapas de los paquetes de control MQTT

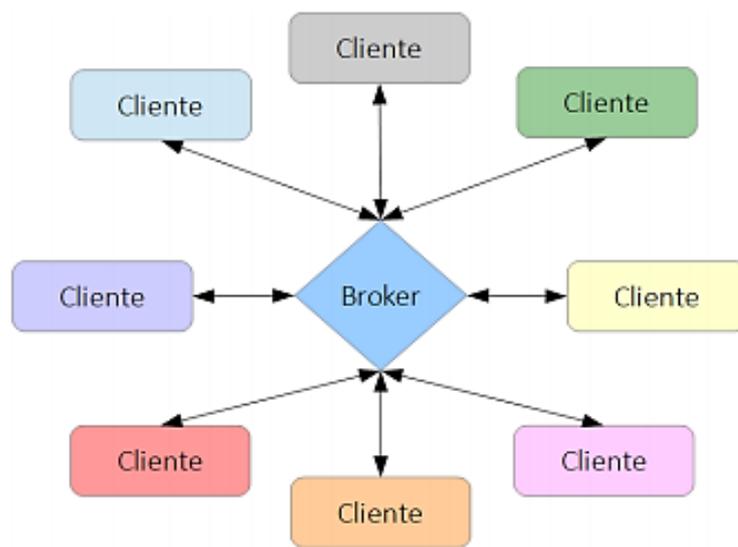


Fuente: (García Soria, 2017)

### 2.3.3 Bróker MQTT

El bróker es la parte más importante del modelo de suscripción, al cual se enlazan los clientes para realizar el envío y recepción de los mensajes. Es el mediador encargado de tomar los mensajes, filtrar en base al tópicos con el fin de despacharlos a los clientes que mantengan una suscripción a dicho tópicos por lo que asume la responsabilidad de autenticar a los nuevos clientes que intenten conectarse a él. En la figura 2.5 se puede visualizar el método de enlace de un bróker (De La Cal Calleja, 2019)

Figura 2. 4 Modelamiento del bróker



Fuente: (De La Cal Calleja, 2019)

### 2.4 ESTABLECIMIENTO DE LA CONEXIÓN

MQTT trabaja mediante capas superiores del modelo OSI con estructura TC/IP. Se asocia un canal de comunicación entre dispositivos mediante un tópicos. Aquellos que soliciten recibir paquetes de datos deben tener una suscripción mediante el bróker y se los asocia mediante un canal de comunicación, al contrario, los módulos que soliciten el envío de datos pedirán ser publicados mediante el tópicos con el que trabaja canal de comunicación. Cuando se presente un nuevo dato se enviarán el nuevo paquete datos a cualquier cliente suscrito mediante su bróker. (Osathanunkul, K., Hantrakul, K., Pramokchon, P., Khoenkaw, P., & Tantitharanukul, N. , 2017)

Figura 2. 5 Modelo OSI mediante el protocolo MQTT

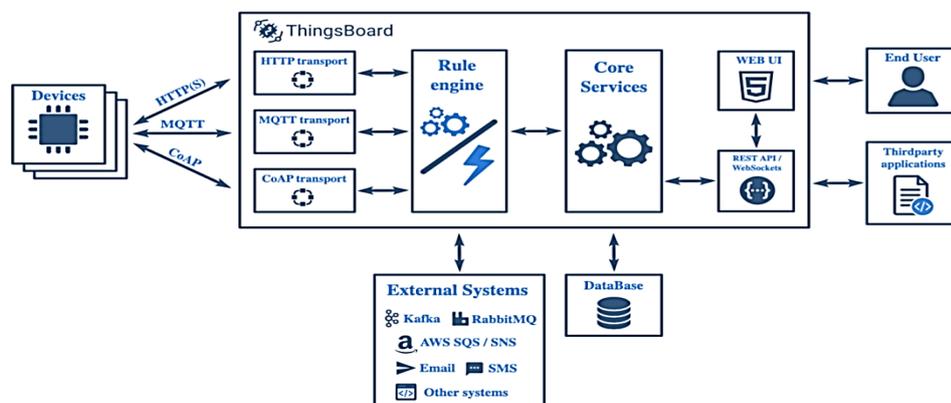


Fuente: (De La Cal Calleja, 2019)

## 2.5 THINGSBOARD

ThingsBoard es una plataforma que suministra una nube IoT para levantar una infraestructura en el servidor para acceder a varias aplicaciones de IoT, provee soporte para todos protocolos como MQTT, CoAP y HTTPs para la conectividad de dispositivos, permite el uso de tres opciones de base de datos diferentes: SQL, bases de datos híbridas y NoSQL (No only SQL). Utiliza estas bases de datos para almacenar entidades (como dispositivos, activos, paneles, usuarios, alarmas, clientes, etc.) y datos de telemetría (atributos, lecturas de sensores de series temporales, estadísticas, eventos, etc.). Presenta dos ediciones diferentes, la Community Edition, que es gratuita y totalmente de código abierto, y la Professional Edition, que tiene características más avanzadas, edición comunitaria es de código abierto y está disponible de forma gratuita tanto en el sitio web oficial de ThingsBoard como en la plataforma de desarrollo de software GitHub en la Figura 2.7. se visualiza la arquitectura básica de ThingsBoard. (Aghenta, L. O., & Iqbal, M. T 4(1), 57., 2020)

Figura 2. 6 Arquitectura Thingsboard

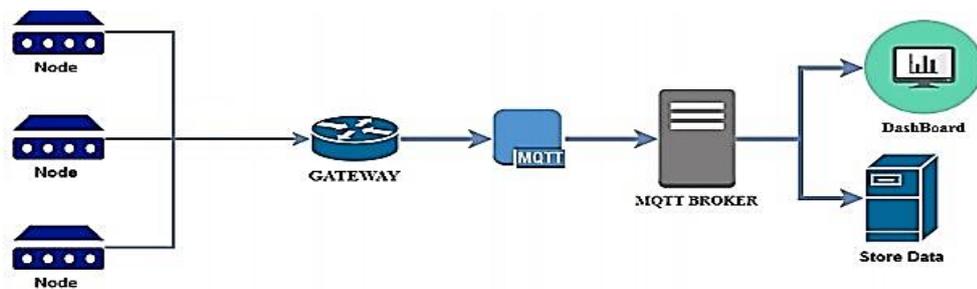


Fuente: (Aghenta, L. O., & Iqbal, M. T 4(1), 57., 2020)

## 2.6 INTERACCIÓN GATEWAY-BROKER MQTT

Para establecer comunicación desde los nodos con el bróker MQTT y la plataforma NODE RED, el Gateway realiza la función de intermediario, cuya función es la de recibir y descifrar los mensajes de los diferentes nodos que se entrelazan en la red. El Gateway debe estar conectado al servidor de la red vía ethernet o wifi según se requiera. Posteriormente, se envía un mensaje a la API de ThingsBoard siguiendo su protocolo de comunicación en el bróker MQTT. Para el proceso de descifración de los mensajes recibidos de los nodos y la creación del mensaje posteriormente, surge la necesidad configurar el Gateway debido a esto Node-RED es la herramienta que se utiliza para realizar esta configuración. En la figura 2.7 se puede visualizar esta interacción. (García Palomino, I., & Hinojosa Pérez, F. J., 2019).

Figura 2. 7 Interacción gateway-broker MQTT



Fuente:\_(García Palomino, I., & Hinojosa Pérez, F. J., 2019)

## 2.7 NODE RED

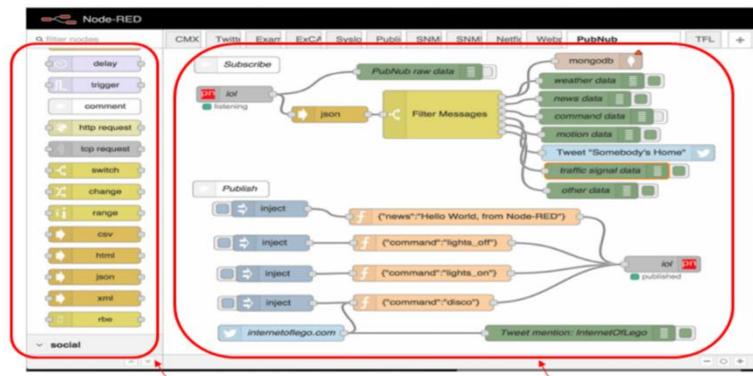
Node-RED es una herramienta de visualización que se basa en el flujo de dispositivos que se manejan mediante protocolos, se lo describe como instrumento de middleware cuya función es ofrecer asistencia a una aplicación para que se comunique con otras aplicaciones. Su estructura se basa nodos, que permite a los usuarios manipular la web y variables necesarias para interconectar los nodos entre sí.

### 2.7.1 Configuración en Node-RED

Node-RED tiene el diseño basado en Node.js ya que este proporciona muchas bibliotecas. En la figura 2.8 se visualiza como se entrelazan los nodos con los diferentes protocolos y aplicaciones.

Otra característica conveniente de Node-RED es el (tablero), incluye varias herramientas GUI y los usuarios pueden usarlo para los diversos flujos de control se pueden convertir en una pantalla de tablero; esta ayuda a la configuración del software y los protocolos, al arrastrar los elementos hacia la derecha se conecta cada entrada / salida y se establecen valores de las variables básicas. (Lekić, M., & Gardašević, G, 2018).

Figura 2. 8 Estructura Node red



Fuente (Oh, S. C., Kim, T. H., & Kim, Y. G , 2018)

En la figura 2.9 se muestra un ejemplo de la implementación real del tablero. Se diseña la pantalla GUI del tablero para este ejemplo en el que se emplean diversas entradas y salidas que son utilizadas en la pantalla de programación. Al obtener la interfaz del tablero esta se puede acceder fácilmente a través de una dirección IP especificada y de forma remota a través de un dispositivo de línea.

Figura 2. 9 Ejemplo de tablero configurado en NODE RED



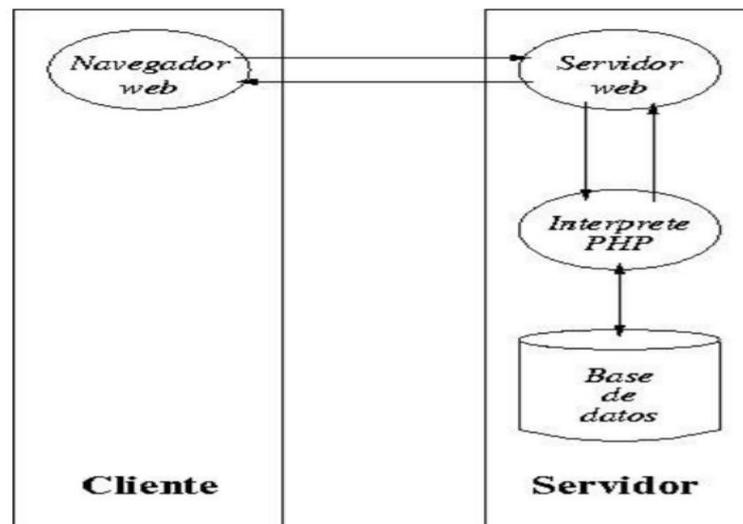
Fuente: (Oh, S. C., Kim, T. H., & Kim, Y. G , 2018)

## 2.8 LENGUAJE DE PROGRAMACIÓN PHP

La aplicación de este lenguaje de programación es más utilizada, en la generación de páginas web siendo esta popular entre los programadores. En concreto, se utiliza en páginas con lenguaje HTML. En caso de que un cliente se caiga se puede seguir utilizando el sistema desde otro equipo que contenga un explorador con conexión al servidor.

Para que el código PHP sea incluido basta con predefinirlo de la etiqueta. Si el servidor web está configurado correctamente, localizará automáticamente el código PHP y, en vez de proveerle el contenido de la página directamente al cliente, el programa lo ejecuta y devuelve su respuesta al navegador. Así pues, el diseño de una petición sería como sigue en la figura 2.10 (Duarte, M. P., & Pérez, I. M, 2017):

Figura 2. 10 Diseño de una petición de navegador en PHP

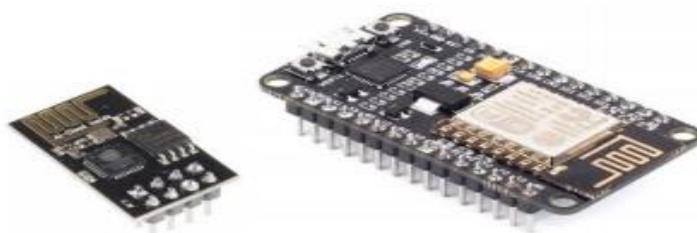


Fuente (Duarte, M. P., & Pérez, I. M, 2017)

## 2.9 TARJETA ESP32

En el mercado de los circuitos integrados existe una gran variedad de productos, varios tipos de módulos para comunicaciones inalámbricas WIFI y en su corazón está colocado un microcontrolador de altas prestaciones y de un muy bajo costo. El tipo de chip que se ocupa permite comunicaciones en redes inalámbricas y poder hacer conexiones en protocolo TCP/IP.

Figura 2. 11 Módulo ESP-01 y NodeMCU. Ambos con el SoC ESP8266

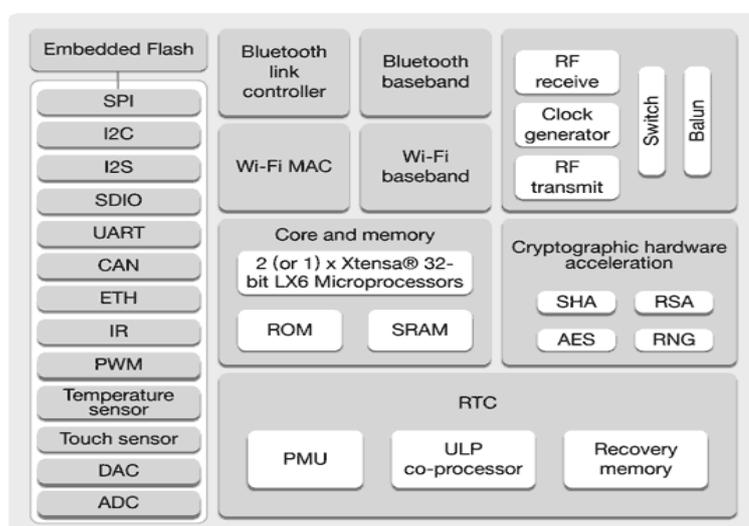


Fuente: (Aghenta, L. O., & Iqbal, M. T 4(1), 57., 2020)

El ESP32 dispone del estándar 802.11 b/g/n con una velocidad máxima de hasta 150 Mbit/s a 2.4GHz. Además, dispone de un gran número de bloques internos para varias funcionalidades y hacen de este módulo un gran dispositivo. ejemplo este módulo puede realizar comunicaciones SPI para poderse comunicar con otros tipos de integrados o a su vez el protocolo I2C que permite poderse incluso comunicar con memorias EEPROM para aumentar la memoria interna.

Cómo se puede observar en la figura 2.12 disponemos todo el esquema interno del módulo ESP32 donde podemos observar los diferentes bloques para funcionalidades extras. Por ejemplo, puede conectarse un módulo ethernet dentro de la misma circuitería y brindar dos tipos de conexiones la primera de forma inalámbrica y la segunda a través de un cable de red.

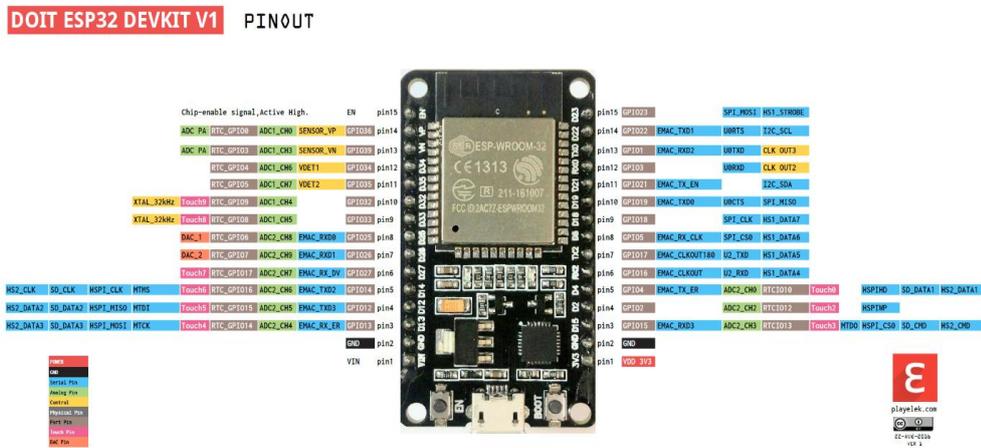
Figura 2. 12 Esquema de la Tarjeta ESP32 .



Fuente: (Lekić, M., & Gardašević, G, 2018)

El ESP32 añade distintas funciones, como mayor potencia sensor táctil, sensor de temperatura, Bluetooth 4.0, sensor hall, variedad de puertos y buses de entrada y salida.

Figura 2. 13 Pinout del ESP32.



Fuente (Aghenta, L. O., & Iqbal, M. T 4(1), 57., 2020)

Para el ESP32 que se usó se detallan las características más importantes de los pines, en la siguiente lista.

- Pines GPIO 34 y 39: pines solamente de entrada. No pueden generar PWM ni actuar como salidas GPIO.
- Pines GPIO 6 hasta GPIO 11: conectados a la flash del SPI interno del chip ESP-WROOM-32, por lo que no se recomienda su uso ya que puede fallar al cargar el código en el dispositivo.
- Pines para el ADC: no todos los pines están conectados a los ADC internos, solo los siguientes:  
0,2,4,12,13,14,15,25,26,27,32,33,34,35,36,39.
- Pines GPIO 25 y GPIO 26: pines que pueden usar el DAC.
- Pines GPIO 21 y GPIO 22: pines para la comunicación I2C. Aunque como se verá posteriormente en el apartado 2.3 se puede abrir otro puerto para una segunda comunicación I2C.

La tarjeta ESP32 dispone en su interior un regulador de voltaje de 3.3 voltios, y requiere ser alimentados con 5 voltios desde una fuente de poder que genere al menos 300 miliamperios o por su defecto el uso de USB para ser alimentado por la computadora.

## **2.10 TELEMETRÍA**

La telemetría es un conjunto de tecnologías que permite realizar el control, manipulación y medición de forma remota. En la actualidad muchos dispositivos disponen de conectividad inalámbrica y de acceso remoto para que se pueda realizar distintas configuraciones y por ello ahorrar tiempo y dinero en logística.

En la actualidad que se vive en una era de gran intercambio de información donde prácticamente muchos de los dispositivos de control siempre se encuentran a una distancia, como por ejemplo el control remoto de un televisor que ya en la actualidad sería algo anticuado o como por medio de una aplicación que está en un teléfono celular se puede realizar las mediciones de temperatura del jardín de una casa.

Por lo indicado anteriormente se deduce que la telemetría es muy importante en este tipo de competencias ya que se necesita un monitoreo constante y a tiempo real, de lo que está sucediendo con los diferentes sistemas del vehículo y poder realizar las correcciones respectivas cuando sean necesarias, ayudando a mejorar los mantenimientos del monoplaza, esto conlleva a que el ciclo de vida útil de las diferentes piezas y sistemas sea el más óptimo.

## **2.11 SISTEMAS DE CONTROL PARA REDES INALÁMBRICA WNCS**

Los sistemas inalámbricos para la transmisión de datos tienen que tomar en consideración un sistema de control en lazo cerrado para la transferencia de datos, esto debido a que las redes de datos presentan desafíos que no pueden ser tomados de forma independiente al sistema de control de realimentación.

Una de las características de los WNCS es la reducción del cableado y por ende la reducción de costos asociados con ella. Las dificultades que puede presentar estos sistemas tiene que ver con los retardos, pérdida de datos aleatorios. Debido a esto se oferta un amplio campo de investigación en estos sistemas de control inalámbrico. (Pillajo Angos, 2017)

## 2.12 CONTROL EN LAZO CERRAZO

En un sistema cuyo controlador se alimenta con la señal de error con el fin de disminuir el error y estabilizar el proceso al valor deseado. El sistema de control en lazo cerrado se compone se los siguientes elementos (RIFFO GUTIÉRREZ, 2018):

- Bloque de comparación: Compara el valor requerido con el valor medido de la salida lo que genera un error.
- Bloque de control: Establece que operación se debe tomar cuando se recibe la señal de error
- Unidad de corrección: se denomina actuador, ocasiona un cambio en el sistema al eliminar el error.
- Proceso: es el sistema en el que se va a aplicar el control de la variable.

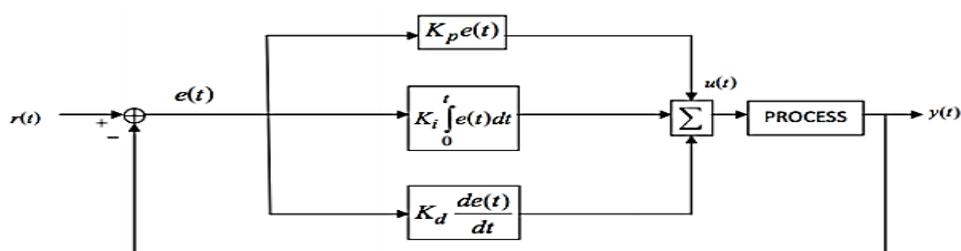
Los controladores PID son los más tradicionales y se aplican en la mayoría de procesos industriales debido a su arquitectura simple, sus principales características van en función al error del estado estacionario y su respuesta transitoria, lo que da paso a que estos controladores puedan controlar la variable controlada cuando se presente cualquier perturbación en el sistema. Sus ajustes de parámetros van de acuerdo con la aplicación y la variable a controlar (Ibrahim, I. N., & Al Akkad, M. A , 2016).

### 2.12.1 Controlador PID

Un PID tradicional donde se compone de los siguientes elementos

- Proporcional P: Este es el error "actual".
- Integral I: se interpreta como la acumulación del error "pasado".
- Parámetro D: Se interpreta la predicción del error "futuro".

Figura 2. 14 Diagrama de bloques PID



Fuente: (Saeteros Ortiz, 2018)

## CAPÍTULO 3

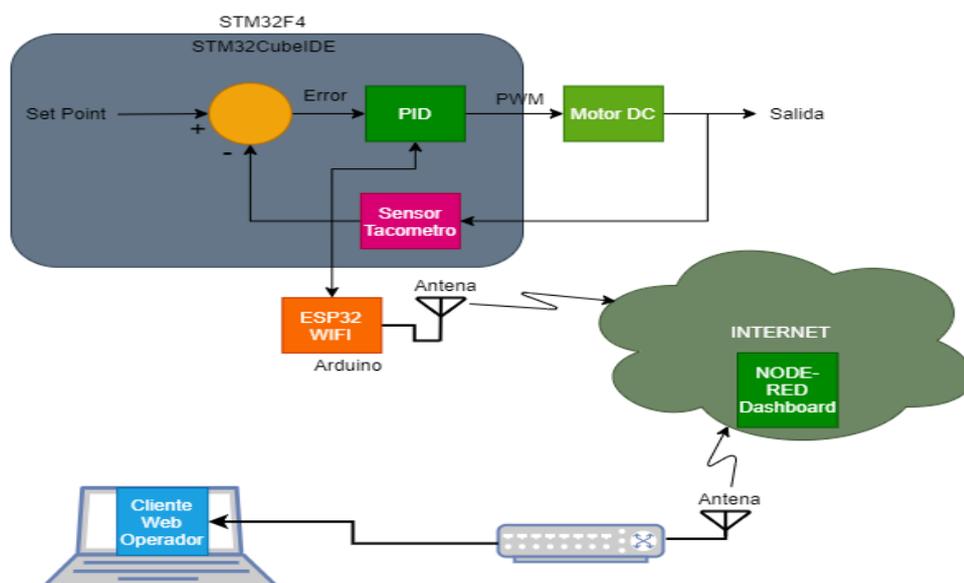
### IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se detalla la parte de implementación del proyecto de titulación, se describe el proceso que se realiza para la comunicación desde la estación de control mediante la tarjeta STM32f4 hacia la plataforma dashboard mediante protocolo MQTT a través de NODE RED para ejecutar el control PID de un motor DC en aplicaciones IoT.

#### 3.1 ESTACIÓN DE CONTROL

En el desarrollo de la estación para el control del motor de corriente continua (DC) se realizó el uso de la tarjeta STM32F4 Discovery de fabricación de la empresa ST para que se incorpore los algoritmos de control PID de la planta. Realizando un muestreo del sensor cada 100 milisegundos y una inmediata actualización de las variables de control, para ser usadas por PWM para generar cambios de las revoluciones del motor. En la figura 3.1 se visualiza de manera general todo el sistema de control, donde se presenta el esquema de la planta iniciando por el set point que es un valor que el operador puede manipular, seguido del cálculo del error propio de un control PID.

Figura 3. 1 Diagrama de bloques general.



Fuente: Autor

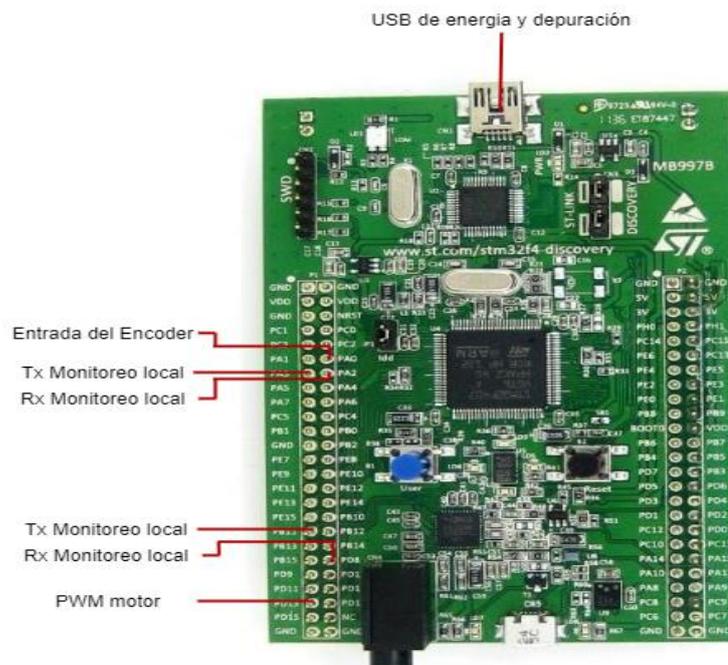
En el bloque Motor DC se incorpora dos etapas, la primera es el puente H que es la interfaz para poder intercalar la etapa digital conjuntamente con la etapa de potencia representada por un motor DC.

### 3.1.1 Tarjeta STM32F4

La tarjeta STM32F4 es un kit de desarrollo de la empresa ST que actualmente tiene un alto impacto en productos de consumo tecnológico. La línea de microcontroladores que produce tiene en su corazón un microprocesador ARM que brinda gran potencia y estabilidad. Además, que los costos de sus microcontroladores son considerablemente económicos que los de la competencia.

El uso que se le dio en el proyecto es para el control principal de la planta, evaluar el valor de los sensores y generar la señal PWM con el fin de regular la velocidad del motor, es importante aclarar que la misma empresa ST dispone de muchas herramientas de carácter libre para que sus consumidores las usen como por ejemplo el entorno de desarrollo integrado (IDE) llamado STMCubeIDE que está desarrollado sobre la plataforma de eclipse para dar soporte a los lenguajes C/C++ para dispositivos embebidos, fue el que se usó para desarrollar los algoritmos de control.

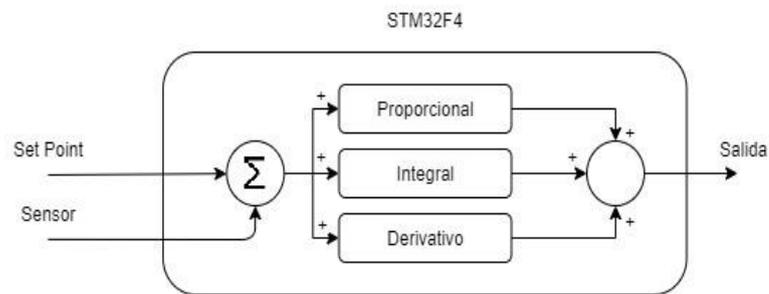
Figura 3. 2 Tarjeta STM32F4 DISCOVERY.



Fuente (Saeteros Ortiz, 2018)

En la figura 3.2 se observa la tarjeta STM32F4 Discovery en una vista superior donde se puede apreciar las diferentes partes que la componen, como por ejemplo en el centro está el microcontrolador STM32F407 con su nomenclatura y referencia a que usa arquitectura ARM. Algunas características de este procesador se enlistan a continuación: Posee una unidad de Punto Flotante (FPU): componente de la CPU, que sirve para el cálculo de operaciones en punto flotante, interacción con herramientas con meta-lenguaje (por ejemplo, Matlab), un sistema (MCU) que posee ultra- baja potencia, un procesador de señales Digitales (DSP) que posee arquitectura Harvard y opera instrucciones MAC de inicio ciclo. En ambos lados de la tarjeta están dispuestos de 100 pines para la interconexión con otros sensores o actuadores.

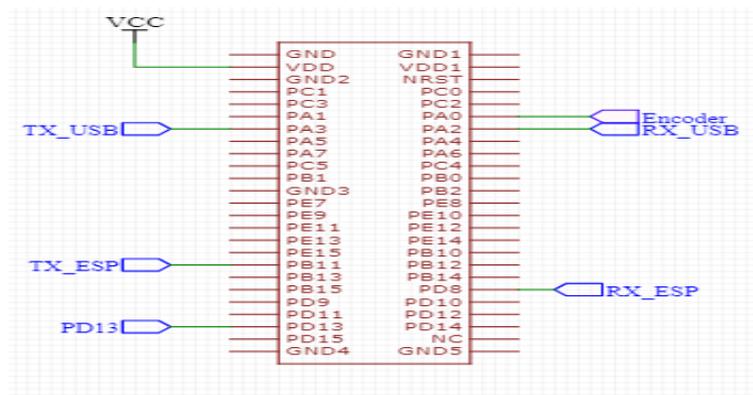
Figura 3. 3 Control PID en la STM32F4 Discovery.



Fuente: Autor

La figura 3.3 es un diagrama de bloques de la integración de control para el cálculo de las operaciones de control proporcional, integral y derivativo que se suman a la salida. El set point es el valor establecido por el operador de la planta, pero ese valor se lo ingresa bajo los términos de normalización, esto quiere decir que los valores deben ingresar en términos de un mínimo de cero y máximo de uno o en su defecto un mínimo de cero y máximo de 100. En ambos casos son equivalentes para realizar los cálculos y operaciones pertinentes.

Figura 3. 4 Pines de entradas y salidas de los periféricos de la tarjeta STM32F4 Discovery..



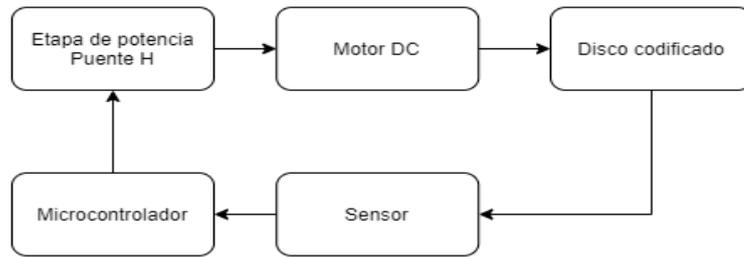
Fuente: Autor

Como se mencionó anteriormente la tarjeta STM32F4 Discovery dispone de terminales para conectarse al mundo, mediante la incorporación de sensores como también periféricos u otros dispositivos según el requerimiento del proyecto. En este caso la figura 3.4 es un diagrama de los pines que serán usados para las conexiones al módulo wifi, comunicación serial a la computadora personal (PC), entrada del tacómetro y salida PWM al puente H.

### 3.1.2 Sensor tacómetro

El dispositivo que se encarga de la interpretación de movimiento rotacional del motor y la digitalización en pulsos para que el microcontrolador mediante sus periféricos pueda recibirlos se denomina en términos generales tacómetro. El tacómetro está constituido por un disco perforado de un número finito que permite tener mayor resolución con mayor número de estas perforaciones, aunque en casos muy sencillos una perforación por revolución es funcional. Con respecto a la figura 3.5 se visualiza las diferentes etapas que conlleva el control de un motor. En primer lugar, el microcontrolador será el dispositivo de control que recibe una cantidad de pulsaciones por revolución en este caso para el proyecto se usó un disco con 100 perforaciones por lo tanto cuando se ejecute una revolución del motor este generaría 100 pulsaciones.

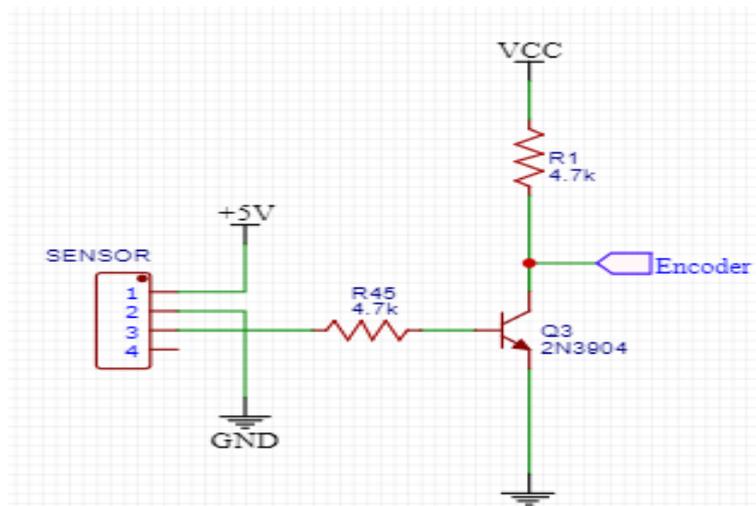
Figura 3. 5 Funcionamiento de un sensor tacómetro.



Fuente: Autor

La salida del sensor tiene niveles de voltaje de 0 a 5 voltios ya que su alimentación es del mismo rango, bajo los criterios de los datos del fabricante. Para realizar la interconectividad del sensor hacia el microcontrolador STM32F407 es importante disponer de un Bypass con la ayuda de un transistor de la serie 2N3904, que entre en estado de saturación cuando reciba en su base un valor de voltaje de 3.3 y pase a estado de corte cuando el valor de voltaje sea 0. Aunque se menciona en valores de voltaje para entrar en estado de corte y saturación del transistor, se debe mencionar que es un transistor bipolar NPN que tiene características basada en corriente, por lo tanto, cuando llegue un voltaje positivo de 3.3, generaría una corriente positiva de base a emisor provocando que entre estado de saturación.

Figura 3. 6 Esquema de conexión del sensor tacómetro con el microcontrolador.

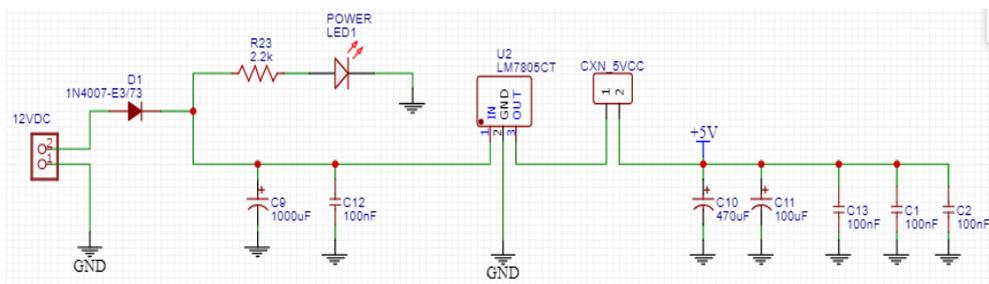


Fuente: Autor

### 3.1.3 Fuente de energía y alimentación

Para el correcto funcionamiento de la planta se debe usar dos fuentes de energía, para propósitos de visualización de las variables de la planta, y en segundo caso para el bloque de potencia que corresponde al puente H y al motor DC, aunque en casos especiales se puede compartir la misma fuente con la parte digital. Como se puede visualizar en la siguiente figura se ve en el diseño de la fuente de energía del circuito, que se describe desde el lado izquierdo que es la bornera donde se conecta la fuente de 12 voltios con un requerimiento mínimo de 1 amperio. A lado de la bornera está provisto de un diodo rectificador para protección de polarización inversa de la fuente y tenemos un diodo LED para indicar que esta con energía la etapa de regulador de voltaje.

Figura 3. 7 Esquema del diseño de la fuente de energía de la planta.



Fuente: Autor

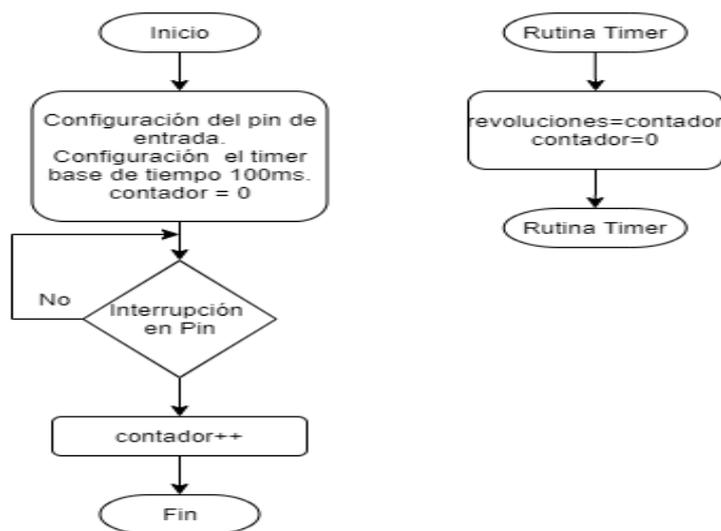
Para regular el voltaje proveniente de la fuente de energía que puede ser un adaptador como se observa en la figura 3.7 que entrega los requerimientos necesarios para el funcionamiento de la planta. Para la etapa de regulador de voltaje se usó el circuito integrado LM7805 y con el jumper CXN se corta la energía del regulador a la etapa digital o se puentea, esto es para los casos que se desee alimentar desde el regulador de voltaje U2 a la STM32F4.

La tarjeta STM32F4 dispone de dos puertos USB, uno de tipo mini USB y el otro es micro USB. El micro USB es para uso de depuración del programa y también sirve para alimentar de energía a la tarjeta desde el computador que sirva de Host. Es importante entender que la tarjeta STM32F4 por medio de sus pines alimenta a la parte digital de la planta, como por ejemplo al sensor tacómetro y el ESP32 que se usa para propósitos de conectividad al internet por medio de la red inalámbrica WIFI.

### 3.1.4 Implementación de medición de revoluciones del motor

En los microcontroladores se puede usar cualquier pin para evaluar el comportamiento de las señales digitales que este recibe, a este procedimiento se lo denomina rutinas por poleo al algoritmo que realiza un estado continuamente para después lanzar a otro proceso de código. Con la modernización de los microcontroladores actualmente existen pines que tienen características especiales para poder generar interrupciones directamente en el microprocesador, para lanzar a otra rutina de rápida ejecución y al terminar vuelva justo en la línea de código que realizó el salto. A esto se lo llama Servicio de Rutinas de Interrupciones que en la mayoría de casos todo microcontrolador dispone de dichas bondades incluso los Arduino disponen de estas funciones de una manera básica.

Figura 3. 8 Algoritmo para medir las revoluciones del motor en función del tiempo.



Fuente: Autor

En la figura 3.8 se ve el algoritmo para la configuración de la interrupción del pin de entrada para recibir los estados lógicos que provienen del sensor tacómetro. El algoritmo espera a que se genere el cambio de estado para lanzar a la rutina de interrupción, mientras que el microprocesador siga ejecutando código como por ejemplo el cálculo del algoritmo PID o también él envío de datos por el puerto serial para la visualización de las variables de la planta. Básicamente el código aumenta una unidad a la variable contador cada vez que se recibe una interrupción como se puede

observar en el segmento de código de la figura 3.9, y esta misma variable se respalda cada 100 milisegundos y se la encera para un nuevo ciclo de monitoreo.

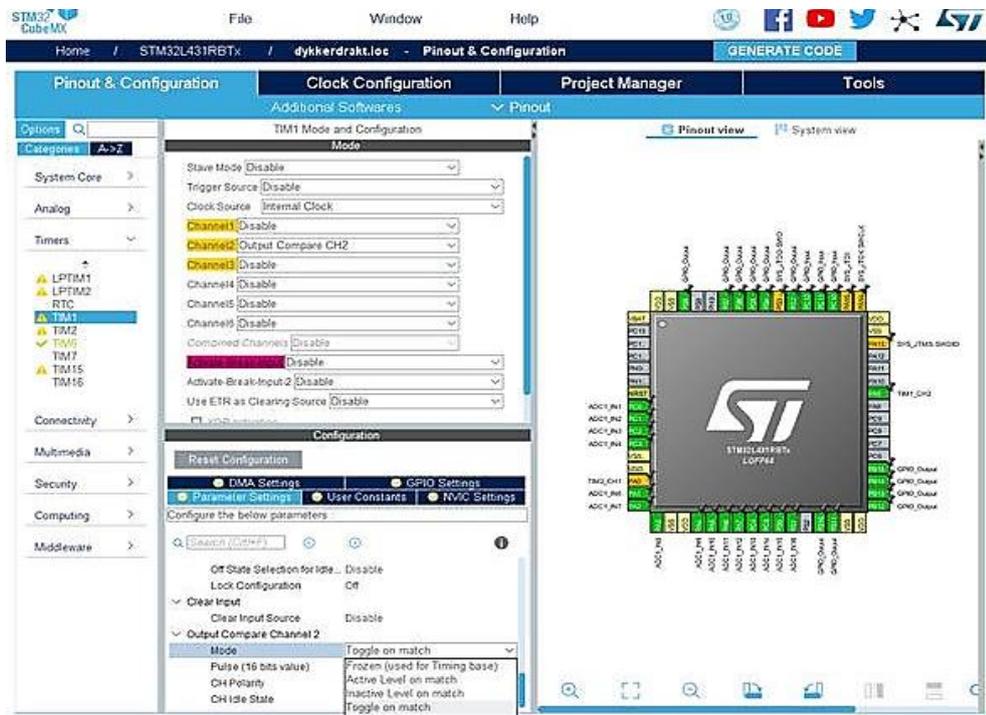
Figura 3. 9 Segmento de código de la interrupción en STM32cube IDE.

```
234 void EXTI0_IRQHandler(void)
235 {
236     /* USER CODE BEGIN EXTI0_IRQn 0 */
237
238     /* USER CODE END EXTI0_IRQn 0 */
239     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
240     /* USER CODE BEGIN EXTI0_IRQn 1 */
241     HAL_GPIO_TogglePin(LD4_GPIO_Port, LD4_Pin);
242     cnt_int++;
243
244     /* USER CODE END EXTI0_IRQn 1 */
245 }
```

Fuente: Autor

Como la herramienta STM32cube IDE dispone de un complemento denominado STM32cube MX como se puede visualizar en la figura 3.10, que es un generador de configuraciones gráfico, que permite establecer parámetros de manera rápida y visual a los distintos periféricos del microcontrolador a usarse, brindando al desarrollador la reducción de tiempos ya que proporciona una plantilla para poder agregar los códigos que se requieran en función del proyecto.

Figura 3. 10 STM32cube MX usado para la configuración de periféricos imagen



Fuente: (life.augmented, 2020)

### 3.1.5 Implementación de control motor por PWM

La señal promedio es el resultado del voltaje máximo y el ciclo de trabajo. El resultado de la expresión:

$$V_{medio} = (V_{cc+} - V_{cc-}) * \frac{DutyCycle}{100} \quad \text{Ec. (2. 1)}$$

De forma similar se tiene

$$DutyCycle = 100 * \frac{V_{medio}}{(V_{cc+} - V_{cc-})} \quad \text{Ec. (2. 2)}$$

Figura 3. 11 Modulación de ancho de pulso.



Fuente: (Zhen, Z., & Yan, G. A. O , 2013)

El puente H está formado por el circuito integrado L298n que tiene entradas para niveles de voltaje TTL y conjuntamente con el pin del STM32F407 que proporciona salidas a un voltaje de 3.3 para el nivel lógico 1, por lo cual se realizó el análisis en los datos del fabricante del puente H y se determinó que son compatibles para su funcionamiento.

Figura 3. 12 Segmento de código para el control PWM.

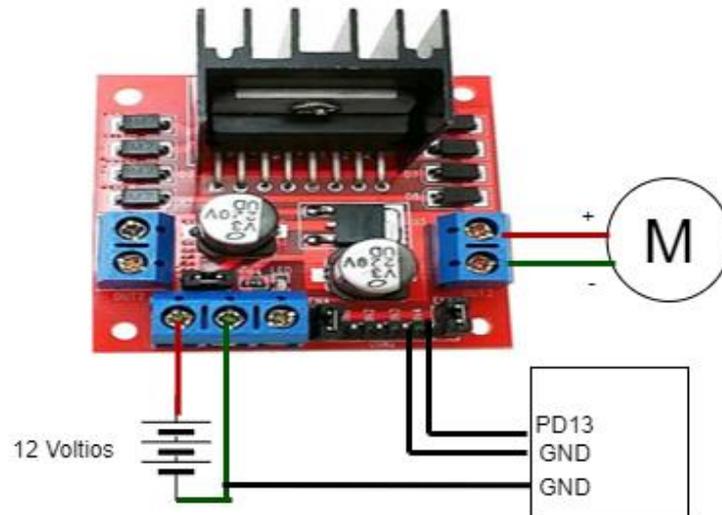
```
262     if(MotorOnFlag==1 && EmergenciaFlag==0){
263         __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_2, duty);
264     }else{
265         __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_2, 0);
266     }
```

Fuente: Autor

Se observa en la figura 3.12 es el segmento de código para el control PWM que gestiona el STM32F4 para la variación de la velocidad del motor. En la línea 262 se

ve un condicionante que permite el control de la ejecución para que el PWM del STM32F4 puede ser controlado por los datos que provienen del módulo WIFI.

Figura 3. 13 Modulo L298n en una placa lista para el uso.



Fuente: (naylampmechatronics, 2020)

El módulo trabaja en el control de los motores, sin la presencia de módulos adicionales. Ya que contiene diodos de protección y un regulador de tensión LM78M05 que es alimentado a 5V en el módulo L298n el diagrama de circuito se lo ve en el anexo 1.

La bornera A posee las líneas de salida OUT1 y OUT2, y la bornera B posee líneas de salida OUT3 y OUT4, es su parte inferior posee 4 señales de entrada y un pin de alimentación de 5v para habilitar las salidas de los módulos A y del módulo B que se denotan por las palabras ENA y ENB que su significado en inglés es enable

### 3.1.6 Conexión de alimentación

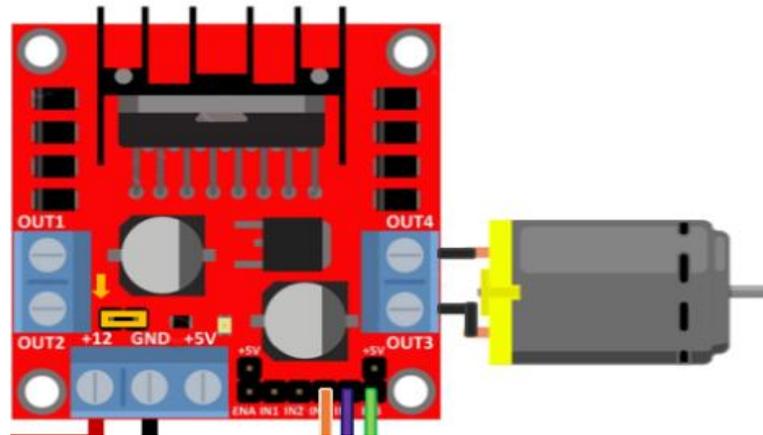
El módulo L298n se alimenta empleando o no el regulador LM7805 que viene en la placa esto se visualiza en el anexo 2.

El módulo ingresa la alimentación entre 6V a 12 voltios corriente continua (7,5V hasta 12V si se usa un 78M05).

Cuando la alimentación se encuentra activa, el pin asignado tendrá un voltaje con salida de 5V.

Cuando el jumper de selección de 5V se lo retira, el módulo permite el ingreso de una alimentación más extensa hacia los motores: de entre 12V a 35V. Usualmente esta alimentación tiene que ser idéntica a la parte de control.

Figura 3. 14 L298N conexión a un motor con las líneas de control.



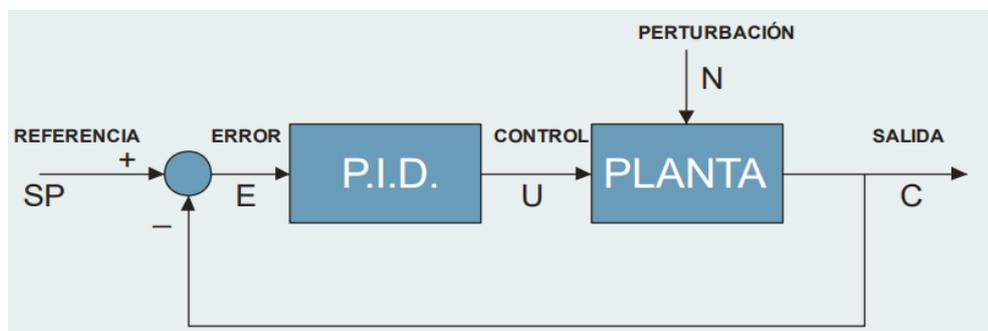
Fuente: (naylampmechatronics, 2020)

### 3.1.7 Sistemas de control en tiempo discreto

Los lazos de control discretos trabajan con señales digitales por lo tanto requieren de un tiempo de muestreo para poder realizar los nuevos cálculos para los valores que se van a usar para el bloque de modulación de ancho de pulso.

La herramienta de transformadas Z es el pilar importante para realizar los cálculos de un control digital partiendo de que se realiza el muestreo a una señal continua y con esa información se analiza en el dominio del tiempo mediante lo antes mencionado.

Figura 3. 15 Esquema básico PID continuo.

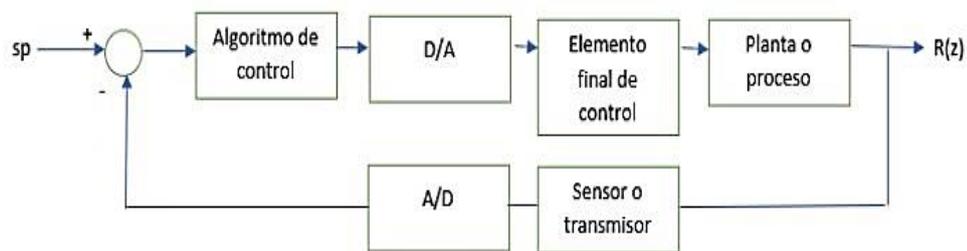


Fuente: Autor

Para la salida de la información el lazo de control se requiere de un bloque que realice la conversión inversa del proceso del sensor. A este proceso se le conoce como un convertidor digital analógico, y básicamente consiste en la modulación de ancho de pulso a una señal analógica. El proceso de conversión se lo puede realizar por medio de filtros pasa bajo para obtener una señal analógica continua o en su lugar enviando la modulación de ancho de pulso directamente al actuador.

Cómo se puede observar en la gráfica siguiente se tiene un diagrama de bloque donde se expone el funcionamiento de un lazo de control digital. Básicamente se dispone de los bloques de conversión análogo-digital y su bloque equivalente inverso. En el bloque del elemento final de control básicamente es el puente H se está usando proyecto, este dispositivo permite realizar la conexión de potencia entre el microcontrolador y maneja niveles de corriente mínimos y elevar a través del circuito integrado L298n, a corrientes de potencia para poder ser aplicados a un motor de corriente continua

Figura 3. 16 Esquema básico PID discreto.



Fuente: Autor

Para el desarrollo de los algoritmos para el control se basó en la base matemática del control continuo.

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad \text{Ec. (2. 3)}$$

Derivando de la ecuación 1 se tiene:

$$\frac{du(t)}{dt} = K_p \frac{de(t)}{dt} + K_i e(t) + K_d \frac{d^2 e(t)}{dt^2}$$

Se tiene las siguientes aproximaciones

$$\frac{du(t)}{dt} \approx \frac{\Delta e[k]}{T} = \frac{e[k] - e[k + 1]}{T} \quad \text{Ec. (2. 4)}$$

$$\frac{d^2 e(t)}{dt} \approx \frac{\Delta}{T} \left( \frac{\Delta e[k]}{T} \right) = \frac{\Delta}{T} \left( \frac{e[k] - e[k - 1]}{T} \right) = \frac{(e[k] - e[k - 1]) - (e[k - 1] - e[k - 2])}{T^2} \quad \text{Ec. (2. 5)}$$

$$\frac{d^2 e(t)}{dt} \approx \frac{(e[k] - 2e[k - 1] + e[k - 2])}{T^2} \quad \text{Ec. (2. 6)}$$

Reemplazando

$$\begin{aligned} \frac{u[k] - u[k - 1]}{T} &= K_p \frac{e[k] - e[k - 1]}{T} + K_i e[k] \\ &+ K_d \frac{(e[k] - 2e[k - 1] + e[k - 2])}{T^2} \end{aligned} \quad \text{Ec. (2. 7)}$$

Se obtiene el siguiente resultado.

$$u[k] = u[k - 1] + K_p(e[k] - e[k - 1]) + K_i T e[k] + K_d \frac{(e[k] - 2e[k - 1] + e[k - 2])}{T} \quad \text{Ec. (2. 8)}$$

Su implementación matemática se la puede observar en la figura siguiente, donde se materializa la ecuación anterior para el uso de control PID.

Figura 3. 17 Segmento de código para el control PID en el algoritmo.

```

304
305   for(uint8_t x=0;x<=18;x++){
306       rps[x]=rps[x+1];
307   }
308   rps[19]=cnt_int;
309   cnt_int=0;
310
311   float suma=0.0;
312   for(uint8_t x=0;x<=19;x++){
313       suma=suma+rps[x];
314   }
315   cnt=suma/19;
316   //cnt=rps[19];
317
318   PV=100*abs(cnt)/PV_MAX;
319
320
321   e[0] = 1.0*Sp-1.0*PV;
322   u[0] = u[1] + Kp*(e[0]-e[1]) + (Kp*0.1/Ti)*e[0] + Kp*Td*(e[0]-2*e[1]+e[2])/0.1;
323
324   e[1] = e[0];
325   u[0] = (u[0]>OUT_MAX)?(OUT_MAX):(u[0]);
326   u[0] = (u[0]<OUT_MIN)?(OUT_MIN):(u[0]);
327   u[2] = u[1];
328   u[1] = u[0];
329
330   duty = 200+800*u[0]/100;

```

Fuente: Autor

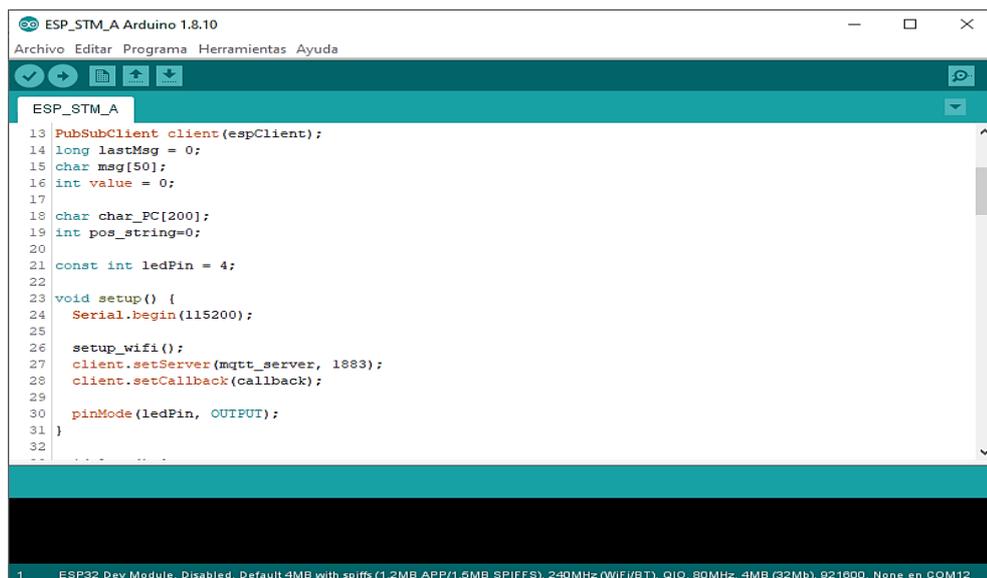
### 3.1.9 Programación de la Tarjeta ESP32

El módulo puede ser programado con el entorno de Arduino realizando previas instalaciones y configuraciones para su compatibilidad. Hay grandes ventajas al ser programado como lo antes mencionado ya que existe un gran respaldo de librerías desarrolladas por la comunidad de acceso libre.

Para realizar las configuraciones básicas del módulo WIFI se requiere establecer la velocidad frecuencia de trabajo, el tipo de partición, la velocidad del monitoreo serial entre varias opciones que de manera inicial vienen establecidas por defecto.

El módulo ESP32 puede trabajar de dos maneras como una estación que genere una señal Wifi o a su vez como un cliente que se conecta a una red Wifi. También existe la posibilidad de que las dos configuraciones estén vigentes para de esta manera generar algún tipo de solución para los proyectos que se requieran, por ejemplo, cuando la estación no tiene conectividad con ninguna red inalámbrica se puede ejecutar que se auto genere una red para que el operador a través de un dispositivo móvil pueda acceder internamente a las configuraciones y establecer el nombre de la red inalámbrica que debe conectarse.

Figura 3. 18 IDE Arduino para programar el ESP32.



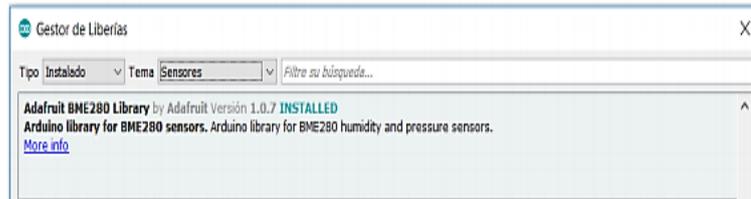
```
ESP_STM_A Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

ESP_STM_A
13 PubSubClient client(espClient);
14 long lastMsg = 0;
15 char msg[50];
16 int value = 0;
17
18 char char_PC[200];
19 int pos_string=0;
20
21 const int ledPin = 4;
22
23 void setup() {
24   Serial.begin(115200);
25
26   setup_wifi();
27   client.setServer(mqtt_server, 1883);
28   client.setCallback(callback);
29
30   pinMode(ledPin, OUTPUT);
31 }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582

```

Se observa en la figura 3.19 es la captura de la ventana del gestor de librerías propio IDE de Arduino. Está conformado por una caja de textos que se puede ingresar el nombre de la librería que el desarrollador desea realizar la búsqueda para que de forma inmediata se enlista en la parte inferior para proceder con la instalación.

Figura 3. 19 Gestor de librerías de Arduino.



Fuente: Autor

En la figura 3.20 se sintetiza el funcionamiento de la ESP32, donde su principal trabajo es la de tramitar los datos de la ESP32 hacia el bróker y del bróker hacia la STM32F4

Figura 3. 20 Algoritmo del funcionamiento de la ESP32.



Fuente: Autor

En la figura 3.21 presenta un segmento del código que es el algoritmo para la conectividad hacia el bróker, por lo tanto, está en la espera de recibir información en estructura de JSON proveniente de la STM32F4 para luego enviar los datos en un payload que es el contenido propio del mensaje enviado bajo el protocolo MQTT.

Figura 3. 21 Segmento de código de la ESP32 programado en el IDE de Arduino.

```

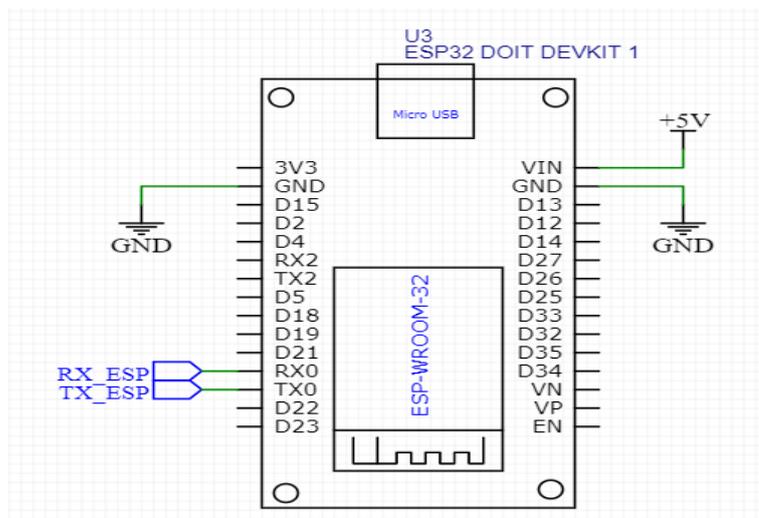
34 | if (!client.connected()) {
35 |     reconnect();
36 | }
37 | client.loop();
38 |
39 | if (Serial.available() > 0) {
40 |     char dat = Serial.read();
41 |     // Serial.write(dat);
42 |     // {"error": "10", "sensor": "15", "motor": "20"}
43 |
44 |     char_PC[pos_string] = dat;
45 |     pos_string++;
46 |
47 |     if (char_PC[0] == '{' && dat == '}') {
48 |         dat = '\n';
49 |         pos_string = 0;
50 |         Serial.println(char_PC);
51 |         client.publish("STM32/sts", char_PC);
52 |     }
53 |     if (dat == '\n') {
54 |         pos_string = 0;
55 |         for (int i = 0; i < sizeof(char_PC); i++) {
56 |             char_PC[i] = (char) 0;
57 |         }
58 |     }
59 | }

```

Fuente: Autor

El ESP32 dispone de varios pines que pueden cumplir muchos propósitos, uno de ellos sería la comunicación serial, en este caso, se usa para intercambiar datos con la STM32F4 para enviar y recibir los datos de las variables de la planta o como también el comportamiento del motor. Como podemos observar en la figura 3.22 es un diagrama donde está señalado los pines usados y sus funciones.

Figura 3. 22 Pines que son usados en la ESP32 para comunicarse con la STM32F4.



Fuente: Autor

### 3.2 CONECTIVIDAD AL BRÓKER MQTT

En la figura 3.23 se observa el segmento de código para la conectividad a un servidor de MQTT denominado Bróker, este código está hecho en el IDE de Arduino. Cuando se realiza la conexión satisfactoria el cliente se suscribe a un topic para recibir los datos del cliente que está gestionando los valores de las variables para el control de la planta.

Figura 3. 23 Segmento de código del algoritmo para conectividad al bróker MQTT.

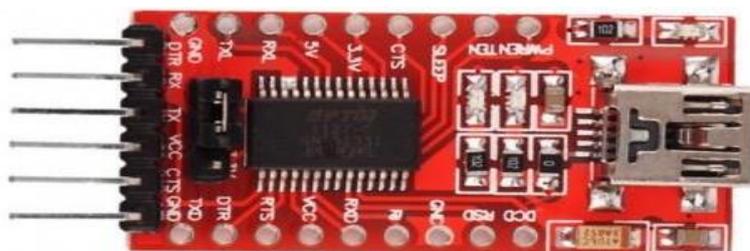
```
94 while (!client.connected()) {
95   Serial.print("Estamos intentado conectarnos al servidor MQTT...");
96   |
97   if (client.connect("STMCliente", mqttUser, mqttPassword)) {
98     Serial.println("connected");
99     // Subscribe
100    client.subscribe("STM32/cmd");
101
102   } else {
103     Serial.print("failed, rc=");
104     Serial.print(client.state());
105     Serial.println(" Intentando en unos 5 seg.");
106
107     delay(5000);
108   }
109 }
```

Fuente: Autor

#### 3.2.1 Salida de datos de la planta por comunicación serial para monitoreo

Para el realizar la comunicación del STM32F4 hacia la computadora se tiene varias opciones como la comunicación por USB o por un puerto serial a niveles TTL. Este último caso es donde se tiene una limitación que requiere hacer el uso de un convertidor de TTL a USB para que la computadora pueda recibir los datos. La figura 3.24 es un módulo para la conversión de datos de comunicación serial con niveles de voltaje TTL a la comunicación más usada en la actualidad que es el USB.

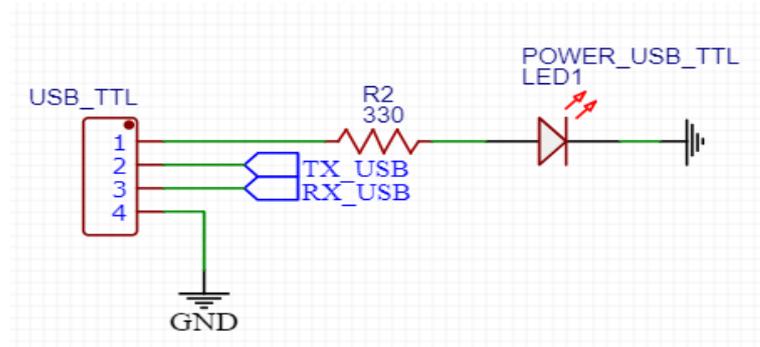
Figura 3. 24 Módulo USB a TTL.



Fuente: (Aghenta, L. O., & Iqbal, M. T 4(1), 57., 2020)

En la tarjeta de control donde se monta todos los módulos para la planta, está implementada de un conector USB\_TTL que es donde se conecta el módulo de comunicación USB-TTL. La tarjeta STM32F4 está configurada para que el puerto 3 de comunicación serial se conecta al módulo convertidor y este último se enlace hacia el computador del operador, para trabajo de manera local.

Figura 3. 25 Módulo USB a TTL.



Fuente: Autor

En la figura 3.29 es el segmento de código que está implementado en el STM32F407 para la salida de los datos de las variables del control PID para ser graficados en la computadora con el objetivo de la visualización por parte del operador local.

Figura 3. 26 Segmento de código de la salida de datos de las variables de control.

```

270     if(HAL_GetTick())>=tickActual+100){
271         tickActual = HAL_GetTick();
272         int a=Pv;
273         int b=e[0];
274         int c= u[0];
275         int d= Sp;
276         n=sprintf (buffer, "%d %d %d %d\n", a, b, c, d);
277         HAL_UART_Transmit(&huart2, buffer, n, 100);
278     }

```

Fuente: Autor

### 3.2.3 Recepción de las constantes para el control PID

Los datos que son transmitidos en el Payload del MQTT se basan en código ASCII, cada carácter tiene su representación entera en el código ASCII. Pongamos el caso del número '2', es el 50 en el ASCII; pero si se quisiera convertir a entero se debería regresar 2.

Justo aquí podemos aplicar un algoritmo; el cual es restar del dígito el número '0' que, bueno, al final no es un número, es un carácter. El '0' tiene el código ASCII 48, entonces cuando hacemos:

$$'2' - '0'$$

El lenguaje opera los caracteres como si de su representación ASCII se tratara; así:

$$50 - 48$$

¿Y cuánto es  $50 - 48$ ? exacto, 2 pero como entero. Y esto funciona para números del 0 al 9, porque en el código ASCII el 0 es el 48, el 1 el 49 y así sucesivamente.

Figura 3. 27 Segmento de código de la recepción de los datos para el control PID.

```

128 void DatosManual(uint8_t* cadena){
129     //Sp=70;
130     //Kp=0.5;
131     //Ti=0.5;
132     //Td=0.00001;
133
134     char *psc_ptr=strstr(cadena, "stm");
135     psc_ptr=psc_ptr+6;
136     Sp=(*psc_ptr-48)*10;
137     psc_ptr++;
138     Sp=Sp+*psc_ptr-48;
139
140     psc_ptr=psc_ptr+9;
141     Kp=*psc_ptr-48;
142     psc_ptr+=2;
143     Kp=Kp+((*psc_ptr-48)*0.1);
144     psc_ptr++;
145     Kp=Kp+((*psc_ptr-48)*0.01);
146
147     psc_ptr=psc_ptr+9;
148     Ti=*psc_ptr-48;
149     psc_ptr+=2;
150     Ti=Ti+((*psc_ptr-48)*0.1);
151     psc_ptr++;
152     Ti=Ti+((*psc_ptr-48)*0.01);
153     psc_ptr++;
154     Ti=Ti+((*psc_ptr-48)*0.001);
155
156     psc_ptr=psc_ptr+9;
157     Td=*psc_ptr-48;
158     psc_ptr+=2;
159     Td=Td+((*psc_ptr-48)*0.1);
160     psc_ptr++;
161     Td=Td+((*psc_ptr-48)*0.01);
162     psc_ptr++;
163     Td=Td+((*psc_ptr-48)*0.001);
164     psc_ptr++;
165     Td=Td+((*psc_ptr-48)*0.0001);

```

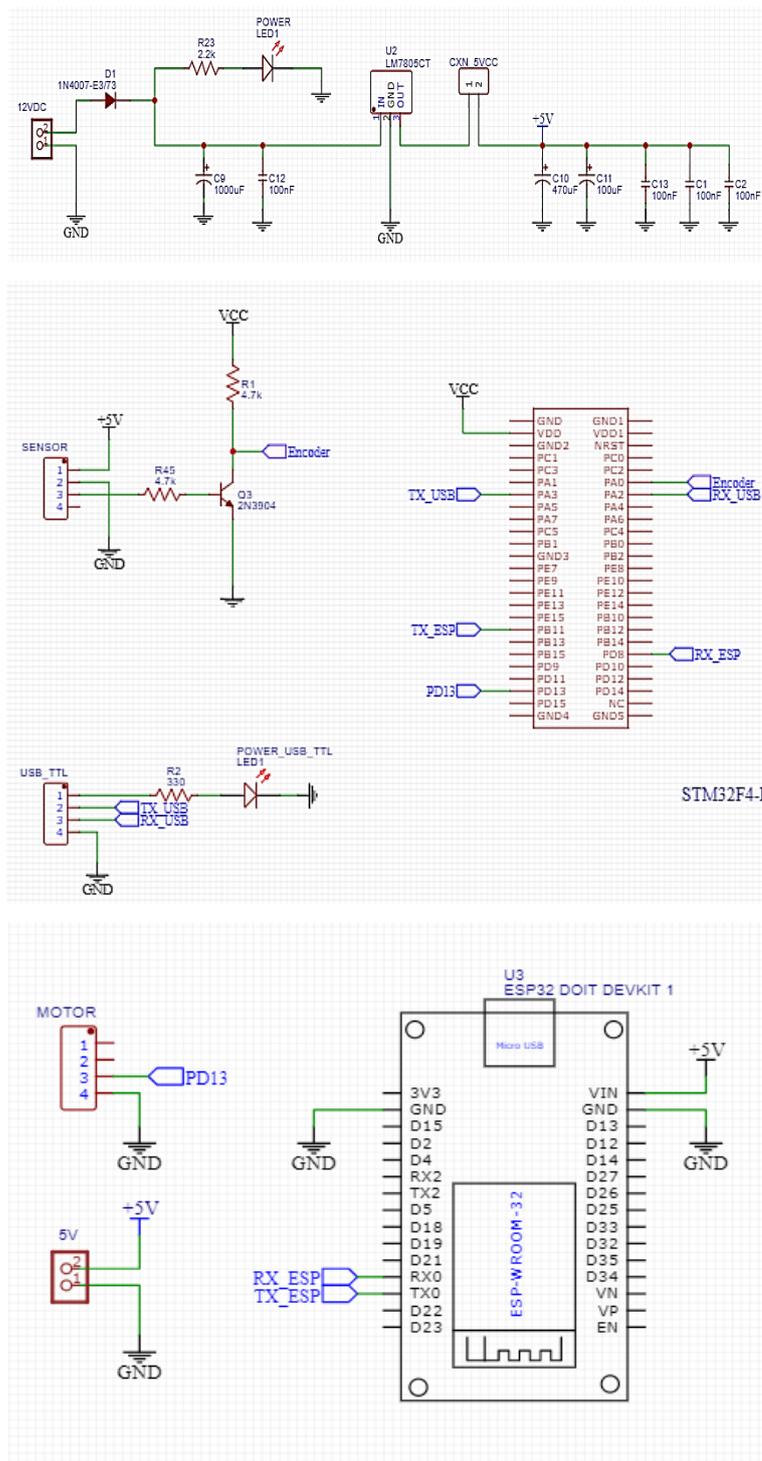
Fuente: Autor

En la figura 3.35 es un segmento de código que se recibe por la comunicación serial los datos que provienen del bróker MQTT por medio del ESP 32, estos datos están en código ASCII y se debe convertir a entero, como se mencionó anteriormente.

### 3.2.4 Diseño de la tarjeta de control

En la figura 3.28 se presenta el diseño del circuito lógico usado para organizar los distintos módulos y elementos electrónicos para el diseño del control en Proteus.

Figura 3. 28 Circuito lógico de la planta.



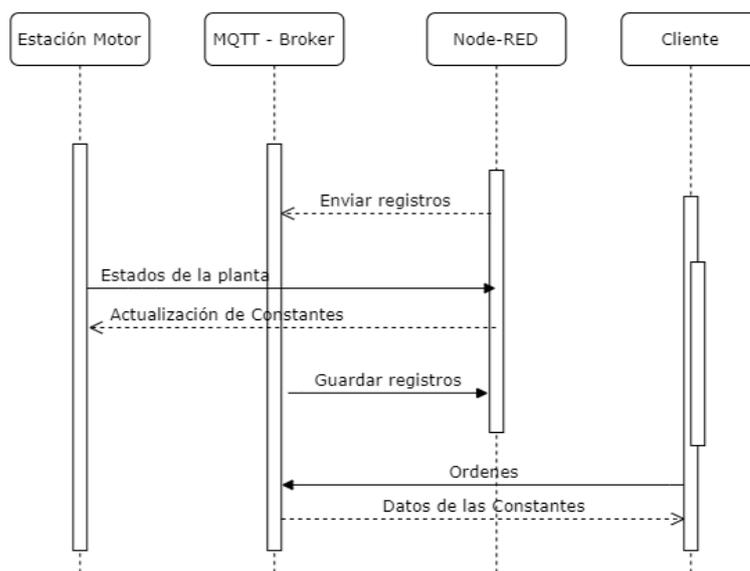
Fuente: Autor

En el Anexo 3 y el Anexo 4 se presenta los resultados del diseño de circuito impreso.

### 3.2.6 Integración de MQTT

El bróker está implementado en un servidor con sistema operativo Ubuntu, la instalación fue realizada con el paquete mosquitto que es el más estable y usado para generar servicios de comunicación de mensajes MQTT. En la figura 3.32 se puede observar un diagrama UML donde se puede entender el comportamiento de las interacciones del cliente con el bróker, para el envío de datos de la página web hasta la planta para la actualización de los datos del controlador PID.

Figura 3. 29 Diagrama UML de la interacción del cliente y bróker MQTT.



Fuente: Autor

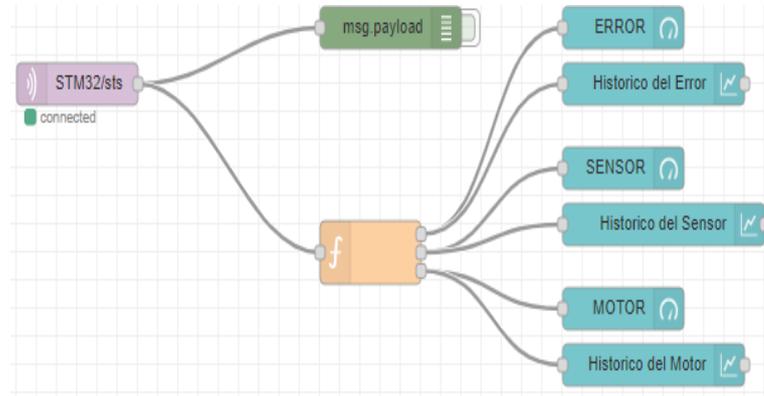
### 3.2.7 Nodo de recepción y representación de Node-red

En Node-Red se implementó los nodos de recepción de los datos enviados por la planta, los datos corresponden al error, sensor y motor, todos ellos representados en porcentajes. El servidor Node-Red se implementa en un VPS Ubuntu con la IP: 207.244.250.117 con el puerto número 1880

Los datos se grafican dos métodos el primero es el tipo indicador analógico y el segundo un historial de datos en función del tiempo como se puede observar en la figura 3.30 se visualiza Node-Red con los nodos para recibir los datos de la planta, el bloque de la izquierda es donde recibe los mensajes MQTT del topic "STM32/sts" y

son procesados en la función en el centro para ser presentada en los graficadores de la izquierda.

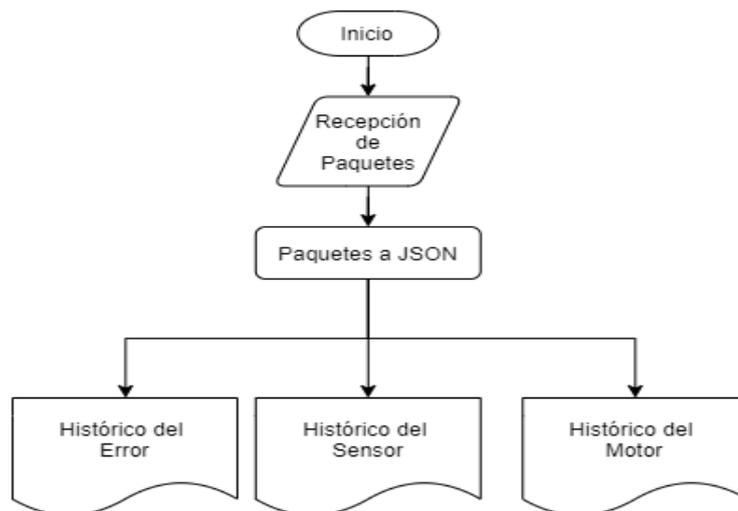
Figura 3. 30 Node-Red con los nodos para recibir los datos de la planta.



Fuente: Autor

El formato JSON es el usado para el intercambio de datos entre los clientes MQTT, para tener una mayor facilidad en la estructuración, como también para la extracción de los datos. Los servicios que emiten información no requieren estar en el mismo lenguaje, lo que quiere decir que, el emisor puede estar programado en lenguaje C++ y el receptor en Node-Red, por tanto, cada uno de ellos maneja su librería.

Figura 3. 31 Diagrama de flujo de la planta para la recepción de paquetes.



Fuente: Autor

### 3.2.8 Nodo de transmisión desde Node-Red

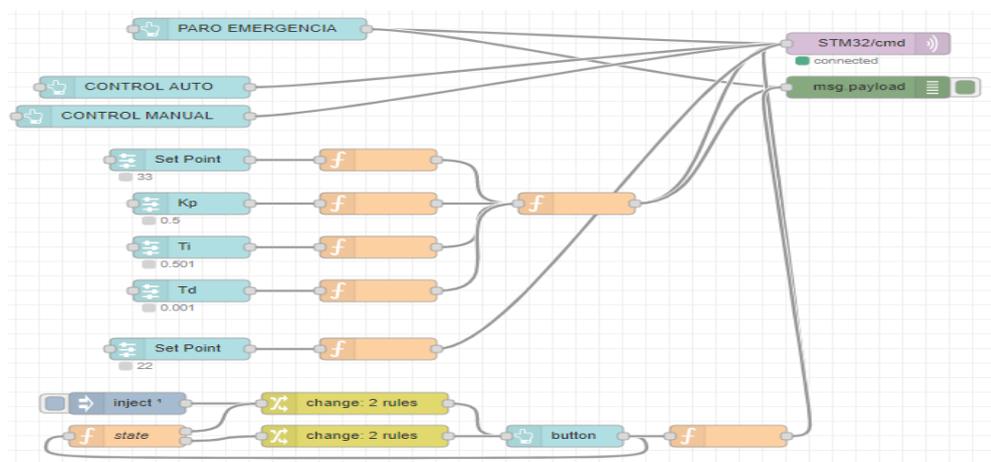
La transmisión de datos básicamente es cuando se genere el movimiento de los Slider por el operador en la pantalla de operación. Las sliders tienen establecidos rangos según la variable a ser alimentados en la planta, y se tiene los siguientes controles que el operador puede cambiar:

- Set point
- Constante de proporcionalidad
- Constante de integración
- Constante de derivación

También tiene controles como:

- Encendido del motor
- Apagado del motor
- Paro de emergencia

Figura 3. 32 Control de los Slider para el cambio de datos de la planta.

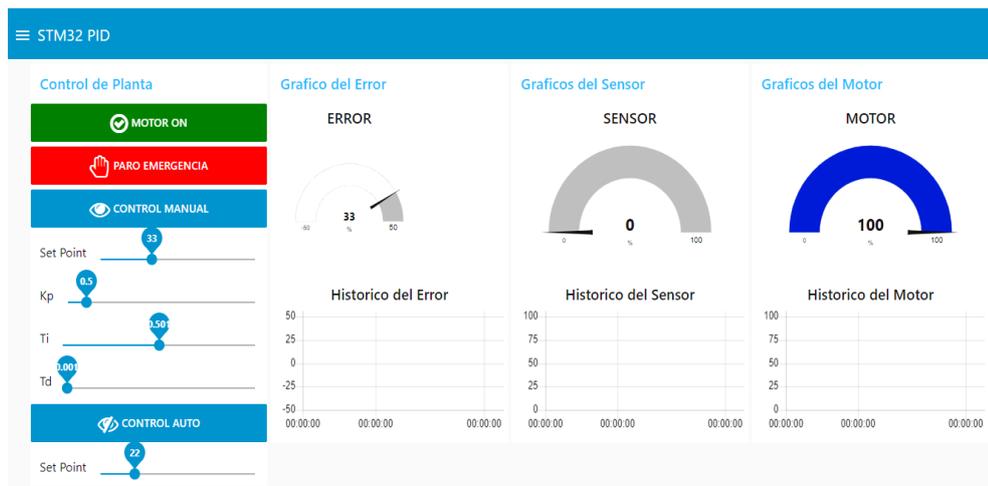


Fuente: Autor

### 3.2.9 Telemetría para el control de constantes

En la siguiente figura se tiene la pantalla que el operador puede acceder y visualizar para realizar la supervisión de la planta de control PID. En el lateral izquierdo se encuentra el panel para el control del motor, donde se puede encender o apagar. Seguido se dispone del botón de paro de emergencia y en la parte inferior la slider para el control de las constantes para la planta.

Figura 3. 33 Pantalla para el control de telemetría de la planta.

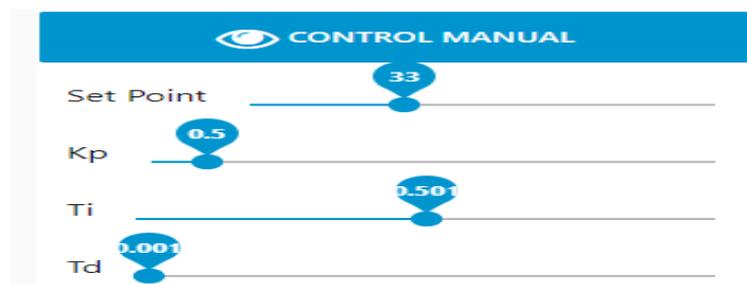


Fuente: Autor

### 3.2.10 Control manual

En esta sección de la pantalla de control, se dispone de la slider para el control de las variables, en donde el operador puede deslizar para cambiar los datos de los distintos datos que se desee cambiar. Las órdenes se envían a la planta tan pronto como deje de presionar el mouse en el movimiento de deslizamiento.

Figura 3. 34 Slider para el control manual.



Fuente: Elaboración Propia

### 3.2.11 Control automático

Para el control automático es más sencillo para el operador tan solo debe deslizar la slider del set point para que la planta apunte a nuevo nivel de control, manteniendo los valores de control PID.

## CAPÍTULO 4

### ANÁLISIS Y RESULTADOS

Este capítulo abarca todos los resultados del proyecto de titulación, además se observa los datos obtenidos de la planta y se analiza cada grafica obtenida con el fin de dar aval y cumplimiento a los objetivos propuestos al inicio del documento y sacar las respectivas conclusiones del proyecto establecido.

#### 4.1 CARACTERÍSTICAS DEL MOTOR DC

El motor de corriente directa tiene las siguientes características.

Tabla 4. 1 Características del motor DC *Transmotec 775-5520F-CC*.

Características	Valor
Voltaje	12 [V]
Corriente	0,2 [A]
Revoluciones por minuto	4000 rpm

Fuente: Autor

Figura 4. 1 Fuente de poder para el motor DC y medición de voltaje y corriente.



Fuente: Autor

## 4.2 PRUEBAS DE ESCALÓN UNITARIO Y ESTABILIZACIÓN DEL MOTOR

Para las pruebas de escalón unitario se usó los siguientes valores de las constantes del control PID.

Tabla 4. 2 Valores para las constantes del control PID usadas.

<b>Característica</b>	<b>Valor [s]</b>
Kp	0,5
Ti	0,501
Td	0,001

Fuente: Autor

Dando como resultado la siguiente gráfica que se puede visualizar de manera local por el operador, donde se tiene un muestreo cada 100 milisegundos o en su defecto 10 Hz. La descripción de los colores son las siguientes:

Tabla 4. 3 Descripción de las señales del gráfico 3.2.

<b>Color</b>	<b>Descripción</b>
Rojo	Error
Amarillo	Set Point
Azul	Sensor
Verde	Motor

Fuente: Autor

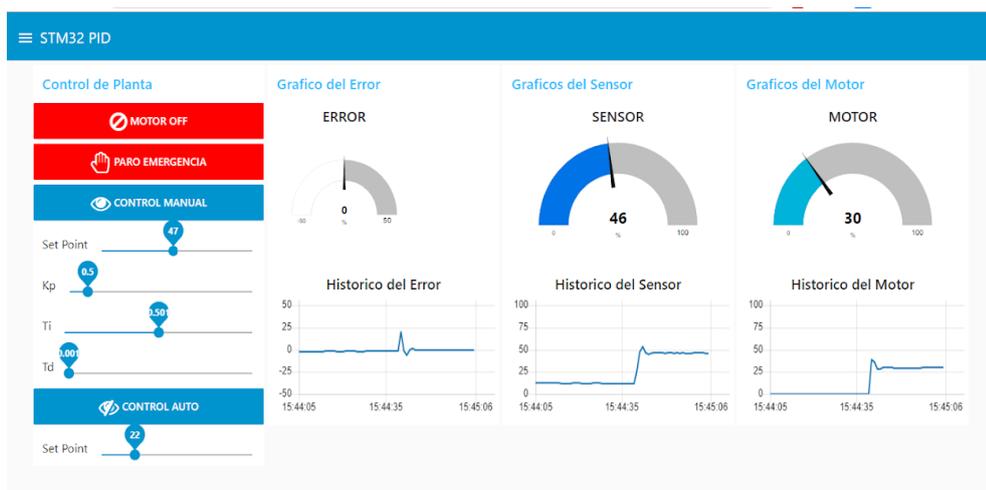
Figura 4. 2 Gráfico de las señales de la planta ante un escalón unitario.



Fuente: Autor

Se visualiza en la figura 4.2 se dispone de 4 señales que son las variables de la planta en pleno funcionamiento, la señal naranja, verde, rojo y azul son el setpoint, señal de salida, señal de error, y señal del sensor. En la muestra 5915 se genera un cambio del setpoint por el operador remoto por medio de la página cliente soportada por Node-Red. El escalón unitario que genera el cambio inmediato del set point genera un comportamiento de la señal del motor, el tiempo de estabilización es de 5.3 segundos esto se refleja en el resultado del modelamiento de señal obtenida de sistema que es del 94.91% como se puede ver figura 4.10.

Figura 4. 3 Interfaz de Node-Red para el operador.



Fuente: Autor

El operador remoto que usaría la página web observará lo que se muestra en la figura 4.3 Como se describió anteriormente en la izquierda está la consola de control donde se puede configurar los valores de las constantes para el control PID, y en las gráficas se puede observar el comportamiento de las señales para ver el comportamiento de la planta ante cualquier cambio. La velocidad de muestreo que se presenta en las señales que están en función del tiempo están a 1 Hz, por lo tanto, la visualización se puede ver un poco disminuida de detalles del comportamiento de la planta, pero para ello existe en la tarjeta de control un módulo para la comunicación serial para poder visualizar de manera local el comportamiento de la planta con un muestreo de 100Hz.

Figura 4. 4 Tarjeta electrónica para el control de la planta.



Fuente: Autor

En la implementación de la parte electrónica de la planta de control PID, se procedió a instalar en una base plástica, donde se atornilla el circuito impreso como se muestra la figura 4.4. La placa electrónica de circuito impreso de color marrón es el circuito impreso que se realizó para la soldadura de los módulos como son:

- STM32F4 Discovery
- ESP32
- Módulo de comunicación serial USB
- Sensor tacómetro
- Fuente de energía
- Puente H

Figura 4. 5 Motor que es el actuador de la planta.



Fuente: Autor

El motor de corriente directa, se le acoplo un disco con 100 perforaciones que conjuntamente con el sensor de infrarrojo, están configurados para la adquisición y medición de las revoluciones del motor. El motor está sujeto con amarras plásticas por sus características cilíndricas y con una base plástica para la mejor sujeción con la caja base. Para el filtrado de ruido eléctrico que se produce por las características mecánicas del motor se le coloca un anillo de ferrita para reducir el mencionado ruido eléctrico.

Figura 4. 6 Vista superior de la planta.



Fuente: Autor

Se observa por medio de la figura 4.6 una vista superior de la implementación de la planta, en la parte superior está la tarjeta electrónica con sus módulos, mientras que la parte inferior se encuentra el motor. La alimentación de 12 voltios se lo hace por una conexión en el interior de la caja.

#### **4.3 MODELAMIENTO**

Para el modelamiento de la planta se usó el software MATLAB, que contiene un conjunto de herramientas propias para el trabajo de sintonización y modelado para aplicaciones PID. En primer lugar, se debe realizar la adquisición de datos desde la tarjeta STM32F4, y se debe de colocar un nuevo código de funcionamiento para la función principal del código. La nueva función está diseñada para que el motor inicie apagado por 5 segundos y después el microcontrolador realiza la activación del motor a una potencia del 70 por ciento.

Figura 4. 7 Segmento de código para la identificación de planta.

```

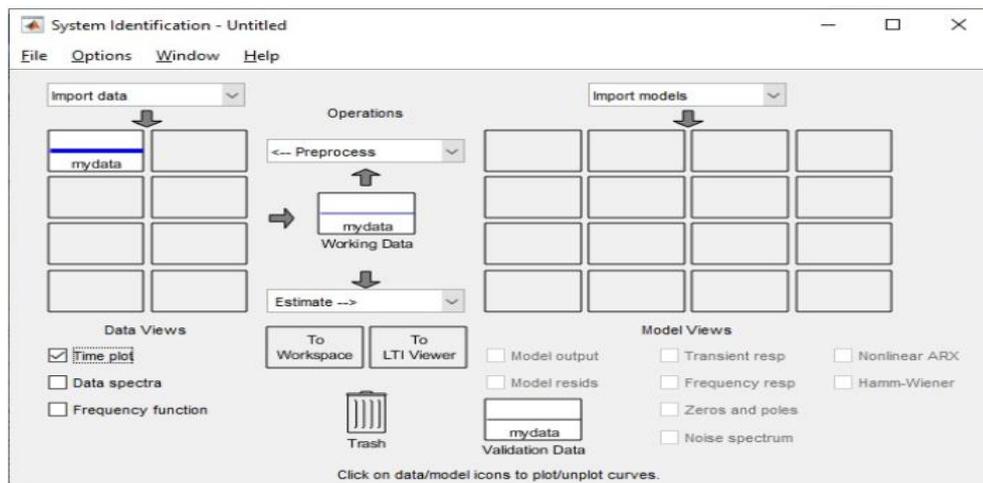
260 uint16_t flag_tiempo=0;
261 __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_2, 0);
262 while (1)
263 {
264     HAL_Delay(10);
265     int a=abs(cnt);
266     n=sprintf (buffer, "%d\n", a);
267     HAL_UART_Transmit(&huart2, buffer, n, 100);
268
269     flag_tiempo++;
270     if(flag_tiempo==500){
271         __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_2, 750);
272     }
273     if(flag_tiempo==1500){
274         while(1){}
275     }
276 }

```

Fuente: Autor

Los datos que se van generando del tacómetro se van enviando en tiempo real por el puerto de comunicación serial con la ayuda del módulo de conversión TTL a USB. Con un monitor de puerto serial en el computador se recibe todos los datos para después pasarlos un arreglo unidireccional de memoria de MATLAB.

Figura 4. 8 Herramienta de MATLAB para el modelamiento de la planta System Identificación.

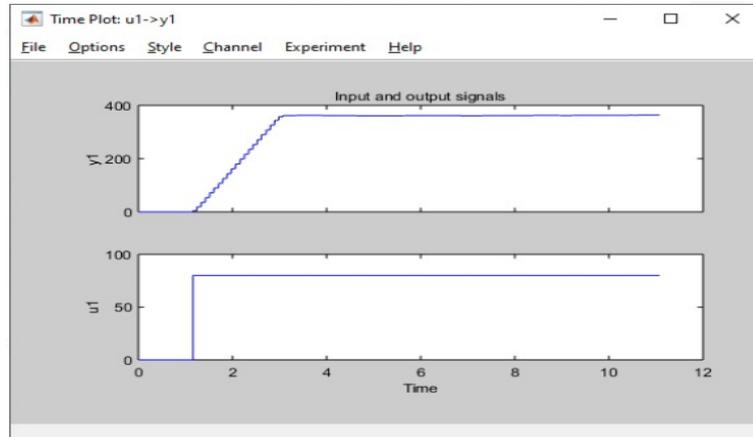


Fuente: Autor

En la figura 4.8 se tiene el System Identification que se usa para extraer el modelo del motor que se está usando. Como ya contamos con los datos del motor en una variable de MATLAB se procede a importar en la herramienta. Para la verificación de los datos que se han importador y tener la certeza de una buena adquisición de datos se mapea en función del tiempo los datos y se presentarán los gráficos como se puede observar en la figura 4.9, donde el gráfico superior es el comportamiento del motor en función

del tiempo bajo las condiciones de un escalón unitario. Y el gráfico inferior es el escalón unitario.

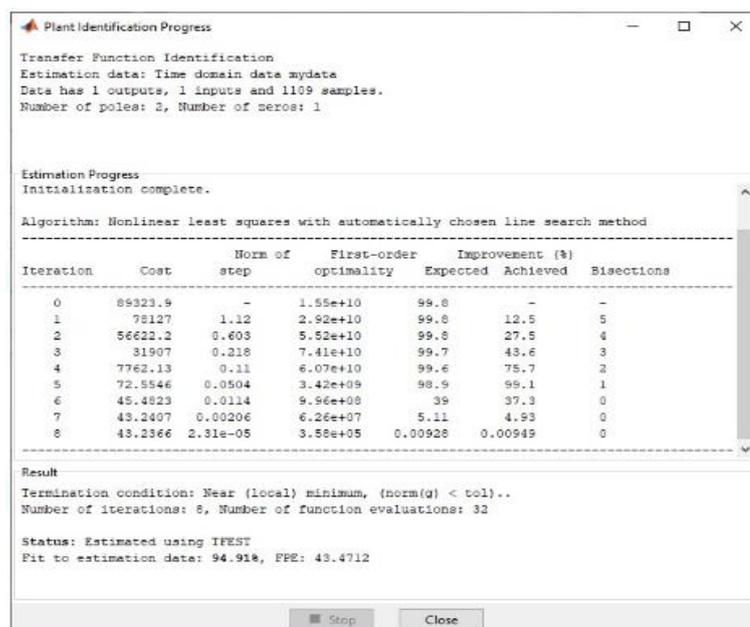
Figura 4. 9 Adquisición de datos de la planta para el modelamiento.



Fuente: Autor

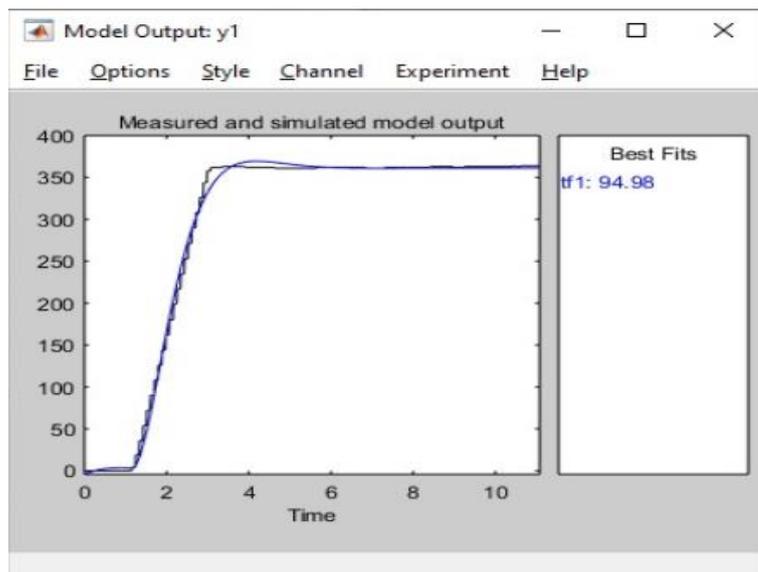
Para el proceso de modelamiento seleccionamos la opción para generar el modelo más aproximado al comportamiento del motor, dándonos como resultado lo que se puede observar en la figura 4.10 que nos indica que tiene una aproximación del 94.91% la función de transferencia.

Figura 4. 10 Resultados del modelamiento con una estimación de 94.91%



Fuente: Autor

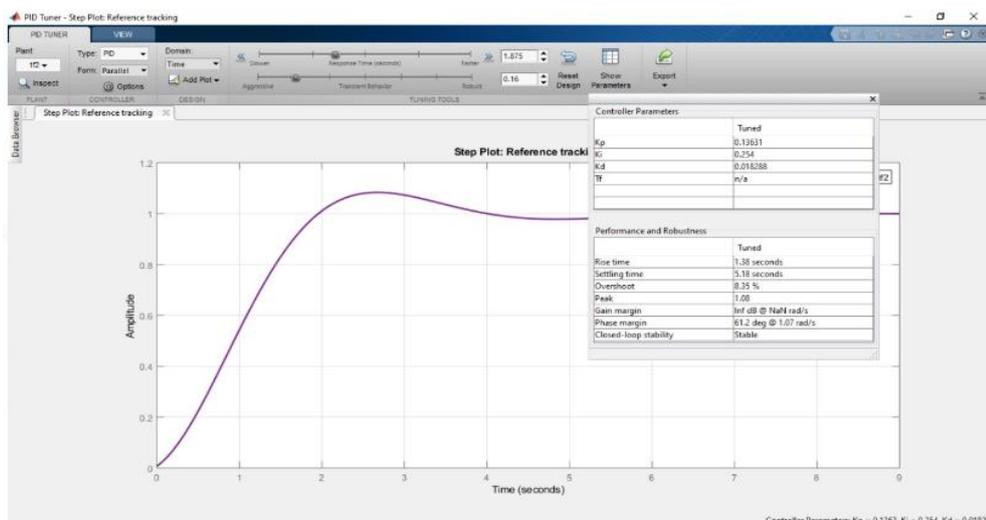
Figura 4. 11 Resultados del modelamiento con una estimación de 94.91%



Fuente: Autor

Finalmente, con el sintonizador PID podemos importar los datos del modelo que ya hemos generado con la herramienta anterior, ahora podremos generar un control PID con las características más cercanas al comportamiento que necesitamos. Para efectos prácticos en la figura 4.12 se denota las constantes usadas para el control PID de nuestra planta.

Figura 4. 12 Resultado de la sintonización PID.

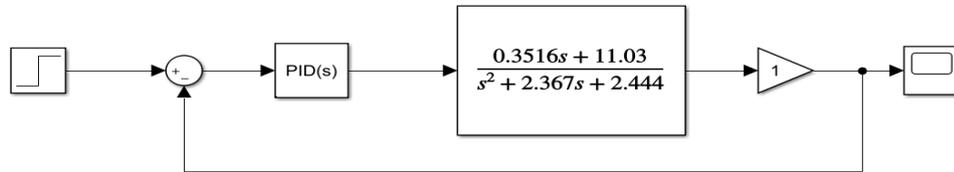


Fuente: Autor

Ya con los valores  $K_p$ ,  $K_i$ ,  $K_d$ , obtenidos en el modelamiento de la planta, se puede generar la función de transferencia usando simulink de Matlab como se ve en la imagen

4.13 generamos nuestra función mediante diagrama de bloques y se observa el PID sintonizado en un osciloscopio digital proporciona la aplicación.

Figura 4. 13 Función Transferencia de la sintonización PID.



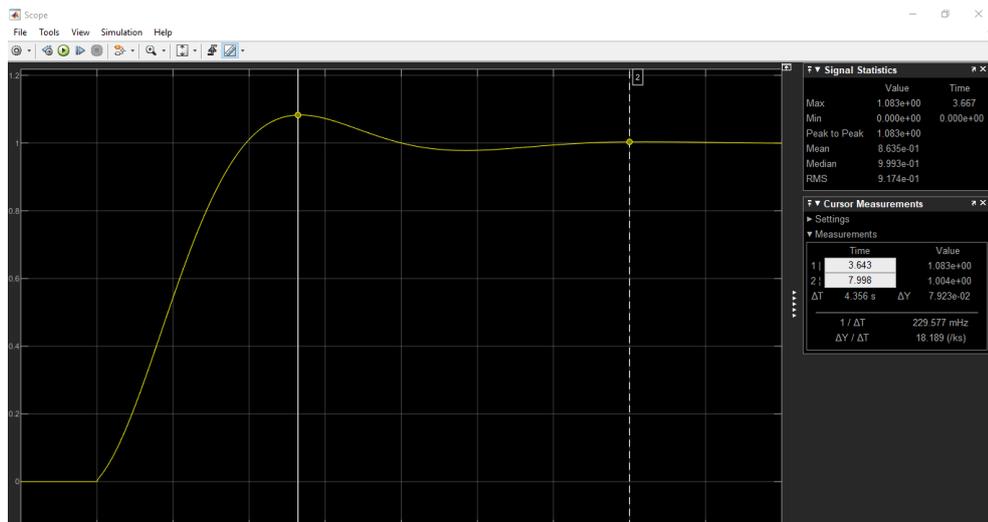
Fuente: Autor

La función de transferencia es:

$$\frac{0,3516s + 11,03}{s^2 + 2,3667s + 2,444} \quad \text{Ec (4. 1)}$$

Finalmente, se verifica el resultado de la sintonización realizada, la misma que con los valores ingresados se determina que la función de transferencia es la idéntica, tanto en la sintonización como en la aplicación de simulink.

Figura 4. 14 Resultado de la sintonización PID de la velocidad en función del tiempo en simulink.



Fuente: Autor

## CONCLUSIONES

Al desarrollar el sistema de control de lazo cerrado se pudo determinar que se puede implementar un control PID con la tarjeta de desarrollo STM32F4 Discovery con el lazo de control con una estabilización de la señal de control en 5.3 segundos y una estimación del 94.91% con respecto a la función de transferencia.

El protocolo que se usó fue el MQTT que brinda una conexión de bajo consumo de datos y tiempo de latencia menor a 100ms según las mediciones que se realizaron para el tipo de trabajo que se realizó, se configuró para tener actualizado las señales de control de la planta a una velocidad de 1 Hz para la dashboard de Node-Red, pero de manera local se puede visualizar los gráficos de las variables de control a una velocidad de 100Hz.

En el desarrollo del trabajo se usó la herramienta de MATLAB con el propósito de realizar el modelamiento de la planta usando el método de Ziegler Nichols por lazo abierto donde el controlador está en modo manual para desarrollar un escalón unitaria en la salida del control PID.

Las velocidades máximas para realizar el monitoreo de las señales de control en el dashboard es 10Hz, y este valor no varía en función del ancho banda del acceso a internet de la planta, sino se debe a la limitación del bróker MQTT y las características del VPS en memoria RAM.

En la comunicación desde la estación de control hasta la página de monitoreo se logró tener un muestreo de las variables del control PID a una velocidad de 1 Hz con el objetivo de no saturar de datos o posibles pérdidas de paquetes de información para optimizar el rendimiento de la ejecución del programa diseñado

## RECOMENDACIONES

El proyecto se desarrolló en un contexto que puede abarcar muchas aplicaciones de control que en general pueden ser aplicados en la industria se podría considerar usar esta misma infraestructura para lograr la implementación de otros métodos de control.

En la comunicación de datos entre el operador remoto y la planta se usó la comunicación MQTT que brinda mucha estabilidad para este tipo de aplicaciones, pero se podría considerar una comunicación usando socket TCP o UDP para generar un streaming de datos para la visualización de las gráficas con mayor detalle, para brindar mejores datos para la toma de decisiones del operador.

En la maqueta se dispone la conexión de una fuente de alimentación que permite el funcionamiento de la etapa de potencia, si esta se la sustituye por una fuente variable se puede incorporar perturbaciones externas para ver el comportamiento de la planta ante cambios de voltaje hacia el motor

## REFERENCIAS

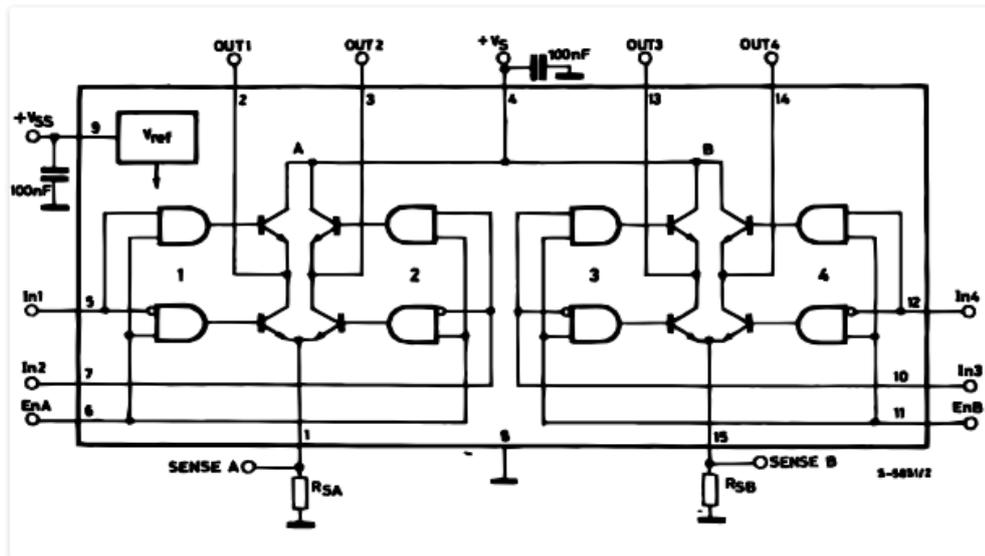
- Aghenta, L. O., & Iqbal, M. T. (2020). Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard and MQTT protocol. *AIMS Electronics and Electrical Engineering*, 4(1), 57.
- Bigtronica. (2020). <https://www.bigtronica.com>. Obtenido de <https://www.bigtronica.com/poblado/fuentes-de-alimentacion/10-adaptador-de-voltaje-12v-1a.html>
- Bonilla-Fabela, I., Tavizon-Salazar, A., Morales-Escobar, M., Guajardo-Muñoz, L. T., & Laines-Alamina, C. I. (2016). IoT, el internet de las cosas y la innovación de sus aplicaciones. *Vinculatégica efan*. (1), 2313-2340.
- Cedeño Nuñez, P & Sanchez Paredes T. (2019). Diseño de una red iot basada en sensores para monitorear el nivel de agua en un reservorio de ESPOL. (*Bachelor's thesis, Espol*).
- Chuquín Balseca S. L. & Pachacama Durán E. P. (2019). Control robusto de un motor AC mediante WNCS (Wireless Network Control System). (*Bachelor's thesis*).
- De La Cal Calleja, J. (2019). Control Domótico basado en el protocolo MQTT.
- Ding Yi, Fan Binwen, Kong Xiaoming, & Ma Qianqian. (2016). Design and implementation of mobile health monitoring system based on MQTT protocol. 2016. *IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, .
- Duarte, M. P., & Pérez, I. M. (2017). Programación en PHP a través de ejemplos. *Departamento de Lenguajes y Sistemas Informáticos. Universidad de Cádiz*.
- García Palomino, I., & Hinojosa Pérez, F. J. (2019). Desarrollo de red de sensores de irradiación solar.

- García Soria, A. (2017). Plataforma domótica basada en Raspberry Pi y el protocolo MQTT. Guellouz, S., Benzina, A., Khalgui, M., & Frey, G. (2016). *Reconfigurable Function Blocks : Extension to the Standard IEC 61499*.
- Houimli, M., Kahloul, L., & Benaoun, S. (2017). Formal Specification , Verification and Evaluation of the MQTT Protocol in the Internet of Things. 2017 . *International Conference on Mathematics and Information Technology*, 214–221.
- Ibrahim, I. N., & Al Akkad, M. A . (2016). Exploiting an intelligent fuzzy-PID system in nonlinear aircraft pitch control. 2016 International Siberian Conference on Control and Communications. *SIBCON 2016*. Obtenido de Proceedings.<https://doi.org/10.1109/SIBCON.2016>.
- Lekić, M., & Gardašević, G. (March de 2018). IoT sensor integration to Node-RED platform. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 1-5.
- life.augmented, S. (2020). <https://www.st.com/>. Obtenido de [https://www.st.com/content/st\\_com/en/products/ecosystems/stm32-open-development-environment/stm32cube.html](https://www.st.com/content/st_com/en/products/ecosystems/stm32-open-development-environment/stm32cube.html)
- Moreno Cerdà. (2018). Demostrador arquitectura publish/subscribe con MQTT . *(Bachelor's thesis, Universitat Politècnica de Catalunya)*.
- naylampmechatronics. (2020). <https://naylampmechatronics.com>. Obtenido de [https://naylampmechatronics.com/blog/11\\_Tutorial-de-Uso-del-M%C3%B3dulo-L298N.html](https://naylampmechatronics.com/blog/11_Tutorial-de-Uso-del-M%C3%B3dulo-L298N.html)
- Oh, S. C., Kim, T. H., & Kim, Y. G . (2018). Implementation of factory monitoring system using MQTT and Node-RED. *The Journal of The Institute of Internet, Broadcasting and Communication*, 18(4), 211-218.
- Osathanunkul, K., Hantrakul, K., Pramokchon, P., Khoenkaw, P., & Tantitharanukul, N. . (2017). Configurable Automatic Smart Urinal Flusher based on MQTT Protocol.

- Pillajo Angos, C. G. (2017). Implementación de un controlador predictivo basado en eventos para un sistema de control en red inalámbrico.
- Rao, K Vaghela, D. J & Gojiya, M. V. (July de 2016). Implementation of SPWM technique for 3- $\Phi$  VSI using STM32F4 discovery board interfaced with MATLAB. *In 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Syst.*
- RIFFO GUTIÉRREZ, C. A. (2018). ANÁLISIS COMPARATIVO DEL CONTROL DE VELOCIDAD DE UN MOTOR DC ENTRE UN CONTROLADOR DIFUSO Y UN CONTROLADOR PID A TRAVÉS DE SIMULACIÓN EN MATLAB.
- Saeteros Ortiz, C. M. (2018). Implementación de un control PID Fuzzy la regulación de nivel del módulo MPS PA Compact workstation mediante la tarjeta STM32F4 Discovery. (*Bachelor's thesis*).
- Salazar, J., & Silvestre, S. (2016). Internet de las cosas. Techpedia. *České vysoké učení technické v Praze Fakulta elektrotechnická.*
- Yacchirema Vargas, D. C. (2019). Arquitectura de Interoperabilidad de dispositivos físicos para el Internet de las Cosas (IoT) .
- Zhen, Z., & Yan, G. A. O . (2013). Zhen, Z., & Yan, G. A. O. (2013). Design of the Fuzzy PID Controller for the Hot Runner Temperature Control System. *Proceedings of the 32nd Chinese Control Conference*, 3464-3469.

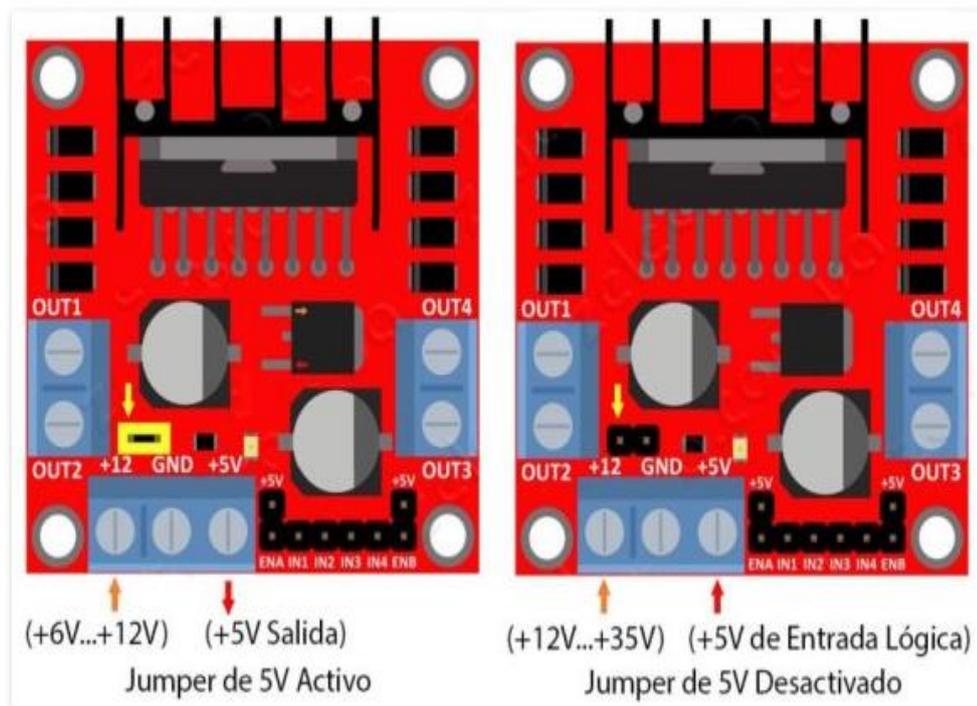
## ANEXOS

Anexo 1 Diagrama de circuito del circuito integrado L298.



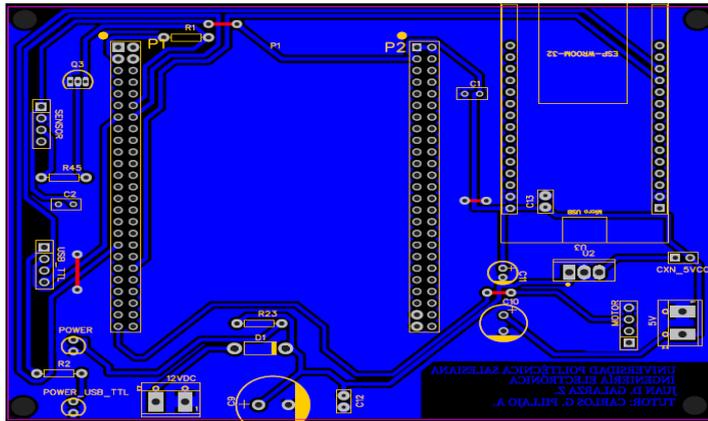
Fuente: (naylampmechatronics, 2020)

Anexo 2 L298n conexión para la fuente para el bloque de potencia.



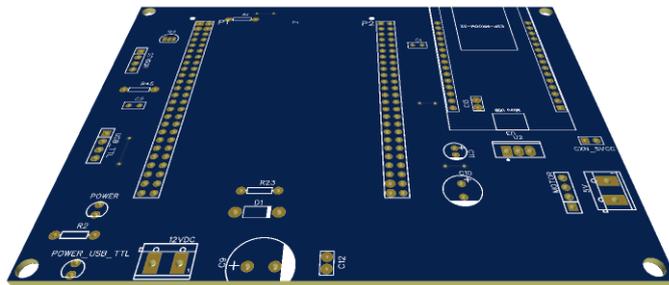
Fuente: (naylampmechatronics, 2020)

Anexo 3 Circuito impreso PCB



Fuente: Autor

Anexo 4 Circuito impreso PCB



Fuente: Autor