

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO**

**FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,  
INFORMÁTICA Y MECÁNICA**

**ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**



**TESIS**

**DISEÑO E IMPLEMENTACION DE UN SISTEMA AUTOMATIZADO DE  
CLORACION CON MONITOREO REMOTO UTILIZANDO REDES IOT PARA LA  
COMUNIDAD DE SAN PEDRO – CANCHIS - CUSCO**

**PRESENTADO POR:**

**Br. PAUL ABRAHAN MONZON CCORIMANYA**

**PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO ELECTRÓNICO**

**ASESOR:**

**Mg. Ing. JORGE LUIS ARIZACA CUSICUNA**

**CUSCO - PERU**

**2024**

## INFORME DE ORIGINALIDAD

(Aprobado por Resolución Nro.CU-303-2020-UNSAAC)

El que suscribe, **Asesor** del trabajo de investigación/tesis titulada: **DISEÑO E IMPLEMENTACION DE UN SISTEMA AUTOMATIZADO DE CLORACION CON MONITOREO REMOTO UTILIZANDO REDES IOT PARA LA COMUNIDAD DE SAN PEDRO - CANCHIS - CUSCO**

Presentado por: **PAUL ABRAHAN MONZON CCORIMANYA** DNI N° **74214006**

presentado por: ..... DNI N°: .....

Para optar el título profesional/grado académico de **INGENIERO ELECTRÓNICO**

Informo que el trabajo de investigación ha sido sometido a revisión por **...2...** veces, mediante el Software Antiplagio, conforme al Art. 6° del **Reglamento para Uso de Sistema Antiplagio de la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de **...9.....%**.

**Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis**

| Porcentaje     | Evaluación y Acciones                                                                                                                                                                                                                                                         | Marque con una (X) |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| Del 1 al 10%   | No se considera plagio.                                                                                                                                                                                                                                                       | X                  |
| Del 11 al 30 % | Devolver al usuario para las correcciones.                                                                                                                                                                                                                                    |                    |
| Mayor a 31%    | El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley. |                    |

Por tanto, en mi condición de asesor, firmo el presente informe en señal de conformidad y **adjunto** las primeras páginas del reporte del Sistema Antiplagio.

Cusco, **5** de ..... **MAYO** ..... de 20**25**.....

  
Firma  
Post firma **JORGE LUIS ARIZACA CUSICUNA**

Nro. de DNI..... **42348906** .....

ORCID del Asesor **0000 - 0003 - 2658 - 5492** .....

**Se adjunta:**

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema Antiplagio: **oid: 27259:451683334** .....

# Abraham Monzon Ccorimanya

## TESIS\_MONZON\_ROTULADO\_FINAL.pdf

 Universidad Nacional San Antonio Abad del Cusco

---

### Detalles del documento

Identificador de la entrega

trn:oid:::27259:451683334

Fecha de entrega

22 abr 2025, 10:31 p.m. GMT-5

Fecha de descarga

22 abr 2025, 10:46 p.m. GMT-5

Nombre de archivo

TESIS\_MONZON\_ROTULADO\_FINAL.pdf

Tamaño de archivo

8.4 MB

177 Páginas

31.933 Palabras

168.993 Caracteres

# 9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- Bibliography
- Quoted Text
- Small Matches (less than 10 words)

## Top Sources

- 8%  Internet sources
- 1%  Publications
- 5%  Submitted works (Student Papers)

## Integrity Flags

### 2 Integrity Flags for Review

-  **Replaced Characters**  
243 suspect characters on 26 pages  
Letters are swapped with similar characters from another alphabet.
-  **Hidden Text**  
14 suspect characters on 7 pages  
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## **Presentación**

El presente trabajo de tesis titulado “Diseño e implementación de un sistema automatizado de cloración con monitoreo remoto utilizando redes IoT para la comunidad de San Pedro-Canchis-Cusco” se desarrolla en el contexto de la mejora de los sistemas de cloración de agua potable en zonas rurales. La calidad del agua es un factor crítico para la salud pública, y su desinfección mediante cloro es un método ampliamente utilizado. Sin embargo, en muchas comunidades, este proceso no se realiza de manera óptima debido a la falta de automatización y monitoreo en tiempo real.

Este estudio aborda la problemática de la cloración inadecuada del agua potable en la comunidad de San Pedro, en la provincia de Canchis, donde los sistemas actuales no consideran las variaciones de caudal y turbidez del agua. Como resultado, las concentraciones de cloro pueden fluctuar fuera de los rangos recomendados, afectando la calidad del agua y la seguridad de los habitantes.

La investigación se enfoca en el diseño e implementación de un sistema embebido de control automático para la regulación de cloro residual, integrando sensores de caudal y turbidez, junto con un sistema de monitoreo remoto basado en redes IoT-LoRa. El objetivo es garantizar una desinfección eficiente del agua, permitiendo además la supervisión en tiempo real de los parámetros de calidad.

Para el desarrollo del sistema, se han considerado tecnologías avanzadas en control automático, sistemas embebidos y comunicación inalámbrica. Se ha utilizado un microcontrolador ARM Cortex M4 para la regulación del proceso y un ESP32 para la gestión de la comunicación con la red IoT. Asimismo, se ha integrado una fuente de alimentación fotovoltaica para asegurar la operatividad del sistema en ubicaciones remotas.

Este trabajo representa una contribución significativa a la automatización de los procesos de tratamiento de agua en comunidades rurales, ofreciendo una solución eficiente y

sostenible. La implementación de este sistema permitirá mejorar la calidad del agua potable y servirá como referencia para futuras investigaciones en el ámbito del control y monitoreo de sistemas de saneamiento mediante tecnologías emergentes.

**Agradecimiento**

*Mi total agradecimiento a la Escuela Profesional de Ingeniería Electrónica, a todos los docentes quienes guiaron mi formación profesional y en especial al ingeniero Jorge Luis Arizaca Cusicuna por su guía en la elaboración del presente proyecto de tesis. También mi reconocimiento a todos los compañeros de la facultad, con quienes pasamos largas y productivas horas de estudio.*

## Introducción

El acceso a agua potable segura es un derecho fundamental y un desafío en muchas comunidades rurales. La correcta cloración del agua es esencial para eliminar microorganismos patógenos y garantizar su calidad para el consumo humano. Sin embargo, en muchas localidades, este proceso no se lleva a cabo de manera óptima debido a la falta de sistemas automatizados que consideren las variaciones de caudal y turbidez. En el distrito de San Pedro, provincia de Canchis, la cloración se realiza manualmente sin un monitoreo en tiempo real, lo que puede generar concentraciones inadecuadas de cloro residual en el agua.

El presente trabajo de investigación titulado *“Diseño e implementación de un sistema automatizado de cloración con monitoreo remoto utilizando redes IoT para la comunidad de San Pedro-Canchis-Cusco”* tiene como objetivo desarrollar un sistema embebido de control para regular la concentración de cloro residual en función de las condiciones del agua. A través de sensores de caudal y turbidez, el sistema podrá ajustar la dosificación de cloro de manera automatizada, optimizando el proceso y garantizando niveles adecuados de desinfección.

Además, se implementará un sistema de monitoreo remoto basado en tecnología IoT-LoRa, permitiendo que los operadores puedan supervisar en tiempo real los parámetros de calidad del agua desde cualquier ubicación. Esto representa una solución innovadora y eficiente para comunidades donde el acceso a los reservorios es complicado debido a su ubicación geográfica.

La metodología utilizada en esta investigación es de tipo experimental cuantitativa, ya que implica la manipulación y medición de variables como caudal, turbidez y concentración de cloro residual. Se realizaron entrevistas con las autoridades locales y responsables del sistema de agua potable para comprender la problemática existente. Posteriormente, se diseñó e implementó un prototipo basado en un microcontrolador ARM Cortex M4 para el control y un

ESP32 para la comunicación, integrando un sistema de alimentación fotovoltaica para garantizar su funcionamiento autónomo.

El desarrollo de este sistema no solo permitirá mejorar la calidad del agua en la comunidad de San Pedro, sino que también servirá como modelo para otras localidades con problemas similares. La automatización y el monitoreo remoto contribuirán a una gestión más eficiente de los recursos hídricos, mejorando la salud y calidad de vida de la población.

## Resumen

El siguiente trabajo de investigación de tesis titulado “Diseño e implementación de un sistema automatizado de cloración con monitoreo remoto utilizando redes IoT para la comunidad de San Pedro-Canchis-Cusco” es una aplicación del área de los sistemas embebidos y las redes IoT, tiene como objetivo principal diseñar e implementar un sistema embebido de control para la correcta regulación de la cloración del agua potable, teniendo en cuenta las variaciones de caudal y turbidez que presenta el agua para consumo humano, durante las diferentes épocas del año, ya que, la cloración que actualmente se realiza no considera estos parámetros y es por eso que se presentan concentraciones inadecuadas de cloro residual a determinadas horas del día; el proyecto de tesis también considera la implementación de un sistema de comunicación basado en tecnologías IoT-LoRa para realizar un adecuado monitoreo de los parámetros que intervienen en la calidad del agua para consumo humano.

Para lo cual, primeramente, se entrevistó a los responsables directos, responsable de ATM de la municipalidad y la JASS (Junta de Administración de Servicios de Saneamiento) de la comunidad, para tener una mejor idea de la problemática, seguidamente se verifico el entorno del sistema de agua potable para tomar evidencias fotográficas y hacer mediciones de los parámetros, como caudal, turbidez en la captación y concentración de cloro residual en reservorio; seguidamente, se identificó los instrumentos necesarios para el desarrollo del proyecto, como sensores, actuadores, instrumentos de medición y calibración; luego se diseñó el sistema de control y sistema de comunicación haciendo uso de simuladores como Proteus y las herramientas de depuración de los microcontroladores.

Finalmente, se implementó el hardware del sistema de control sobre un microcontrolador de arquitectura ARM Cortex -M4 y un ESP32 para la comunicación entre el dispositivo LoRa y la red de internet; una vez implementado el sistema completo se hizo las pruebas en campo para validar su correcto funcionamiento.

**Palabras clave:** Sistema automatizado, control de cloración, regulación automática; Redes IoT, monitoreo remoto, IoT-LoRa, comunicación inalámbrica; Cloración de agua, cloro residual, turbidez del agua, calidad del agua, Microcontrolador ARM Cortex-M4, ESP32, sistemas embebidos, sensores y actuadores, Sistemas Embebidos.

## **Abstract**

The following thesis research work entitled "Design and implementation of an automated chlorination system with remote monitoring using IOT networks for the community of San Pedro-Canchis-Cusco" is an application of the area of embedded systems and IoT networks and has The main objective is to design and implement an embedded control system for the correct regulation of the chlorination of drinking water, taking into account the variations in flow and turbidity that water for human consumption presents, during the different times of the year, since the Chlorination that is currently carried out does not consider these parameters and that is why inadequate concentrations of residual chlorine occur at certain hours of the day; The thesis project also considers the implementation of a communication system based on IoT-LoRa technologies to carry out an adequate monitoring of the parameters that intervene in the quality of water for human consumption.

For which, first, the direct managers were interviewed, responsible for the ATM of the municipality and the JASS of the community, to have a better idea of the problem, then the environment of the drinking water system was verified to take photographic evidence and make measurements of the parameters, such as flow, turbidity in the catchment and concentration of residual chlorine in the reservoir; Next, the necessary instruments for the development of the project were identified, such as sensors, actuators, measurement and calibration instruments; then the control system and communication system were developed using simulators such as PROTEUS and microcontroller debugging tools.

Finally, the control system hardware was implemented on an ARM Cortex -M4 architecture microcontroller and an ESP32 for communication between the LoRa device and the Internet network; Once the complete system was implemented, field tests were carried out to validate its correct operation.

**Keyword:** Automated system, chlorination control, automatic regulation, IoT Networks, remote monitoring, IoT-LoRa, wireless communication, Water chlorination, residual chlorine, water turbidity, water quality, ARM Cortex-M4 Microcontroller, ESP32, embedded systems, sensors and actuators, Embedded Systems.

## Índice

|                                                                            |     |
|----------------------------------------------------------------------------|-----|
| Presentación.....                                                          | II  |
| Agradecimiento.....                                                        | III |
| Introducción.....                                                          | IV  |
| Resumen.....                                                               | V   |
| Abstract.....                                                              | IV  |
| 1. Capítulo I: Generalidades .....                                         | 19  |
| 1.1. Problemática.....                                                     | 19  |
| 1.2. Problema General.....                                                 | 21  |
| 1.3. Problemas Específicos.....                                            | 21  |
| 1.4. Objetivos .....                                                       | 22  |
| 1.4.1. Objetivo General.....                                               | 22  |
| 1.4.2. Objetivos Específicos .....                                         | 22  |
| 1.5. Metodología .....                                                     | 22  |
| 1.6. Variables.....                                                        | 22  |
| 1.6.1. Variables Independientes.....                                       | 22  |
| 1.6.2. Variables Dependientes .....                                        | 23  |
| 1.7. Justificación.....                                                    | 23  |
| 1.8. Limitaciones .....                                                    | 28  |
| 2. Capítulo II: Marco Teórico .....                                        | 29  |
| 2.1. Antecedentes .....                                                    | 29  |
| 2.2. Bases Teóricas.....                                                   | 30  |
| 2.2.1. Métodos de Desinfección del Agua Potable.....                       | 30  |
| 2.2.2. Sistema de Cloración por Goteo Constante con Doble Recipiente ..... | 31  |
| 2.2.2.1. Terminología .....                                                | 31  |
| 2.2.2.2. Partes del Sistema de Cloración por Goteo.....                    | 32  |
| 2.2.2.3. Cálculo del Caudal de Dosificación.....                           | 33  |
| 2.2.3. Sistema de Adquisición de Datos .....                               | 34  |
| 2.2.3.1. Concepto de Sensores .....                                        | 34  |
| 2.2.3.2. Sensor de Caudal de Efecto Hall.....                              | 34  |
| 2.2.3.3. Sensor de Turbidez.....                                           | 36  |
| 2.2.3.4. Sensor de Cloro Residual .....                                    | 40  |
| 2.2.4. Sistema de Control.....                                             | 42  |
| 2.2.4.1. Sistemas de Control en Lazo Abierto.....                          | 42  |
| 2.2.4.2. Sistemas de Control en Lazo Cerrado .....                         | 42  |
| 2.2.5. Sistema de Comunicación Inalámbrica IoT.....                        | 42  |
| 2.2.5.1. Internet de las Cosas (Internet of Thing IoT).....                | 42  |
| 2.2.6. Tecnologías Inalámbricas de Comunicación.....                       | 43  |
| 2.2.7. SigFox.....                                                         | 45  |
| 2.2.8. LoRaWAN.....                                                        | 46  |
| 2.2.8.1. Arquitectura de Red de LoRaWAN .....                              | 46  |
| 2.2.8.2. LoRaWAN MAC Layer.....                                            | 47  |
| 2.2.8.3. LoRaWAN Gateway .....                                             | 48  |

|           |                                                                       |     |
|-----------|-----------------------------------------------------------------------|-----|
| 2.2.9.    | LoRa .....                                                            | 49  |
| 2.2.9.1.  | Modulación LoRa.....                                                  | 49  |
| 2.2.9.2.  | Spread Spectrum .....                                                 | 50  |
| 2.2.9.3.  | Chirp Spread Spectrum .....                                           | 51  |
| 2.2.10.   | Virtualización y Computación en la Nube para Aplicaciones IoT .....   | 52  |
| 2.2.11.   | Amazon Web Service (AWS).....                                         | 52  |
| 2.2.12.   | Node-RED.....                                                         | 53  |
| 2.2.13.   | Marco Legal de las Bandas no Licenciadas en el Perú.....              | 53  |
| 2.2.14.   | Sistemas Fotovoltaicos.....                                           | 54  |
| 2.2.14.1. | Componentes Básicos de un Sistema Fotovoltaico.....                   | 56  |
| 2.2.14.2. | Panel Solar .....                                                     | 56  |
| 2.2.14.3. | Baterías y Regulador de Carga.....                                    | 58  |
| 2.2.15.   | Sistemas Embebidos .....                                              | 58  |
| 2.2.15.1. | Arquitectura de un Sistema Embebido.....                              | 58  |
| 2.2.15.2. | Microcontroladores ARM .....                                          | 59  |
| 2.2.15.3. | Familia de Procesadores Cortex-M.....                                 | 59  |
| 2.2.15.4. | Microcontrolador STM32F401RE .....                                    | 61  |
| 2.2.15.5. | Disposición y Configuración del Hardware del STM32F401RE .....        | 61  |
| 2.2.15.6. | Memoria del Sistema de la STM32.....                                  | 62  |
| 2.2.15.7. | Microcontrolador ESP32.....                                           | 62  |
| 2.2.16.   | Lenguaje Ensamblador y Lenguaje C .....                               | 62  |
| 2.2.16.1. | STM32Cube IDE .....                                                   | 65  |
| 3.        | CAPITULO III: Diseño e Implementación.....                            | 66  |
| 3.1.      | Diseño del Sistema de Control .....                                   | 67  |
| 3.1.1.    | Implementación del Sistema de Medición del Caudal en Tiempo Real..... | 70  |
| 3.1.1.1.  | Calibración del Sensor de Caudal .....                                | 70  |
| 3.1.1.2.  | Diseño y Acondicionamiento del Sensor de Caudal .....                 | 72  |
| 3.1.1.3.  | Configuración de los Periféricos del STM32.....                       | 73  |
| 3.1.2.    | Implementación del Sensor de Turbidez .....                           | 74  |
| 3.1.2.1.  | Modelado del Sensor de Turbidez.....                                  | 74  |
| 3.1.2.2.  | Calibración del Sensor de Turbidez .....                              | 76  |
| 3.1.2.3.  | Implementación del Sistema de Medición de la Turbidez .....           | 82  |
| 3.1.2.4.  | Configuración del ADC del STM32 .....                                 | 83  |
| 3.1.3.    | Implementación del Sensor de Cloro Residual.....                      | 86  |
| 3.1.3.1.  | Calibración del Sensor de Cloro Residual .....                        | 86  |
| 3.1.3.2.  | Configuración del ADC del STM32 .....                                 | 91  |
| 3.1.4.    | Diseño del Regulador de Suministro de la Solución Hipoclorosa.....    | 91  |
| 3.1.5.    | Diseño e Implementación del Tiempo de Toma de Muestra.....            | 95  |
| 3.1.5.1.  | Determinación del Tiempo de Extinción del Cloro Residual .....        | 95  |
| 3.1.5.2.  | Implementación del Periférico RTC .....                               | 98  |
| 3.2.      | Diseño del Sistema de Comunicación .....                              | 100 |
| 3.2.1.    | Diseño del Sistema de Transmisión LoRa.....                           | 102 |
| 3.2.1.1.  | Implementación del Módulo RAK811.....                                 | 102 |
| 3.2.1.2.  | Implementación del STM32 para la Transmisión de Información .....     | 103 |

|             |                                                                          |     |
|-------------|--------------------------------------------------------------------------|-----|
| 3.2.2.      | Diseño del Sistema Receptor LoRa .....                                   | 106 |
| 3.2.3.      | Diseño de la Plataforma Node-RED Sobre AWS .....                         | 109 |
| 3.2.3.1.    | Diseño del Entorno de Visualización en Node-RED .....                    | 109 |
| 3.3.        | Diseño de la Etapa de Potencia del Sistema de Control.....               | 113 |
| 3.3.1.      | Diseño de la Etapa de Regulación de Tensión .....                        | 113 |
| 3.3.2.      | Diseño de la Etapa de Potencia STM32 – Motobomba.....                    | 114 |
| 3.4.        | Diseño de la Etapa de Alimentación Fotovoltaica .....                    | 115 |
| 3.4.1.      | Diseño e Implementación del Panel Solar .....                            | 116 |
| 4.          | CAPITULO IV: Resultados y Pruebas .....                                  | 119 |
| 4.1.        | Pruebas y Resultados del Funcionamiento del Sistema de Control .....     | 119 |
| 4.2.        | Prueba y Resultados del Sistema de Comunicación.....                     | 131 |
| 4.2.1.      | Comprobación del Sistema de Comunicación en Laboratorio .....            | 131 |
| 4.2.2.      | Comprobación del Sistema de Comunicación en Campo .....                  | 138 |
| 5.          | Conclusiones .....                                                       | 143 |
| 6.          | Recomendaciones .....                                                    | 144 |
| 7.          | Referencias.....                                                         | 145 |
| Apéndice A. | Hoja de presupuesto.....                                                 | 146 |
| Apéndice B. | Código de programación del microcontrolador ESP32 .....                  | 147 |
| Apéndice C. | Código de programación del microcontrolador STM32F401RE .....            | 150 |
| Apéndice C. | Implementación de Node-RED .....                                         | 168 |
| Apéndice E. | Sensor de cloro residual .....                                           | 171 |
| Apéndice F. | Sensor de turbidez .....                                                 | 172 |
| Apéndice G. | Sensor de caudal .....                                                   | 173 |
| Apéndice H. | Diagramas esquemáticos de los dispositivos electrónicos utilizados. .... | 174 |

## Índice de Figuras

|                                                                                              |    |
|----------------------------------------------------------------------------------------------|----|
| <b>Figura 1</b> Nivel estático dentro de un reservorio .....                                 | 20 |
| <b>Figura 2</b> Captación de Ladera.....                                                     | 21 |
| <b>Figura 3</b> Porcentaje de población que consume agua clorada.....                        | 26 |
| <b>Figura 4</b> Grafica de cloro agregado en función al cloro residual .....                 | 31 |
| <b>Figura 5</b> Sistema de cloración por goteo con doble recipiente.....                     | 33 |
| <b>Figura 6</b> Efecto Hall .....                                                            | 35 |
| <b>Figura 7</b> Sensor de caudal de efecto Hall .....                                        | 36 |
| <b>Figura 8</b> Diagrama simplificado de un sensor de turbidez .....                         | 37 |
| <b>Figura 9</b> Diagrama eléctrico .....                                                     | 37 |
| <b>Figura 10</b> Corriente directa en función de la temperatura .....                        | 38 |
| <b>Figura 11</b> Distribución espectral led infrarrojo 5 mm .....                            | 38 |
| <b>Figura 12</b> Intensidad de radiación en función de la corriente directa .....            | 39 |
| <b>Figura 13</b> Patrón de radiación diodo led infrarrojo .....                              | 39 |
| <b>Figura 14</b> Corriente de colector en función de la intensidad de radiación .....        | 40 |
| <b>Figura 15</b> Sensor de cloro residual CL2 .....                                          | 41 |
| <b>Figura 16</b> Sistema de control de lazo abierto .....                                    | 42 |
| <b>Figura 17</b> Sistema de control de lazo cerrado .....                                    | 42 |
| <b>Figura 18</b> Evolución del IoT.....                                                      | 43 |
| <b>Figura 19</b> Clasificación de las tecnologías inalámbricas.....                          | 44 |
| <b>Figura 20</b> Arquitectura de una red LoRaWAN .....                                       | 47 |
| <b>Figura 21</b> Capa de protocolos LoRaWAN .....                                            | 48 |
| <b>Figura 22</b> Chirps spread spectrum.....                                                 | 51 |
| <b>Figura 23</b> Modulación LoRa .....                                                       | 52 |
| <b>Figura 24</b> Frecuencia de canales AU915-928 .....                                       | 54 |
| <b>Figura 25</b> Ángulos solares.....                                                        | 55 |
| <b>Figura 26</b> Unión PN de una celda solar.....                                            | 56 |
| <b>Figura 27</b> Modelo eléctrico de una celda solar.....                                    | 57 |
| <b>Figura 28</b> Relación del voltaje y la corriente.....                                    | 57 |
| <b>Figura 29</b> Arquitectura Von Neumann (izquierda) y arquitectura Harvard (derecha) ..... | 59 |
| <b>Figura 30</b> Familia de microprocesadores ARM Cortex M.....                              | 60 |
| <b>Figura 31</b> Estructura de la STM32 .....                                                | 62 |
| <b>Figura 32</b> Compilación de código en C a código máquina .....                           | 64 |
| <b>Figura 33</b> Esquema del sistema de control y comunicación implementado en campo .....   | 66 |
| <b>Figura 34</b> Implementación final del proyecto .....                                     | 67 |

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| <b>Figura 35</b> Cálculo del caudal del agua en el reservorio.....                         | 68 |
| <b>Figura 36</b> Diagrama para la calibración del sensor de caudal .....                   | 71 |
| <b>Figura 37</b> Circuito de medición y calibración del sensor de caudal.....              | 71 |
| <b>Figura 38</b> Acondicionamiento de la señal cuadrada.....                               | 72 |
| <b>Figura 39</b> Acondicionamiento de la señal .....                                       | 73 |
| <b>Figura 40</b> Fuente de interrupción EXTI1 .....                                        | 73 |
| <b>Figura 49</b> Registro SYSCFG_EXTICR1. ....                                             | 74 |
| <b>Figura 42</b> Simulación del sensor de turbidez .....                                   | 75 |
| <b>Figura 43</b> Recta del fototransistor .....                                            | 76 |
| <b>Figura 44</b> Sensor HANNA HI93703 .....                                                | 77 |
| <b>Figura 45</b> Muestras para calibración de sensor de turbidez (0, 10, 500 NTU).....     | 78 |
| <b>Figura 46</b> Medición de la turbidez del agua potable 1.2 NTU .....                    | 78 |
| <b>Figura 47</b> Muestra de agua con nivel de turbidez de 53 NTU.....                      | 79 |
| <b>Figura 48</b> Muestra de agua con nivel de turbidez de 977 NTU.....                     | 79 |
| <b>Figura 49</b> Medición de voltaje a la salida del sensor (977 NTU).....                 | 80 |
| <b>Figura 50</b> Medición de voltaje a la salida del sensor (1.2 NTU).....                 | 80 |
| <b>Figura 51</b> Grafica turbidez (NTU) en función del voltaje .....                       | 81 |
| <b>Figura 52</b> Sistema de medición de turbidez .....                                     | 82 |
| <b>Figura 53</b> Sistema de control y medición instalado.....                              | 83 |
| <b>Figura 54</b> Periférico ADC .....                                                      | 84 |
| <b>Figura 55</b> Modo de operación del ADC simple y continuo .....                         | 84 |
| <b>Figura 56</b> Registro ADC_CR2 .....                                                    | 85 |
| <b>Figura 57</b> Registro ADC1_SQR1 .....                                                  | 85 |
| <b>Figura 58</b> Registro ADC1_SQR3.....                                                   | 86 |
| <b>Figura 59</b> Registro ADC_CR1 .....                                                    | 86 |
| <b>Figura 60</b> Niveles de coloración en presencia del reactivo DPD .....                 | 87 |
| <b>Figura 61</b> Medición del nivel cloro residual.....                                    | 87 |
| <b>Figura 62</b> Muestra de agua con intensidad de coloración de 0.2 mg/L .....            | 88 |
| <b>Figura 63</b> Muestra de agua con intensidad de coloración correspondiente a 2 mg/L .   | 88 |
| <b>Figura 64</b> Concentración de cloro residual en función de la corriente de salida..... | 89 |
| <b>Figura 65</b> Modelado del sensor de cloro residual y su acondicionamiento.....         | 90 |
| <b>Figura 66</b> Concentración (mg/Litro) en función del voltaje de salida.....            | 91 |
| <b>Figura 67</b> Sistema de control On/Off para el suministro de cloro.....                | 92 |
| <b>Figura 68</b> Motor paso a paso .....                                                   | 93 |
| <b>Figura 69</b> Simulación en Proteus.....                                                | 93 |

|                                                                                        |     |
|----------------------------------------------------------------------------------------|-----|
| <b>Figura 70</b> Sección de estrangulamiento.....                                      | 95  |
| <b>Figura 71</b> Decaimiento del cloro en función del tiempo .....                     | 96  |
| <b>Figura 72</b> Modelo de la concentración de cloro residual .....                    | 97  |
| <b>Figura 73</b> Concentración en función del tiempo .....                             | 97  |
| <b>Figura 74</b> Registro RCC_CFGR.....                                                | 98  |
| <b>Figura 75</b> Registro RTC_DR.....                                                  | 99  |
| <b>Figura 76</b> Registro RTC_TR .....                                                 | 99  |
| <b>Figura 77</b> Cobertura de SigFox en el Perú .....                                  | 100 |
| <b>Figura 79</b> Implementación del sistema de comunicación IoT.....                   | 102 |
| <b>Figura 80</b> Interfaz RAK Serial Port Tool .....                                   | 103 |
| <b>Figura 81</b> Registro USART_BRR.....                                               | 104 |
| <b>Figura 82</b> Comunicación STM32 – RAK811 en modo transmisor.....                   | 105 |
| <b>Figura 83</b> Implementación RAK811 en modo transmisión.....                        | 105 |
| <b>Figura 94</b> Comunicación RAK811 en modo receptor – ESP32 – Internet.....          | 106 |
| <b>Figura 85</b> Interfaz RAK811 Serial Port Tool en modo receptor .....               | 107 |
| <b>Figura 86</b> Implementación RAK811 - ESP32 en modo recepción.....                  | 108 |
| <b>Figura 87</b> Implementación módulos LoRa transmisor – receptor .....               | 108 |
| <b>Figura 89</b> Comunicación Node-RED – ESP32 .....                                   | 109 |
| <b>Figura 90</b> Estructura JSON.....                                                  | 110 |
| <b>Figura 91</b> Desarrollo del sistema de adquisición de información en Node-RED..... | 110 |
| <b>Figura 92</b> Implementación del payload .....                                      | 111 |
| <b>Figura 93</b> Implementación en código de programación .....                        | 112 |
| <b>Figura 94</b> Visualización grafica en Node-RED.....                                | 112 |
| <b>Figura 95</b> Sistema regulador del voltaje. ....                                   | 113 |
| <b>Figura 96</b> Etapa de potencia motobomba1 .....                                    | 114 |
| <b>Figura 97</b> Etapa de potencia motobomba2 .....                                    | 115 |
| <b>Figura 98</b> Etapa de alimentación fotovoltaica .....                              | 116 |
| <b>Figura 99</b> Mapa solar de la ubicación del sistema de cloración .....             | 117 |
| <b>Figura 100</b> Ubicación de la posición del sol en el transcurso de un año.....     | 118 |
| <b>Figura 101</b> Angulo de elevación del panel solar .....                            | 118 |
| <b>Figura 102</b> Sistema de control instalado en la caseta de cloración .....         | 119 |
| <b>Figura 102</b> Sistema de control en funcionamiento .....                           | 120 |
| <b>Figura 103</b> Medición de cloro residual en el reservorio.....                     | 121 |
| <b>Figura 104</b> Grafica de concentración de cloro residual día 1.....                | 129 |
| <b>Figura 105</b> Grafica de concentración de cloro residual día 2.....                | 130 |

|                   |                                                           |     |
|-------------------|-----------------------------------------------------------|-----|
| <b>Figura 106</b> | Grafica de concentración de cloro residual día 3.....     | 130 |
| <b>Figura 107</b> | Grafica de concentración de cloro residual día 4.....     | 131 |
| <b>Figura 108</b> | Paquete de señal UART en modulo transmisor .....          | 132 |
| <b>Figura 109</b> | Señal UART en el módulo transmisor .....                  | 132 |
| <b>Figura 110</b> | Modulo transmisor – receptor .....                        | 133 |
| <b>Figura 111</b> | Paquete UART en el transmisor y receptor .....            | 134 |
| <b>Figura 112</b> | Señales en el transmisor y receptor .....                 | 134 |
| <b>Figura 113</b> | Tiempo de transmisión entre ambos módulos LoRa.....       | 135 |
| <b>Figura 114</b> | Módulo RAK811-ESP32 – Internet.....                       | 135 |
| <b>Figura 115</b> | Nodo transmisor sobre el reservorio de agua potable ..... | 138 |
| <b>Figura 116</b> | Modulo transmisor en campo.....                           | 141 |
| <b>Figura 117</b> | Entorno grafico sobre Node-RED.....                       | 142 |

## Índice de Tablas

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| Tabla 1. Parámetros de calidad de agua. ....                                        | 24  |
| Tabla 2. Calidad de agua en la población proveniente de la red publica .....        | 26  |
| Tabla 3. Métodos de desinfección de agua potable .....                              | 30  |
| Tabla 4. Frecuencias de operación Sigfox.....                                       | 45  |
| Tabla 6. Rangos de frecuencias identificados para servicios móviles avanzados. .... | 53  |
| Tabla 7. Valores de caudal. ....                                                    | 68  |
| Tabla 9. Salida de voltaje para diferentes niveles de turbidez. ....                | 81  |
| Tabla 10. Secuencia de bit para giro del motor a pasos. ....                        | 94  |
| Tabla 11. Concentración del cloro residual en función del tiempo. ....              | 97  |
| Tabla 12. Primer monitoreo de cloro residual. ....                                  | 122 |
| Tabla 13. Segundo monitoreo de cloro residual. ....                                 | 122 |
| Tabla 14. Tercer monitoreo de cloro residual. ....                                  | 123 |
| Tabla 15. Cuarto monitoreo de cloro residual. ....                                  | 123 |
| Tabla 16. Valores medidos de cloro residual y nivel de turbidez .....               | 124 |

## **Capítulo I: Generalidades**

### **1.1. Problemática**

La cloración y monitoreo del agua potable en el Perú, en las zonas urbanas está a cargo de prestadores de servicio privados, dichas organizaciones cuentan con profesionales especializados y sistemas de cloración altamente efectivos para brindar agua potable de calidad.

La situación del agua y saneamiento en las comunidades rurales del Perú está a cargo de las mismas organizaciones comunales las cuales a través de una asamblea nombran a sus integrantes de JASS (Junta Administradora de Servicios de Saneamiento), quienes son los responsables de realizar la cloración y monitoreo del agua potable. Muchas veces la desinfección del agua potable no se efectúa de manera eficaz debido a que, en las comunidades campesinas no se cuenta con un profesional especialista con dedicación exclusiva y los integrantes de las JASS realizan esta tarea de manera no remunerada y voluntaria.

Los mecanismos de cloración usados actualmente, en la mayoría de las comunidades del Perú, son hipocloradores del tipo goteo constante, los cuales funcionan bastante bien y son relativamente baratos en comparación a otras tecnologías de cloración, pero requieren un permanente monitoreo y calibración debido a las variaciones de los parámetros del agua como se estudiará durante el desarrollo del presente trabajo de tesis.

Esta misma problemática se presenta en una de las comunidades del distrito de San Pedro, la cual, tiene 80 usuarios, quienes se abastecen de agua potable de una captación de ladera; cuentan también con un reservorio con un nivel estático y un sistema de cloración por goteo con doble recipiente con regulador de presión.

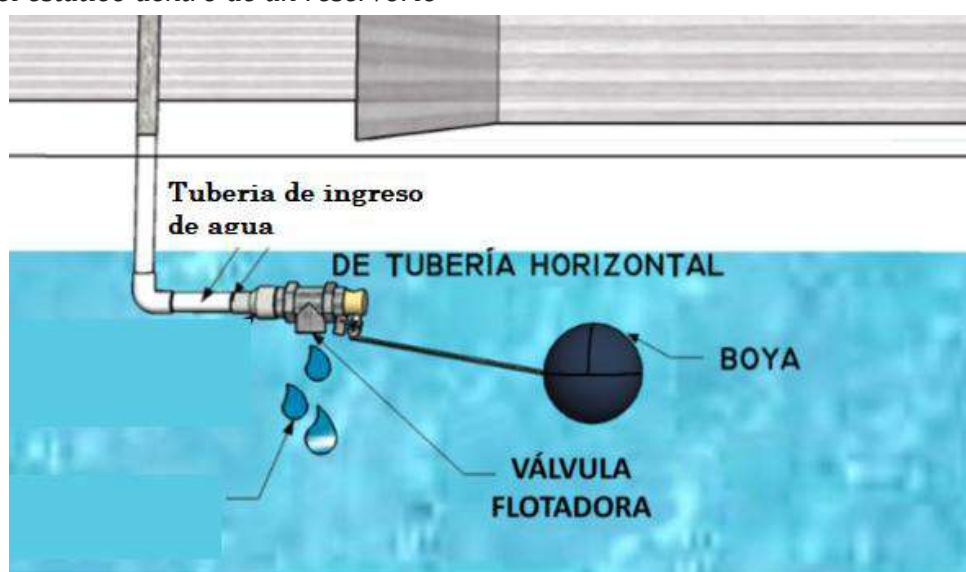
El principal problema de los usuarios en este centro poblado son las altas dosis de cloro residual en el agua, debido a las variaciones del caudal de agua que ingresa al reservorio, el cual, es regulado por un nivel estático On/Off que se encuentra dentro del reservorio, por lo tanto, el caudal de agua que ingresa al reservorio varía en el tiempo, y depende del consumo

del agua potable, pero el sistema de cloración suministra constantemente la solución hipoclorosa, por esta razón después de periodos largos en donde el consumo de agua potable es bajo (por ejemplo, en las noches) se presentan altas concentraciones de cloro residual en el agua potable.

La siguiente figura 1 muestra un diagrama de un nivel estático dentro de un reservorio y tiene la función de cerrar el ingreso de agua cuando el reservorio se encuentra lleno.

**Figura 1**

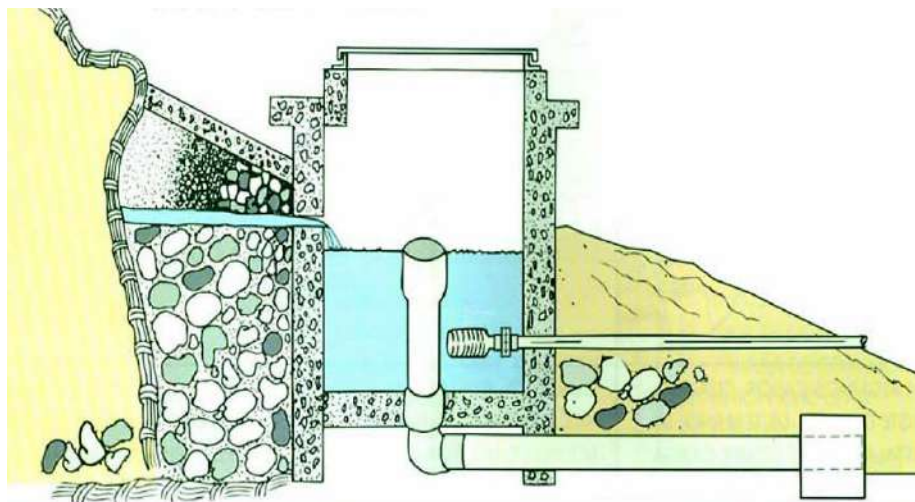
*Nivel estático dentro de un reservorio*



*Nota. Adaptado de Saneamiento Básico (p. 24), MEF, 2022*

El segundo problema surge debido a la ubicación de la captación de ladera de la comunidad, encontrándose ubicada en medio de áreas de cultivo, por lo tanto, en épocas de riego y lluvia la turbiedad del agua que se utiliza para el riego de los cultivos se filtra a la fuente de agua de la captación y por esa razón el nivel de turbidez del agua que llega al reservorio es inadecuado para realizar la cloración.

La siguiente figura 2 se muestra un esquema de las partes y la estructura interna de una captación de ladera, pudiéndose observar la posibilidad de una filtración de líquidos externos hacia la fuente de agua.

**Figura 2***Captación de ladera*

*Nota.* Adaptado de *Captación de ladera*, SSWM (<https://sswm.info/es>)

Finalmente, no existe una manera de realizar un monitoreo en tiempo real de los parámetros que determinan la calidad del agua y de ese modo tomar las decisiones adecuadas según el estado del sistema de agua potable para así garantizar un agua potable de calidad.

## 1.2. Problema General

En los centros poblados rurales, la cloración de agua potable es inadecuada, debido a que, no se está considerando las variaciones de caudal y turbidez; tampoco se cuenta con un sistema de medición de los parámetros del agua en tiempo real el cual sea accesible desde cualquier lugar para un adecuado monitoreo.

## 1.3. Problemas Específicos

- Los sistemas de cloración actuales no consideran las variaciones de caudal y turbidez del agua que ingresa al reservorio.
- Los sistemas de cloración actuales son difíciles de calibrar manualmente y requieren muchos ensayos de prueba y error hasta garantizar un valor adecuado de concentración de cloro.
- El monitoreo de cloro residual es muy difícil para los operadores ya que el reservorio de agua se encuentra muy lejos de la población y en un lugar elevado.

## **1.4. Objetivos**

### **1.4.1. *Objetivo General***

Diseñar e implementar un sistema de control y monitoreo en tiempo real de la concentración de cloro residual, para mantener la concentración dentro de los valores recomendados, en el agua potable de la comunidad del distrito de San Pedro en la provincia de Canchis.

### **1.4.2. *Objetivos Específicos***

- Diseñar un sistema de medición y adquisición de datos de las variables caudal y turbidez del agua que ingresa al reservorio.
- Regular la concentración de cloro residual a través de un sistema de control, diseñado en base a las mediciones previamente realizadas.
- Realizar un monitoreo remoto de los parámetros de la calidad de agua, utilizando un sistema de comunicación IoT.
- Integrar a nivel de prototipo el sistema completo de control y comunicación con su respectiva alimentación fotovoltaica.

## **1.5. Metodología**

El método empleado para desarrollar la siguiente tesis será del tipo experimental cuantitativa porque será necesario la manipulación y medición de variables como el caudal y turbidez para regular y medir la variable de concentración de cloro y de ese modo, poder hacer el monitoreo remoto a través del sistema de comunicación.

## **1.6. Variables**

### **1.6.1. *Variables Independientes***

- Caudal de agua que ingresa al reservorio.
- Turbidez.

### **1.6.2. Variables Dependientes**

- Concentración de cloro residual.

### **1.7. Justificación**

“Una adecuada cloración del agua potable disminuye significativamente los índices de anemia y las enfermedades infecciosas e intestinales” (INEI, 2020), por lo tanto, un adecuado servicio por parte de los prestadores de servicio de saneamiento en las comunidades campesinas (Junta Administradora de Servicio de Saneamiento) más alejadas del país es una dificultad en la que los organismos directamente involucrados como la SUNASS, Ministerio de Vivienda Construcción y Saneamiento y los Gobiernos Locales vienen trabajando, buscando mejorar la calidad del agua con una adecuada cloración, desde hace ya un buen tiempo por ejemplo en la Región del Cusco se tiene implementado el “Programa Presupuestal 0083 (Programa de Incentivos para la Mejora de los Servicios de Saneamiento)” (MEF, 2022) y el “Sello Regional Tinkuy” (CUSCO, 2022), en los cuales, se premia a las municipalidades que mejoran la prestación de los servicios de saneamiento en sus respectivas comunidades campesinas con un presupuesto anual. “A consecuencia de estos esfuerzos por parte de los encargados en agua y saneamiento, en la mayoría de las comunidades rurales ya se viene avanzando un trabajo social de concientización sobre la importancia de la adecuada desinfección del agua potable” (Peru, 2021). Pero todos estos esfuerzos muchas veces son insuficientes.

En la Resolución Ministerial nro. 647 – 2010/MINSA se aprueba la “Guía Técnica para la Implementación Operación y Mantenimiento del Sistema de Tratamiento Intradomiciliario de Agua para Consumo Humano”, en donde se fijan los valores adecuados de los parámetros de agua potable para consumo humano tal como se muestra en la siguiente tabla.

**Tabla 1***Parámetros de calidad de agua*

| <b>Parámetro</b>                                                                                                                                                                                                                                                                                | <b>Unidad</b>     | <b>Valores</b>           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------------------|
| Cloro Residual Libre (1)                                                                                                                                                                                                                                                                        | mg/L              | $\geq 0.5$ mg/L – 1 mg/L |
| Turbiedad                                                                                                                                                                                                                                                                                       | UNT               | $< 5$                    |
| E. Coli o Coliformes                                                                                                                                                                                                                                                                            | UNF/100 mL 44.5°C | 0 (2)                    |
| Huevos Larvas de Helmintos, quistes de protozoarios patógenos.                                                                                                                                                                                                                                  | Nº org/L          | 0                        |
| <p>(1) En caso de que el valor de cloro residual libre es <math>&lt; 0.5</math> mg/L tomar muestra de agua para el análisis microbiológico.</p> <p>(2) En caso de analizar por la técnica de NMP por tubos múltiples = <math>&lt; 2.2/100</math> ml UNT = Unidad Nefelométrica de Turbiedad</p> |                   |                          |

*Nota.* Adaptado de Resolución Ministerial N.º647 - 2010 - MINSA, p. 11.

En el cuadro anterior podemos apreciar el rango de valores aceptables de los parámetros cloro residual libre y turbiedad para garantizar agua potable de calidad.

En el Reglamento de Calidad de Agua para Consumo Humano, en el artículo 66 del Título IX afirma lo siguiente:

*Antes de la distribución del agua para consumo humano, el proveedor realizará la desinfección con un desinfectante eficaz para eliminar todo microorganismo y dejar un residual a fin de proteger el agua de posible contaminación microbiológica en la distribución. En caso de usar cloro o solución clorada como desinfectante, las muestras tomadas en cualquier punto de la red de distribución no deberán contener menos de 0.5 mg/L de cloro residual libre en el noventa por ciento (90 %) del total de muestras tomadas durante un mes. Del diez por ciento (10 %) restante, ninguna debe contener menos de 0.3 mg/L y la turbiedad deberá ser menor de 5 unidad nefelométrica de turbiedad (UNT).” (Ministerio de Salud, Lima, 2011), en el artículo anteriormente citado también tenemos los valores de cloro residual y turbidez para agua de consumo humano.*

También tenemos en la Guía para la *Calidad del agua para consumo humano*. Pág. 7:

*“La desinfección tiene una importancia incuestionable en la seguridad del abastecimiento de agua de consumo humano. La eliminación de microorganismos patógenos es una operación fundamental que muy frecuentemente se realiza con productos químicos reactivos como el cloro.” ... “El uso de productos químicos desinfectantes en el tratamiento del agua suele generar subproductos. No obstante, los riesgos para la salud que ocasionan estos subproductos son extremadamente pequeños en comparación con los riesgos asociados a una desinfección insuficiente.” ... “La concentración de algunos desinfectantes del agua de consumo humano, como el cloro, se puede monitorear y controlar fácilmente, y se recomienda el monitoreo frecuente cuando se aplica la cloración del agua.” (OMS, Ginebra, 2011).*

Como se puede deducir de los párrafos citados anteriormente, tanto los organismos nacionales (MINSA) e internacionales (OMS) recomiendan implementar un mecanismo de desinfección del agua potable antes de ser distribuida a los usuarios finales, con el fin de disminuir la transmisión de enfermedades bacteriológicas; también nos detalla que los valores de concentración de cloro residual deben estar dentro del rango de 0.5 mg/L a 1 mg/L para de ese modo, garantizar una adecuada desinfección del agua que podría contaminarse durante su distribución; también nos recomienda un valor máximo de turbidez de 5 UNT para evitar la formación de subproductos derivados de la mezcla del cloro residual y los sólidos disueltos, los cuales sí podrían generar daños a la salud si no se tienen presente al momento de realizar la desinfección del agua potable. En la Encuesta Nacional de Hogares (ENAHOG), se realizó la medición de la concentración de cloro residual en los hogares que tengan acceso al agua potable y según los resultados se puede apreciar bajos indicadores de cloración ( $> 0.5$  mg/L,  $< 1$  mg/L) a nivel nacional, tal como se puede apreciar en la Tabla 2. Estos datos tienen como fecha de corte el inicio del aislamiento social por la propagación de la pandemia debido al COVID19.

**Tabla 2**

Calidad de agua en la población de la red pública Año: abril 2019 – marzo 2020

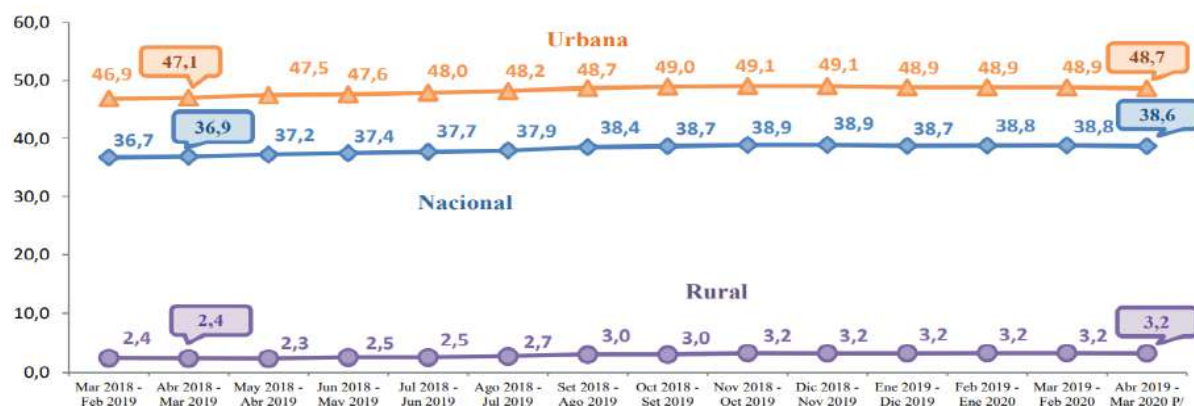
| Año móvil                  | Con algún nivel de cloro | Adecuado ( $\geq 0,5$ mg/l) | Inadecuado ( $\geq 0,1$ mg/l ^ $< 0,5$ mg/l) |
|----------------------------|--------------------------|-----------------------------|----------------------------------------------|
| <b>Indicadores anuales</b> |                          |                             |                                              |
| Mar 2018 - Feb 2019        | 52,6                     | 36,7                        | 15,8                                         |
| Abr 2018 - Mar 2019        | 52,6                     | 36,9                        | 15,7                                         |
| May 2018 - Abr 2019        | 52,8                     | 37,2                        | 15,6                                         |
| Jun 2018 - May 2019        | 53,0                     | 37,4                        | 15,6                                         |
| Jul 2018 - Jun 2019        | 53,2                     | 37,7                        | 15,5                                         |
| Ago 2018 - Jul 2019        | 53,1                     | 37,9                        | 15,2                                         |
| Set 2018 - Ago 2019        | 53,4                     | 38,4                        | 15,0                                         |
| Oct 2018 - Set 2019        | 53,5                     | 38,7                        | 14,9                                         |
| Nov 2018 - Oct 2019        | 53,8                     | 38,9                        | 14,9                                         |
| Dic 2018 - Nov 2019        | 53,9                     | 38,9                        | 15,0                                         |
| Ene 2019 - Dic 2019        | 53,8                     | 38,7                        | 15,0                                         |
| Feb 2019 - Ene 2020        | 53,5                     | 38,8                        | 14,8                                         |
| Mar 2019 - Feb 2020        | 53,7                     | 38,8                        | 14,9                                         |
| Abr 2019 - Mar 2020 P/     | 53,7                     | 38,6                        | 15,0                                         |

Nota. Adaptado de “Encuesta Nacional de Hogares (ENAHOG)”, INEI, 2022, p. 15

La siguiente figura 3 muestra los porcentajes de cloración a nivel nacional según el área de residencia rural y urbano.

**Figura 3**

Porcentaje de población que consume agua clorada



Nota. Adaptado de “Encuesta Nacional de Hogares (ENAHOG)”, INEI, 2022, p. 16

Como se puede apreciar en la figura anterior, el porcentaje de población en la zona rural que consume agua con una adecuada cloración solo alcanza a 3.2 % a finales de marzo del 2020, por lo tanto, se concluye que existe una necesidad de mejorar los métodos de desinfección y cloración en las zonas rurales del Perú.

Por una parte, vimos que una concentración baja de cloro residual es inadecuada para la salud, pero, por otro lado, tenemos la siguiente información acerca de la formación de los trihalometanos debido a un exceso de la concentración del cloro residual en el agua potable:

*“Este cloro residual es importante que se encuentre en niveles seguros para el consumo humano. Si este se encuentra en exceso, el cloro puede resultar tóxico para el consumo. Además, por ser una sustancia tan activa, un exceso de cloro puede reaccionar con distintos compuestos orgánicos, por lo que aumenta el riesgo de que se produzcan trihalometanos, que son compuestos carcinógenos para el humano.”* (Uso del Cloro en la Desinfección del Agua, U. Puerto Rico, Facultad de Ciencias Naturales).

También tenemos lo siguiente: *“Además de partículas inertes en suspensión, el agua turbia puede contener también materia orgánica en suspensión, lo que inhibe el efecto del biocida, que se pierde al reaccionar con la materia orgánica, y esta reacción puede dar lugar a la formación de subproductos de la desinfección, como los trihalometanos y las tricloroaminas, nocivos para la salud humana.”* (Higiene y Salud Ambiental, A. Sanches Z, 2018), por lo tanto, según los artículos anteriormente citados, un nivel elevado de cloro residual puede generar enfermedades cancerígenas, es por eso es muy importante mantener la concentración de cloro residual dentro de los valores seguros.

En conclusión, en el Perú, a nivel nacional, el porcentaje de población que consume agua con una adecuada desinfección es muy bajo, dicho porcentaje es aún más preocupante si solo consideramos la población que habita las zonas rurales. El consumo de agua sin desinfectar es un problema de salud pública debido a que puede generar el contagio de enfermedades anémicas e infecciosas, debido al agua con un nivel de cloro residual menor al necesario puede retener bacterias, protozoos y virus patógenos que son dañinos para la salud del ser humano, mientras que una cloración con niveles de turbidez inadecuados o una concentración elevada conlleva la formación de trihalometanos los cuales son cancerígenos.

Por estas razones, en este proyecto de tesis se propone el diseño e implementación de un sistema de medición, control y monitoreo de los parámetros que determinan la calidad del agua para consumo humano.

### **1.8. Limitaciones**

- El sistema de control está diseñado solo para sistemas de cloración por goteo constante con tanque de regulación de presión.
- El sistema control está diseñado para una concentración de cloro máxima de 2000 ppm en el tanque de solución madre, para garantizar de mejor manera el control de flujo de suministro de cloro.
- El único sistema de alimentación que se considera en el diseño es la energía solar debido a la ubicación remota del proceso a controlar.
- No se tiene en cuenta la variación de la concentración de cloro residual debido a otros parámetros como pH y conductividad.
- En el presente trabajo de tesis solo se considera el diseño del sistema fotovoltaico de alimentación mas no se considera su implementación por razones presupuestales.
- Solo se considera un nodo central de recolección de datos por lo tanto el sistema debe ser implementado en un lugar que tenga línea de vista libre de obstáculos.

## Capítulo II: Marco Teórico

### 2.1. Antecedentes

En el año 2021, la SUNASS y el Centro de Investigación y Tecnología de Agua desarrollaron un sistema de monitoreo para medir la concentración de cloro residual en reservorios del Perú. El sistema de monitoreo remoto está diseñado para detectar concentraciones inadecuadas de cloro residual, el proyecto se instaló en reservorios de prestadores de servicio de 10 pequeñas ciudades. (Centro de Investigación y Tec. de Agua).

En el año 2020, Valdez Monzon, N. en su proyecto de tesis “Desarrollo e implementación de un sistema de monitoreo de los parámetros de cloro residual y PH, para la calidad del agua mediante un sistema de comunicación GPRS, en la Comunidad Campesina de Sucso Auccaylle – San Jerónimo – Cusco” desarrollo un sistema de monitoreo remoto de parámetros de agua potable. A continuación, se muestra un cuadro con las similitudes y diferencias entre la presente tesis y el trabajo desarrollado por Valdez Monzon N.

| Diseño e implementación de un sistema automatizado de cloración... (Presente tesis)                                                                                                                                                                                                                                                                                     | Desarrollo e implementación de un sistema de monitoreo de los parámetros de cloro residual y PH... (Valdez Monzon N.)                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Diseño de un sistema control y regulación de la concentración residual de cloro para realizar una adecuada desinfección del agua potable en una comunidad del distrito de San Pedro para evitar el conflicto de los pobladores y el Centro de Salud debido a los niveles inadecuados de concentración residual de cloro en el agua especialmente en horas de la mañana. | ---                                                                                                                                                               |
| Diseño de un sistema de monitoreo remoto basado en la tecnología inalámbrica LoRa para un adecuado monitoreo de los parámetros: Concentración residual de cloro y nivel de turbidez                                                                                                                                                                                     | Diseño de un sistema de monitoreo remoto basado en la tecnología inalámbrica GPRS para un monitoreo remoto de los parámetros: Concentración residual de cloro pH. |

En el año 2020 se de desarrollo el trabajo de tesis: “Sistemas de tele monitoreo del nivel de cloro presente en el agua potabilizada” en la República del Ecuador, en donde se desarrolla un sistema de monitoreo de cloro residual en el agua potable.

En el año 2013 en la República del Ecuador se desarrolló un trabajo de tesis denominado: “Implementación de un sistema de control y monitoreo de cloración de agua en forma remota mediante tecnología ZigBee para la planta de potabilización El Calzado de la EPMAPAL” en el cual desarrolla un sistema de control de cloro residual y su monitoreo para sistemas de agua potable urbanos.

## 2.2. Bases Teóricas

### 2.2.1. Métodos de Desinfección del Agua Potable

En la actualidad existen varios métodos de desinfección del agua potable y estos varían de acuerdo con la ubicación geográfica, tipo de fuente de agua, composición físico- químico-orgánico, volumen de agua, etc. A continuación, se muestra los principales métodos de desinfección del agua potable utilizados en la actualidad.

**Tabla 3**

*Métodos de desinfección de agua potable*

| Físicos                                                                                           |                                                                                   | Químicos                                                                            |                                                                                         |
|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| Ultrafiltración<br>Ultrasonido<br>Ósmosis inversa<br>Electroforético<br>Ebullición<br>Congelación |                                                                                   | Cloro                                                                               | Gas                                                                                     |
|                                                                                                   |                                                                                   |                                                                                     | Hipoclorito <ul style="list-style-type: none"> <li>○ Sodio</li> <li>○ Calcio</li> </ul> |
|                                                                                                   |                                                                                   |                                                                                     | Dióxido de cloro<br>Cloraminas                                                          |
|                                                                                                   |                                                                                   | Permanganato de potasio<br>Yodo<br>Bromo<br>Ozono<br>Peróxido de hidrógeno<br>Plata |                                                                                         |
| Radiación ionizante                                                                               | <ul style="list-style-type: none"> <li>○ Gamma</li> <li>○ Ultravioleta</li> </ul> |                                                                                     |                                                                                         |

El tipo de desinfección que trataremos durante el desarrollo de la presente tesis será el método químico mediante cloro; por lo tanto, a continuación, se describen los componentes de esta tecnología de desinfección.

### 2.2.2. Sistema de Cloración por Goteo Constante con Doble Recipiente

Los sistemas de cloración por goteo constante son ampliamente utilizados en las zonas rurales del Perú especialmente en las regiones andinas, estos sistemas son fáciles de implementar y su costo es relativamente barato en comparación a otras tecnologías.

#### 2.2.2.1. Terminología

**Cloro:** Elemento químico altamente toxico, comúnmente encontrado en estado gaseoso de color amarillo verdoso, siendo aproximadamente 2.5 veces más pesado que el aire.

**Cloro Residual:** Es la concentración de cloro en el agua luego del proceso de desinfección, es también conocido como cloro libre.

**Demanda de Cloro:** Es la cantidad de la solución hipoclorosa que se consume durante el proceso de desinfección del agua. En la siguiente figura se muestra la diferencia entre demanda de cloro y cloro residual. En la siguiente figura 4 se puede apreciar la relación entre la demanda de cloro y el cloro residual.

**Figura 4**

*Grafica de cloro agregado en función al cloro residual*



Nota. Adaptado de *Hipoclorito de calcio*, USAL (<https://cidta.usal.es/cloracion>)

### 2.2.2.2. Partes del Sistema de Cloración por Goteo

**a) Tanque de solución madre:** Es un tanque de una capacidad de 250, 600 o 1200 Litros, su tamaño se elige teniendo en cuenta el volumen del reservorio y el caudal de agua que ingresa al reservorio.

En este tanque, es donde se almacena la solución hipoclorosa que será dosificada hacia el agua del reservorio, la concentración máxima recomendable de la solución hipoclorosa es de 5000 ppm para evitar la solidificación en las mangueras debido a las bajas temperaturas.

La concentración de hipoclorito en este tanque se calcula con la siguiente ecuación:

$$C = \frac{10 * \%HCl}{V}$$

Donde:

*C: Concentración de cloro en ppm o mg/L.*

*%HCl: Porcentaje de concentración de hipoclorito de calcio (70%).*

*V: Volumen del tanque de solución madre en Litros.*

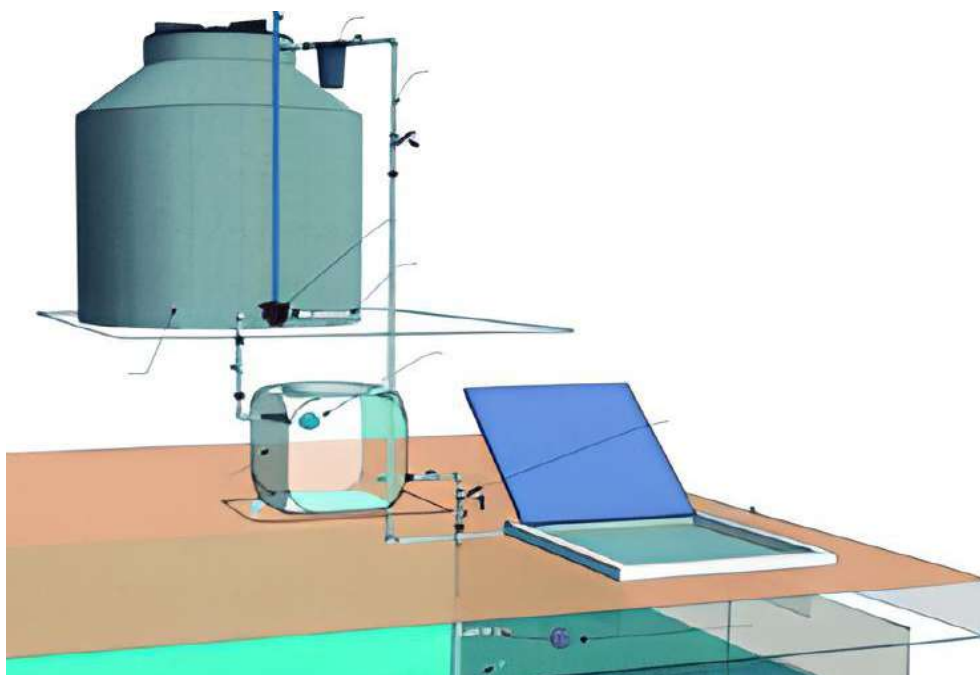
**b) Tanque de regulación de presión constante:** Es el tanque de la siguiente etapa después del tanque de solución madre, tiene la función de mantener una presión constante en el fondo del tanque manteniendo constante el nivel del líquido, mediante una boya On/Off.

**c) Válvula de regulación:** Es una válvula manual que se conecta después del tanque de regulación y antes de la entrada al reservorio, tiene la finalidad de regular el caudal de ingreso al reservorio.

En la siguiente figura se muestra el sistema de cloración por goteo con doble recipiente completo, pudiéndose apreciar en la parte superior el tanque de solución madre seguido del tanque de regulación de presión y finalmente se muestra la válvula de regulación de caudal de goteo instalado después del tanque de regulación de presión.

**Figura 5**

*Sistema de cloración por goteo con doble recipiente*



*Nota. Adaptado de Saneamiento Básico (p. 56), MEF, 2022*

### 2.2.2.3. Cálculo del Caudal de Dosificación

Para calcular la correcta dosificación de la solución hipoclorosa hacia el agua en el reservorio será necesario realizar algunas mediciones aplicando el siguiente procedimiento:

**Primero.** Se calcula el caudal de ingreso de agua al reservorio, para lo cual se utiliza un pequeño balde de 4 Litros y se mide el tiempo de llenado y después se aplica la siguiente ecuación:

$$Q(\text{Litros/segundo}) = \frac{\text{Volumen(Litros)}}{\text{Tiempo(Segundos)}}$$

**Segundo.** Se determina el tiempo de recarga de cloro en días, y seguidamente se determina el volumen total de agua a desinfectar:

$$V_{\text{Total}}(\text{Litros}) = Q * \text{nro\_dias} * 86400$$

**Tercero.** Se calcula la cantidad de hipoclorito de calcio a utilizar teniendo en cuenta la siguiente ecuación:

$$P = \frac{C * V_{\text{Total}}}{\%HCl * 10}$$

Donde:

P: Es el peso de hipoclorito de calcio a utilizar (Gramos).

C: Concentración de cloro residual que se desea (ppm).

%HCl: Concentración comercial de hipoclorito de calcio.

Finalmente: se calcula el caudal de goteo al reservorio utilizando la siguiente ecuación:

$$q_{Goteo}(mL/minuto) = \frac{V_{Tanque}}{nro. Dias} * \frac{1}{0.69}$$

### **2.2.3. Sistema de Adquisición de Datos**

#### **2.2.3.1. Concepto de Sensores**

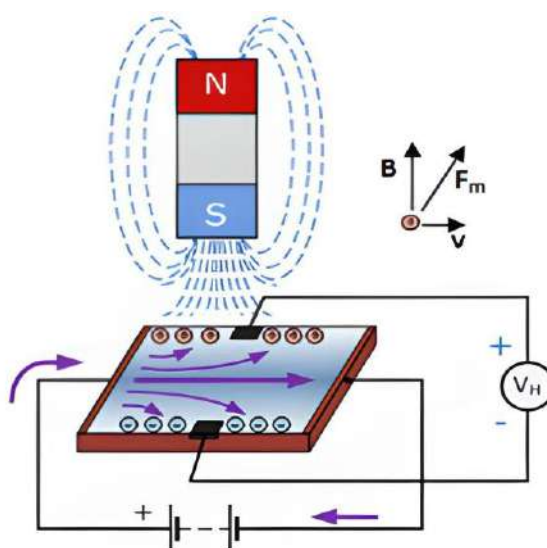
Los sensores son dispositivos electrónicos que varían sus propiedades eléctricas en respuesta a variaciones del ambiente exterior que los rodea.

Por lo tanto, los sensores responden a variaciones físicas como presión, intensidad luminosa, fuerza, concentración de cloro, caudal, etc. Y los transforman en señales eléctricas que pueden ser medidas.

#### **2.2.3.2. Sensor de Caudal de Efecto Hall**

Los sensores de flujo de agua o caudalímetros de efecto Hall son dispositivos electromecánicos, utilizados para la medición de caudal o flujo volumétrico de un fluido. Su principio de funcionamiento se basa en el fenómeno físico llamado efecto Hall, el cual describe la generación de una tensión eléctrica debido al paso de una corriente eléctrica a través de un campo magnético transversal. La siguiente figura 6 muestra el efecto Hall.

**Figura 6**  
*Efecto Hall*

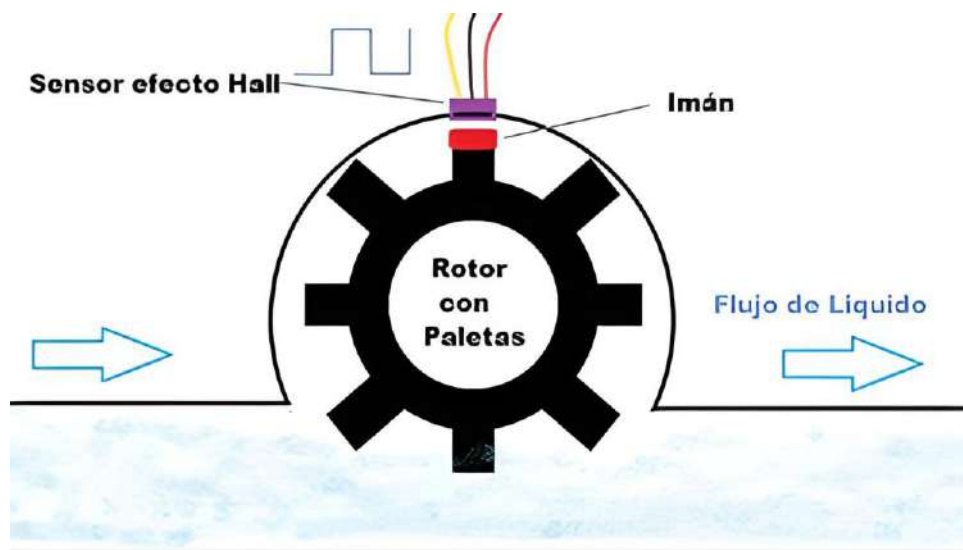


*Nota.* Adaptado de *Efecto Hall*, RSEF (<http://rsefalicante.umh.es/>)

En la figura anterior se puede observar la generación de una diferencia de potencial en los extremos de la cinta, la cual conduce una determinada intensidad de corriente eléctrica, el potencial es generado por la presencia del campo magnético y las cargas en movimiento están sometidas a las fuerzas de Lorentz, esta diferencia de potencia es proporcional a la intensidad de la corriente eléctrica y del campo magnético.

La estructura interna de sensor de efecto Hall consta de un eje rotatorio con una determinada cantidad de paletas imantadas, las cuales generan un pulso de voltaje en cada ciclo de rotación, por lo tanto, el caudal del fluido será proporcional a la frecuencia de la señal cuadrada que se genera a la salida del sensor.

En la siguiente figura 7 se puede observar los componentes internos de un sensor de caudal de efecto Hall.

**Figura 7***Sensor de caudal de efecto Hall*

*Nota.* Adaptado de *Sensor de caudal y partes*, Arduino (<http://www.arduino.com>)

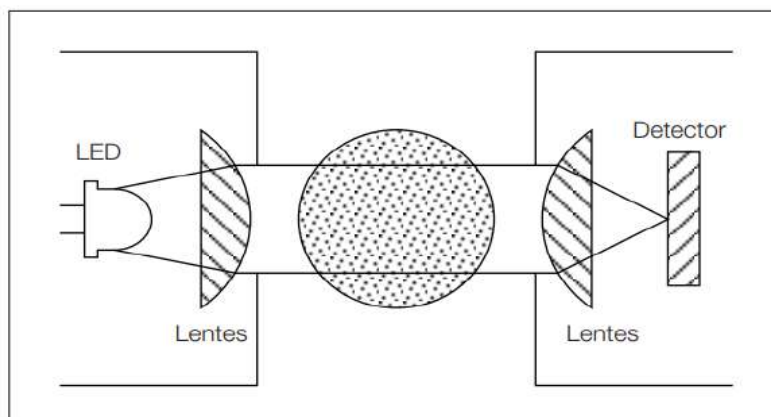
### 2.2.3.3. Sensor de Turbidez

Este dispositivo de medición permite determinar de forma sencilla la calidad de agua haciendo una medición de su nivel turbidez. Este sensor utiliza un diodo led infrarrojo y un fototransistor ubicado en contra posición, de ese modo la corriente de colector en el fototransistor dependerá de la cantidad de radiación infrarroja que logra atravesar la muestra de agua, el voltaje de salida se medirá en una resistencia en el emisor del fototransistor.

A continuación, en la figura 8 se muestra un diagrama simplificado del funcionamiento de la medición de la turbidez mediante la técnica de adsorción, el cual consiste en un fotodiodo led infrarrojo que emite un haz de luz que atraviesa una muestra y al otro extremo un fototransistor el cual mide la intensidad de radiación que incide sobre él.

**Figura 8**

Diagrama simplificado de un sensor de turbidez

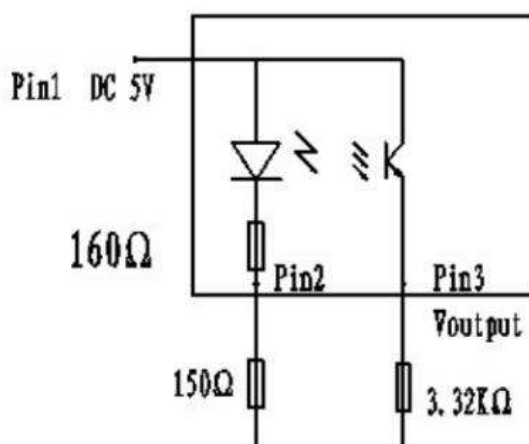


*Nota.* Adaptado de *Sensor de turbidez*, Hach (<https://essupport.hach.com>)

Según la figura anterior se puede observar en la parte izquierda un led, el cual emite una radiación infrarroja que atraviesa una muestra del fluido, debido al nivel de turbidez solo una parte de esta radiación atravesará y será detectado por un fototransistor, y la otra parte de la radiación se perderá en las partículas en suspensión en el líquido. En la figura 9 se muestra el diagrama eléctrico del sensor de turbidez utilizado en la implementación del presente proyecto.

**Figura 9**

Diagrama eléctrico



*Nota.* Adaptado de *Sensor de turbidez*, Hach (<https://essupport.hach.com>)

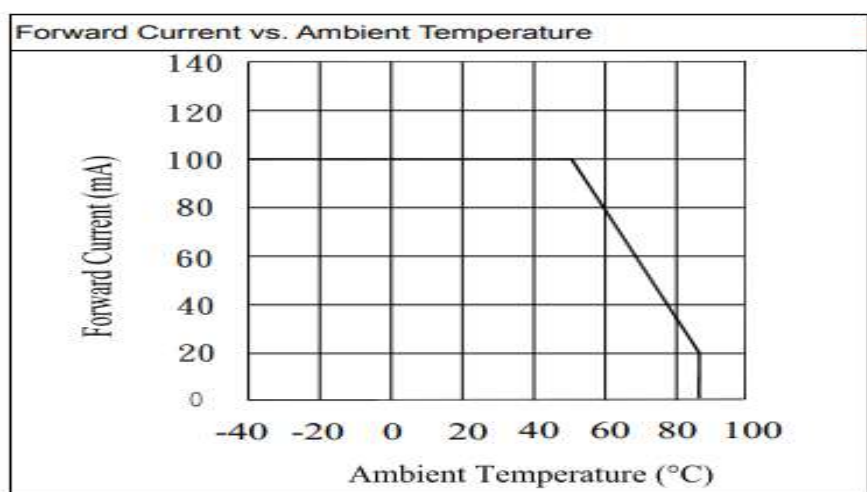
**a). Led Infrarrojo de 5 mm**

Es un dispositivo semiconductor, el cual genera una radiación infrarroja cuando atraviesa una corriente eléctrica en su zona de operación directa.

La variación de la corriente en polarización directa clásica de un led infrarrojo en función de la variación de temperatura sigue la curva de la siguiente figura 10.

**Figura 10**

*Corriente directa en función de la temperatura*

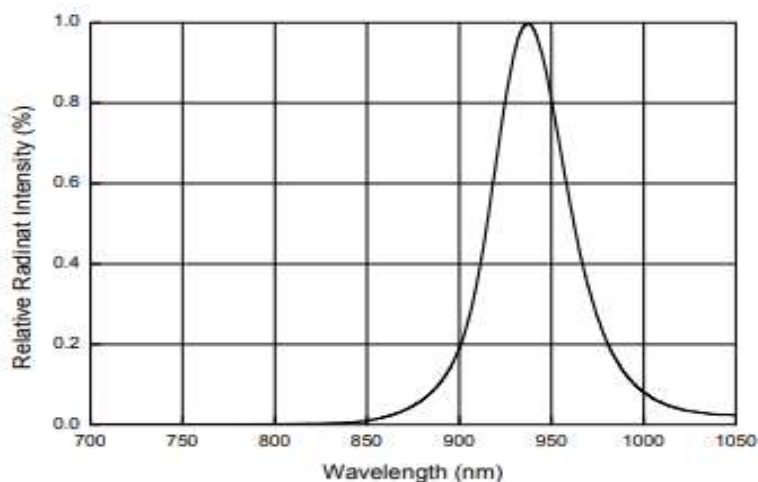


*Nota.* Adaptado de *The 5 mm infrared LED*, Everlight (<https://everlightamericas.com>)

La distribución espectral de un led infrarrojo es mostrada en la siguiente figura 11.

**Figura 11**

*Distribución espectral led infrarrojo 5 mm*

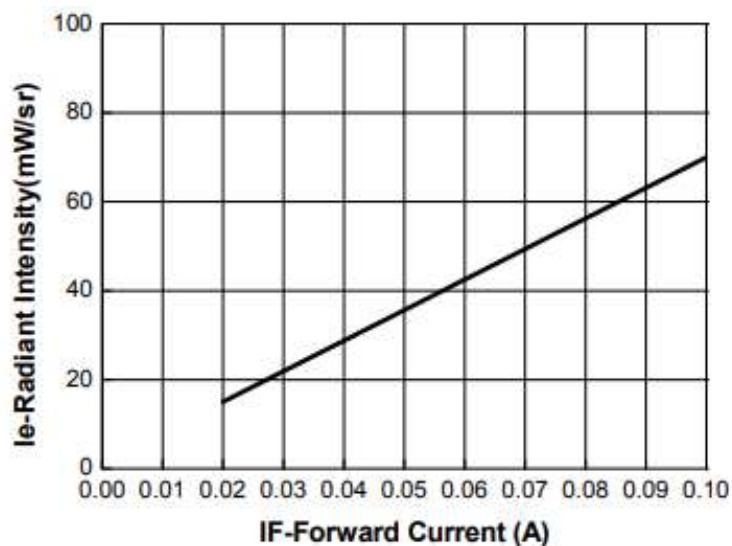


*Nota.* Adaptado de *The 5 mm infrared LED*, Everlight (<https://everlightamericas.com>)

También tenemos la relación de la intensidad de radiación luminosa vs. Corriente directa a través del diodo led tal como se muestra en la siguiente figura 12.

**Figura 12**

*Intensidad de radiación en función de la corriente directa*

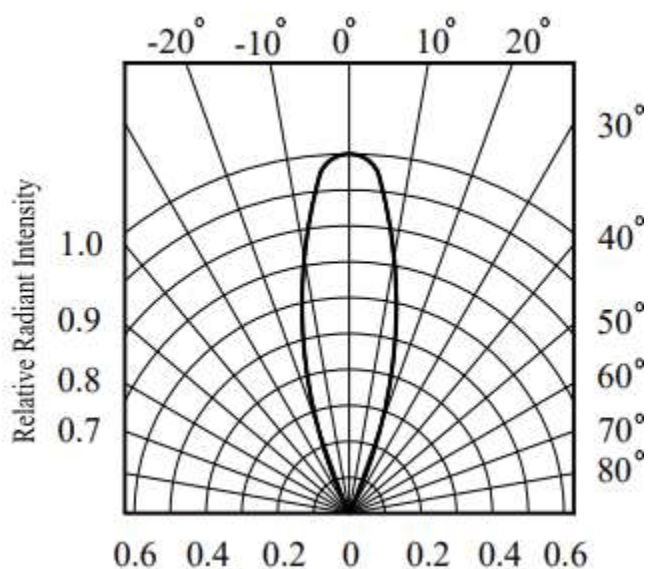


*Nota.* Adaptado de *The 5 mm infrared LED*, Everlight (<https://everlightamericas.com>)

Finalmente, tenemos el patrón de radiación del diodo LED Infrarrojo tal como se muestra en la siguiente figura 13.

**Figura 13**

*Patrón de radiación diodo led infrarrojo*



*Nota.* Adaptado de *The 5 mm infrared LED*, Everlight (<https://everlightamericas.com>)

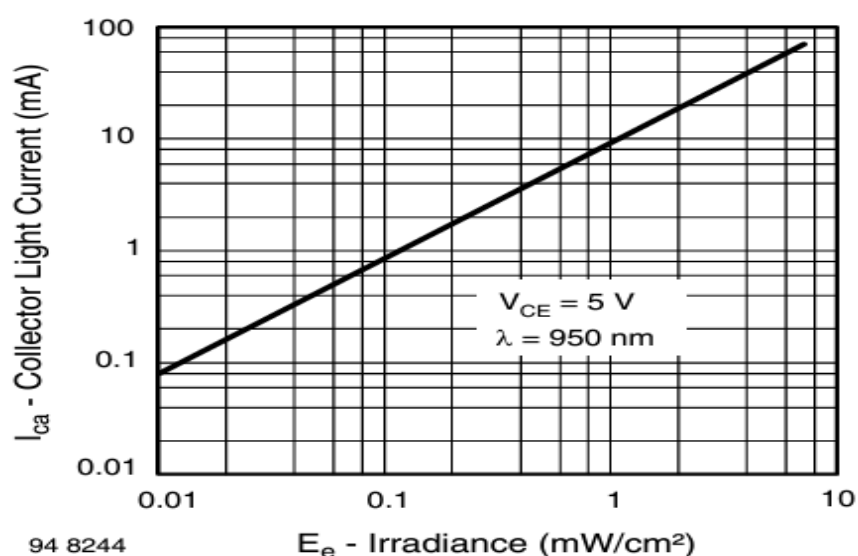
### b). Fototransistor BPV11 NPN Phototransistor

Un fototransistor es un dispositivo sensible a la radiación infrarroja. La radiación incide sobre la región de la base, generando portadores libres en el transistor, estos portadores libres en la base permiten que el transistor entre en un estado de conducción.

A continuación, en la siguiente figura 14 se muestra la relación de la corriente de colector en función de la intensidad de radiación.

**Figura 14**

*Corriente de colector en función de la intensidad de radiación*



*Nota.* Adaptado de *The 5 mm phototransistor*, Everlight (<https://everlightamericas.com>)

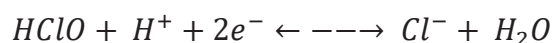
Para el desarrollo del presente proyecto de tesis, se utilizará el sensor de turbidez Gravity Turbidity Sensor, el cual es un dispositivo de uso comercial desarrollado por la empresa DFROBOT – DRIVE THE FUTURE.

#### 2.2.3.4. Sensor de Cloro Residual

El sensor de cloro libre es un dispositivo electroquímico que detecta la presencia de niveles bajos de cloro libre en el agua potable.

Este sensor utiliza como principio de funcionamiento la medición de la corriente eléctrica generada en el proceso de reducción y oxidación de un analito al aplicarle una diferencia de potencial eléctrica adecuada.

La ecuación estequiometríca del análisis de cloro libre residual:

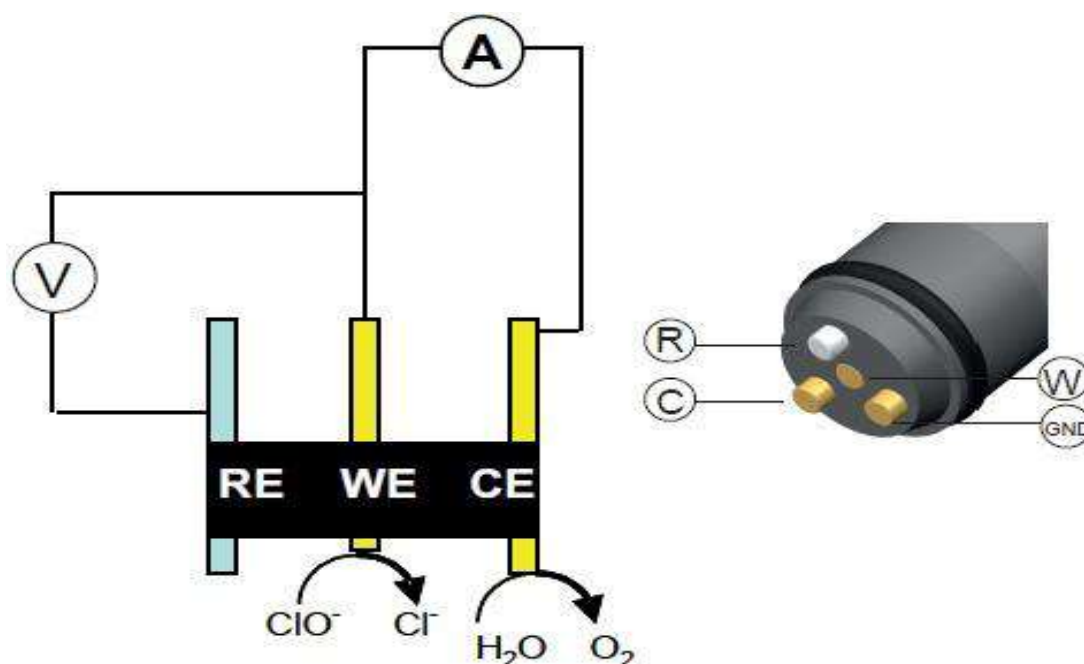


El proceso de reducción del cloro residual se realiza en el electrodo de trabajo de oro, en donde, se aplica un voltaje eléctrico adecuado y a partir de la medición en el electrodo de referencia de Ag/AgCl. El circuito eléctrico para la medición de la corriente eléctrica se bloquea con un electrodo auxiliar (contraelectrodo) de oro. Finalmente, la corriente eléctrica generada es de muy baja intensidad, con el fin de tener una señal eléctrica muy estable se integra un cuarto electrodo de oro, que actúa como puesta a tierra para eliminar cualquier corriente eléctrica residual que pueda estar presente en el agua potable.

El sensor utilizado para la implementación del presente proyecto de tesis es el modelo CL2 sensor de cloro amperométrico, el cual está diseñado para aplicaciones de medición de calidad de agua en el rango de 0 ppm hasta los 3 ppm, con una salida de señal analógica normalizada de 4 – 20 mA. y salida digital en formato UART a 9600 baudios.

**Figura 15**

*Sensor de cloro residual CL2*



*Nota.* Adaptado de *Sensor de cloro residual CL2*, CRF (<https://crfinstruments.com/sensores/>)

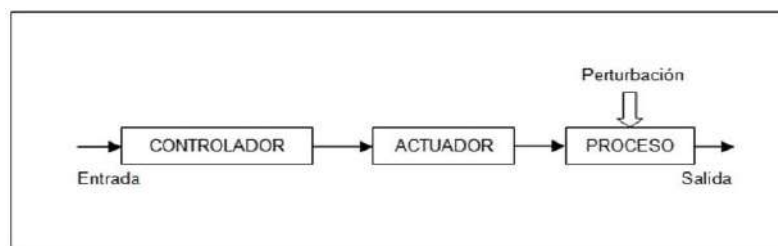
## 2.2.4. Sistema de Control

### 2.2.4.1. Sistemas de Control en Lazo Abierto

Los sistemas de control en lazo abierto se caracterizan porque no presentan realimentación de la variable controlada. A continuación, la figura 16 se muestra un sistema de control en lazo abierto con sus diferentes etapas.

**Figura 16**

*Sistema de control de lazo abierto*

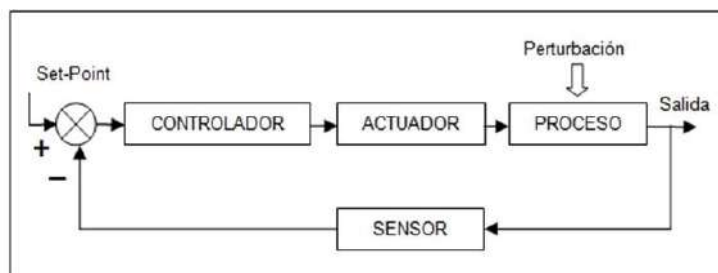


### 2.2.4.2. Sistemas de Control en Lazo Cerrado

Los sistemas de control en lazo cerrado se caracterizan por recibir información de la salida del sistema, por lo tanto, el sistema tendrá un mecanismo de autorregulación. La siguiente figura 17 muestra el diagrama de bloques de un sistema de control en lazo cerrado.

**Figura 17**

*Sistema de control de lazo cerrado*



## 2.2.5. Sistema de Comunicación Inalámbrica IoT

### 2.2.5.1. Internet de las Cosas (Internet of Thing IoT)

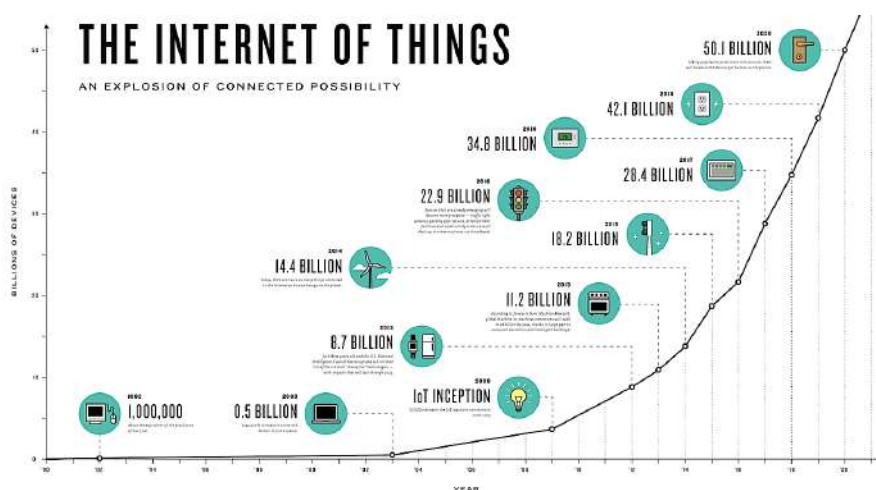
Internet de las cosas es un concepto reciente que describe la interconexión de objetos físicos, los cuales, implementan sensores y sistemas de comunicación para el intercambio de datos en tiempo real.

Se estima que el crecimiento del Internet de las Cosas será exponencial en los siguientes años. Por lo tanto, es correcto afirmar que IoT es la nueva revolución industrial del presente siglo, ya que, evolucionara la economía mundial, en estos tiempos, cuando la tecnología va de la mano con la industrialización de las empresas y los mismos gobiernos tendrán que promover su desarrollo, con el fin de ofrecer a la población una vida más práctica, fácil y cómoda, gracias al avance acelerado del Internet en estos últimos años.

En la siguiente figura 18 se muestra la evolución del IoT en estos últimos años, en donde, claramente se puede observar un crecimiento exponencial, lo cual solo será sostenible en Internet gracias a la implementación de IPv6.

**Figura 18**

*Evolución del IoT*



*Nota:* Adaptado de *Evolución del IoT*, IoT ([www.internetofthing.com](http://www.internetofthing.com))

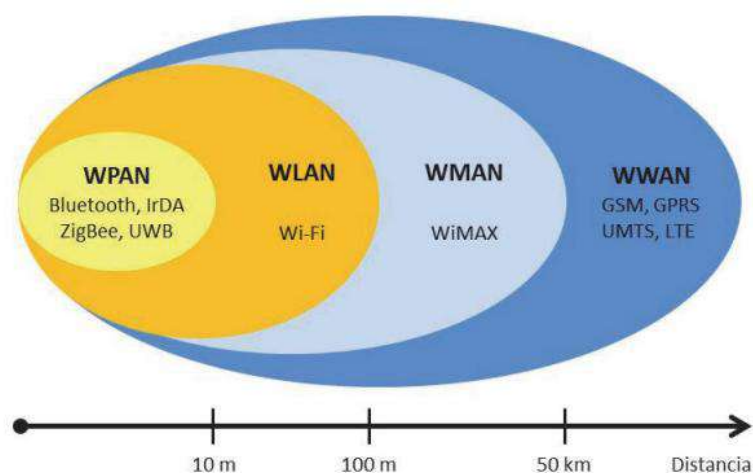
## 2.2.6. Tecnologías Inalámbricas de Comunicación

Las redes de comunicaciones inalámbricas se clasifican en cuatro grupos específicos según su área de aplicación y la amplitud de cobertura geográfica de la señal. Las redes Inalámbricas de Área Personal son redes de pequeño alcance, Redes Inalámbricas de Área Local para campus universitarios o edificios de empresas, Redes Inalámbricas de Área Metropolitana para la cobertura de ciudades y las Redes Inalámbricas de Área Ampla están diseñadas para

cubrir amplias áreas geográficas como países o incluso departamentos. En la siguiente figura 19 se muestra la clasificación de las tecnologías inalámbricas.

**Figura 19**

*Clasificación de las tecnologías inalámbricas*



*Nota.* Adaptado de *Clasificación de tecnologías inalámbricas, IoT*

([www.tecnologiasinalambricas.com](http://www.tecnologiasinalambricas.com))

También, las redes de comunicaciones inalámbricas pueden ser divididas en dos grupos de acuerdo con su alcance: de corto y de largo radio de cobertura.

Las redes inalámbricas de corto radio de cobertura o alcance hacen referencia a las redes destinadas a cubrir un área limitada de solo varios metros cuadrados, las redes de área local se limitan a cubrir áreas como campus universitarios o edificios de empresas; en las redes de área personal (PAN) los dispositivos electrónicos necesitan estar muy cerca entre sí para iniciar el intercambio de información. Estas redes generalmente utilizan un espectro no licenciado y reservado para uso de investigación, industrial y científico. Las bandas de frecuencia no licenciadas más comúnmente utilizadas son la de 2,4 GHz y 5 GHz estas bandas son libres en la mayoría de los países. La disponibilidad para el libre uso de estas bandas no licenciadas permite a los usuarios utilizar estas frecuencias con redes inalámbricas de manera libre y gratuita, ello ha facilitado el crecimiento de este tipo de redes inalámbricas de comunicación.

### 2.2.7. SigFox

La tecnología y protocolo SigFox fue fundada en Francia en el año 2010 con el fin de crear un sistema de comunicación global que se enfocara al desarrollo de Internet de las cosas y operara con un costo muy bajo y un consumo mínimo de energía. La tecnología Sigfox se fundamenta en un enlace por radio de largo alcance, con el fin de que cualquier dispositivo provisto de un sensor pueda comunicarse con un costo energético muy bajo, extrayendo y enviando mensajes muy pequeños.

Características del protocolo de SigFox:

- Autonomía: su consumo de energía extremadamente bajo que permite años de duración de la batería.
- Simplicidad: sin una configuración complicada, solicitud de conexión o señalización.
- Eficiencia de precio: bajo costo del hardware utilizado en los dispositivos y bajas tasas de suscripción al servicio a la red de SigFox.

SigFox es una red de cobertura mundial, que funciona en la banda ISM en todo el mundo. Las operaciones mundiales se dividen actualmente en seis áreas geográficas. Cada área tiene un conjunto diferente de parámetros que limitan claramente la implementación del hardware del dispositivo, principalmente rango de frecuencias, frecuencia central ( $F_c$ ) y potencia máxima irradiada. A continuación, se muestra una tabla con las diferentes regiones RC en la red de SigFox.

**Tabla 4**

*Frecuencias de operación SigFox*

|                                    | RC1     | RC2     | RC3     | RC4     | RC5     | RC6     |
|------------------------------------|---------|---------|---------|---------|---------|---------|
| $F_c$ <i>uplink</i> (MHz)          | 868.130 | 902.200 | 923.200 | 920.800 | 923.300 | 865.200 |
| $F_c$ <i>downlink</i> (MHz)        | 869.525 | 905.200 | 922.200 | 922.300 | 922.300 | 866.300 |
| Velocidad <i>uplink</i> (bits/s)   | 100     | 600     | 100     | 600     | 100     | 100     |
| Velocidad <i>downlink</i> (bits/s) | 600     | 600     | 600     | 600     | 600     | 600     |
| EIRP (dBm)                         | 16      | 24      | 16      | 24      | 14      | 16      |

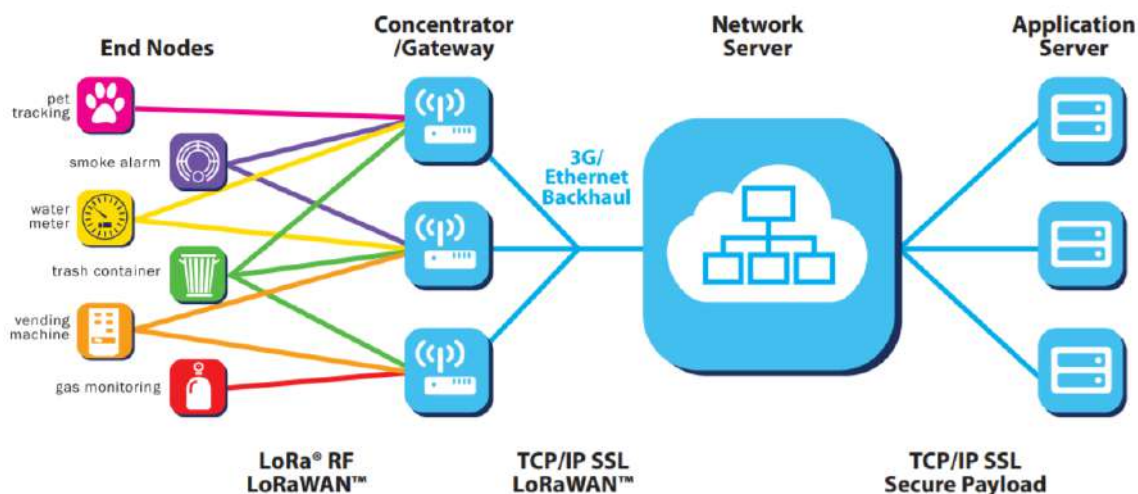
*Nota.* Adaptado de *Frecuencias de operación*, SigFox ([www.sigfox.com](http://www.sigfox.com))

### **2.2.8. LoRaWAN**

**LoRaWAN** es una pila de protocolos de red de comunicaciones inalámbricas que utiliza la tecnología LoRa, para transmisores de baja potencia y amplia cobertura, LPWAN (Low Power Wide Area Network) es empleado para interconectar y administrar dispositivos LoRa desde redes locales, regionales o globales hacia Internet.

#### **2.2.8.1. Arquitectura de Red de LoRaWAN**

Las redes de comunicación LoRaWAN generalmente se encuentran desplegadas en una topología de tipo estrella, donde la puerta de enlace o gateWay transmite los mensajes entre los dispositivos finales y un servidor central en el backend. En una red de comunicación inalámbrica LoRaWAN, los dispositivos finales de transmisión no están necesariamente conectados con un gateWay específico. En cambio, los datos transmitidos por un dispositivo son generalmente recibidos por varios gateWay. Cada puerta de enlace será la encargada de reenviar el paquete recibido desde el dispositivo final al servidor de red el cual generalmente se encuentra alojado sobre la nube a través de algún medio de transporte como el protocolo TCP/IP. El sistema de comunicación LoRaWAN soporta transmisiones bidireccionales, aunque la comunicación de enlace ascendente (UpLink) desde un dispositivo final (sensor) al servidor de la red se espera que sea el tráfico predominante. La siguiente figura 20 muestra la arquitectura clásica de una red LoRaWAN.

**Figura 20***Arquitectura de una red LoRaWAN*

*Nota.* Adaptado de *Arquitectura LoRaWAN*, LoRa Alliance (<https://loro-alliance.org/>)

Los protocolos de seguridad de una red LoRaWAN permite garantizar la autenticidad de la información en el dispositivo receptor de la red, mientras que la seguridad de la capa de superior garantiza que el operador de la red no tenga acceso a los datos de la aplicación de los usuarios finales. El cifrado AES es utilizado para la interconexión de datos mediante identificadores IEEE EUI164.

#### **2.2.8.2. LoRaWAN MAC Layer**

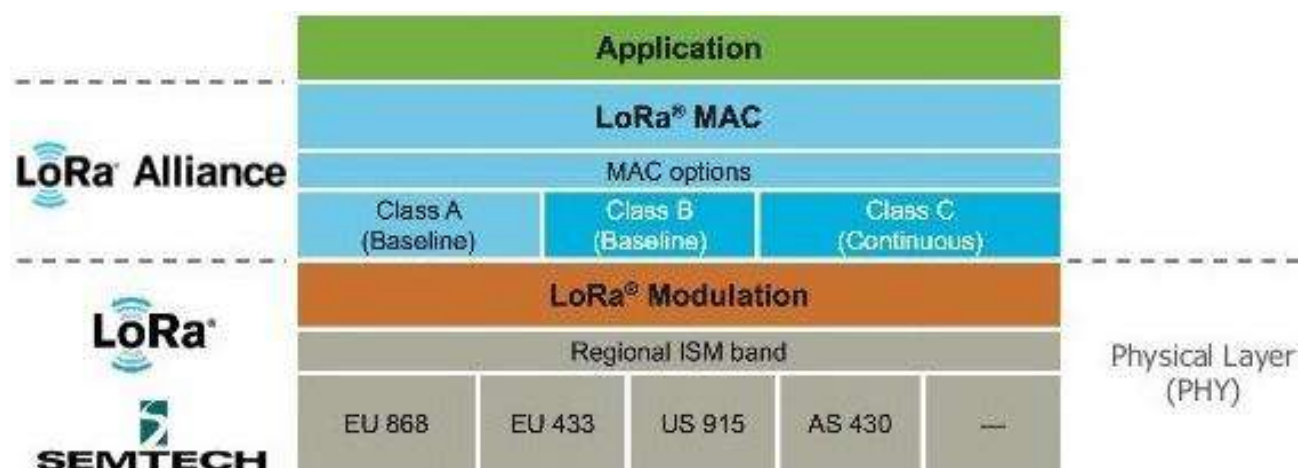
LoRaWAN define el protocolo de la capa MAC (Media Access Control) y la arquitectura de la red de comunicación, mientras que la capa inferior o física está definida por LoRa, esta capa permite una comunicación de gran alcance.

Los protocolos de comunicación y las arquitecturas de las redes de trabajo están orientados a la optimización de la vida útil de la batería de un dispositivo, la capacidad de transmisión y recepción de la red, la calidad del servicio brindado, la seguridad de la información transmitida y la variedad de aplicaciones permitidas por la red. LoRaWAN es de código abierto y está ensamblado por LoRa Alliance.

La siguiente figura 21 muestra el stack de protocolos de LoRa, el cual consta de la capa de Aplicación, la capa MAC definida por LoRaWAN y la capa de física definida por LoRa.

**Figura 21**

*Capa de protocolos LoRaWAN*



*Nota.* Adaptado de *Arquitectura LoRaWAN*, LoRa Alliance (<https://loro-alliance.org/>)

En donde, los datos de la capa de Aplicación requieren el establecimiento de una conexión entre los dispositivos finales (sensores) y el gateway, estos datos son transportados por la capa MAC como un *payload* de datos, seguidamente la capa de MAC construye una trama con el *payload* y un encabezado MAC.

Seguidamente, la capa física utiliza el *MAC payload* para construir un paquete añadiendo un encabezado propio de la capa física, un preámbulo y un cálculo CRC del total del paquete.

Finalmente, este paquete es transmitido por el aire mediante una portadora a una determinada frecuencia. Los parámetros de RF, incluido  $F_c$ , potencia de transmisión, banda de transmisión, etc. Son determinados por la capa física LoRa.

### 2.2.8.3. LoRaWAN Gateway

Como ya se mencionó anteriormente los gateway tienen la función de interconectar los dispositivos finales y los servidores Backend en la nube de internet. Los gateway tienen

implementados dentro de su hardware un *SX-1301* el cual es un chip de banda base digital que administra y procesa una gran cantidad de señales digitales.

Cuando los dispositivos gateway reciben una trama no importa el dispositivo final que los envía, simplemente verifica el CRC de la capa física y luego lo envía hacia el servidor de red por medio de un enlace IP (Ethernet, 5G, LTE, WIFI).

### **2.2.9. LoRa**

LoRa es una tecnología de comunicación inalámbrica de manera similar a WiFi, Bluetooth, etc. define la capa física de la pila de protocolos. LoRa utiliza una técnica de modulación en radiofrecuencia patentada por Semtech.

Esta técnica de modulación es llamada Chirp Spread Spectrum (CSS), y es comúnmente usada en aplicaciones de comunicación militar y aeroespacial desde hace ya muchos años. Su gran ventaja, es que puede lograr transferencia de datos a largas distancias (varios kilómetros) y puede soportar grandes interferencias.

#### **2.2.9.1. Modulación LoRa**

Según la teoría de la información, la relación entre la máxima capacidad de transmisión de información, en ancho de banda y relación de señal a ruido está determinada por la ecuación de Shannon, según la siguiente ecuación:

$$C = B \log_2 \left( 1 + \frac{S}{N} \right)$$

Donde:

C: Capacidad del canal en bps.

B: Ancho de banda en Hz.

S: Potencia de la señal de información en Watt.

N: Potencia de ruido.

Si consideramos el ruido blanco unilateral, luego la potencia total de ruido N en un ancho de banda B es  $B \times N_0$ .

$$C = B \log_2 \left( 1 + \frac{S}{B \cdot N_0} \right)$$

Manipulando la ecuación anterior podemos obtener:

$$C = \left( \frac{S}{N_0} \right) \left( \frac{B \cdot N_0}{S} \right) \log_2 \left( 1 + \frac{S}{B \cdot N_0} \right)$$

$$C = \left( \frac{S}{N_0} \right) \log_2 \left( 1 + \frac{S}{B \cdot N_0} \right)^{\left( \frac{B \cdot N_0}{S} \right)}$$

Debido a que el B será muy grande del orden de los KHz.

Aplicamos el siguiente límite:

$$\lim_{n \rightarrow \infty} \left( 1 + \frac{1}{n} \right)^n = e$$

Por lo tanto, la ecuación quedaría como sigue:

$$C = \left( \frac{S}{N_0} \right) \log_2 e = \left( \frac{S}{N_0} \right) \cdot 1.44$$

De la ecuación anterior podemos deducir que la máxima tasa de transmisión para una determinada potencia no está limitada por el ancho de banda. Para alcanzar este límite solo es necesario incrementar en ancho de banda.

### 2.2.9.2. Spread Spectrum

Según el análisis realizado previamente, al incrementar el BW de la señal, es posible disminuir la degradación de la relación señal a ruido del canal inalámbrico de comunicación.

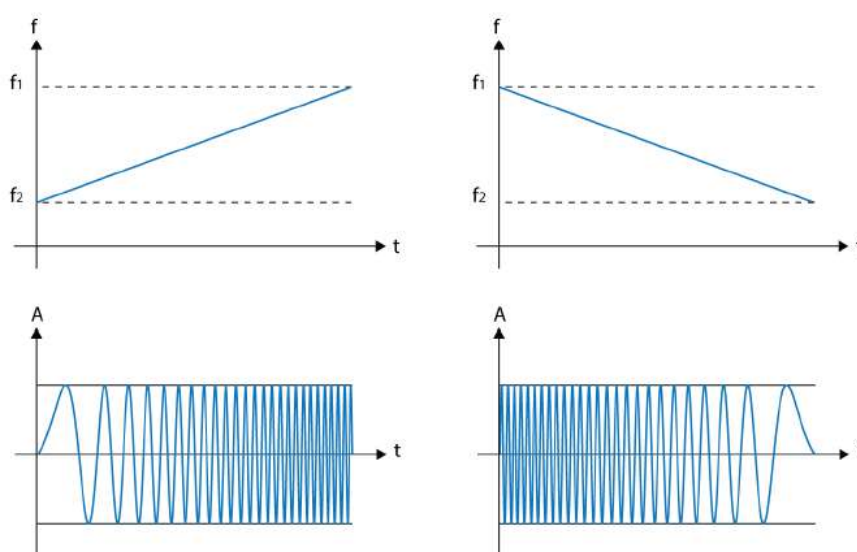
Las técnicas de Spread Spectrum son métodos mediante los cuales se generan señales que gradualmente varían su frecuencia de manera lineal, dando como resultado una señal modulada con un BW más amplio. En la modulación LoRa, el incremento del espectro se obtiene mediante la generación de una señal chirp que varía linealmente de manera continua en frecuencia. Una de las ventajas de la modulación mediante este método es el desplazamiento de la frecuencia y el tiempo, entre el transmisor y el receptor de forma equivalente, reduciendo la dificultad de la implementación del receptor. El BW de la señal generada mediante esta modulación es equivalente al BW espectral de la señal.

### 2.2.9.3. Chirp Spread Spectrum

La modulación LoRa, utiliza la técnica de Chirp Spread Spectrum (CSS), la cual funciona desplazando la frecuencia de un tono RF a través del tiempo de una manera lineal. Las transmisiones LoRa funcionan mediante chirps, interrumpiendo los chirps en diferentes momentos de tiempo y frecuencia para codificar un símbolo. Una de las características importantes de LoRa es la capacidad de generar chirps estables utilizando un bucle enganchado de fase (PLL). A continuación, en la gráfica se muestra señales sinusoidales moduladas mediante CSS; a la izquierda, CSS Up-Chirp y a la derecha, CSS Down-Chirp, ambas señales varían linealmente su frecuencia. En la siguiente figura 22 se muestra la modulación CSS.

**Figura 22**

*Chirps Spread Spectrum*



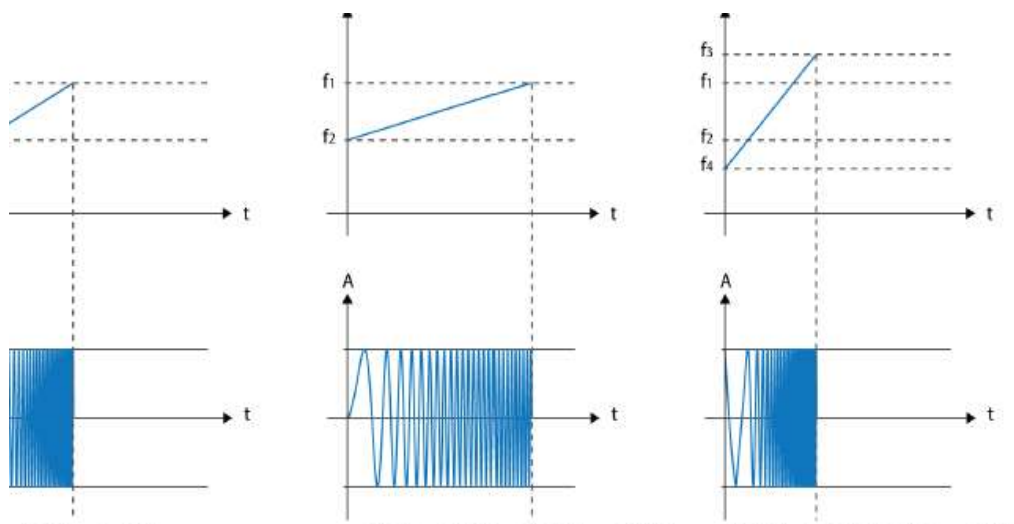
*Nota.* Adaptado de *Modulación LoRa*, LoRa (<https://lora-panama.com>)

Según el ancho de banda (BW) y el factor de dispersión (Spreading Factor SF) seleccionado, el tiempo de símbolo llevará más o menos tiempo. Un incremento en el SF seleccionado duplicará el tiempo del símbolo, por lo que un símbolo con un SF12 será 32 veces más largo que un símbolo enviado en SF7. Sin embargo, BW será inversamente proporcional con el tiempo, duplicar el ancho de banda dividirá el tiempo del símbolo a la mitad.

En la siguiente figura 23, se muestra la relación entre las variaciones del BW, SF y el tiempo de símbolo.

**Figura 23**

*Modulación LoRa*



*Nota.* Adaptado de *Modulación LoRa*, LoRa (<https://lora-panama.com>)

En la figura 23 se puede observar el tiempo de símbolo TS, para una frecuencia de 125KHz con un SF=7; seguidamente se observa un incremento del valor de SF a 8, incrementándose también el tiempo de bit al doble, y finalmente se duplica el BW al doble y se puede reducir el tiempo de bit a la mitad.

#### **2.2.10. Virtualización y Computación en la Nube para Aplicaciones IoT**

La virtualización y la computación en la nube, es el servicio de un conjunto de servidores remotos interconectados a la red de internet para almacenar y procesar datos. En lugar de depender de un servidor físico instalado localmente, la computación en la nube nos permite realizar el proceso de virtualización a través de hipervisores. Existen dos clases de hipervisores que son ampliamente usados en la computación en la nube.

#### **2.2.11. Amazon Web Service (AWS)**

Es un conjunto de servicios de virtualización en la nube que ofrece Amazon como la virtualización, almacenamiento de base de datos, procesamiento de datos, etc.

### 2.2.12. Node-RED

Es una herramienta de desarrollo y programación gráfica está orientada a una mejor visualización gráfica de los datos.

Es ideal para entornos industriales en donde se tienen instalados varios sensores y existe la necesidad de visualizarlos en tiempo real.

Esta herramienta permite visualizar el flujo de la información para una mejor comprensión del estado del sistema.

### 2.2.13. Marco Legal de las Bandas no Licenciadas en el Perú

En el “Texto Único Ordenado del Reglamento General de la Ley de Telecomunicaciones”, el artículo 28 titulado “Bandas no licenciadas”, se define “Asimismo, todos aquellos servicios cuyos equipos, utilizando la banda de 915 - 928 MHz transmiten con una potencia no superior a un watt (1 W) o 30 dBm en antena (potencia efectiva irradiada).”

El Perú adopta la clasificación de los rangos de frecuencias para aplicaciones en servicios móviles avanzados de las Conferencias Mundiales en Radiocomunicaciones:

**Tabla 6**

*Rangos de frecuencias identificados para servicios móviles avanzados*

| Denominación de Banda | Rango de Frecuencias                                                                                                 |
|-----------------------|----------------------------------------------------------------------------------------------------------------------|
| Banda de 450 MHz      | Banda 450-470 MHz                                                                                                    |
| Banda de 700 MHz      | Banda 698 - 806 MHz                                                                                                  |
| Banda de 800 MHz      | Bandas 806 – 821 MHz y 851 – 866 MHz<br>Bandas 821 – 824 MHz y 866 – 869 MHz<br>Bandas 824 - 849 MHz y 869 - 894 MHz |
| Banda de 900 MHz      | Bandas 890 – 915 MHz y 935 – 960 MHz                                                                                 |
| Banda de 1900 MHz     | Bandas 1 850 - 1 910 MHz y 1 930 – 1 990 MHz                                                                         |
| Banda de 1,7/2,1 GHz  | Bandas 1 710 - 1 770 MHz y 2 110 – 2 170 MHz                                                                         |
| Banda de 2,3 GHz      | Banda 2 300 – 2 400 MHz                                                                                              |
| Banda de 2,5 GHz      | Banda 2 500 – 2 690 MHz                                                                                              |
| Banda de 3,5 GHz      | Banda 3 400 – 3 600 MHz                                                                                              |

*Nota.* Adaptado de Resolución ministerial N° 015-2012-MTC, MTC

LoRa Alliance define las especificaciones y parámetros para cada región geográfica, según la clasificación geográfica de LoRa Alliance el Perú se encuentra en la región AU915 la

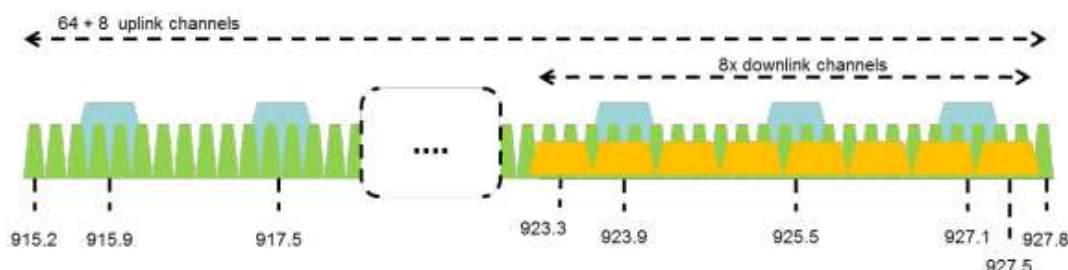
cual comprende frecuencias entre los 915 Mhz y 928 Mhz, correspondiendo a las bandas no licenciadas del Perú.

Para el tráfico de datos ascendente LoRa Alliance divide el espectro en 64 canales numerados del 0 al 63 utilizando un BW de 125 KHz, variando de DR0 a DR5 utilizando una tasa de codificación de 4/5, iniciando en 915.2 Mhz con un incremento lineal de 200 KHz hasta los 927.8 Mhz, además se tiene 8 canales numerados del 64 al 71 con un BW de 500 KHz en DR6, con un incremento lineal de 1.6 Mhz hasta 927.1 Mhz.

Para el tráfico descendente se tienen 8 canales numerados del 0 al 7 con un BW de 500 KHz (DR8 a DR13), iniciando en 923.3 Mhz con un incremento lineal de 600 KHz hasta los 927.5 Mhz.

**Figura 24**

*Frecuencia de canales AU915-928*



*Nota.* Adaptado de *Distribución de frecuencias*, LoRa Alliance (<https://lora-alliance.org/>)

#### **2.2.14. Sistemas Fotovoltaicos**

Son sistemas eléctricos de generación de energía fotovoltaica, mediante la transformación de la energía solar a energía eléctrica, la cual puede ser almacenada en baterías.

A continuación, se describen los conceptos más importantes de los sistemas de generación fotoeléctrica.

*Potencia* es la cantidad de energía entregada o consumida por unidad de tiempo, su unidad de medición son los *Watt*. En donde la relación entre voltaje corriente y potencia está determinada por la siguiente ecuación:

$$W = V * A$$

En donde W: potencia

V: Voltaje

A: Corriente

*Energía* Es la cantidad de potencia entregada multiplicado por las horas de funcionamiento.

$$Wh = W * h$$

Donde Wh: Watt-hora.

W: potencia.

h: tiempo en horas.

*Radiación solar* Es la energía proveniente del sol en forma de radiación luminosa. Su unidad de medida es KWh/m<sup>2</sup>.

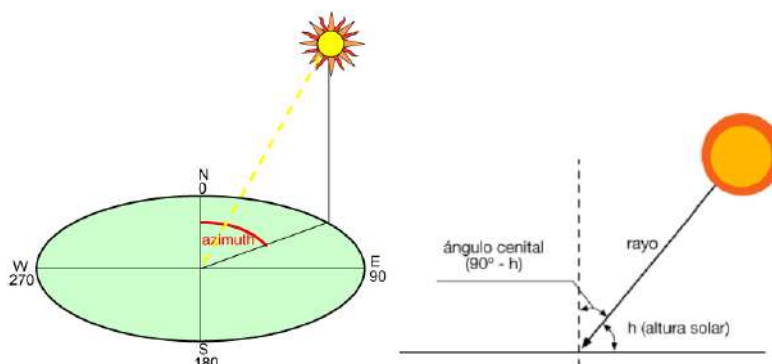
*Movimiento solar* Según estudios se determina que el sol presenta diferentes tipos de movimiento según la época del año. Para describir su recorrido se emplea un sistema coordenado de dos ángulos.

- Altura solar: es el ángulo formado entre la horizontal y la posición aparente del sol.
- Azimut solar: es el ángulo formado entre la proyección horizontal de la posición del sol y el norte verdadero.

En la siguiente figura 25 se muestra gráficamente el concepto de altura y azimut solares.

**Figura 25**

*Ángulos solares*



*Nota.* Adaptado de *Ángulo solar y ángulo azimut*, 2022, Geosystem ([www.geosystem.com](http://www.geosystem.com))

### 2.2.14.1. Componentes Básicos de un Sistema Fotovoltaico

Las partes de un sistema de generación de energía fotovoltaica están formadas principalmente por tres subsistemas.

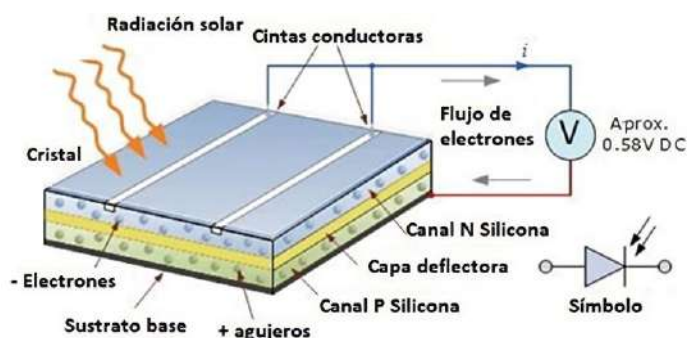
- Subsistema de generación, es la parte encargada de la transformación de la energía solar en energía eléctrica, está conformado principalmente por el panel solar.
- Subsistema de almacenamiento, encargado del almacenamiento de la energía eléctrica generada por el panel solar durante el día, está conformado por la batería.
- Subsistema de control se encarga del monitoreo de todo el sistema fotovoltaico y la supervisión de su correcto funcionamiento.

### 2.2.14.2. Panel Solar

Es el dispositivo encargado de la transformación de la radiación solar en energía eléctrica, para su almacenamiento en baterías y su posterior aprovechamiento. Están conformados por celdas fotovoltaicas de unión PN, las cuales, generan una corriente eléctrica en presencia de fotones, tal como se muestra en la siguiente figura 26.

**Figura 26**

*Unión PN de una celda solar*

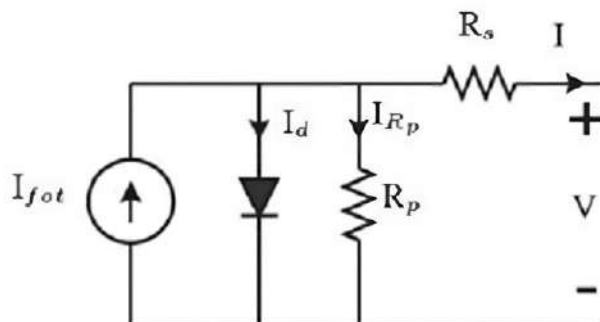


*Nota.* Adaptado de *Celda solar*, 2014, Litio (<https://www.bateriasdelitio.net/?p=1306>)

También es posible modelar una celda solar mediante su equivalente eléctrico, tal como se muestra en la siguiente figura 27.

**Figura 27**

Modelo eléctrico de una celda solar



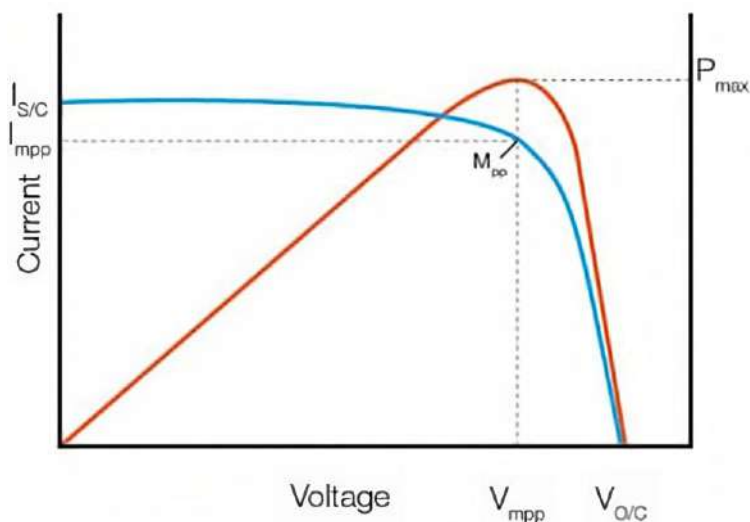
Nota. Adaptado de *Celda solar*, 2015, Litio (<https://www.bateriasdelitio.net/?p=1306>)

Donde el valor de  $V$  e  $I$  representan el voltaje y la corriente respectivamente a la salida de una celda fotovoltaica,  $R_s$  y  $R_p$  representan las resistencias en serie y paralelo de la celda,  $I_{fot}$  representa la corriente producida por el efecto fotovoltaico.

La relación entre la corriente, voltaje y potencia a la salida de un panel solar está representada por la siguiente figura 28.

**Figura 28**

Relación del voltaje y la corriente



Nota. Adaptado de *Celda fotovoltaica*, Baethge Erick ([www.uct.com](http://www.uct.com))

En la gráfica anterior, se puede visualizar la relación entre voltaje vs. corriente (curva azul), el valor de  $I_{s/c}$  es la corriente máxima en corto circuito, el valor de  $V_{o/c}$  es voltaje en circuito abierto; también se puede observar la curva característica de la potencia entregada por

el panel solar vs. voltaje de salida (curva roja), en valor de  $V_{mpp}$  es el voltaje de máxima potencia entregada por el panel solar.

### **2.2.14.3. Baterías y Regulador de Carga**

Las baterías son elementos de almacenamiento de energía eléctrica, están constituidas por celdas electroquímicas, estas celdas pueden generar un suministro de corriente eléctrica a una tensión constante debido a las reacciones químicas dentro de la batería.

En las instalaciones fotovoltaicas es necesario la implementación de un regulador de voltaje o carga el cual monitorea de manera constante la carga y descarga de la batería evitando de ese modo, una posible sobre carga que pueda dañar los componentes del sistema.

### **2.2.15. Sistemas Embebidos**

Son sistemas computacionales basados en un microcontrolador, están diseñados para realizar, una o algunas pocas tareas dedicadas, frecuentemente son utilizados en sistemas de computación en tiempo real. A diferencia de los ordenadores de propósito general (laptop o PC) basados en microprocesador, diseñados para cubrir una amplia cantidad de necesidades, los sistemas embebidos están diseñados para cubrir necesidades específicas.

Generalmente, los sistemas embebidos pueden ser programados en lenguaje ensamblador el cual, se encuentra incorporado en el mismo microcontrolador, también es posible programarlos utilizando lenguaje C mediante un determinado compilador.

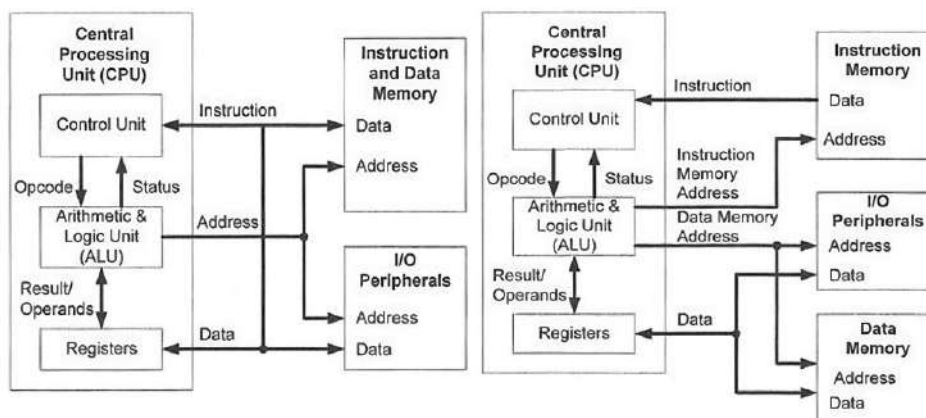
#### **2.2.15.1. Arquitectura de un Sistema Embebido**

Existen dos tipos de arquitectura de microprocesador, tal como se muestra en la figura 29; en la arquitectura Von Neumann la memoria de datos y la memoria de instrucciones comparten una sola memoria física, por lo tanto, solo existe un bus de direcciones de memoria y un solo bus de direcciones de datos. En la arquitectura Harvard, la memoria de datos y la memoria de instrucciones son dos memorias físicamente separadas, por lo tanto, existe un bus

de direcciones de datos y otro bus de instrucciones, lo cual permite un acceso en paralelo a hacia ambas memorias de manera simultánea.

**Figura 29**

*Arquitectura Von Neumann (izquierda) y arquitectura Harvard (derecha)*



*Nota.* Adaptado de *Embedded Systems with ARM Cortex-M* (p. 56), Yiung E., 2014

### 2.2.15.2. Microcontroladores ARM

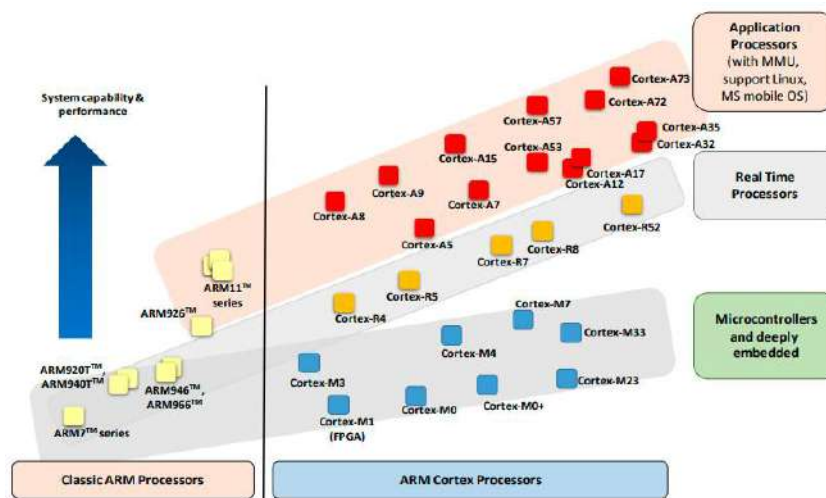
Los procesadores *ARM*, específicamente la familia de los *Cortex-M3* y *Cortex-M4* fueron diseñados para trabajar a una frecuencia de operación del sistema superior a los 80 Mhz a 32 bit de tamaño de palabra. El procesador *Cortex-M3* fue la primera generación de procesadores, liberado por *ARM* en el año 2005. Los procesadores *Cortex-M4* fueron liberados recién en el año 2010. Ambos procesadores utilizan un banco de registros interno, bus de datos y el bus de interfaces de 32 bits.

### 2.2.15.3. Familia de Procesadores Cortex-M

Los procesadores *ARM* según la aplicación a la cual están orientados, son agrupados en tres grandes familias, tal como se muestra en la siguiente figura 30.

**Figura 30**

Familia de microprocesadores ARM Cortex-M



Nota. Adaptado de *Microcontroladores y Microprocesadores* (p. 77), Joseph Yiu, 2010

Según la figura 30 se puede diferenciar tres grupos de familia de microprocesadores Cortex.

*Cortex-A*: Especialmente diseñados para aplicaciones con sistemas operativos embebidos como Android, Linux, Windows etc. Los cuales requieren un poder de procesamiento muy alto como SmartPhone, Tablet, PC, incluso servidores Web.

*Cortex-R*: Orientados para aplicaciones en tiempo real, donde el tiempo de respuesta es muy crítico, las aplicaciones como controladores de disco duro, controladores de comunicaciones de banda base para comunicaciones móviles, y sistemas automotrices, en los que un alto poder de procesamiento y una alta confiabilidad son esenciales para tener una baja latencia

*Cortex-M*: Procesadores diseñados para aplicaciones de menor escala como microcontroladores y sistemas embebidos, donde criterios como bajo costo, bajo consumo de potencia, la eficiencia energética y la baja latencia de interrupción son importantes. Al mismo tiempo, el diseño del procesador debe ser fácil de usar y capaz de proporcionar comportamiento requerido en muchos sistemas de control en tiempo real.

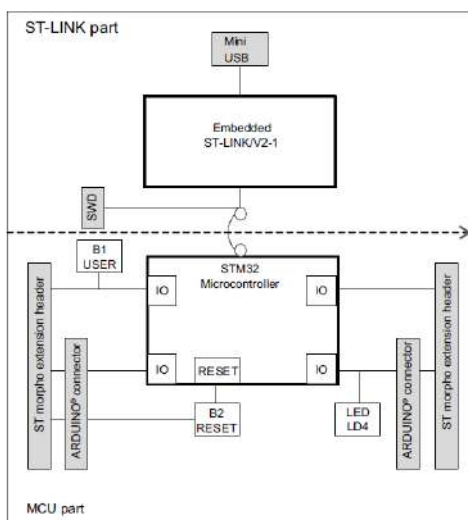
#### **2.2.15.4. Microcontrolador STM32F401RE**

Los microcontroladores STM32F401RE utilizan un núcleo RISC de 32 bits ARM Cortex -M4 de alto rendimiento que opera a una frecuencia del procesador de hasta 84 MHz. Su núcleo Cortex-M4 puede operar unidades de variable de punto flotante (FPU) de precisión simple que admite todas las instrucciones de procesamiento de datos y tipos de datos de precisión simple. También tiene implementado un conjunto completo de instrucciones para procesamiento digital de señales (DSP) y una unidad de protección de memoria (MPU) que brinda mayor seguridad a las aplicaciones. El STM32F401RE cuenta con memorias embebidas de alta velocidad (512 Kbytes de memoria Flash, 96 Kbytes de SRAM) y una gran cantidad de pines de Entrada/Salida y periféricos optimizados conectados a dos buses principales APB, dos buses AHB.

Todos los dispositivos de esta serie ofrecen un ADC de 12 bits de precisión, un RTC de bajo consumo energético, seis temporizadores de 16 bits de uso general, incluido un temporizador PWM, dos temporizadores de 32 bits de uso general. También tiene interfaces de comunicación estándar y avanzadas.

#### **2.2.15.5. Disposición y Configuración del Hardware del STM32F401RE**

La tarjeta de desarrollo STM32F401RE está dividida en dos partes, en la parte inferior se encuentra el microcontrolador STM32 con todos sus periféricos de acceso. El ST-LINK para la programación y depuración se encuentra en la parte superior, tal como se muestra en la figura 31.

**Figura 31***Estructura de la STM32*

*Nota.* Adaptado de *RM0368 Reference manual* (p. 16), STMicroelectronics, 2018

#### 2.2.15.6. Memoria del Sistema de la STM32

Esta tarjeta de desarrollo soporta un amplio rango de características de memoria:

- Espacio de memoria es de 4 GB, con un direccionamiento lineal de 32 bit.
- Característica opcional de bit band.
- Coherencia en la definición del mapa de memoria arquitectónica.
- Unidad de protección de memoria.

#### 2.2.15.7. Microcontrolador ESP32

El ESP32 es el sucesor del ESP8266, el cual integra varias nuevas características como WIFI y BLUETOOTH en una misma tarjeta de desarrollo.

Este microcontrolador integra un procesador *dual core*, ADC de una precisión de 12 bit, comunicación UART, I2C, SPI, sensor de efecto Hall y un DAC de 8 bit de precisión.

#### 2.2.16. Lenguaje Ensamblador y Lenguaje C

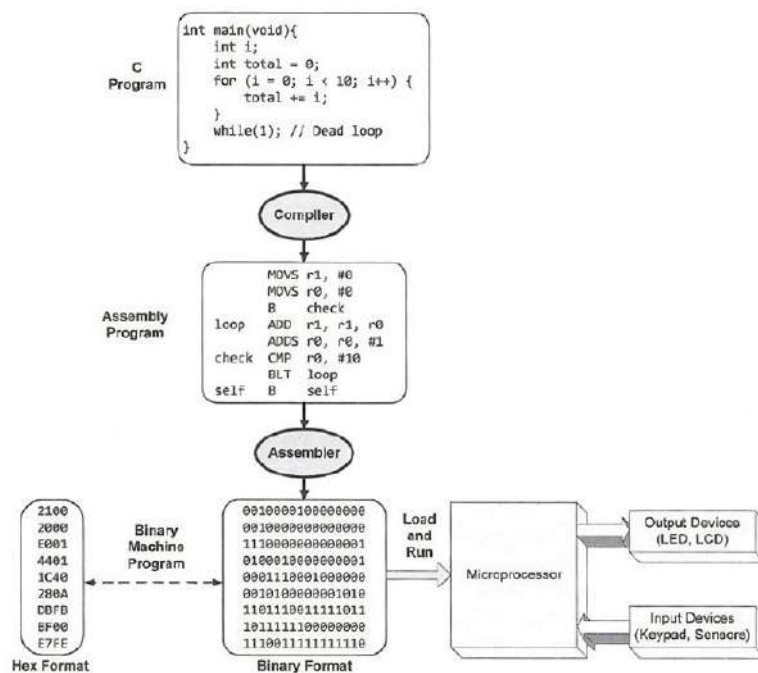
El lenguaje de programación ensamblador (.asm) es un lenguaje de bajo nivel, el cual, consiste en un grupo de mnemónicos que representan instrucciones básicas que son interpretadas por el procesador, mediante una representación simbólica de los códigos binarios

interpretados por el procesador, cada procesador tiene su propio lenguaje ensamblador el cual depende su arquitectura y está definida por el fabricante.

El lenguaje de programación C es un lenguaje de alto nivel, está orientado a la implementación de sistemas operativos, es un lenguaje estructurado con una alta eficiencia lo cual, es ideal para el desarrollo de software para sistemas embebidos, ya que, es posible acceder a direcciones de memoria específicas mediante punteros. La primera estandarización del lenguaje C fue en ANSI, más conocido popularmente como ANSI C

Por lo tanto, para realizar una programación de un procesador en C, es necesario el uso de un compilador, el cual, transforma un lenguaje de programación de alto nivel como C entendible por los humanos, a un archivo de bajo nivel codificado en binario ejecutable y entendible por un computador.

Los archivos ejecutables creados por los compiladores están orientados a una determinada plataforma, por lo tanto, un archivo ejecutable para un determinado microprocesador no puede ser ejecutado directamente en otro microprocesador de una arquitectura diferente como PIC, Atmel AVR porque estos tienen un diferente set de instrucciones. Para los compiladores de C, el programa intermedio es parecido a un lenguaje ensamblador, tal como se muestra en la figura 32, finalmente se traslada el programa ensamblador a un programa máquina, también llamado binario ejecutable que puede ser ejecutado por una determinada plataforma.

**Figura 32***Compilación de código en C a código máquina*

*Nota.* Adaptado de *Microprocesadores y Microcontroladores ARM* (p. 34), Dr. Yifeng Zhu

El lenguaje de programación C utiliza un enfoque estructurado y ordenado para el diseño de programas informáticos. Todo programa en lenguaje C consta, principalmente, de un conjunto de funciones secundarias, y una función principal denominada “main”, la cual es la primera que se ejecuta al comenzar el programa, llamándose desde está a las demás funciones secundarias del programa.

Desde su creación, existen distintas versiones de lenguaje C, que incluyen características como diferentes palabras reservadas. Por este motivo, surgió la necesidad de normalizar el lenguaje C bajo un mismo estándar, y es por ello que surgió un standard de lenguaje C, llamado ANSI-C, que establece un conjunto de características que debe cumplir cualquier lenguaje de programación en C. Gracias a ello, todo programa que se desarrolle siguiendo el standard ANSI C será fácilmente compilado por cualquier modelo de compilador.

### **2.2.16.1.STM32Cube IDE**

Es un software desarrollado por STMicroelectronics para el desarrollo de aplicaciones en sus microcontroladores, es posible programar tanto en lenguaje C y ensamblador. Sus características más importantes son:

- Plataforma de desarrollo para microprocesadores y microcontroladores.
- STM32CubeMX, es una herramienta de programación de software gráfico que permite la generación de código en C de manera automática, para la inicialización mediante asistentes gráficos.
- STM32Cube integra una Capa de abstracción de hardware (HALL), lo que garantiza una facilidad de programación sin la necesidad de manipular los registros internos.
- Un conjunto de instrucciones para trabajar con RTOS, TCP/IP, USB y archivos FAT.
- Soporte para ST-LINK (STMicroelectronics) y J-Link (SEGGER) para realizar prueba de depuración.
- Soporte para Windows, Linux, Mac solo para 64 bit.

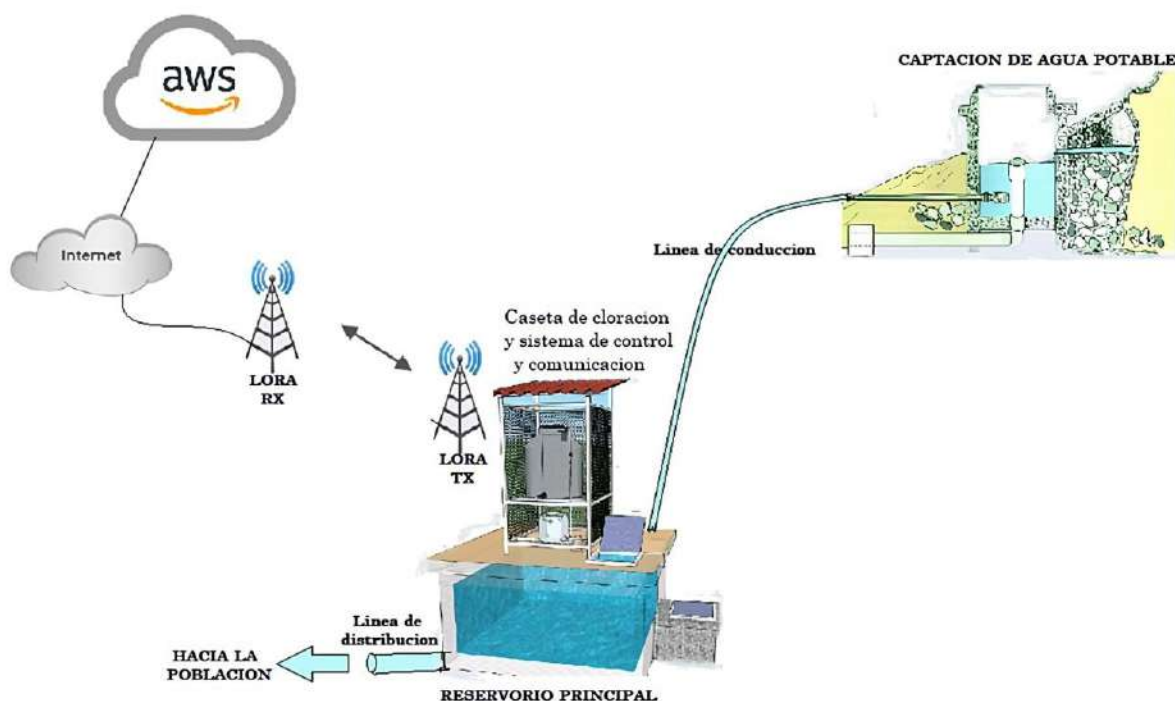
### CAPITULO III: Diseño e Implementación

El sistema de control, comunicación y monitoreo será diseñado e implementado en laboratorio y las pruebas se realizarán sobre el sistema de agua potable de la comunidad de San Pedro.

El sistema de agua potable cuenta con una captación de agua tipo ladera, una línea de conducción de 1" de diámetro, un reservorio de agua de 2.0 m. x 2.0 m. y 2.2 m. de altura, el sistema de desinfección de agua utilizado es el de goteo constante con doble recipiente; el reservorio de agua internamente cuenta con un nivel estático accionado por una boya, el cual, cierra la entrada de agua al reservorio cuando el nivel del agua llega a cierta altura; la figura 33 muestra el sistema de agua con la respectiva implementación del sistema de control y comunicación.

**Figura 33**

*Esquema del sistema de control y comunicación implementado en campo*



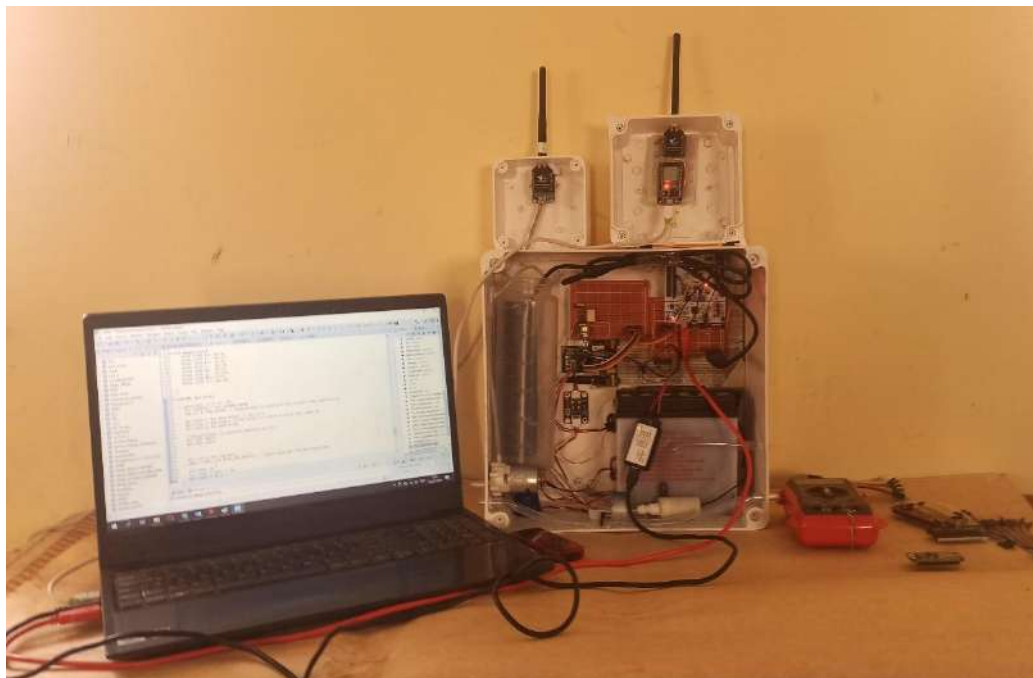
En la figura 33, se muestra la implementación del sistema de comunicación, mediante el cual, las mediciones realizadas por los sensores serán enviadas a una plataforma de Node-RED desplegada sobre AWS cada 30 minutos; para comunicar el punto de ubicación remoto

del sistema de agua con un nodo central se utilizará la tecnología LoRa, seguidamente la información será enviada por la red de Internet mediante API REST.

En la siguiente figura 34 se muestra la implementación final del hardware del proyecto con sus respectivos módulos de comunicación de transmisión y recepción.

**Figura 34**

*Implementación final del proyecto*



### **3.1. Diseño del Sistema de Control**

El sistema de control será diseñado e implementado en hardware, teniendo en cuenta la identificación de las diferentes variables de medición (cloro residual y turbidez), controlador (microcontrolador) y el dispositivo final de control (actuador) el cual, influirá de manera directa en el proceso (sistema de agua potable y sistema de cloración).

Para determinar la cantidad de hipoclorito de calcio necesario para garantizar un adecuado nivel de concentración de cloro residual será necesario realizar el cálculo de la dosificación o caudal de goteo de la solución hipoclorosa hacia el reservorio, el caudal de goteo es una variable que depende del valor del caudal máximo de agua que ingresa al reservorio,

para lo cual, el valor del caudal de agua será medido dentro del reservorio del sistema de agua, tal como se muestra en la siguiente figura 35.

Para determinar un valor más exacto del caudal, se tomarán 5 muestras de agua en un balde de 12 litros y con ayuda de un cronometro se medirá el tiempo de llenado del balde, los resultados de estas mediciones se muestran en la tabla 7, y el caudal del agua se calculará determinado el valor promedio de estas 5 mediciones.

**Figura 35**

*Cálculo del caudal del agua en el reservorio*



**Tabla 7**

*Valores de caudal*

|                 |      |     |      |      |      |
|-----------------|------|-----|------|------|------|
| Volumen (L)     | 12   | 12  | 12   | 12   | 12   |
| Tiempo (Seg.)   | 34.2 | 30  | 37.5 | 33.3 | 34.2 |
| Caudal (L/Seg.) | 0.35 | 0.4 | 0.32 | 0.36 | 0.35 |

Por lo tanto, aplicamos la ecuación para calcular el promedio de la sumatoria de  $n$  muestras.

$$P_{Prom.} = \frac{1}{n} \cdot \sum_{i=1}^n p_i$$

$$Q_{Prom} = \frac{1}{5} (0.35 + 0.4 + 0.32 + 0.36 + 0.35)$$

$$Q_{Prom} = 0.35 \text{ Litros/Segundo}$$

Teniendo el valor exacto del caudal de agua calculamos el volumen total de agua a desinfectar para un tiempo de recarga de 15 días.

$$V_{Total} = Q_{Prom} \times T_{Total}$$

$$V_{Total} = 0.35 \times 15 \times 86400$$

$$V_{Total} = 453600 \text{ Litros}$$

Por lo tanto, ahora aplicamos la ecuación para determinar el peso del hipoclorito de calcio a utilizar.

$$P_{Hcl} = \frac{V \cdot C}{10 \times \%}$$

Donde:

V es el volumen de agua a desinfectar en gramos.

C es la concentración de cloro residual deseado en el reservorio (ppm o mg/L).

% es la concentración del hipoclorito de calcio.

Entonces, como el valor adecuado del cloro residual en las viviendas debe estar entre 0.5 a 1 mg/L, el valor de la concentración C en el reservorio será de 1.2 mg/L, debido a la posible contaminación del agua durante su distribución y de ese modo evitar la disminución de la concentración de cloro residual a valores inadecuados; el valor de % es comercialmente conocido, y su valor casi siempre es de 70.

Reemplazando los valores determinados anteriormente en la ecuación se tiene lo siguiente.

$$P_{Hcl} = \frac{453600 \times 1.2}{10 \times 70}$$

$$P_{Hcl} = 777 \text{ gramos}$$

Ahora, sabemos que el volumen del tanque de solución madre es de 250 Litros, por lo tanto, podemos calcular la concentración de cloro en dicho tanque.

$$P_{Utilizado} = \frac{V_{Tanque} \cdot C_{Tanque}}{10x\%}$$

$$777 = \frac{250x C_{Tanque}}{10x70}$$

$$C_{Tanque} = 2175.6 \text{ mg/Litro}$$

Según la guía técnica en el “Manual del Usuario: Desinfección y cloración” del Ministerio de Vivienda Construcción y Saneamiento, recomienda no exceder 5000 mg/Litro, la concentración de cloro en el tanque de solución madre, para evitar la solidificación de la solución hipoclorosa debido a las bajas temperaturas en la Región del Cusco.

Entonces debemos tener en cuenta:

$$C_{Tanque} < 5000 \text{ mg/Litro}$$

Finalmente calculamos el caudal de goteo de la solución hipoclorosa, para lo cual, tenemos que el tiempo de recarga es de 15 días.

$$q_{goteo} = \frac{250x1000 \text{ miliLitros}}{15x86400 \text{ Segundos}}$$

$$q_{goteo} = 11.5 \text{ mililitros/minuto}$$

### **3.1.1. Implementación del Sistema de Medición del Caudal en Tiempo Real**

#### **3.1.1.1. Calibración del Sensor de Caudal**

Para realizar la calibración del sensor de caudal tenemos la ecuación de caudal:

$$Q = \frac{V}{t} (\text{Litros/segundo})$$

Y también tenemos la ecuación del fabricante dado por:

$$Q = K_p * f$$

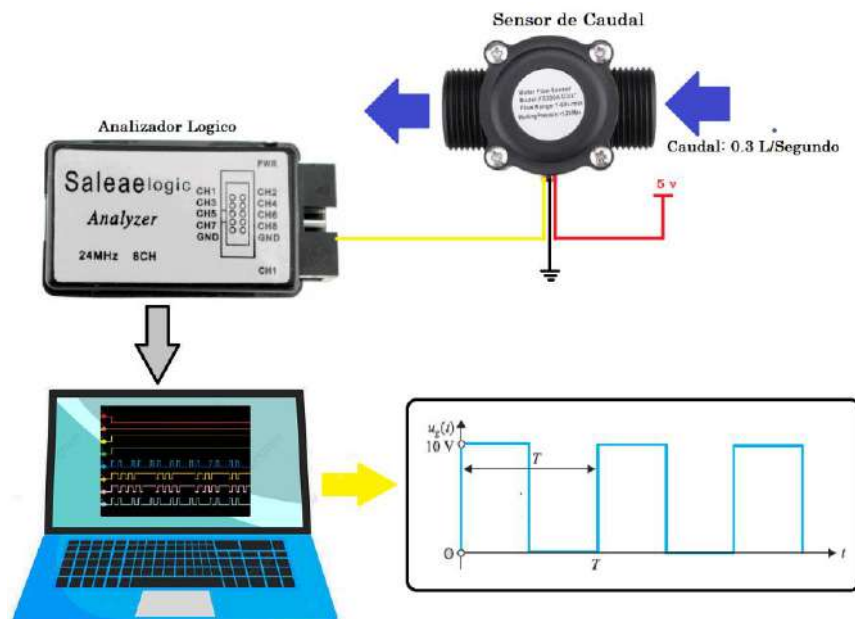
Donde:

Q: es caudal, Kp: constante de calibración y f: es la frecuencia de los pulsos cuadrados generados por el sensor.

Según la hoja de datos del fabricante el valor aproximado de la constante de calibración es de 4.8, pero recomienda realizar un ajuste más preciso. Por lo tanto, se implementa el siguiente diagrama de medición y calibración.

**Figura 36**

Diagrama para la calibración del sensor de caudal



En la siguiente figura 37 se muestra la implementación en laboratorio del sistema de medición y calibración del sensor de caudal.

**Figura 37**

Circuito de medición y calibración del sensor de caudal



Por lo tanto, se realizarán mediciones de la frecuencia a la salida del sensor de caudal tal como se muestra en la siguiente tabla 8.

**Tabla 8**

*Cálculo del factor de calibración*

|                           |      |    |     |      |     |
|---------------------------|------|----|-----|------|-----|
| Caudal (L/min).           | 6    | 12 | 24  | 36   | 48  |
| Frecuencia (Hz).          | 29   | 60 | 112 | 170  | 234 |
| Facto de calibración (K). | 4.83 | 5  | 4.6 | 4.72 | 4.8 |

De la tabla anterior se calcula el valor medio de factor de calibración

$$K_{med} = 4.79$$

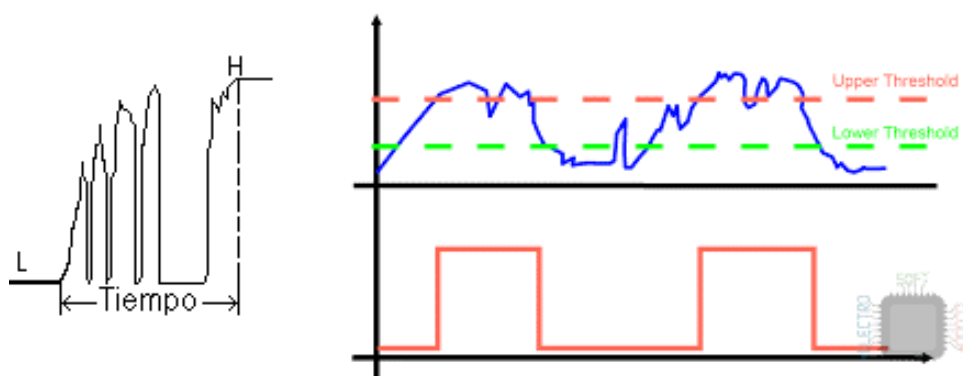
Este valor será utilizado durante la implementación del código en lenguaje C para el STM32 tal como se observa en el anexo B.

### 3.1.1.2. Diseño y Acondicionamiento del Sensor de Caudal

Para evitar falsas mediciones del valor del caudal será necesario realizar un acondicionamiento de la señal cuadrada generada por el sensor y de ese modo evitar el efecto de los falsos contactos o transitorios, en la siguiente figura 38 se muestran las señales reales y su corrección mediante la implementación de un Schmitt Trigger.

**Figura 38**

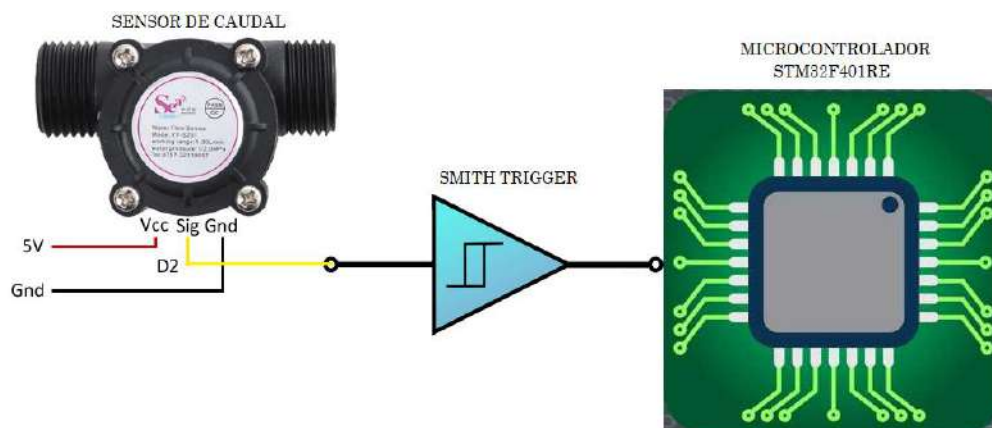
*Acondicionamiento de la señal cuadrada*



Por lo tanto, el circuito que será necesario implementar se muestra en la figura 39, el cual consiste en un circuito integrado Schmitt Trigger (74LS14) conectado entre la salida del sensor de caudal y la entrada digital del microcontrolador.

**Figura 39**

*Acondicionamiento de la señal*

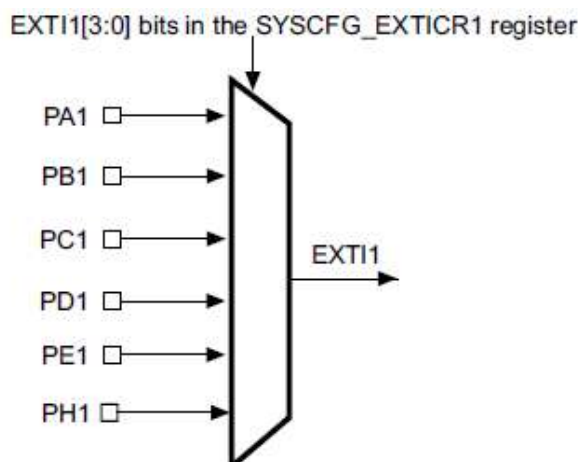


### 3.1.1.3. Configuración de los Periféricos del STM32

Para realizar la medición de la frecuencia de la señal cuadrada generada por el sensor de caudal se utilizará el Pin C1 del microcontrolador, por lo tanto, en dicho Pin se medirá el número de interrupciones por cada flanco de subida generado por el sensor, el número de interrupciones en cada segundo será acumulado en una variable *count* de tipo *uint16\_t* en lenguaje de programación C, en el anexo x se muestra su implementación en STMCubeIDE.

**Figura 40**

*Fuente de interrupción EXTI1*



*Nota.* Adaptado de *RM0368 Reference manual* (p. 45), STMicroelectronics, 2018

El SMT32 cuenta con 84 fuentes de interrupción externa, el PinC1 está conectado a la fuente de interrupción denominada EXTI1, tal como se muestra en la figura 49, para seleccionar el PinC1 como fuente de interrupción será necesario acceder al registro SYSCFG\_EXTICR1 y configurar el valor de los bits EXTI1[3:0] en 0b0001, tal como se muestra en la figura 49.

**Figura 49**

*Registro SYSCFG\_EXTICR1.*

|            |     |     |     |            |     |     |     |            |     |     |     |            |     |     |     |
|------------|-----|-----|-----|------------|-----|-----|-----|------------|-----|-----|-----|------------|-----|-----|-----|
| 31         | 30  | 29  | 28  | 27         | 26  | 25  | 24  | 23         | 22  | 21  | 20  | 19         | 18  | 17  | 16  |
| Reserved   |     |     |     |            |     |     |     |            |     |     |     |            |     |     |     |
| 15         | 14  | 13  | 12  | 11         | 10  | 9   | 8   | 7          | 6   | 5   | 4   | 3          | 2   | 1   | 0   |
| EXTI3[3:0] |     |     |     | EXTI2[3:0] |     |     |     | EXTI1[3:0] |     |     |     | EXTI0[3:0] |     |     |     |
| r/w        | r/w | r/w | r/w | r/w        | r/w | r/w | r/w | r/w        | r/w | r/w | r/w | r/w        | r/w | r/w | r/w |

*Nota.* Adaptado de *RM0368 Reference manual* (p. 46), STMicroelectronics, 2018

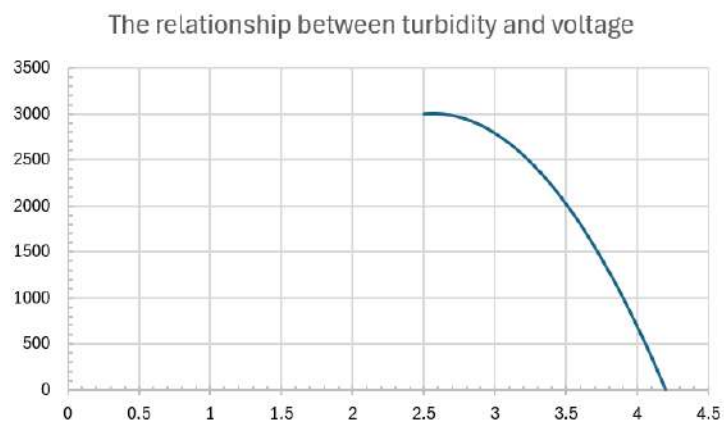
### 3.1.2. Implementación del Sensor de Turbidez

#### 3.1.2.1. Modelado del Sensor de Turbidez

Según la hoja del fabricante el sensor de turbidez DFROBOT está compuesto por un led infrarrojo y un fototransistor, ambos separados 1 cm. Además, cuenta con componentes eléctricos de limitación de corriente como resistencias tal como se muestra en la figura 42; teniendo en cuenta la sensibilidad no lineal del fototransistor con respecto a la radiación infrarroja que incide sobre su base tal como podemos observar en la figura 41, considerando que los niveles de turbidez en el agua potable son bajos (max. 5 UNT para consumo humano) podemos aproximar una correspondencia lineal entre el voltaje de salida del sensor y el nivel de turbidez para un rango de medición de 0 - 1500 UNT tal como se puede observar en la parte inferior de la figura 41.

**Figura 41**

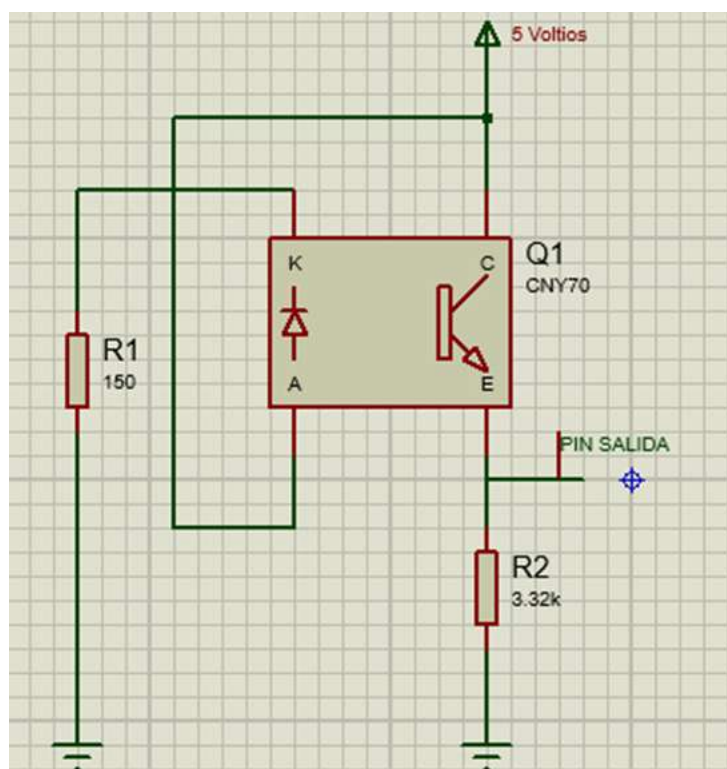
*Grafica voltaje en función al nivel de turbidez*



*Nota. Adaptado de RM0368 Reference manual (p. 67), STMicroelectronics, 2018*

**Figura 42**

*Simulación del sensor de turbidez*



Teniendo en cuenta el rango de operación del sensor de turbidez podemos aproximar una relación lineal entre la corriente en el colector del transistor y la radiación que incide sobre el fototransistor como sigue:

$$I_{colector} = k \cdot E_e$$

Considerando un bloqueo total del paso de la radiación infrarroja el fototransistor se ubicará en su zona de corte, pero a medida que la intensidad de radiación se vaya incrementado (disminuyendo la turbidez), la corriente del colector aumentará proporcionalmente, entonces el transistor entrara a su zona de amplificación.

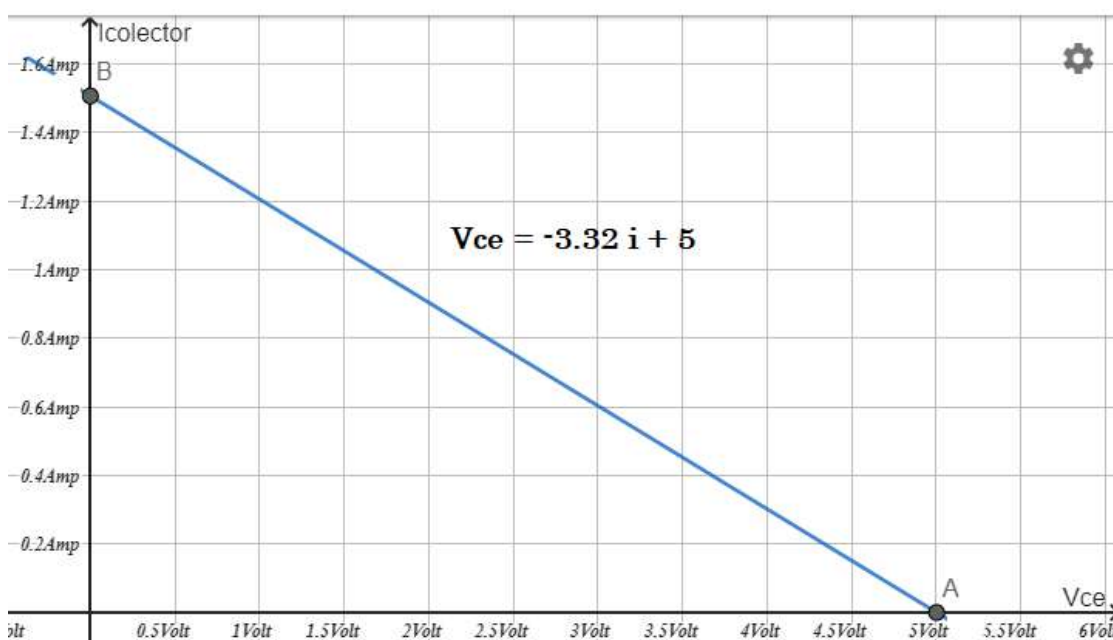
$$5 = v_{CE} + 3.32 i_{colector}$$

Ordenado la ecuación tenemos la recta de polarización del fototransistor:

$$v_{CE} = -3.32 i_{colector} + 5$$

**Figura 43**

*Recta del fototransistor*



El voltaje a la salida del sensor de turbidez estará determinado por la siguiente ecuación.

$$v_{Sensor} = R_2 * i_C$$

### 3.1.2.2. Calibración del Sensor de Turbidez

Para realizar la calibración del sensor de turbidez se utilizará otro sensor profesional ya previamente calibrado de la marca *HANNAinstruments* serie HI93703, tal como se muestra en la figura 44, las especificaciones técnicas de este sensor son las siguientes:

Marca: Hanna Instruments

Modelo: HI93703

**Rango:** 0.00 a 1000 FTU

**Resolución:** 0.01 (0.00 a 50.00 FTU); 1 (50 a 1000 FTU)

**Exactitud:**  $\pm 0.5$  FTU ó  $\pm 5\%$  lectura

Calibración: 2 o 3 puntos (0,10 y 500 FTU)

Alimentación: AA (X4)

*Nota.* Adaptado de *Manual de fabricante, Hannainstruments, 2010*

**Figura 44**

Sensor HANNA HI93703



*Nota.* Adaptado de *Manual de fabricante, Hannainstruments, 2010*

Según el manual de usuario este sensor requiere una calibración previa antes de utilizarlo, para lo cual, se utilizará muestras de fluido con valores de turbidez de 0, 10 y 500 NTU de la marca *HANNAinstruments*, tal como se muestra en la figura 45.

**Figura 45**

*Muestras para calibración de sensor de turbidez (0, 10, 500 NTU)*



Por lo tanto, una vez calibrado el sensor de turbidez, se tomará una muestra del sistema de agua potable y se medirá su nivel de turbidez dando como resultado un valor de 1.2 NTU, tal como se muestra en la figura 46.

**Figura 46**

*Medición de la turbidez del agua potable 1.2 NTU*



Seguidamente, se aumentará de manera gradual el nivel de turbidez para tener varias muestras con diferentes niveles de turbidez, tal como se muestran en las siguientes figuras 47 y 48.

**Figura 47**

*Muestra de agua con nivel de turbidez de 53 NTU*



**Figura 48**

*Muestra de agua con nivel de turbidez de 977 NTU*



Teniendo los valores de turbidez de las diferentes muestras de agua, se procede a medir la salida de voltaje en el sensor para todas las muestras de agua, en las figuras 49 y 50 se observan las mediciones para los niveles de turbidez de 1.2 y 977 NTU.

**Figura 49**

*Medición de voltaje a la salida del sensor (977 NTU)*



**Figura 50**

*Medición de voltaje a la salida del sensor (1.2 NTU)*



Teniendo los valores de las salidas de voltaje para las 7 muestras con diferentes niveles de turbidez se construye la siguiente tabla.

**Tabla 9**

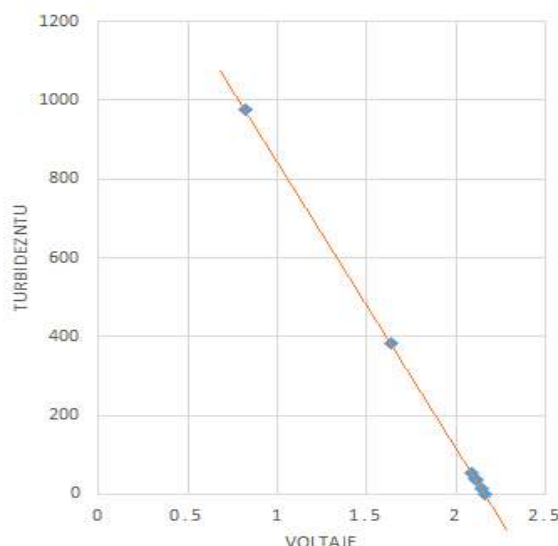
*Salida de voltaje para diferentes niveles de turbidez*

| Nro. Muestra   | 1    | 2     | 3     | 4     | 5    | 6    | 7    |
|----------------|------|-------|-------|-------|------|------|------|
| Turbidez (NTU) | 1.2  | 14.41 | 34.11 | 40.44 | 53   | 383  | 977  |
| Voltaje        | 2.16 | 2.14  | 2.11  | 2.10  | 2.08 | 1.63 | 0.82 |

Según los datos mostrados en la tabla 9 se puede observar una disminución del voltaje a medida que aumenta la turbidez de la muestra de agua, lo cual tiene sentido debido a las características ópticas del sensor, por lo tanto, la salida de voltaje será nula cuando se bloquee completamente el paso de la luz infrarroja (nivel de turbidez infinito). Con los valores obtenidos de la tabla 9, en la figura 51 se muestra la gráfica Turbidez (NTU) vs. Voltaje para el sensor de turbidez alimentado con un nivel de tensión de 3.3 voltios.

**Figura 51**

*Grafica turbidez (NTU) en función del voltaje*



Mediante el método de mínimos cuadrados se calcula los parámetros de la recta que mejor se ajusta a los puntos mostrados en la figura 51.

Se tiene la ecuación de la recta:  $y = m * x + b$ , donde  $b = \frac{\sum y \sum x^2 - \sum x \sum xy}{N \sum x^2 - \sum x \sum x}$  y  $m =$

$\frac{N \sum yx - \sum x \sum y}{N \sum x^2 - \sum x \sum x}$ , se calcula los valores de  $m = -728.2$  y  $b = 1574$ , por lo tanto tenemos la ecuación

de la recta que mejor se ajusta los puntos mostrados en la gráfica anterior:

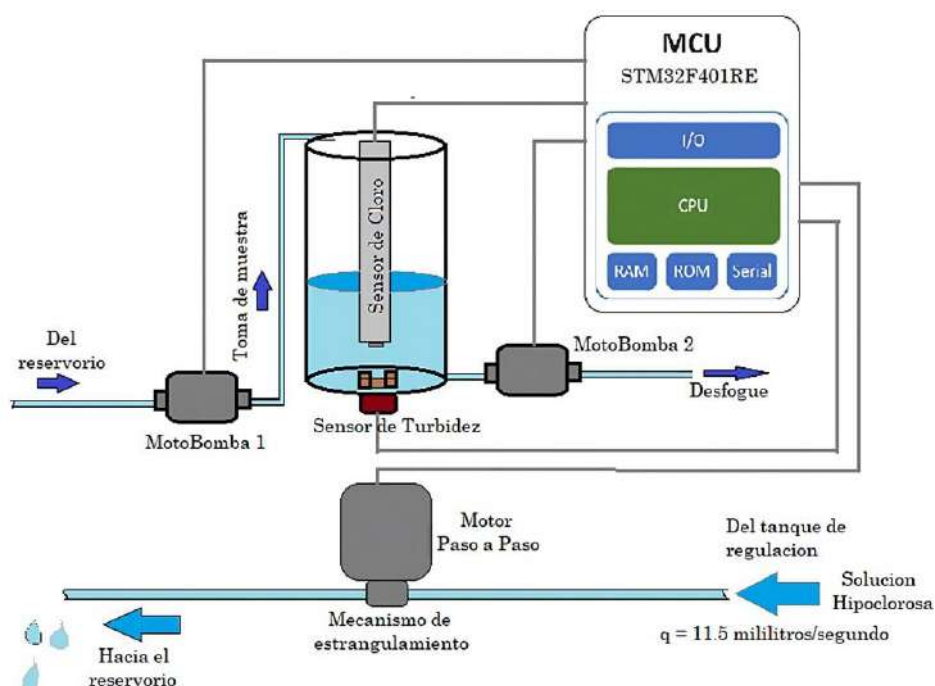
$$T_{(v)} = -728.2v + 1574$$

### 3.1.2.3. Implementación del Sistema de Medición de la Turbidez

El sistema de medición de turbidez, será una parte del sistema de control completo tal como se muestra en la figura 52, el sensor de turbidez estará ubicado en la parte inferior del recipiente de toma de muestra, dicha muestra será obtenida directamente del reservorio del sistema de agua mediante la motobomba 1, una vez realizada la medición de la turbidez y el cloro residual, el microcontrolador encenderá la motobomba 2 para realizar el desfogue y la eliminación de la muestra; según las mediciones realizadas por el microcontrolador, también se controlara la posición del motor paso a paso, el cual abrirá y cerrara el flujo de la solución hipoclorosa hacia el reservorio.

**Figura 52**

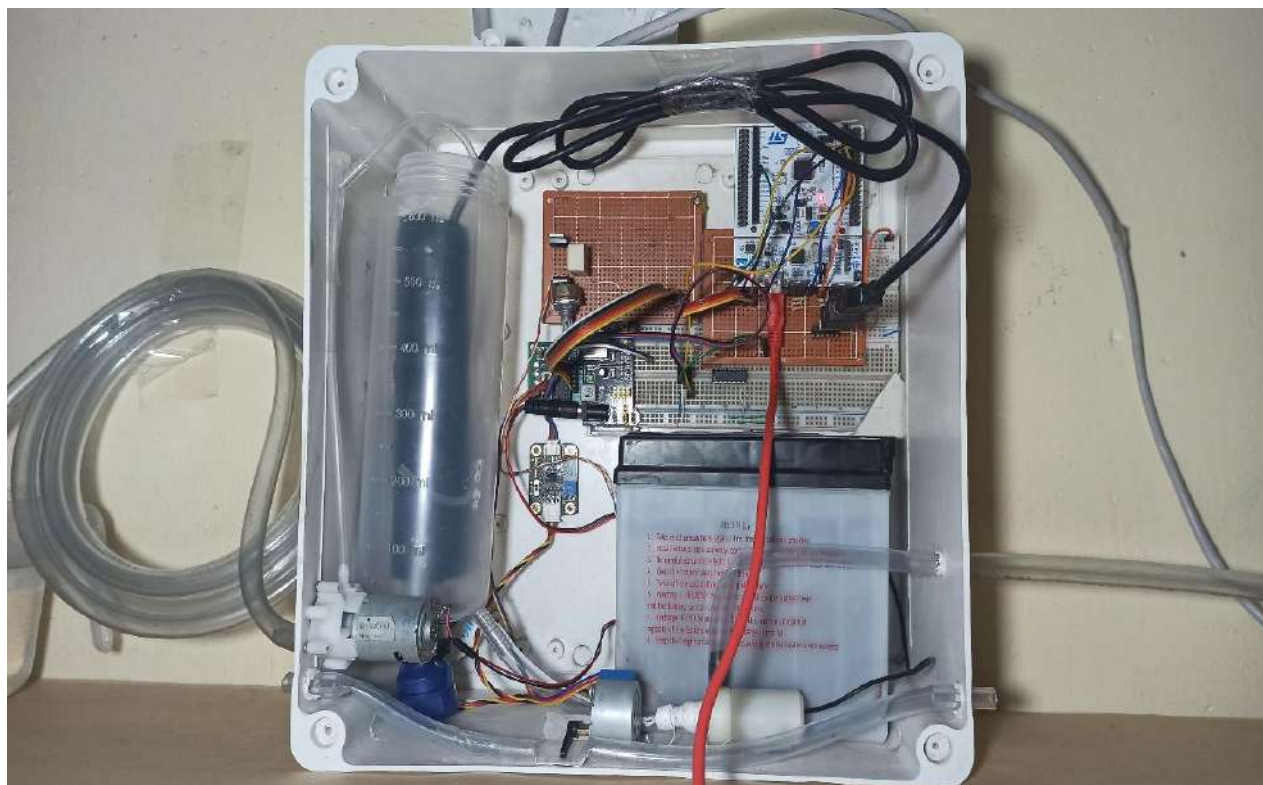
*Sistema de medición de turbidez*



En la siguiente figura 53 se muestra la implementación de hardware del sistema de medición y control, con sus respectivas conexiones al microcontrolador, su alimentación mediante una batería, sensor de cloro, sensor de turbidez, mini bombas de agua y su carcasa de protección.

**Figura 53**

*Sistema de control y medición instalado*



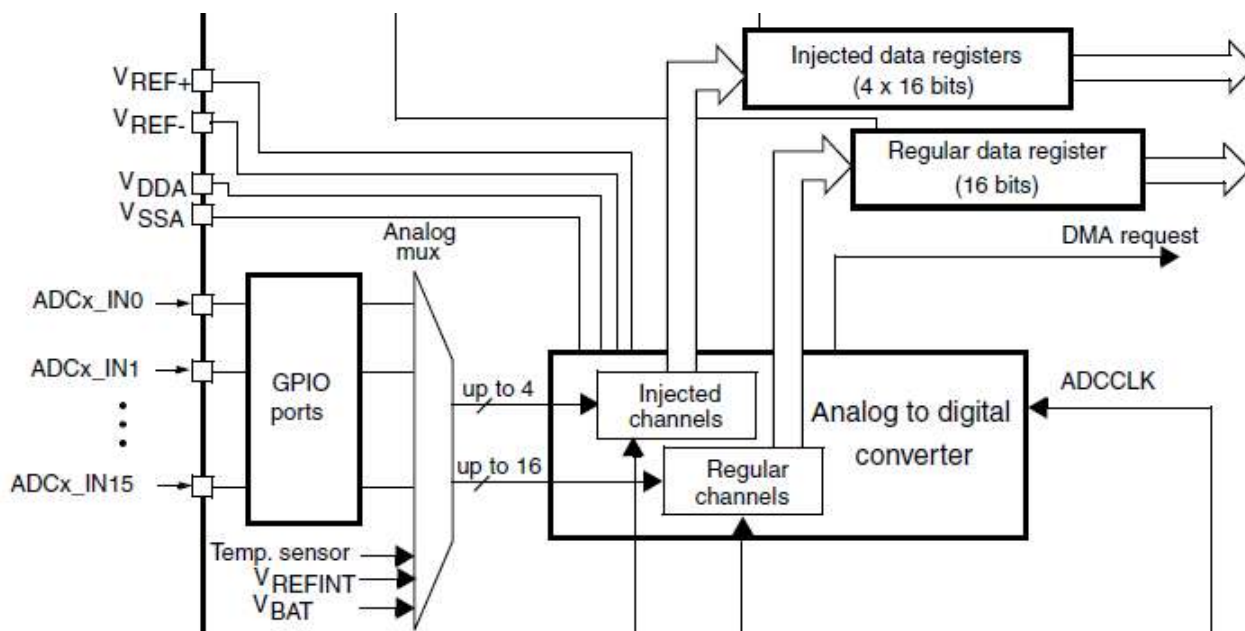
#### **3.1.2.4. Configuración del ADC del STM32**

El STM32 cuenta con solamente un periférico ADC, el cual, puede ser utilizado con una resolución de 12, 10, 8 y 6 bits, en el desarrollo de este proyecto se utilizara solamente 8 bits de resolución para poder empaquetar la medición realizada en una trama de UART de 8 bits de palabra sin bit de paridad y un bit de stop; el método de conversión utilizado es el de aproximaciones sucesivas, este ADC cuenta con varios canales de entrada que pueden ser multiplexados uno a la vez, mediante la configuración de determinados registros, en la siguiente figura 54 se muestra una parte del periférico ADC del STM32, con sus respectivos 16 canales

analógicos externos y tres canales internos, los cuales son seleccionados mediante un multiplexor.

**Figura 54**

*Periférico ADC*

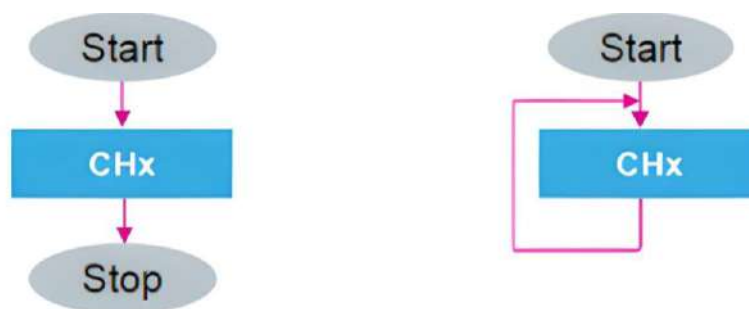


*Nota.* Adaptado de RM0368 Reference manual (p. 67), STMicroelectronics, 2018

El ADC tiene dos métodos de funcionamiento: *Single Mode* y *Continuous Mode*, la figura 55 muestra el esquema de funcionamiento de ambos métodos, en el modo de operación Simple una vez que termina la conversión esta se detiene hasta una señal de inicio de conversión mediante software, mientras en el modo de operación continuo la conversión reinicia inmediatamente después de terminada la anterior conversión.

**Figura 55**

*Modo de operación del ADC simple y continuo*



*Nota.* Adaptado de RM0368 Reference manual (p. 56), STMicroelectronics, 2018

Para la implementación del presente proyecto de tesis se utilizará el método de conversión simple, para lo cual será necesario colocar el bit SWSTART en 1 lógico del registro ADC\_CR2, tal como se muestra en la siguiente figura 56.

**Figura 56**

*Registro ADC\_CR2*

|          |             |       |     |       |             |     |     |          |          |              |        |     |     |              |      |
|----------|-------------|-------|-----|-------|-------------|-----|-----|----------|----------|--------------|--------|-----|-----|--------------|------|
| 31       | 30          | 29    | 28  | 27    | 26          | 25  | 24  | 23       | 22       | 21           | 20     | 19  | 18  | 17           | 16   |
| reserved | SWST<br>ART | EXTEN |     |       | EXTSEL[3:0] |     |     |          | reserved | JSWST<br>ART | JEXTEN |     |     | JEXTSEL[3:0] |      |
|          | r/w         | r/w   | r/w | r/w   | r/w         | r/w | r/w | r/w      |          | r/w          | r/w    | r/w | r/w | r/w          | r/w  |
| 15       | 14          | 13    | 12  | 11    | 10          | 9   | 8   | 7        | 6        | 5            | 4      | 3   | 2   | 1            | 0    |
| reserved |             |       |     | ALIGN | EOCS        | DDS | DMA | Reserved |          |              |        |     |     | CONT         | ADON |
|          |             |       |     | r/w   | r/w         | r/w | r/w |          |          |              |        |     |     | r/w          | r/w  |

*Nota.* Adaptado de *RM0368 Reference manual* (p.57), STMicroelectronics, 2018

Este bit es puesto a cero por hardware inmediatamente después de iniciada la conversión, para iniciar otra conversión será necesario ponerlo a uno lógico mediante software. En el presente proyecto la salida del sensor de turbidez está conectado al PinB0 el cual está asignado a canal analógico 8 (chx = 8), los bits L[3:0] del registro ADC1\_SQR1 nos permite seleccionar el número de conversiones regulares, en nuestro caso solo realizaremos una conversión, por lo tanto, según la hoja de datos correspondería colocar el valor de 0b0000, la siguiente figura 57 muestra el registro ADC1\_SQR1.

**Figura 57**

*Registro ADC1\_SQR1*

|          |           |     |     |     |     |           |     |        |     |     |           |           |     |     |     |
|----------|-----------|-----|-----|-----|-----|-----------|-----|--------|-----|-----|-----------|-----------|-----|-----|-----|
| 31       | 30        | 29  | 28  | 27  | 26  | 25        | 24  | 23     | 22  | 21  | 20        | 19        | 18  | 17  | 16  |
| Reserved |           |     |     |     |     |           |     | L[3:0] |     |     |           | SQ16[4:1] |     |     |     |
|          |           |     |     |     |     |           |     | r/w    | r/w | r/w | r/w       | r/w       | r/w | r/w | r/w |
| 15       | 14        | 13  | 12  | 11  | 10  | 9         | 8   | 7      | 6   | 5   | 4         | 3         | 2   | 1   | 0   |
| SQ16_0   | SQ15[4:0] |     |     |     |     | SQ14[4:0] |     |        |     |     | SQ13[4:0] |           |     |     |     |
| r/w      | r/w       | r/w | r/w | r/w | r/w | r/w       | r/w |        |     |     | r/w       | r/w       | r/w | r/w | r/w |

*Nota.* Adaptado de *RM0368 Reference manual* (p. 57), STMicroelectronics, 2018

Las secuencias de las conversiones están definidas por los bits SQR1- SQR16 de los registros ADC1\_SQR1 - ADC1\_SQR3, para realizar la primera conversión será necesario indicar el valor del canal (chx = 8) en los bits SQR1 del registro ADC1\_SQR3, tal como se muestra en la siguiente figura 58.

**Figura 58****Registro ADC1\_SQR3**

|          |     |          |     |     |     |     |          |     |     |     |     |          |     |     |     |
|----------|-----|----------|-----|-----|-----|-----|----------|-----|-----|-----|-----|----------|-----|-----|-----|
| 31       | 30  | 29       | 28  | 27  | 26  | 25  | 24       | 23  | 22  | 21  | 20  | 19       | 18  | 17  | 16  |
| Reserved |     | SQ6[4:0] |     |     |     |     | SQ5[4:0] |     |     |     |     | SQ4[4:1] |     |     |     |
|          |     | r/w      | r/w | r/w | r/w | r/w | r/w      | r/w | r/w | r/w | r/w | r/w      | r/w | r/w | r/w |
| 15       | 14  | 13       | 12  | 11  | 10  | 9   | 8        | 7   | 6   | 5   | 4   | 3        | 2   | 1   | 0   |
| SQ4_0    |     | SQ3[4:0] |     |     |     |     | SQ2[4:0] |     |     |     |     | SQ1[4:0] |     |     |     |
| r/w      | r/w | r/w      | r/w | r/w | r/w | r/w | r/w      | r/w | r/w | r/w | r/w | r/w      | r/w | r/w | r/w |

*Nota.* Adaptado de *RM0368 Reference manual* (p. 57), STMicroelectronics, 2018

Finalmente, la comunicación UART utilizada entre el microcontrolador y el RAK811, será de 8 bits por palabra, sin bit de paridad y 1 bit de parada (stop), por lo tanto, la resolución del convertidor analógico digital tendrá que ser de 8 bits, teniendo  $2^8$  niveles de codificación.

Según la hoja de datos del fabricante el registro necesario para configurar la codificación en 8 bits de palabra es el ADC\_CR1, escribiendo los bits RES[0:1] en 0b10.

**Figura 59****Registro ADC\_CR1**

|              |     |     |         |        |       |        |      |        |        |          |            |     |     |     |     |
|--------------|-----|-----|---------|--------|-------|--------|------|--------|--------|----------|------------|-----|-----|-----|-----|
| 31           | 30  | 29  | 28      | 27     | 26    | 25     | 24   | 23     | 22     | 21       | 20         | 19  | 18  | 17  | 16  |
| Reserved     |     |     |         |        | OVRIE | RES    |      | AWDEN  | JAWDEN | Reserved |            |     |     |     |     |
|              |     |     |         |        | r/w   | r/w    | r/w  | r/w    | r/w    |          |            |     |     |     |     |
| 15           | 14  | 13  | 12      | 11     | 10    | 9      | 8    | 7      | 6      | 5        | 4          | 3   | 2   | 1   | 0   |
| DISCNUM[2:0] |     |     | JDISCEN | DISCEN | JAUTO | AWDSSL | SCAN | JEOCIE | AWDIE  | EOCIE    | AWDCH[4:0] |     |     |     |     |
| r/w          | r/w | r/w | r/w     | r/w    | r/w   | r/w    | r/w  | r/w    | r/w    | r/w      | r/w        | r/w | r/w | r/w | r/w |

*Nota.* Adaptado de *RM0368 Reference manual* (p. 58), STMicroelectronics, 2018

### 3.1.3. Implementación del Sensor de Cloro Residual

Para la medición de la concentración de cloro residual se utilizará la misma muestra de agua obtenida durante la medición de la turbidez, el sensor de cloro también está adaptado al mismo recipiente en donde se toma la muestra de agua tal como ya se mostró en la figura 42,

#### 3.1.3.1. Calibración del Sensor de Cloro Residual

Para realizar la calibración del sensor de cloro residual, se utilizarán muestras de agua con diferentes niveles de cloro residual, dichos niveles serán medidos mediante el uso de un colorímetro manual, el cual, mide la coloración del agua en presencia de un reactivo DPD añadido a una muestra de agua de 10 mL, en la siguiente figura 60 se muestra los niveles de coloración para muestras desde cero a 3.5 mg/L de concentración, correspondiendo una

coloración transparente a 0 mg/L y un incremento gradual de una coloración rojiza para muestras de cloro residual sucesivamente mayor. En la siguiente figura 60 se muestra un colorímetro manual tipo disco.

**Figura 60**

*Niveles de coloración en presencia del reactivo DPD*



*Nota. Adaptado de Colorímetro manual, Hanna ([www.hannainstrument.com](http://www.hannainstrument.com))*

**Figura 61**

*Medición del nivel cloro residual*



En la figura 61 en el lado derecho se observa la muestra de agua transparente con cloro residual, y en la parte izquierda la misma muestra con presencia del reactivo DPD, se puede observar claramente el nivel de coloración.

Por lo tanto, la intensidad de la coloración en presencia del reactivo DPD será proporcional a la concentración de cloro residual en el agua.

Para realizar la calibración del sensor CL2, se tiene las muestras de agua con diferentes niveles de concentración de cloro residual tal como se muestran en las siguientes figuras 62 y 63.

**Figura 62**

*Muestra de agua con intensidad de coloración de 0.2 mg/L*



**Figura 63**

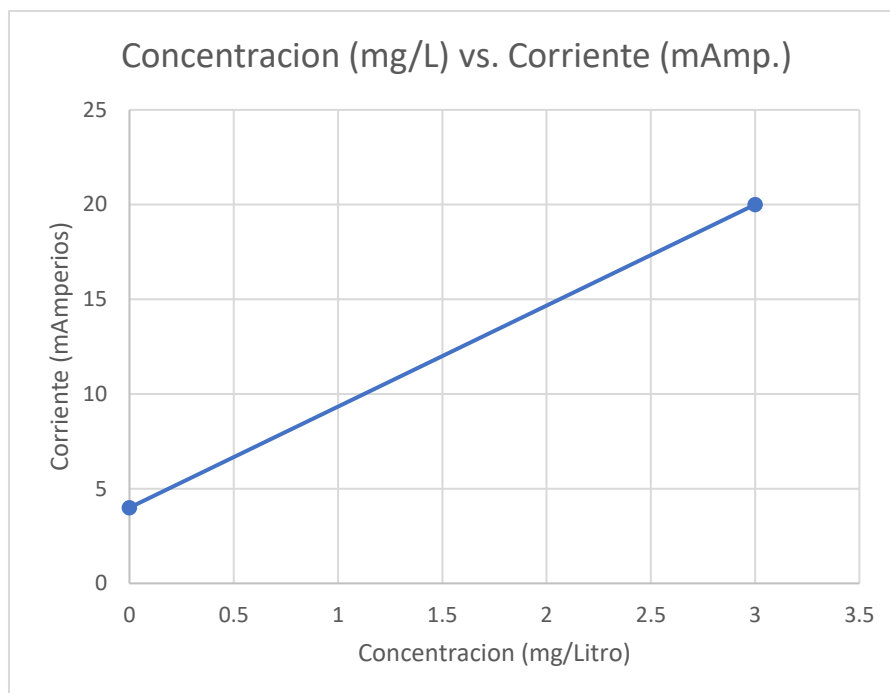
*Muestra de agua con intensidad de coloración correspondiente a 2 mg/L*



El sensor de cloro cl420MB-ES es un sensor de instrumentación industrial, por lo que se comprobó que para valores de concentración de 0 – 3 mg/L genera una corriente de salida proporcional de 4 – 20 mA, correspondiendo una salida de 4 mA para un valor de 0 mg/L y 20 mA para un valor de 3 mg/L, tal como se muestra en la siguiente figura 64.

**Figura 64**

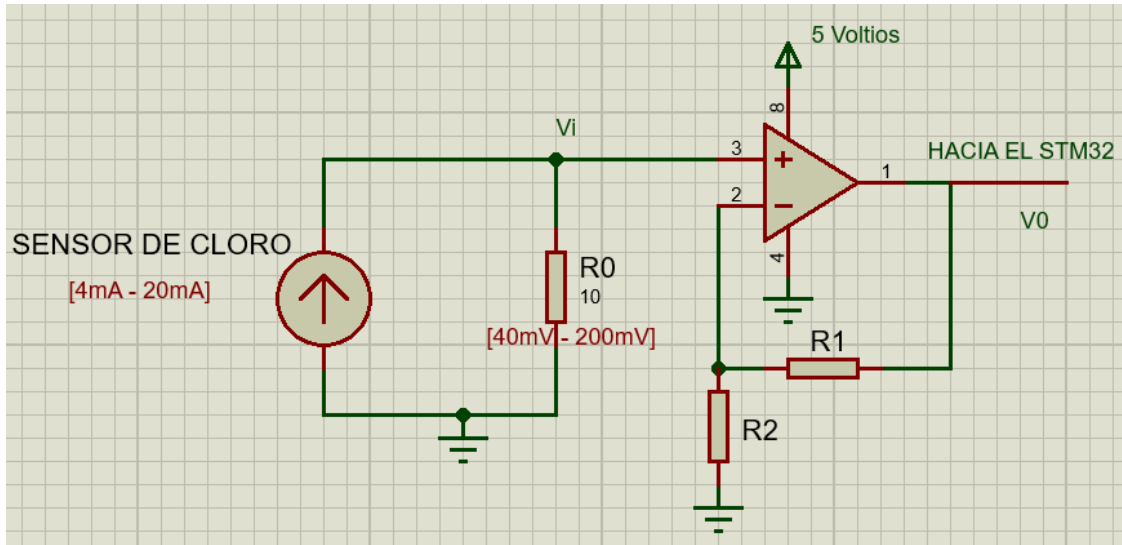
*Concentración de cloro residual en función de la corriente de salida*



Teniendo en cuenta el intervalo de valores de corriente en la salida del sensor, será necesario utilizar una resistencia de bajo valor para generar un voltaje que pueda ser medido por el microcontrolador, la siguiente figura 65 muestra la simulación del sensor de cloro residual como una fuente de corriente DC y su acondicionamiento en Proteus.

**Figura 65**

*Modelado del sensor de cloro residual y su acondicionamiento*



Debido al bajo nivel de tensión entregado por el sensor de cloro residual será necesario una etapa de amplificación de mediante un OPAM configurado en modo no inversor.

$$v_0 = v_i \left( 1 + \frac{R_1}{R_2} \right); \quad v_0 = k * v_i$$

$$\text{Donde, el valor de } k = \left( 1 + \frac{R_1}{R_2} \right)$$

El microcontrolador detecta un máximo valor de tensión de 3.3 volt, por lo tanto, será necesario ajustar el valor de escalamiento  $k$  de tal forma que, para un valor de  $v_i$  igual a 200 mV, el valor de  $v_0$  sea igual a 3.3 volt.

$$k = \frac{3.3 \text{ Volt}}{200 \text{ mVolt}}$$

$$k = 16.5$$

$$\frac{R_1}{R_2} = 15.5$$

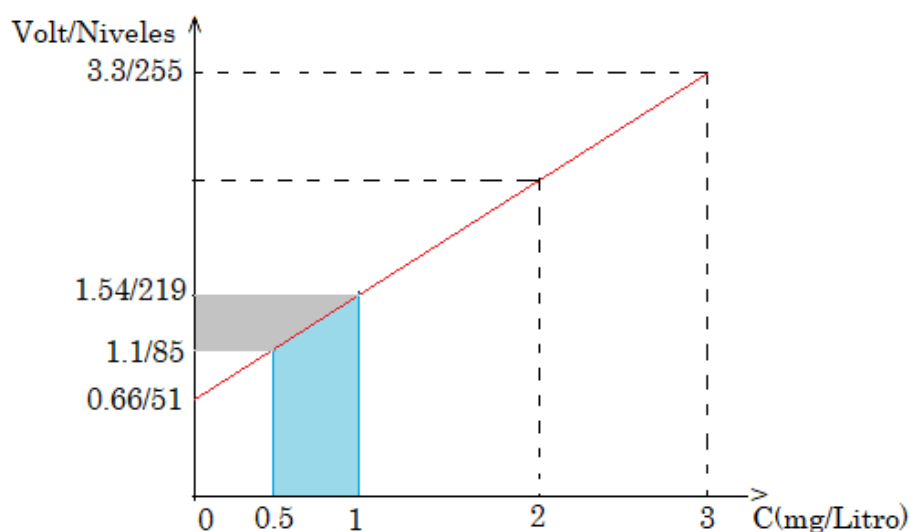
$$R_1 = 15.5 \text{ Kohm}, R_2 = 1 \text{ Kohm}$$

Por lo tanto, teniendo en cuenta una resolución de 8 bits del ADC, tendremos una  $2^8 = 256$  niveles de codificación, con una resolución de  $r = \frac{3.3 \text{ Volt}}{256 \text{ niveles}} = 12.89 \text{ mVolt/nivel}$ .

En la siguiente figura 66 se muestra la gráfica de correspondencia entre la concentración de cloro residual vs. voltaje/niveles de codificación.

**Figura 66**

*Concentración (mg/Litro) en función del voltaje de salida*



### 3.1.3.2. Configuración del ADC del STM32

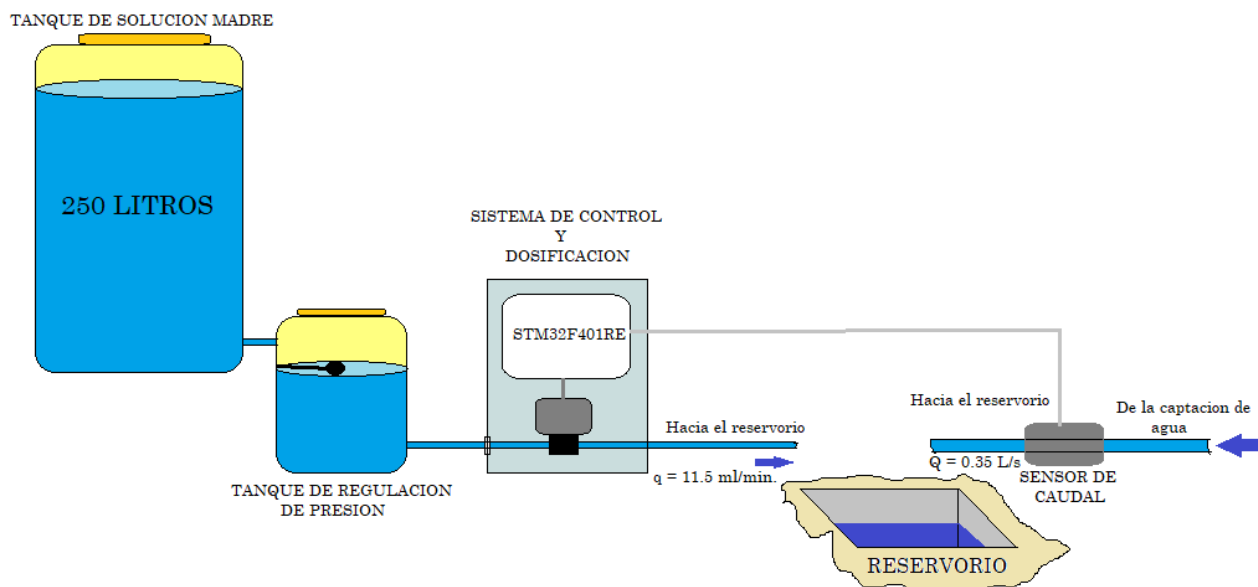
La configuración del ADC del STM32 para la medición del sensor de cloro residual será la misma realizada para el sensor de turbidez, solo se tendrá en cuenta el cambio del Pin donde está conectado el sensor de cloro, para este parámetro le corresponde el PinB1 el cual está asociado al canal analógico 9 (chx =9), por lo tanto, para realizar la medición será necesario establecer los bits SQ1[3:0] del registro ADC1\_SQR3 en un valor de 9.

### 3.1.4. Diseño del Regulador de Suministro de la Solución Hipoclorosa

El sistema de control ON/OFF estará accionado por el microcontrolador STM32, el cual en base a las mediciones del nivel de turbidez y el caudal, abrirá o cerrará el paso de la solución hipoclorosa mediante un mecanismo de estrangulamiento, este mecanismo consiste en un motor paso a paso adaptado a la manguera de goteo que sale del tanque regulador de presión, tal como se muestra en la figura 67.

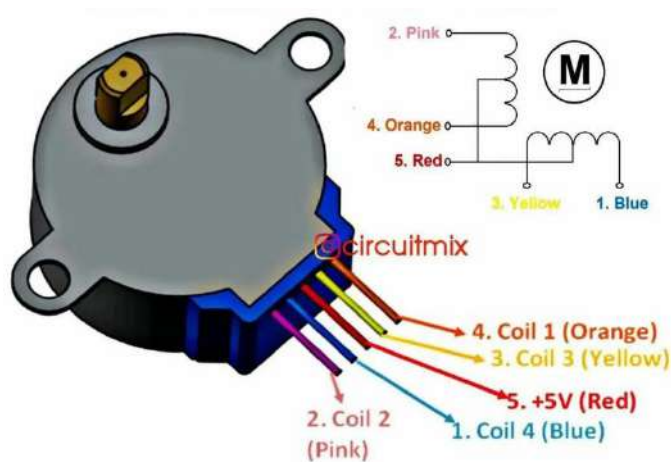
**Figura 67**

*Sistema de control On/Off para el suministro de cloro*

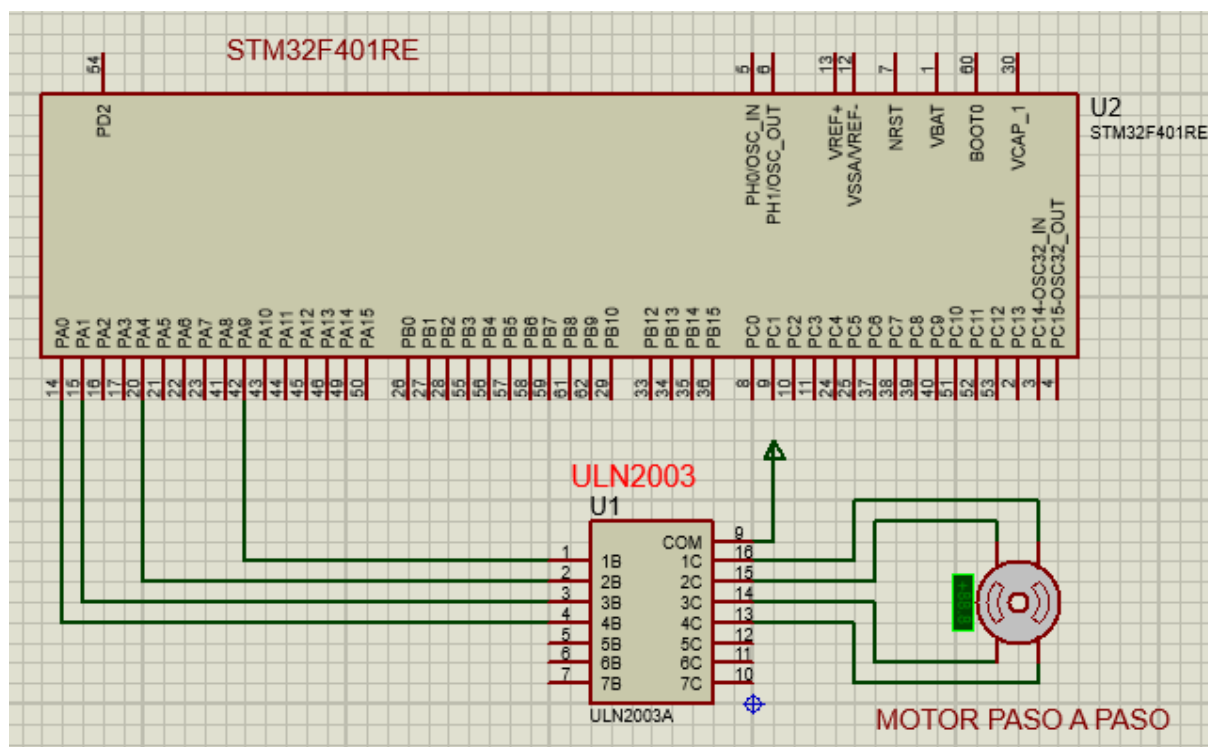


Como se puede apreciar en la figura 67 anterior, el suministro de cloro dependerá de la posición del eje del motor paso a paso, para la implementación del proyecto se utilizará un motor paso a paso 28BYJ-4S el cual se muestra en la siguiente figura 68.

El motor 28BYJ-4S es un motor a pasos de 4 bobinas, tiene un alto consumo de corriente, para adaptarlo al microcontrolador se utilizará una etapa de potencia mediante el integrado lm57, la simulación en Proteus se muestra en la figura 69; también será necesario definir la secuencia de pasos hacia la izquierda y la derecha según se desee abrir o cerrar el paso de la solución hipoclorosa, en la tabla 10 muestra la secuencia de los estados de los bits para cada giro.

**Figura 68***Motor paso a paso*

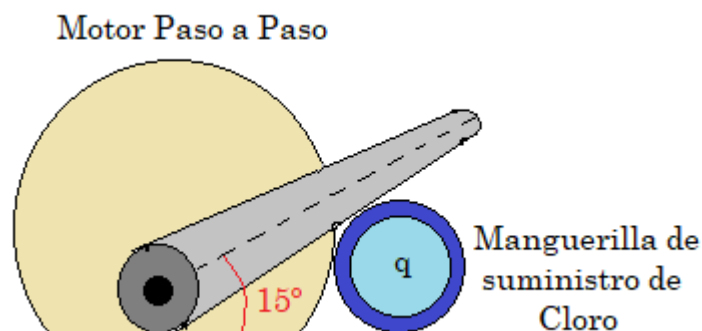
Nota. Adaptado de *Motor paso a paso*, Circuitmix (circuitmix.com)

**Figura 69***Simulación en Proteus*

**Tabla 10***Secuencia de bit para giro del motor a pasos*

| <i>Giro derecha</i>   |           |           |           |
|-----------------------|-----------|-----------|-----------|
| <i>I1</i>             | <i>I2</i> | <i>I3</i> | <i>I4</i> |
| <i>1</i>              | <i>1</i>  | <i>0</i>  | <i>0</i>  |
| <i>0</i>              | <i>1</i>  | <i>1</i>  | <i>0</i>  |
| <i>0</i>              | <i>0</i>  | <i>1</i>  | <i>1</i>  |
| <i>1</i>              | <i>0</i>  | <i>0</i>  | <i>1</i>  |
| <i>Giro izquierda</i> |           |           |           |
| <i>I1</i>             | <i>I2</i> | <i>I3</i> | <i>I4</i> |
| <i>0</i>              | <i>0</i>  | <i>1</i>  | <i>1</i>  |
| <i>0</i>              | <i>1</i>  | <i>1</i>  | <i>0</i>  |
| <i>1</i>              | <i>1</i>  | <i>0</i>  | <i>0</i>  |
| <i>1</i>              | <i>0</i>  | <i>0</i>  | <i>1</i>  |

En donde, este motor de pasos 28BYJ-4S requiere 512 pasos para dar una vuelta completa, por lo tanto, será necesario la multiplicación de un factor de conversión a grados sexagesimales tal como se ve en la demostración de la ecuación, este factor deberá ser considerado al momento de la implementación en código como se ve en el anexo B; la figura 70 muestra una vista transversal de la sección de estrangulamiento, se puede apreciar fácilmente que se requiere aproximadamente un desplazamiento angular de 15° para abrir el paso del suministro de cloro.

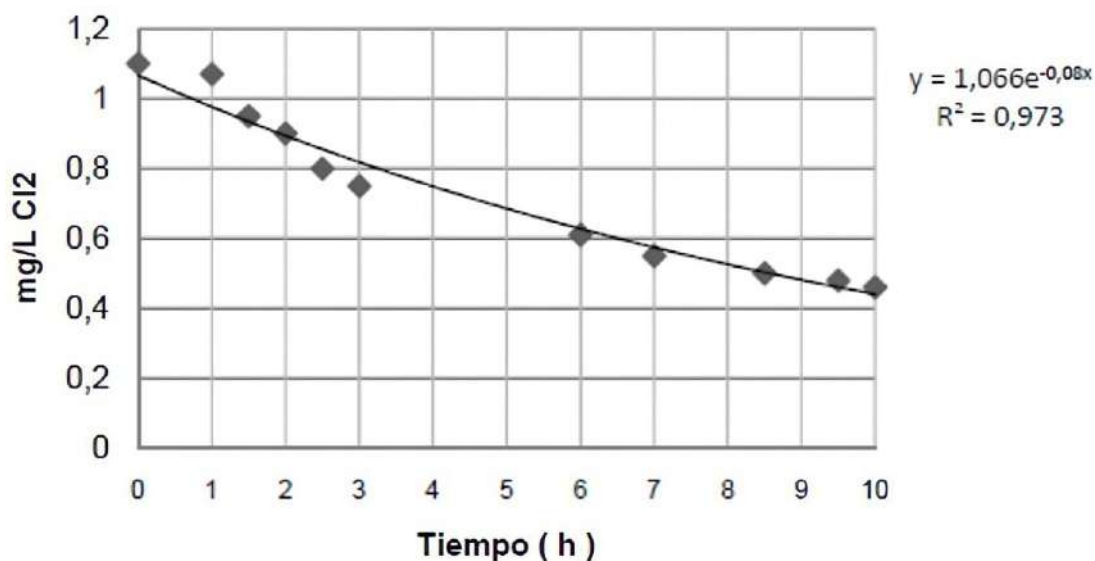
**Figura 70***Sección de estrangulamiento*

### **3.1.5. Diseño e Implementación del Tiempo de Toma de Muestra**

#### **3.1.5.1. Determinación del Tiempo de Extinción del Cloro Residual**

Para determinar el tiempo adecuado de toma de cada muestra de agua para la medición de la turbidez y concentración de cloro residual será necesario la medición de la velocidad de cambio de la concentración de cloro en el agua potable con respecto al tiempo, para lo cual, primero se verificarán estudios previos con respecto a este tema y seguidamente se realizará una medición del cambio de la concentración en el mismo sistema de agua potable donde se realizará la implementación y la prueba del sistema de control.

En el proyecto de investigación “Modelo Experimental para Evaluar Cloro Residual en el Agua de las Redes de Distribución del Acueducto Metropolitano de Bucaramanga S.A. ESP” (Barragán Cardozo, 2010) Universidad Industrial de Santander. El autor calcula modelos experimentales para la determinación de la concentración de cloro residual en función del tiempo en condiciones estáticas. Este modelo le permite finalmente graficar el decaimiento del cloro residual según los tiempos de residencia del agua en la red de agua estática.

**Figura 71***Decaimiento del cloro en función del tiempo*

*Nota.* Adaptado de *Modelamiento y determinación de parámetros en una planta de agua potable*, B. Cardozo, 2014

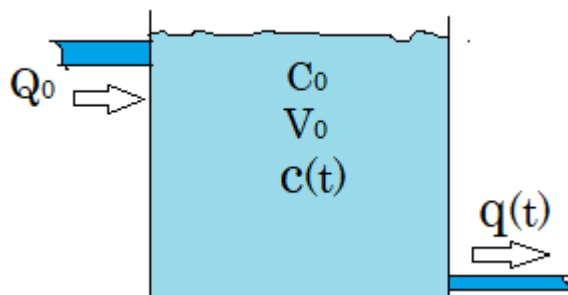
Según la gráfica anterior se puede observar que en condiciones estáticas el decaimiento de cloro residual es muy lento y para intervalos de tiempo de una hora la concentración permanece aproximadamente constante mientras los factores externos permanezcan constantes.

En la práctica en un sistema de agua potable el caudal de consumo de agua no es constante, siendo casi nulo a avanzadas horas de la noche y presentando picos máximos de consumo a determinadas horas del día.

Por lo tanto, para determinar la velocidad de cambio del cloro residual en función del tiempo en condiciones reales de ingreso y consumo de agua del reservorio se realizó una medición constante y periódica de la concentración de cloro residual en el reservorio cada 5 minutos, iniciando la medición cuando la concentración de cloro era de 1 mg/L y seguidamente cortando el suministro constante de la solución hipoclorosa, tal como se muestra en la siguiente figura 72.

**Figura 72**

*Modelo de la concentración de cloro residual*



La medición de la concentración de cloro residual en función del tiempo se realizó en las primeras horas de la mañana debido a la alta demanda de agua potable a esas horas del día; la siguiente tabla muestra los valores obtenidos.

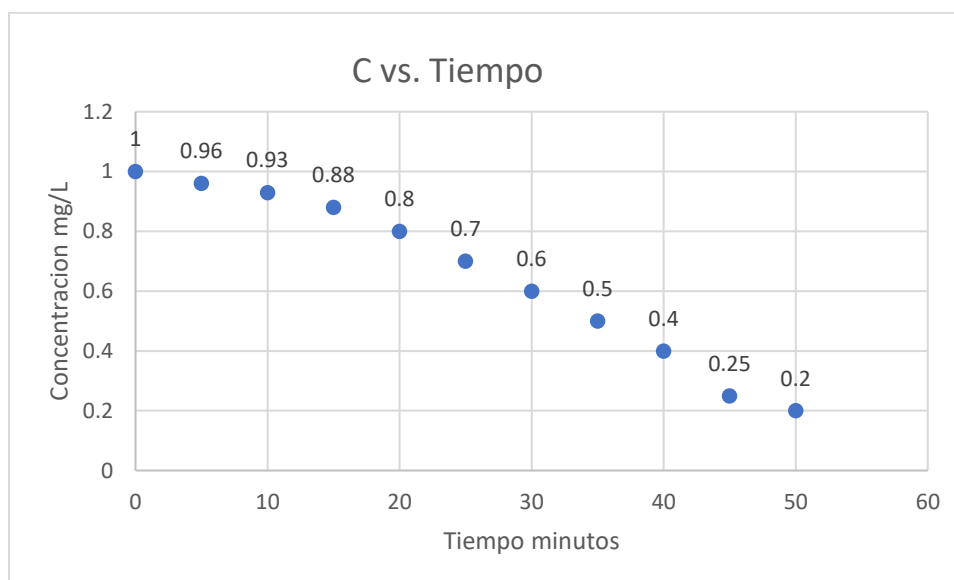
**Tabla 11**

*Concentración del cloro residual en función del tiempo*

|                     |   |   |      |      |     |     |     |     |     |      |     |
|---------------------|---|---|------|------|-----|-----|-----|-----|-----|------|-----|
| $C(\text{ppm})$     | 1 | 1 | 0.96 | 0.95 | 0.8 | 0.7 | 0.6 | 0.5 | 0.3 | 0.25 | 0.2 |
| $T(\text{minutos})$ | 0 | 5 | 10   | 15   | 20  | 25  | 30  | 35  | 40  | 45   | 50  |

**Figura 73**

*Concentración en función del tiempo*



Teniendo en cuenta el rango recomendado de concentración del cloro residual [0.5 – 1 mg/L], en la gráfica anterior se puede notar el tiempo que demora en descender de 1 mg/L a 0.5

mg/L es de aproximadamente 35 minutos, por lo tanto, la medición de los parámetros del agua deberá medirse aproximadamente cada media hora para detectar cambios significativos en la concentración.

### 3.1.5.2. Implementación del Periférico RTC

Como se observó anteriormente la velocidad de cambio de concentración de cloro residual es relativamente lenta, por lo tanto, será necesario implementar en el microcontrolador un sistema de medición del tiempo, y de ese modo, el microcontrolador obtenga la muestra de agua cada 30 minutos y realice la respectiva medición de los parámetros de calidad de agua, para lo cual, será de utilidad la correcta configuración del periférico RTC y de ese modo el microcontrolador tenga en cuenta el momento adecuado en el cual se debe realizar la medición.

Para su implementación, según el manual del fabricante, primero será necesario escoger la fuente de reloj RTCCLK, en este caso se utiliza la fuente HSE el cual, se encuentra configurado a 8 Mhz, el valor recomendado del HSE\_RTC debe ser máximo 4 Mhz, por lo tanto, será necesario seleccionar un factor de escalamiento para obtener un valor de HSE\_RTC igual a 1 Mhz, los bits RTCPRE del registro RCC\_CFGR permiten dividir la frecuencia HSE a un valor deseado, en la figura 74 se muestra el registro RCC\_CFGR donde los bits RTCPRE se configuro en 0b01000 (8 decimal) para obtener una división de la frecuencia del sistema en un factor de 8.

**Figura 74**

*Registro RCC\_CFGR*

|            |    |               |            |    |               |          |    |            |      |    |             |      |      |     |     |
|------------|----|---------------|------------|----|---------------|----------|----|------------|------|----|-------------|------|------|-----|-----|
| 31         | 30 | 29            | 28         | 27 | 26            | 25       | 24 | 23         | 22   | 21 | 20          | 19   | 18   | 17  | 16  |
| MCO2       |    | MCO2 PRE[2:0] |            |    | MCO1 PRE[2:0] |          |    | I2SSC<br>R | MCO1 |    | RTCPRE[4:0] |      |      |     |     |
| rw         |    | rw            | rw         | rw | rw            | rw       | rw | rw         | rw   |    | 0           | 1    | 0    | 0   | 0   |
| 15         | 14 | 13            | 12         | 11 | 10            | 9        | 8  | 7          | 6    | 5  | 4           | 3    | 2    | 1   | 0   |
| PPRE2[2:0] |    |               | PPRE1[2:0] |    |               | Reserved |    | HPRE[3:0]  |      |    |             | SWS1 | SWS0 | SW1 | SW0 |
| rw         | rw | rw            | rw         | rw | rw            |          |    | rw         | rw   | rw | rw          | r    | r    | rw  | rw  |

*Nota.* Adaptado de *RM0368 Reference manual* (p. 80), STMicroelectronics, 2018

Teniendo el valor de RTCCLK (1Mhz), también se tendrá que dividir entre factores de escalamiento definidos por los bits PREDIV\_A y PREDIV\_S del registro RTC\_PRER para obtener una frecuencia de salida de 1 hz, los valores calculados son: PREDIV\_A = 0x63 =99 y PREDIV\_S = 0x270F = 9999, estos valores serán utilizados en la siguiente ecuación:

$$f_{CK\_SPRE} = \frac{f_{RTCCLK}}{(PREDIV\_S + 1) \times (PREDIV\_A + 1)}$$

Por lo tanto, una vez obtenido el valor de frecuencia de 1hz, los diferentes bits de los registros RTC\_DR y RTC\_TR almacenaran los valores de la fecha y hora respectivamente.

La siguiente figura 75 muestra el registro RTC\_DR y la figura 76 muestra el registro RTC\_TR.

**Figura 75**

*Registro RTC\_DR*

|          |    |    |    |         |    |    |    |          |    |         |    |         |    |    |    |
|----------|----|----|----|---------|----|----|----|----------|----|---------|----|---------|----|----|----|
| 31       | 30 | 29 | 28 | 27      | 26 | 25 | 24 | 23       | 22 | 21      | 20 | 19      | 18 | 17 | 16 |
| Reserved |    |    |    |         |    |    |    | YT[3:0]  |    |         |    | YU[3:0] |    |    |    |
|          |    |    |    |         |    |    |    | rw       | rw | rw      | rw | rw      | rw | rw | rw |
| 15       | 14 | 13 | 12 | 11      | 10 | 9  | 8  | 7        | 6  | 5       | 4  | 3       | 2  | 1  | 0  |
| WDU[2:0] |    |    | MT | MU[3:0] |    |    |    | Reserved |    | DT[1:0] |    | DU[3:0] |    |    |    |
| rw       | rw | rw | rw | rw      | rw | rw | rw |          |    | rw      | rw | rw      | rw | rw | rw |

*Nota.* Adaptado de *RM0368 Reference manual* (p. 81), STMicroelectronics, 2018

**Figura 76**

*Registro RTC\_TR*

|          |          |    |    |          |    |    |    |      |         |         |    |         |    |    |    |
|----------|----------|----|----|----------|----|----|----|------|---------|---------|----|---------|----|----|----|
| 31       | 30       | 29 | 28 | 27       | 26 | 25 | 24 | 23   | 22      | 21      | 20 | 19      | 18 | 17 | 16 |
| Reserved |          |    |    |          |    |    |    | PM   |         | HT[1:0] |    | HU[3:0] |    |    |    |
|          |          |    |    |          |    |    |    | rw   | rw      | rw      | rw | rw      | rw | rw | rw |
| 15       | 14       | 13 | 12 | 11       | 10 | 9  | 8  | 7    | 6       | 5       | 4  | 3       | 2  | 1  | 0  |
| Res.     | MNT[2:0] |    |    | MNU[3:0] |    |    |    | Res. | ST[2:0] |         |    | SU[3:0] |    |    |    |
|          | rw       | rw | rw | rw       | rw | rw | rw |      | rw      | rw      | rw | rw      | rw | rw | rw |

*Nota.* Adaptado de *RM0368 Reference manual* (p. 82), STMicroelectronics, 2018

### 3.2. Diseño del Sistema de Comunicación

Como se estudió anteriormente existen tecnologías de comunicación inalámbrica licenciadas y no licenciadas, para la implementación del presente proyecto se utilizará una tecnología de comunicación no licenciada; dentro de las tecnologías no licenciadas se destaca las tecnologías de LoRa y SigFox, ambas funcionan muy bien para aplicaciones de IoT en donde el ancho de banda requerido es muy pequeño, pero de acuerdo al modelo de negocio de SigFox esta tecnología no es conveniente para la implementación de este proyecto de tesis debido a que su operación y funcionamiento se basa en los servicios de un operador de telecomunicaciones de SigFox que brinde cobertura en el área donde se implementara el proyecto, en la siguiente figura 77 se muestra el despliegue de SigFox sobre el territorio peruano.

**Figura 77**

*Cobertura de SigFox en el Perú*



*Nota. Adaptado de Cobertura SigFox, SigFox (SigFox.com/Coverage)*

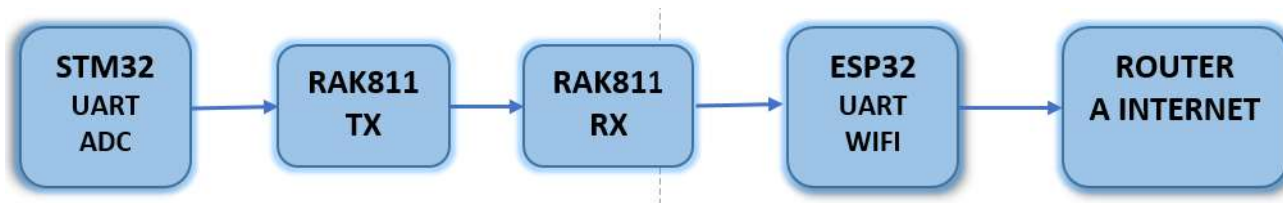
Como se puede notar en la figura 77, la cobertura de SigFox es muy amplia en la costa central del Perú y siendo casi escasa en la parte sur, en la siguiente figura 78 se muestra la cobertura de SigFox en la región del Cusco.

**Figura 78***Cobertura de SigFox en la región de Cusco*

*Nota.* Adaptado de *Cobertura SigFox*, SigFox ([SigFox.com/Coverage](http://SigFox.com/Coverage))

Como se puede observar en la gráfica anterior, la cobertura de la red SigFox en la provincia de Cuzco es nula.

A diferencia de la tecnología de SigFox, LoRa no requiere los servicios de cobertura en una determinada zona por parte de un proveedor de servicios de telecomunicaciones, es posible desplegar LoRa utilizando solo un nodo remoto y un gateway con acceso a internet. Por estas razones el sistema de comunicación utilizado para realizar el monitoreo de los parámetros de calidad de agua estará implementado sobre la tecnología LoRa, para la comunicación del nodo remoto ubicado sobre la caseta de cloración y un nodo central con acceso a Internet mediante un microcontrolador con soporte a WiFi, en microcontrolador accederá a la red de Internet mediante una puerta de enlace o gateway, la figura 79 muestra el diagrama de bloques del sistema de comunicación IoT.

**Figura 79***Implementación del sistema de comunicación IoT*

LoRaWAN define las bandas no licenciadas de acuerdo con el país donde se desplegará esta tecnología, en el Perú se le asigna la banda AU915-928, la cual permite una operación desde los 915 MHz hasta los 928 MHz, en el Perú esta es una banda libre.

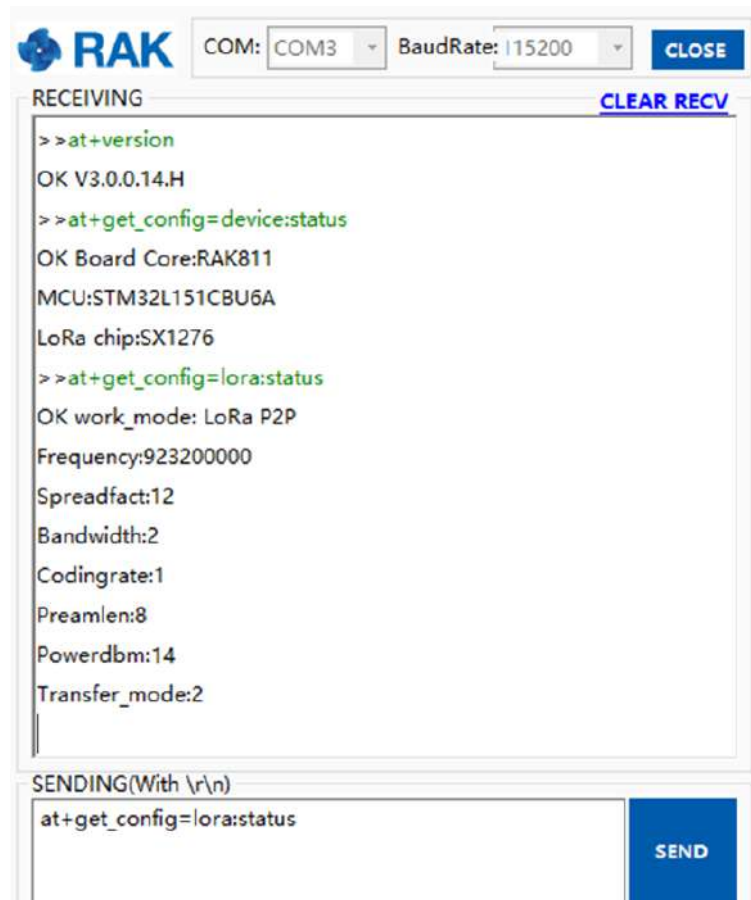
### 3.2.1. Diseño del Sistema de Transmisión LoRa

#### 3.2.1.1. Implementación del Módulo RAK811

Para la transmisión de la información obtenida por el STM32 mediante los sensores de turbidez y cloro residual, se utilizará una comunicación serial hacia un módulo LoRa RAK811 el cual, utiliza comandos AT para su configuración y funcionamiento, la siguiente tabla se muestra los principales comandos at utilizados por los dispositivos RAK.

| Comando                                                      | Operación        | Respuesta                       |
|--------------------------------------------------------------|------------------|---------------------------------|
| <i>at+versión</i>                                            | <i>Lectura</i>   | <i>OK &lt;Vesion number&gt;</i> |
| <i>at+set_config=device:sleep</i>                            | <i>Escritura</i> | <i>OK &lt;Status&gt;</i>        |
| <i>at+get_config=device:status</i>                           | <i>Lectura</i>   | <i>OK &lt;Information&gt;</i>   |
| <i>at+set_config=device:uart::&lt;index&gt;:&lt;rate&gt;</i> | <i>Escritura</i> | <i>OK</i>                       |
| <i>at+send=uart::&lt;data&gt;:&lt;rate&gt;</i>               | <i>Escritura</i> | <i>OK</i>                       |

Para realizar la configuración del dispositivo RAK será necesario el uso del software Serial Port Tool, mediante esta aplicación se realizará la configuración de los parámetros como velocidad de transmisión UART a 115200 baudios, frecuencia de transmisión a 9.2 Mhz, modo de operación 2 (transmisión), región de operación, dirección del dispositivo, contraseña del dispositivo, intervalo de envío de datos, etc. en la figura 80 se muestra la interfaz de programación del software con un resumen de los parámetros configurados.

**Figura 80***Interfaz RAK Serial Port Tool*

### 3.2.1.2. Implementación del STM32 para la Transmisión de Información

En el microcontrolador STM32F401RE se tiene un periférico USART, el cual, mediante la configuración adecuada de sus parámetros mediante el acceso a sus registros de programación, se utilizará en modo UART para su comunicación con el módulo RAK811, la velocidad de transmisión serial será de 115200 baudios, el tamaño de palabra será de 8 bits mediante la configuración del bit M en el registro USART\_CR1, sin paridad, un bit de stop y una tasa de sobre muestreo de 16.

Por lo tanto, según el manual de fabricante tenemos que la velocidad de tx/rx para una tasa de sobre muestreo de 16 está dado por:

$$Tx/Rx \text{ baud} = \frac{f_{CK}}{16 \times USARTDIV}$$

En donde, el valor de  $f_{CK}$  es la frecuencia de operación del microcontrolador configurado en 16 MHz y el USARTDIV es el valor almacenado en el registro USART\_BRR, tal como se muestra en la siguiente figura 81.

**Figura 81**

*Registro USART\_BRR*

|                    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |
|--------------------|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|
| 31                 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19                | 18 | 17 | 16 |
| Reserved           |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |
| 15                 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3                 | 2  | 1  | 0  |
| DIV_Mantissa[11:0] |    |    |    |    |    |    |    |    |    |    |    | DIV_Fraction[3:0] |    |    |    |
| rw                 | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw                | rw | rw | rw |

*Nota.* Adaptado de *RM0368 Reference manual* (p. 101), STMicroelectronics, 2018

En donde DIV\_Mantissa[11:0] corresponde a la parte entera y DIV\_Fraction[3:0] a la parte fraccionaria del valor de USARTDIV, entonces, para un valor de Tx/Rx de 115200 baudios , tenemos lo siguiente:

$$USARTDIV = 8.68 \text{ (decimal)}$$

Por lo tanto, haciendo las conversiones tenemos los valores de:

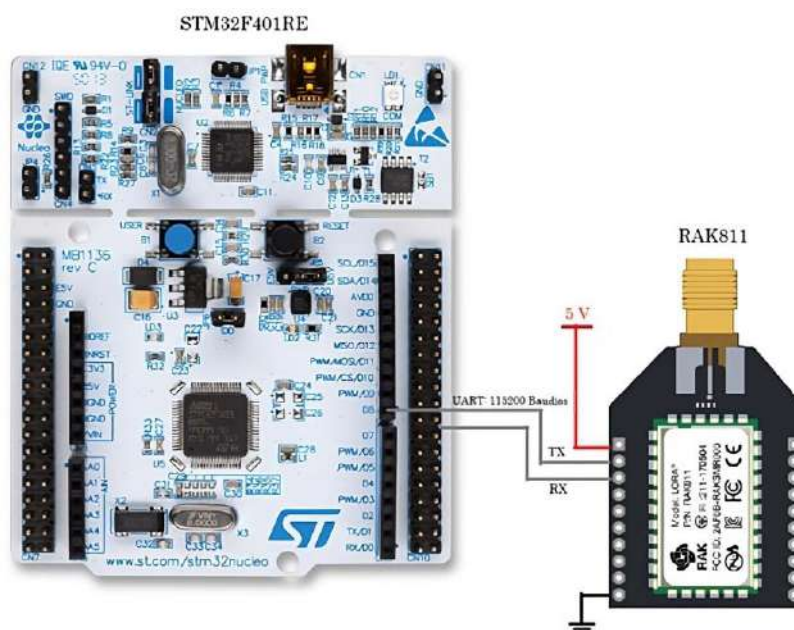
$$DIV\_Mantissa[11:0] = 0x8 \text{ y } DIV\_Fraction[3:0] = 0x1$$

Los valores calculados anteriormente deberán ser implementados sobre la configuración del microcontrolador, en el anexo b se tiene una explicación más detallada de su implementación en lenguaje de programación C.

La siguiente figura 82 se muestra el diagrama de interconexión entre el STM32 y el dispositivo LoRa RAK811.

**Figura 82**

*Comunicación STM32 – RAK811 en modo transmisor.*



En la siguiente figura 83 se muestra la implementación del módulo RAK811 configurado en modo transmisor con su respectiva caja de protección.

**Figura 83**

*Implementación RAK811 en modo transmisión*

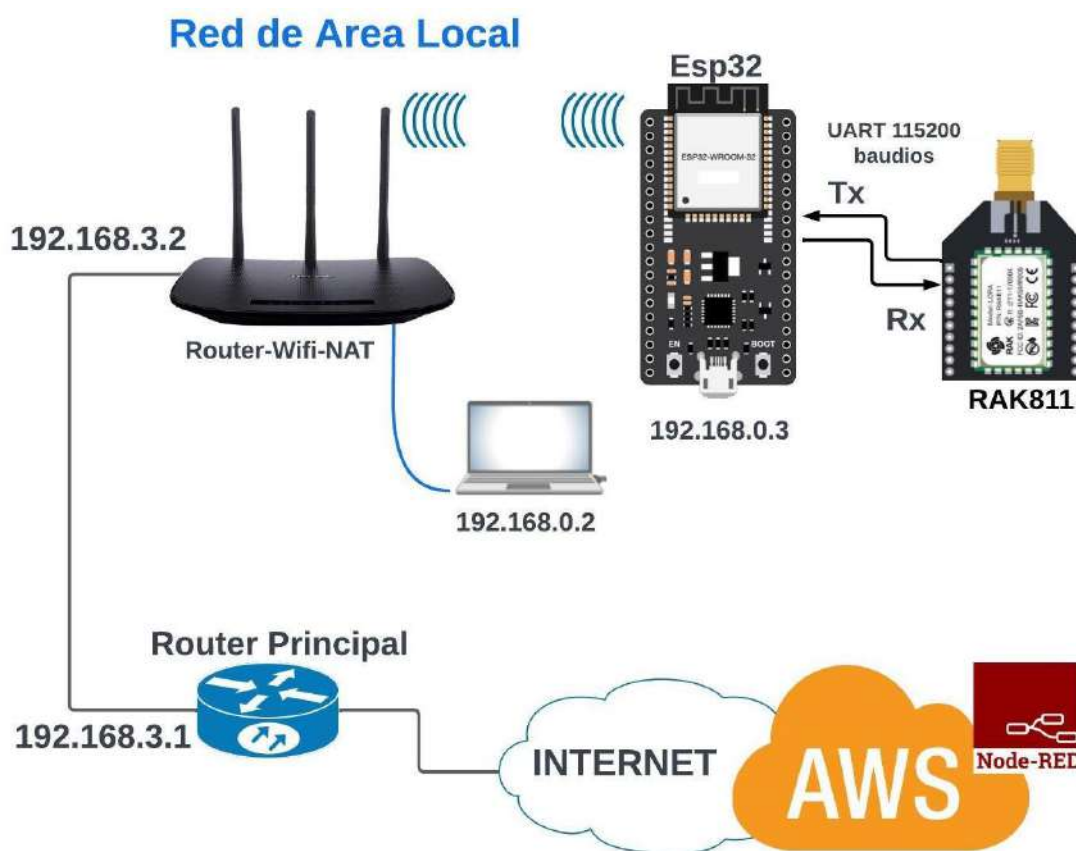


### 3.2.2. Diseño del Sistema Receptor LoRa

En la etapa de recepción del sistema de comunicación, se utilizará otro módulo RAK811 configurado en modo receptor, este a su vez, tendrá una comunicación cableada mediante UART con el microcontrolador ESP32, tal como se muestra en la figura 94, el microcontrolador ESP32 tiene un periférico de WiFi mediante el cual, tendrá acceso a Internet por medio de la puerta de enlace ROUTER-WiFi-NAT con dirección IP 192.168.3.1.

**Figura 94**

*Comunicación RAK811 en modo receptor – ESP32 – Internet*



El microcontrolador ESP32 accederá a los valores medidos por los sensores por medio de la comunicación UART a una velocidad tx/rx de 115200 baudios, estos valores serán enviados a través de Internet a una plataforma en Node-RED ubicado en un servidor remoto de Raspberry, en el anexo x se detalla la implementación en código de programación del ESP32 utilizando el IDE de Arduino.

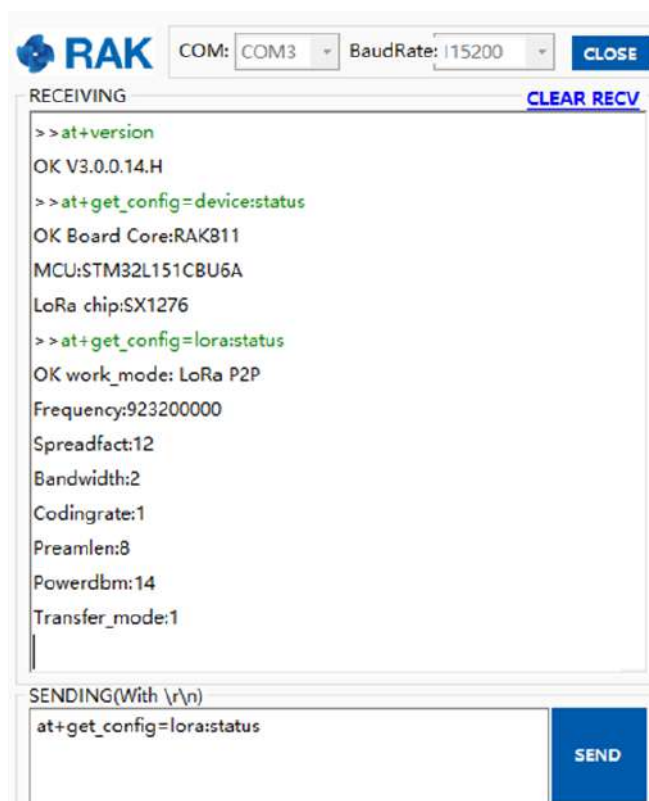
Con respecto al módulo RAK811, será configurado en modo de trabajo punto a punto, modo de operación receptor, a una frecuencia de 923.2 Mhz, a una potencia máxima de transmisión de 20 dbm y un Spreadfact de 12.

En la figura 85, se muestra un resumen de configuración de los parámetros del RAK811 en modo receptor, sobre la plataforma Serial Port Tool.

En la siguiente figura 85, tenemos la implementación en hardware del módulo receptor RAK811 – ESP32 sobre protoboard en estado operativo con su respectiva caja de protección.

### **Figura 85**

*Interfaz RAK811 Serial Port Tool en modo receptor*



**Figura 86**

*Implementación RAK811 - ESP32 en modo recepción*



Y finalmente, en la figura 87, se muestra la implementación de los módulos LoRa de transmisión y recepción operando correctamente.

**Figura 87**

*Implementación módulos LoRa transmisor – receptor*

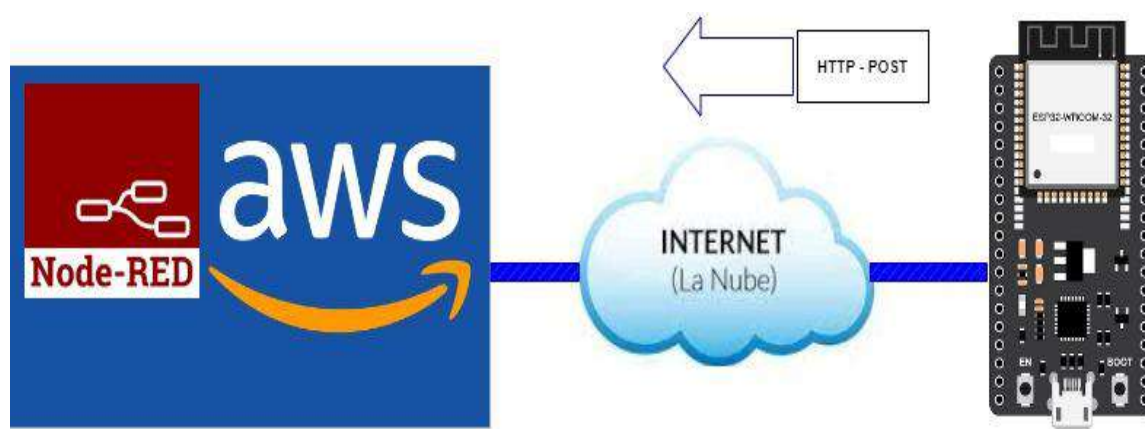


### 3.2.3. Diseño de la Plataforma Node-RED Sobre AWS

El software de desarrollo de recolección de información Node-RED será implementado sobre una plataforma de AWS alojado en internet, el microcontrolador ESP32 enviará los datos recibidos de los sensores a la plataforma remota mediante el método POST del protocolo HTTP, la siguiente figura 89 muestra el esquema de funcionamiento.

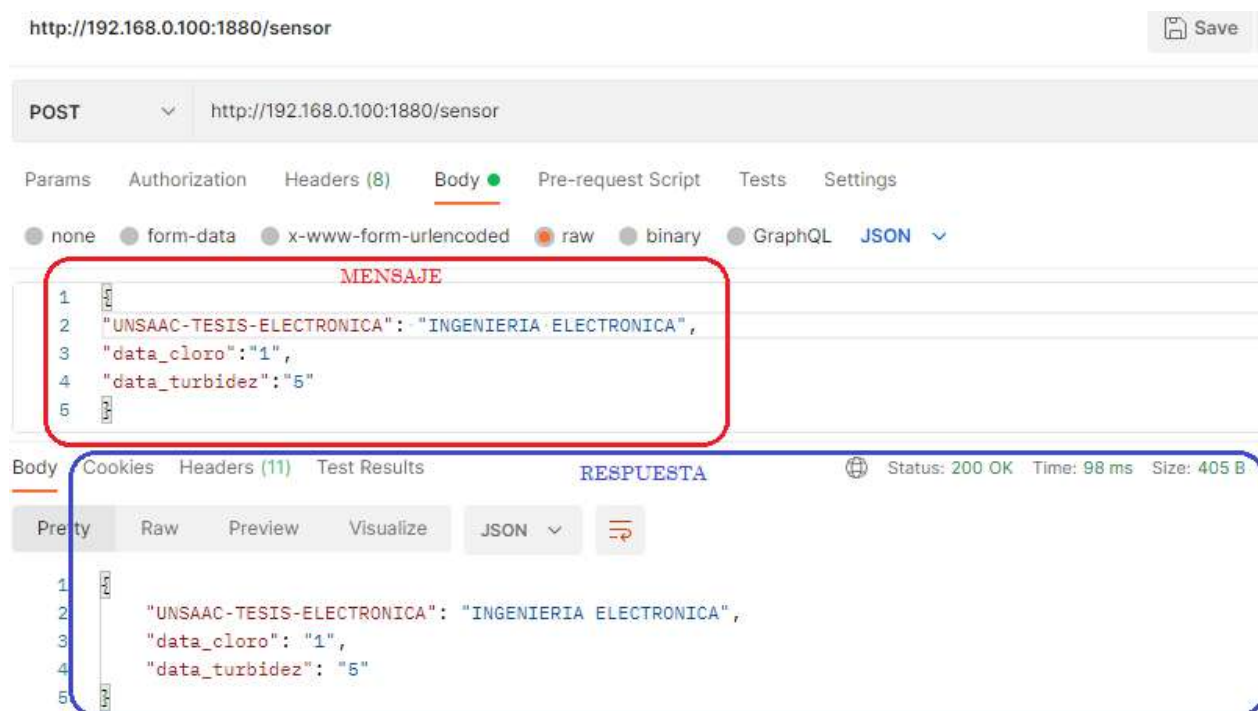
**Figura 89**

*Comunicación Node-RED – ESP32*

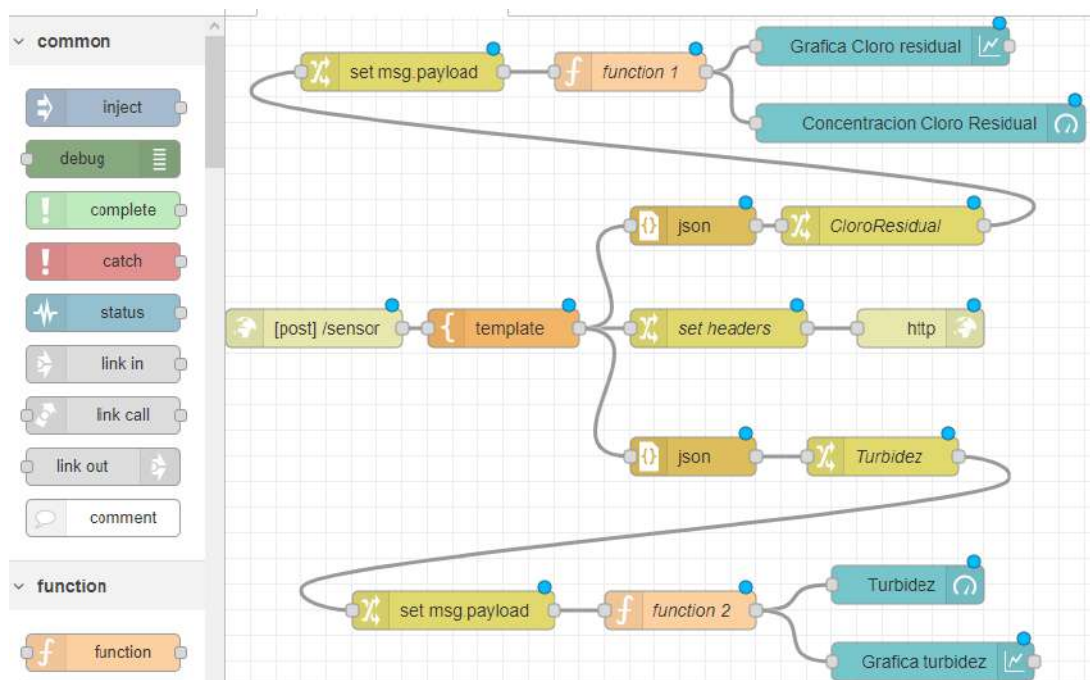


#### 3.2.3.1. Diseño del Entorno de Visualización en Node-RED

El desarrollo de la plataforma deberá mostrar mediante una interfaz gráfica los valores de los parámetros medidos, recibidos desde los sensores de cloro residual y turbidez, para lo cual el microcontrolador construirá una estructura de datos en formato JSON, en la siguiente figura 90 se muestra la simulación del mensaje en formato JSON construido por el microcontrolador y enviado a Node-RED, el cual, se encuentra ubicado sobre un servidor en la red 192.168.0.0/24 y dirección de la capa de transporte 1880; en la parte superior se muestra la construcción del mensaje de prueba y en la parte inferior se puede observar la respuesta del servidor a una petición HTTP de tipo POST, con mensaje HTTP 200, un tiempo de respuesta de 98 ms y tamaño de palabra de 405 byte.

**Figura 90****Estructura JSON**

Seguidamente, se diseña el sistema de adquisición de datos y su visualización.

**Figura 91****Desarrollo del sistema de adquisición de información en Node-RED**

En donde se tiene:



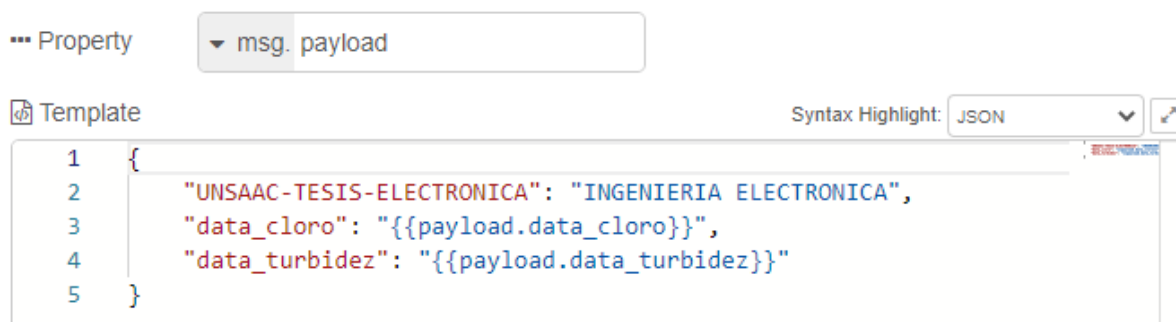
Nodo HTTP POST, crea el servicio HTTP sobre una carpeta de almacenamiento /sensor.



Nodo Template, se implementa el mensaje payload en formato JSON y las variables “payload.data\_cloro” y “payload.data\_turbidez” para el almacenamiento de los valores cloro residual y turbidez, tal como se muestra en la siguiente figura 92.

**Figura 92**

*Implementación del payload*



Nodo JSON, convierte el mensaje en formato STRING recibido la red TCP/IP a formato JSON.



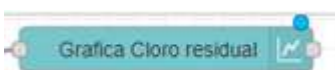
Nodo CloroResidual, extrae el valor de “payload.data\_cloro” del mensaje payload en formato JSON.



Nodo Change, convierte el formato STRING del valor de “payload.data\_cloro” a formato NUMBER.



Nodo function 1, se implementa el código de programación de ajuste de los valores recibidos, tal como se muestra en la figura 93.



Nodo Grafica Cloro Residual, implementa en el DASHBOARD una gráfica de los valores recibidos de la concentración de cloro residual.



Nodo Concentración de cloro residual, implementa una visualización grafica del valor de cloro residual.



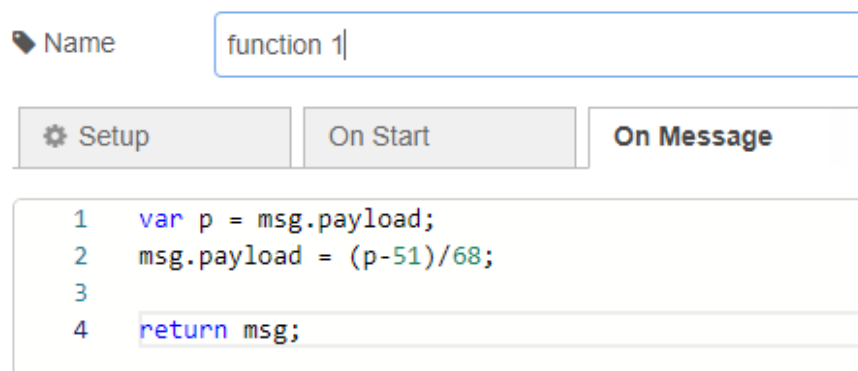
Nodo function 2, se implementa el código de programación de ajuste de los valores de turbidez recibidos.



Nodo Grafica turbidez, implementa en el DASHBOARD una gráfica de los valores recibidos del nivel de turbidez.

**Figura 93**

*Implementación en código de programación*



**Figura 94**

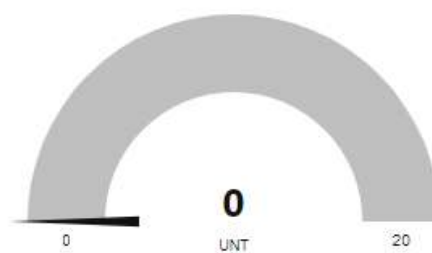
*Visualización grafica en Node-RED*

### Sistema de Monitoreo de Calidad de Agua

Concentracion Cloro Residual



Turbidez



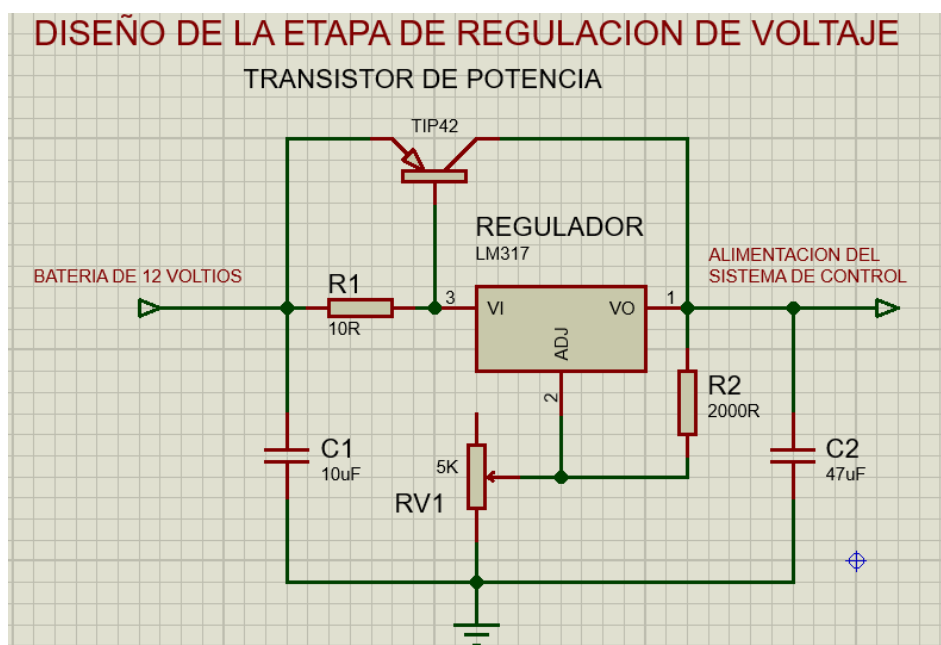
### 3.3. Diseño de la Etapa de Potencia del Sistema de Control

#### 3.3.1. Diseño de la Etapa de Regulación de Tensión

El voltaje entregado por la batería es de 12 voltios, pero la mayoría de los componentes del sistema funcionan a una tensión de 5 voltios, por lo tanto, será necesario el diseño de una etapa de regulación, el cual, debe ser capaz de entregar una corriente de 2 Amperios debido al alto consumo de potencia de las dos motobombas, en la figura 95 se muestra el diseño y su simulación en Proteus de la etapa de regulación de tensión.

**Figura 95**

*Sistema regulador del voltaje*



Según la hoja de fabricante del LM317, puede regular diferentes niveles de tensión a una corriente máxima de solo 1 Amperio, entonces será necesario la incorporación de un transistor PNP de potencia tal como se muestra en la figura 95, el transistor escogido para la implementación de este proyecto de tesis es el TIP42 el cual es un transistor PNP que soporta una corriente de colector de hasta 6 Amperios.

De la ecuación del regulador LM317 tenemos:

$$V_{OUT} = 1.25V \left( 1 + \frac{R_2}{R_1} \right) + I_{ADJ} \cdot R_2$$

Considerando  $I_{adj} = 0$ , tenemos:

$$V_{OUT} = 1.25V(1 + \frac{R_2}{R_1})$$

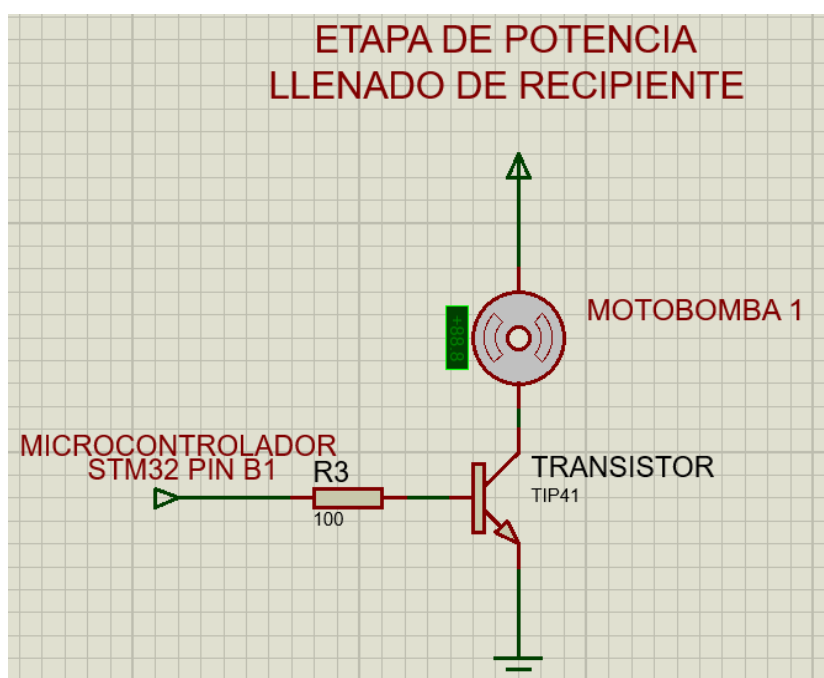
Para el diseño de este regulador se escogió una resistencia fija de 2 Kohm para R2 y una resistencia variable de 2 Kohm para R1, despejando el valor de R1 en la ecuación anterior se calcula  $R1 = 6 \text{ Kohm}$ .

### 3.3.2. Diseño de la Etapa de Potencia STM32 – Motobomba

Un nivel alto en el PIN B1 del STM32, activará la motobomba 1, la cual, será la encargada de llenar el recipiente de toma de muestra hasta un determinado volumen, pero debido al alto consumo de corriente eléctrica será necesario realizar una adaptación mediante un transistor de potencia, tal como se muestra en la siguiente figura 96.

**Figura 96**

*Etapa de potencia motobomba1*



En la figura 96 se muestra la simulación en Proteus, el transistor TIP41 NPN es un transistor de potencia con una corriente máxima de colector de 6 Amperios, en esta configuración cuando el nivel de salida sea 1 lógico (3.3 v), el transistor entrara en su zona de

saturación, teniendo un voltaje colector – emisor de aproximadamente cero y cuando la salida lógica del microcontrolador sea 0 (0 volt.), el transistor entrara en su zona de corte.

De la hoja del fabricante tenemos que el valor máximo de  $h_{FE}$  es 75, entonces podemos calcular el valor de la resistencia de base.

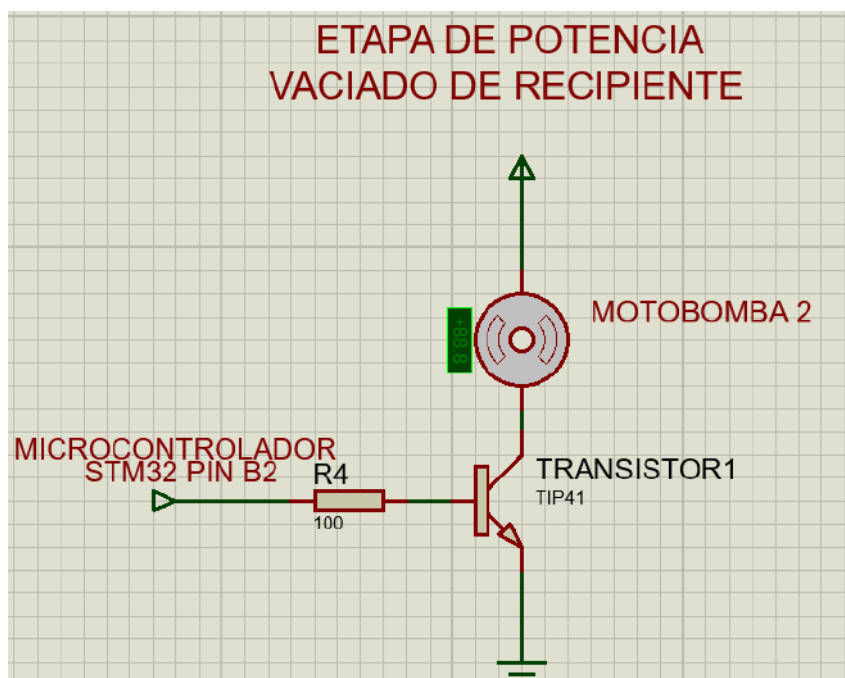
$$3.3 \text{ v} = I_b R_b + 0.7$$

$$I_b = \frac{I_c}{h_{FE}}$$

$I_{cmax} = 1 \text{ Amp}$ . Entonces  $R_b = 195 \text{ Ohm}$ . El análisis y cálculo para la motobomba 2 será el mismo, tal como se muestra en figura 97.

**Figura 97**

*Etapas de potencia motobomba2*



### 3.4. Diseño de la Etapa de Alimentación Fotovoltaica

Para el diseño de la alimentación fotovoltaica en la figura 98 se muestran las etapas del sistema mismo, primeramente, se tiene el panel solar, luego la etapa de acondicionamiento y regulación de la carga, seguidamente la batería y finalmente la etapa de potencia del sistema de control.

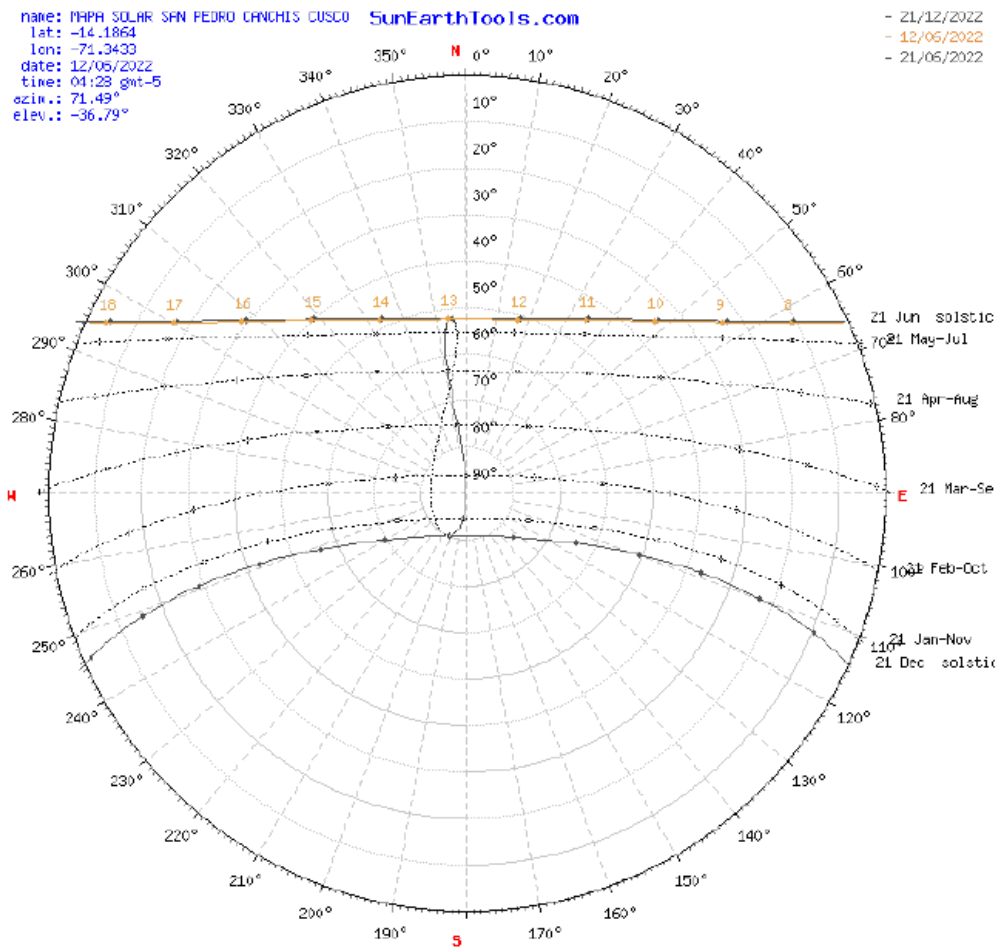
**Figura 98***Etapas de alimentación fotovoltaica***3.4.1. Diseño e Implementación del Panel Solar**

Para realizar un diseño efectivo de la ubicación y orientación del panel solar será necesario tener las coordenadas exactas del lugar donde se ubica el sistema de cloración, de ese modo, se podrá determinar el ángulo de elevación del panel y su orientación para tener una máxima transferencia de potencia.

Para obtener la información de las coordenadas de la ubicación del sistema se utilizará la página de Google Earth, los resultados son los siguientes:

Latitud: -14.1864, Longitud: -71.3433, Altura: 3500 m.s.n.m.

Ahora teniendo esta información, se utilizará la página de SunEarthTools.com para obtener el mapa solar del punto de ubicación del sistema de cloración, tal como se muestra en la figura 99.

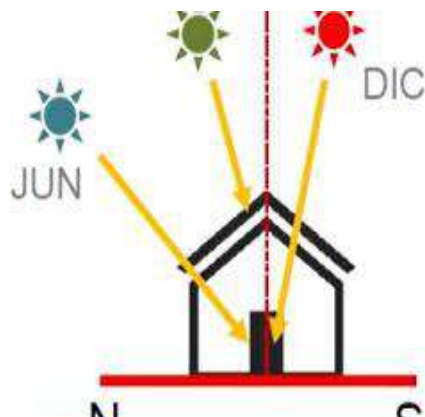
**Figura 99***Mapa solar de la ubicación del sistema de cloración*

De la figura 99 se observa que el ángulo máximo del solsticio de diciembre es de aproximadamente de  $110^\circ$  con respecto a la horizontal, y el ángulo de menor elevación del sol se da en el solsticio de junio y es aproximadamente  $52^\circ$ .

Por lo tanto, el desplazamiento del sol en el transcurso del año realizara un barrido de  $58^\circ$ , la siguiente figura 100 muestra las variaciones de la posición del sol con respecto a la ubicación del sistema de cloración.

**Figura 100**

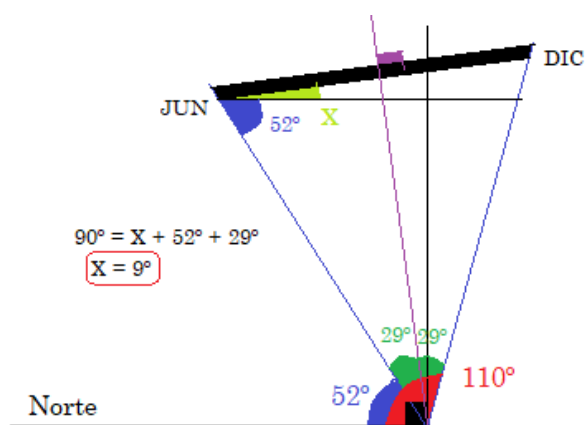
*Ubicación de la posición del sol en el transcurso de un año*



Finalmente calculamos el ángulo de elevación del panel solar con respecto a la horizontal para obtener la máxima conversión de energía solar a eléctrica, tal como se muestra en la figura 101.

**Figura 101**

*Angulo de elevación del panel solar*



Como se puede observar de la figura 101, mediante geometría elemental se puede demostrar que el ángulo de elevación del panel solar debe ser de aproximadamente  $9^\circ$  por encima de la horizontal con una orientación hacia el norte.

## CAPITULO IV: Resultados y Pruebas

### 4.1. Pruebas y Resultados del Funcionamiento del Sistema de Control

Las pruebas del funcionamiento del sistema de control se realizaron en la caseta de cloración del reservorio del sistema de agua potable tal como se muestra en la siguiente figura 102.

**Figura 102**

*Sistema de control instalado en la caseta de cloración*



**Figura 102**

*Sistema de control en funcionamiento*



Para realizar la validación del sistema de control del suministro de cloro residual se realizaron cuatro mediciones de la concentración según la recomendación técnica del “*Manual de Usuario: Cloración y Desinfección*” del PNSR, las mediciones serán realizadas un día anterior a la instalación y prueba del sistema de control, la primera medición será realizada en las primeras horas de la mañana para así obtener muestras de agua potable después de la acumulación de cloro residual durante toda la noche y la segunda medición será realizada aproximadamente al medio día para obtener muestras después de la normalización del funcionamiento del sistema de cloración.

La tercera y cuarta, será realizada al día siguiente de la instalación del sistema de control, aproximadamente en la misma hora, para así obtener muestras de agua a las mismas condiciones ambientales y de ese modo poder realizar una comparación de los resultados obtenidos de ambos días.

Según las recomendaciones técnicas del “*Manual de Usuario: Cloración y Desinfección*” las mediciones de la concentración del cloro residual deben realizarse en los siguientes puntos del sistema de agua potable:

1. Una medición en el reservorio del agua potable.
2. Una medición en la primera vivienda de la red de distribución o vivienda más próxima al reservorio de agua potable.
3. Una medición en la vivienda intermedia de la red de distribución de agua potable.
4. Una medición en la última vivienda o vivienda más alejada del reservorio de agua potable.

La siguiente figura 103 muestra las mediciones de cloro residual en el reservorio utilizando el sensor de cloro.

**Figura 103**

*Medición de cloro residual en el reservorio*



Las mediciones serán realizadas en coordinación con el responsable de ATM de la Municipalidad Distrital de San Pedro, para tener el registro de las mediciones se tendrá en cuenta el llenado de la Ficha de Monitoreo de Cloro Residual, en el anexo x se adjunta las fichas de monitoreo con los resultados obtenidos, en la siguiente tabla se muestran las mediciones obtenidas el día anterior a la instalación del sistema de control.

**Tabla 12**

*Primer monitoreo de cloro residual*

| <i>Hora.</i> | <i>Ubicación</i>    | <i>Nombre de usuario.</i> | <i>Concentración</i> |
|--------------|---------------------|---------------------------|----------------------|
| 7:00 am      | Reservorio          |                           | 2 mg/Litro           |
| 7:10 am      | Primera vivienda    | Javier Quispe Quispe.     | 1.8 mg/Litro         |
| 7:20 am      | Vivienda intermedia | Antonia Mamani Quispe     | 1.8 mg/Litro         |
| 7:25 am      | Ultima vivienda     | Inocencio Tintaya Laura   | 1.6 mg/Litro         |

Según las mediciones mostradas en la tabla anterior se puede observar una elevada concentración del cloro residual esto debido a su acumulación durante toda la noche, también se puede observar un ligero decremento de la concentración a medida que la distancia del punto de operación y el reservorio se incrementa.

La siguiente tabla muestra las mediciones realizadas el mismo día, pero aproximadamente al medio día.

**Tabla 13**

*Segundo monitoreo de cloro residual*

| <i>Hora.</i> | <i>Ubicación</i>    | <i>Nombre de usuario.</i> | <i>Concentración</i> |
|--------------|---------------------|---------------------------|----------------------|
| 12:00 pm     | Reservorio          |                           | 1 mg/Litro           |
| 12:10 pm     | Primera vivienda    | Javier Quispe Quispe.     | 0.8 mg/Litro         |
| 12:20 pm     | Vivienda intermedia | Antonia Mamani Quispe     | 0.6 mg/Litro         |
| 12:25 pm     | Ultima vivienda     | Inocencio Tintaya Laura   | 0.5 mg/Litro         |

Según los datos obtenidos en la tabla 13, se puede observar una disminución de la concentración de cloro residual a valores aceptables que se encuentran dentro del rango

permisible, esto debido a que a esas horas del día ya existe un consumo constante del agua, por lo tanto, el caudal de agua que ingresa al reservorio es constante; de los datos obtenidos anteriormente se concluye que los cálculos de la cantidad de hipoclorito de calcio son adecuados.

Seguidamente, las mediciones se realizarán después de la instalación del sistema automatizado de control, la tercera medición será también en las primeras horas de la mañana y la cuarta y última en la tarde del mismo día, los resultados obtenidos son mostrados en las siguientes tablas 14 y 15.

**Tabla 14**

*Tercer monitoreo de cloro residual*

| <i>Hora.</i> | <i>Ubicación</i>    | <i>Nombre de usuario.</i> | <i>Concentración</i> |
|--------------|---------------------|---------------------------|----------------------|
| 12:00 pm     | Reservorio          |                           | 1 mg/Litro           |
| 12:10 pm     | Primera vivienda    | Javier Quispe Quispe.     | 0.8 mg/Litro         |
| 12:20 pm     | Vivienda intermedia | Antonia Mamani Quispe     | 0.6 mg/Litro         |
| 12:25 pm     | Ultima vivienda     | Inocencio Tintaya Laura   | 0.5 mg/Litro         |

**Tabla 15**

*Cuarto monitoreo de cloro residual*

| <i>Hora.</i> | <i>Ubicación</i>    | <i>Nombre de usuario.</i> | <i>Concentración</i> |
|--------------|---------------------|---------------------------|----------------------|
| 4:00 pm      | Reservorio          |                           | 1 mg/Litro           |
| 4:10 pm      | Primera vivienda    | Javier Quispe Quispe.     | 0.8 mg/Litro         |
| 4:20 pm      | Vivienda intermedia | Antonia Mamani Quispe     | 0.6 mg/Litro         |
| 4:25 pm      | Ultima vivienda     | Inocencio Tintaya Laura   | 0.5 mg/Litro         |

Según los datos mostrados en la tabla 15 y 16 se puede observar la corrección de los valores de cloro residual, tanto en la medición realizada en horas de la mañana y la tarde.

Para tener una mejor visualización de los valores de los parámetros medidos y controlados por el sistema de control y comunicación se realizó una toma de muestra de valores durante tres días con intervalos de medición de cada 30 minutos. En el primer día de toma de

muestra solo se instaló el sistema de comunicación y medición para obtener la evolución de los parámetros durante el transcurso del día. En el día 2 y el día 3 de toma de muestra se instaló el sistema de control y comunicación completo y de ese modo se obtuvo una medición de valores de parámetros antes y después de la instalación del proyecto de tesis. En la siguiente tabla se tienen los valores obtenidos.

**Tabla 16**

*Valores medidos de cloro residual y nivel de turbidez*

| <i>Marca temporal</i> | <i>Valor Cloro Residual</i> | <i>Valor Nivel de Turbidez</i> |
|-----------------------|-----------------------------|--------------------------------|
| 18/07/2022 10:50      | 1,6                         | 0                              |
| 18/07/2022 11:20      | 1,55                        | 0,1                            |
| 18/07/2022 11:50      | 1,5                         | 0                              |
| 18/07/2022 12:20      | 1,3                         | 0                              |
| 18/07/2022 12:50      | 1,2                         | 0                              |
| 18/07/2022 13:20      | 1                           | 0                              |
| 18/07/2022 13:50      | 1,1                         | 0                              |
| 18/07/2022 14:20      | 1                           | 0                              |
| 18/07/2022 14:50      | 1                           | 0                              |
| 18/07/2022 15:20      | 0,9                         | 0                              |
| 18/07/2022 15:50      | 1                           | 0                              |
| 18/07/2022 16:20      | 1                           | 0                              |
| 18/07/2022 16:50      | 0,9                         | 0                              |
| 18/07/2022 17:20      | 1,1                         | 0                              |
| 18/07/2022 17:50      | 1                           | 0                              |
| 18/07/2022 18:20      | 1                           | 0                              |
| 18/07/2022 18:50      | 0,9                         | 0                              |
| 18/07/2022 19:20      | 1                           | 0                              |
| 18/07/2022 19:50      | 1                           | 0                              |
| 18/07/2022 20:20      | 1                           | 0                              |
| 18/07/2022 20:50      | 1,1                         | 0                              |
| 18/07/2022 21:20      | 1                           | 0                              |
| 18/07/2022 21:50      | 1                           | 0                              |
| 18/07/2022 22:20      | 1,1                         | 0                              |

---

|                  |     |   |
|------------------|-----|---|
| 18/07/2022 22:50 | 1,1 | 0 |
| 18/07/2022 23:20 | 1,2 | 0 |
| 18/07/2022 23:50 | 1,3 | 0 |
| 19/07/2022 00:20 | 1,3 | 0 |
| 19/07/2022 00:50 | 1,4 | 0 |
| 19/07/2022 01:20 | 1,5 | 0 |
| 19/07/2022 01:50 | 1,7 | 0 |
| 19/07/2022 02:20 | 1,8 | 0 |
| 19/07/2022 02:50 | 2   | 0 |
| 19/07/2022 03:20 | 2,1 | 0 |
| 19/07/2022 03:50 | 2,1 | 0 |
| 19/07/2022 04:20 | 2,4 | 0 |
| 19/07/2022 04:50 | 2,6 | 0 |
| 19/07/2022 05:20 | 3   | 0 |
| 19/07/2022 05:50 | 2,9 | 0 |
| 19/07/2022 06:20 | 3   | 0 |
| 19/07/2022 06:50 | 3   | 0 |
| 19/07/2022 07:20 | 3   | 0 |
| 19/07/2022 07:50 | 2,8 | 0 |
| 19/07/2022 08:20 | 2,6 | 0 |
| 19/07/2022 08:50 | 2,3 | 0 |
| 19/07/2022 09:20 | 2   | 0 |
| 19/07/2022 09:50 | 2   | 0 |
| 19/07/2022 10:20 | 1,7 | 0 |
| 19/07/2022 10:50 | 1,7 | 0 |
| 19/07/2022 11:20 | 1,4 | 0 |
| 19/07/2022 11:50 | 1,4 | 0 |
| 19/07/2022 12:20 | 1,2 | 0 |
| 19/07/2022 12:50 | 1,1 | 0 |
| 19/07/2022 13:20 | 1   | 0 |
| 19/07/2022 13:50 | 1   | 0 |
| 19/07/2022 14:20 | 0,9 | 0 |
| 19/07/2022 14:50 | 0,8 | 0 |

---

---

|                  |     |   |
|------------------|-----|---|
| 19/07/2022 15:20 | 0,9 | 0 |
| 19/07/2022 15:50 | 1   | 0 |
| 19/07/2022 16:20 | 1   | 0 |
| 19/07/2022 16:50 | 1,1 | 0 |
| 19/07/2022 17:20 | 1   | 0 |
| 19/07/2022 17:50 | 1   | 0 |
| 19/07/2022 18:20 | 1,1 | 0 |
| 19/07/2022 18:50 | 1   | 0 |
| 19/07/2022 19:20 | 0,9 | 0 |
| 19/07/2022 19:50 | 1   | 0 |
| 19/07/2022 20:20 | 1,1 | 0 |
| 19/07/2022 20:50 | 1   | 0 |
| 19/07/2022 21:20 | 0,9 | 0 |
| 19/07/2022 21:50 | 1,1 | 0 |
| 19/07/2022 22:20 | 1   | 0 |
| 19/07/2022 22:50 | 1   | 0 |
| 19/07/2022 23:20 | 1,1 | 0 |
| 19/07/2022 23:50 | 1   | 0 |
| 20/07/2022 00:20 | 1   | 0 |
| 20/07/2022 00:50 | 0,9 | 0 |
| 20/07/2022 01:20 | 1   | 0 |
| 20/07/2022 01:50 | 1   | 0 |
| 20/07/2022 02:20 | 1   | 0 |
| 20/07/2022 02:50 | 0,9 | 0 |
| 20/07/2022 03:20 | 0,9 | 0 |
| 20/07/2022 03:50 | 0,9 | 0 |
| 20/07/2022 04:20 | 0,8 | 0 |
| 20/07/2022 04:50 | 0,8 | 0 |
| 20/07/2022 05:20 | 0,9 | 0 |
| 20/07/2022 05:50 | 0,8 | 0 |
| 20/07/2022 06:20 | 0,8 | 0 |
| 20/07/2022 06:50 | 0,8 | 0 |
| 20/07/2022 07:20 | 0,9 | 0 |

---

---

|                  |     |   |
|------------------|-----|---|
| 20/07/2022 07:50 | 0,9 | 0 |
| 20/07/2022 08:20 | 0,9 | 0 |
| 20/07/2022 08:50 | 1   | 0 |
| 20/07/2022 09:20 | 1   | 0 |
| 20/07/2022 09:50 | 1   | 0 |
| 20/07/2022 10:20 | 1   | 0 |
| 20/07/2022 10:50 | 1   | 0 |
| 20/07/2022 11:20 | 1   | 0 |
| 20/07/2022 11:50 | 0,9 | 0 |
| 20/07/2022 12:20 | 1,1 | 0 |
| 20/07/2022 12:50 | 1   | 0 |
| 20/07/2022 13:20 | 1,1 | 0 |
| 20/07/2022 13:50 | 1   | 0 |
| 20/07/2022 14:20 | 1   | 0 |
| 20/07/2022 14:50 | 1   | 0 |
| 20/07/2022 15:20 | 1   | 0 |
| 20/07/2022 15:50 | 0,9 | 0 |
| 20/07/2022 16:20 | 0,9 | 0 |
| 20/07/2022 16:50 | 1   | 0 |
| 20/07/2022 17:20 | 0,9 | 0 |
| 20/07/2022 17:50 | 1   | 0 |
| 20/07/2022 18:20 | 0,9 | 0 |
| 20/07/2022 18:50 | 1   | 0 |
| 20/07/2022 19:20 | 1   | 0 |
| 20/07/2022 19:50 | 1   | 0 |
| 20/07/2022 20:20 | 1   | 0 |
| 20/07/2022 20:50 | 1   | 0 |
| 20/07/2022 21:20 | 0,9 | 0 |
| 20/07/2022 21:50 | 0,9 | 0 |
| 20/07/2022 22:20 | 1   | 0 |
| 20/07/2022 22:50 | 1   | 0 |
| 20/07/2022 23:20 | 1   | 0 |
| 20/07/2022 23:50 | 1   | 0 |

---

|                  |     |   |
|------------------|-----|---|
| 21/07/2022 00:20 | 1,1 | 0 |
| 21/07/2022 00:50 | 1   | 0 |
| 21/07/2022 01:20 | 1   | 0 |
| 21/07/2022 01:50 | 0,9 | 0 |
| 21/07/2022 02:20 | 0,9 | 0 |
| 21/07/2022 02:50 | 0,9 | 0 |
| 21/07/2022 03:20 | 0,9 | 0 |
| 21/07/2022 03:50 | 0,8 | 0 |
| 21/07/2022 04:20 | 0,8 | 0 |
| 21/07/2022 04:50 | 0,8 | 0 |
| 21/07/2022 05:20 | 0,9 | 0 |
| 21/07/2022 05:50 | 0,9 | 0 |
| 21/07/2022 06:20 | 0,8 | 0 |
| 21/07/2022 06:50 | 0,8 | 0 |
| 21/07/2022 07:20 | 0,9 | 0 |
| 21/07/2022 07:50 | 0,9 | 0 |
| 21/07/2022 08:20 | 1   | 0 |
| 21/07/2022 08:50 | 1   | 0 |
| 21/07/2022 09:20 | 0,9 | 0 |
| 21/07/2022 09:50 | 1   | 0 |
| 21/07/2022 10:20 | 1   | 0 |
| 21/07/2022 10:50 | 1   | 0 |
| 21/07/2022 11:20 | 1   | 0 |
| 21/07/2022 11:50 | 1,1 | 0 |
| 21/07/2022 12:20 | 1   | 0 |
| 21/07/2022 12:50 | 1   | 0 |
| 21/07/2022 13:20 | 1   | 0 |

A continuación, se muestra una gráfica de la evolución de los parámetros medidos por día de medición en función de las horas del día.

En la figura 104 correspondiente al primer día de medición se muestra la variación de la concentración de cloro residual desde las 11:00 am, se puede observar que inicialmente existe

una concentración de cloro partiendo desde 1.6 ppm lo cual va disminuyendo a medida que transcurren las horas; también en la parte final de la gráfica a partir de las 22:00 horas se puede notar un incremento de la concentración llegando a un máximo de 1.2 ppm al final de la gráfica.

En la gráfica y corresponde al segundo día de medición y se muestra la continuación de la evolución de la concentración de cloro residual, se puede observar un incremento excesivo de la concentración en las primeras horas del día llegando a un máximo de 3 ppm y manteniéndose en ese valor hasta 7:00 am. Luego de esa hora se puede observar que la concentración de cloro residual ira disminuyendo a medida que transcurren las horas del día.

La instalación y el funcionamiento del sistema de regulación y control de cloro residual se realizó en el segundo día, esperando la estabilidad de la concentración en su valor ideal en 1 ppm. Aproximadamente se a las 15:00 horas inicio su funcionamiento.

**Figura 104**

*Gráfica de concentración de cloro residual día 1*



**Figura 105***Gráfica de concentración de cloro residual día 2*

Las gráficas 106 y 107 corresponden a los días tercero y cuarto, se puede observar que la concentración de cloro residual se estabilizo en el valor de 1 ppm aproximadamente, este valor permanece constante a medida que transcurren las horas del día.

**Figura 106***Gráfica de concentración de cloro residual día 3*

**Figura 107***Grafica de concentración de cloro residual día 4*

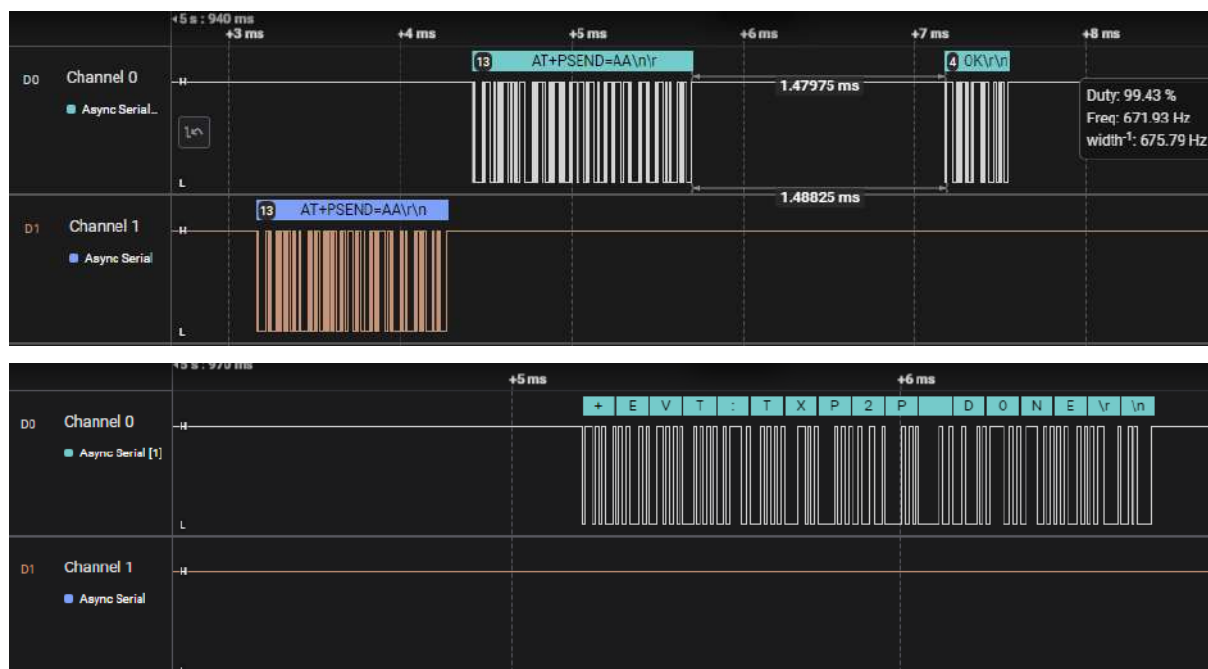
## 4.2. Prueba y Resultados del Sistema de Comunicación

Para la comprobación del funcionamiento del sistema de comunicación, primeramente, se realizarán pruebas en laboratorio para validar su correcto funcionamiento y se tomarán medidas las cuales serán comparadas con las pruebas realizadas en campo.

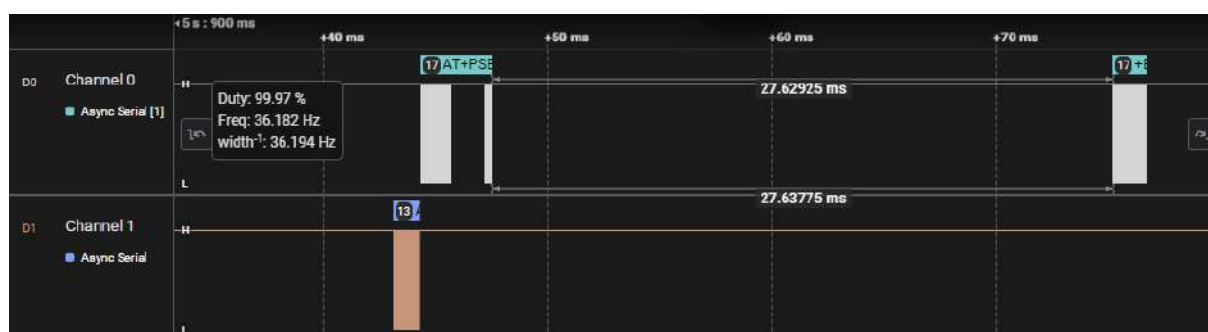
### 4.2.1. Comprobación del Sistema de Comunicación en Laboratorio

En esta primera etapa se realizarán las pruebas de conectividad en laboratorio del módulo transmisor y el microcontrolador STM32 mediante comunicación UART y después de validar su funcionamiento se realizará la respectiva prueba en campo.

En la figura 108 se muestra la estructura del paquete UART generado por el microcontrolador STM32 (*Channel 1*) y enviado por comunicación serial al módulo LoRa, en este mensaje de prueba se envía la señal de prueba “AA” mediante el comando AT+PSEND, en el canal 0 (*Channel 0*) se puede observar la respuesta del módulo LoRa en donde se observan tres señales: (1) Comando recibido (AT+PSEND=AA), (2) Señal de confirmación de recepción (OK) y (3) Señal de ejecución de envío (EVT:TXP2P DONE).

**Figura 108***Paquete de señal UART en modulo transmisor*

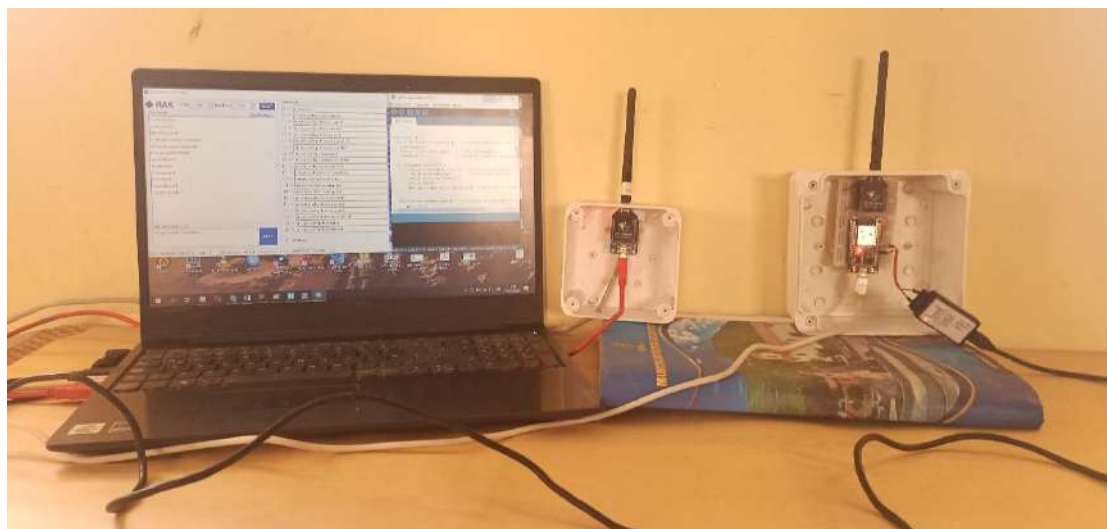
En la siguiente figura 109 se muestra la ubicación de las señales de comunicación UART desde una perspectiva más amplia con respecto al tiempo, en este diagrama se puede observar que el módulo LoRa requiere por lo menos 30 ms ( $> 27.6$  ms) de tiempo de respuesta desde el envío del comando (Channel 1) hasta la respuesta de confirmación de envío (Channel 2).

**Figura 109***Señal UART en el módulo transmisor*

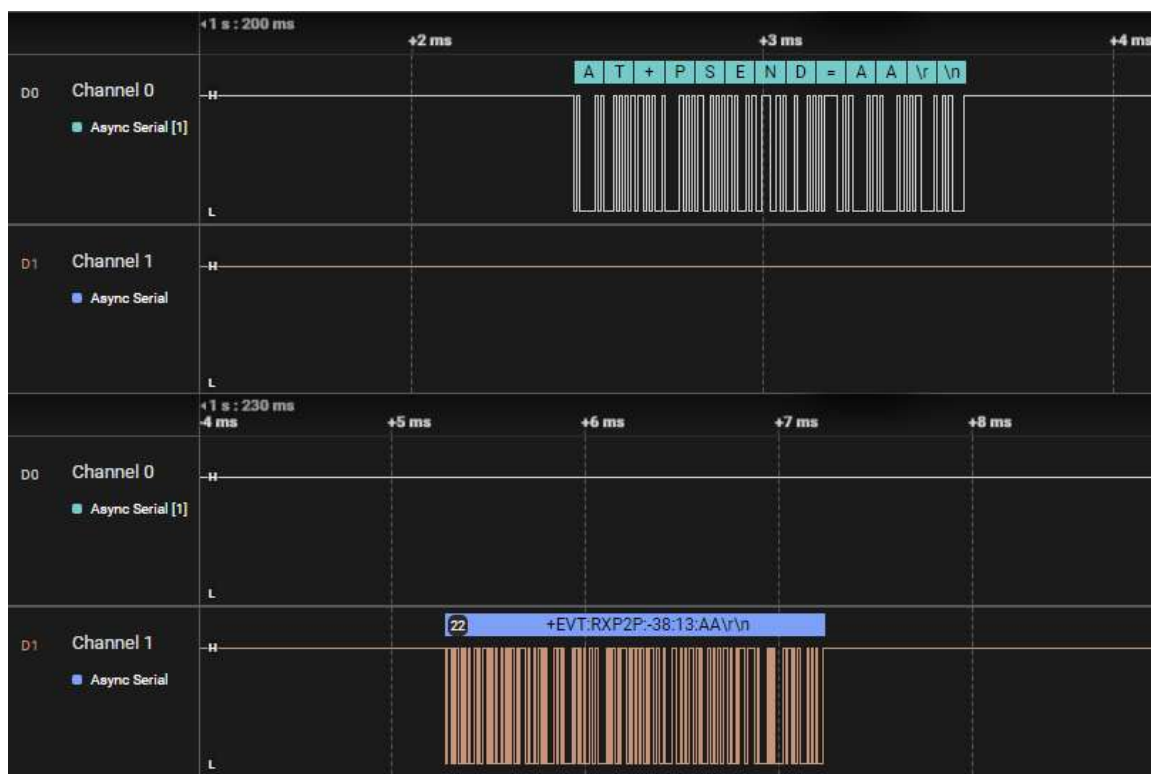
En la siguiente figura 110 se muestra la implementación del módulo transmisor - receptor basado en el módulo LoRa RAK811, estos módulos se encuentran separados aproximadamente 30 cm, en estas condiciones se inicia la comprobación de la comunicación entre ambos módulos enviando la misma señal de prueba “AA”.

**Figura 110**

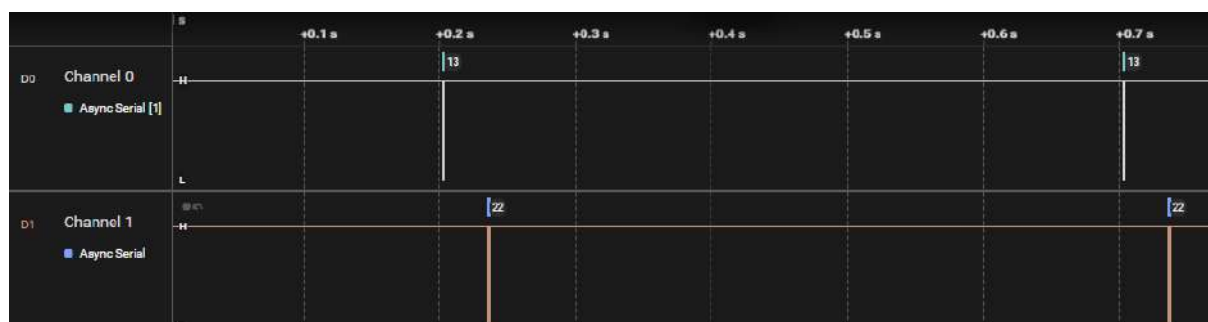
*Modulo transmisor – receptor*



En la siguiente figura 111 se muestra las señales de envío (modulo LoRa TX ) y recepción (modulo LoRa RX), en el canal 0 se pueden capturar el paquete UART generado por el microcontrolador STM32 y enviado al transmisor LoRa ubicado en la zona remota donde se encuentra implementado el proyecto, en el canal 1 se muestra la captura del paquete UART generado por el módulo receptor en el instante que recibe la señal RF, este paquete es enviado al microcontrolador ESP32 mediante comunicación serial a 115200 baudios.

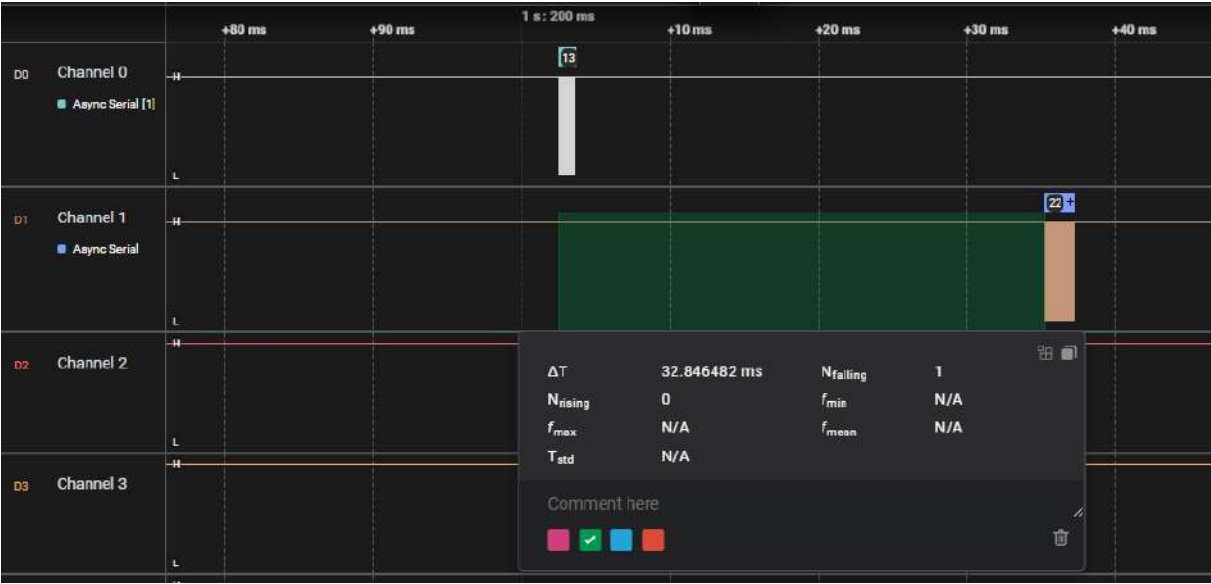
**Figura 111***Paquete UART en el transmisor y receptor*

En la siguiente figura 112 se muestra la ubicación estas señales desde una perspectiva más amplia con respecto al tiempo en donde Channel 0 corresponde al comando AT+PSEND y en Channel 1 corresponde a la señal recibida por el módulo RX.

**Figura 112***Señales en el transmisor y receptor*

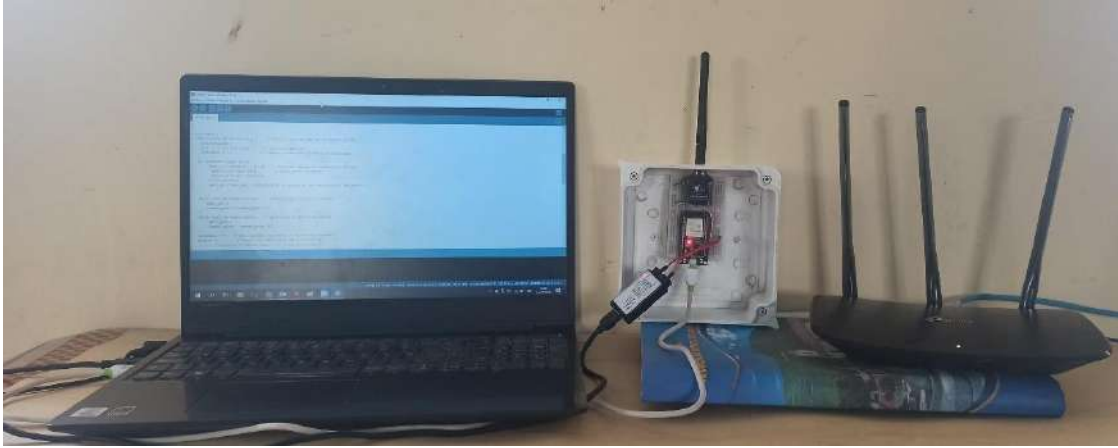
En la figura 113 se muestra el tiempo requerido para la transmisión entre ambos módulos manteniendo la misma distancia de 30 cm, se observa una medición de 32.8 ms de tiempo requerido entre ambas señales.

**Figura 113**  
*Tiempo de transmisión entre ambos módulos LoRa*



En la siguiente figura 114 se muestra la implementación del módulo receptor ESP32-RAK811 y el Access Point por medio del cual el sistema tendrá acceso a internet.

**Figura 114**  
*Módulo RAK811-ESP32 – Internet*



Manteniendo la misma distancia de separación de ambos módulos se procede a comprobar la comunicación entre ambos módulos mediante el envío de una ráfaga de datos desde el transmisor hacia el receptor, esta ráfaga de datos es una secuencia de números del 0 al 99, estos datos son empaquetados en formato UART 8N1, la separación entre cada paquete será de 50 ms, en la siguiente tabla se muestran los paquetes recibidos en el módulo receptor.

| Transmisior | Receptor             |
|-------------|----------------------|
| AT+PSEND=00 | +EVT:RXP2P:-32:14:00 |
| AT+PSEND=01 | +EVT:RXP2P:-31:14:01 |
| AT+PSEND=02 | +EVT:RXP2P:-32:13:02 |
| AT+PSEND=03 | +EVT:RXP2P:-32:13:03 |
| AT+PSEND=04 | +EVT:RXP2P:-31:14:04 |
| AT+PSEND=05 | +EVT:RXP2P:-31:14:05 |
| AT+PSEND=06 | +EVT:RXP2P:-31:14:06 |
| AT+PSEND=07 | +EVT:RXP2P:-32:13:07 |
| AT+PSEND=08 | +EVT:RXP2P:-31:14:08 |
| AT+PSEND=09 | +EVT:RXP2P:-31:14:09 |
| AT+PSEND=10 | +EVT:RXP2P:-31:13:10 |
| AT+PSEND=11 | +EVT:RXP2P:-31:13:11 |
| AT+PSEND=12 | +EVT:RXP2P:-31:13:12 |
| AT+PSEND=13 | +EVT:RXP2P:-31:13:13 |
| AT+PSEND=14 | +EVT:RXP2P:-31:13:14 |
| AT+PSEND=15 | +EVT:RXP2P:-31:14:15 |
| AT+PSEND=16 | +EVT:RXP2P:-31:13:16 |
| AT+PSEND=17 | +EVT:RXP2P:-31:13:17 |
| AT+PSEND=18 | +EVT:RXP2P:-33:14:18 |
| AT+PSEND=19 | +EVT:RXP2P:-31:14:19 |
| AT+PSEND=20 | +EVT:RXP2P:-32:14:20 |
| AT+PSEND=21 | +EVT:RXP2P:-32:14:21 |
| AT+PSEND=22 | +EVT:RXP2P:-32:13:22 |
| AT+PSEND=23 | +EVT:RXP2P:-32:14:23 |
| AT+PSEND=24 | +EVT:RXP2P:-32:14:24 |
| AT+PSEND=25 | +EVT:RXP2P:-32:14:25 |
| AT+PSEND=26 | +EVT:RXP2P:-32:13:26 |
| AT+PSEND=27 | +EVT:RXP2P:-32:14:27 |
| AT+PSEND=28 | +EVT:RXP2P:-32:14:28 |
| AT+PSEND=29 | +EVT:RXP2P:-32:14:29 |
| AT+PSEND=30 | +EVT:RXP2P:-33:14:30 |
| AT+PSEND=31 | +EVT:RXP2P:-32:14:31 |
| AT+PSEND=32 | +EVT:RXP2P:-32:14:32 |
| AT+PSEND=33 | +EVT:RXP2P:-32:14:33 |
| AT+PSEND=34 | +EVT:RXP2P:-32:14:34 |
| AT+PSEND=35 | +EVT:RXP2P:-32:14:35 |
| AT+PSEND=36 | +EVT:RXP2P:-32:14:36 |
| AT+PSEND=37 | +EVT:RXP2P:-33:14:37 |
| AT+PSEND=38 | +EVT:RXP2P:-33:14:38 |
| AT+PSEND=39 | +EVT:RXP2P:-32:14:39 |
| AT+PSEND=40 | +EVT:RXP2P:-32:14:40 |
| AT+PSEND=41 | +EVT:RXP2P:-32:14:41 |
| AT+PSEND=42 | +EVT:RXP2P:-32:14:42 |
| AT+PSEND=43 | +EVT:RXP2P:-32:14:43 |
| AT+PSEND=44 | +EVT:RXP2P:-32:14:44 |
| AT+PSEND=45 | +EVT:RXP2P:-32:14:45 |
| AT+PSEND=46 | +EVT:RXP2P:-32:14:46 |
| AT+PSEND=47 | +EVT:RXP2P:-32:14:47 |
| AT+PSEND=48 | +EVT:RXP2P:-32:14:48 |

---

|             |                      |
|-------------|----------------------|
| AT+PSEND=49 | +EVT:RXP2P:-32:14:49 |
| AT+PSEND=50 | +EVT:RXP2P:-32:15:50 |
| AT+PSEND=51 | +EVT:RXP2P:-32:14:51 |
| AT+PSEND=52 | +EVT:RXP2P:-32:14:52 |
| AT+PSEND=53 | +EVT:RXP2P:-32:15:53 |
| AT+PSEND=54 | +EVT:RXP2P:-32:14:54 |
| AT+PSEND=55 | +EVT:RXP2P:-32:14:55 |
| AT+PSEND=56 | +EVT:RXP2P:-32:15:56 |
| AT+PSEND=57 | +EVT:RXP2P:-32:14:57 |
| AT+PSEND=58 | +EVT:RXP2P:-32:14:58 |
| AT+PSEND=59 | +EVT:RXP2P:-32:14:59 |
| AT+PSEND=60 | +EVT:RXP2P:-32:14:60 |
| AT+PSEND=61 | +EVT:RXP2P:-32:14:61 |
| AT+PSEND=62 | +EVT:RXP2P:-32:14:62 |
| AT+PSEND=63 | +EVT:RXP2P:-32:14:63 |
| AT+PSEND=64 | +EVT:RXP2P:-32:14:64 |
| AT+PSEND=65 | +EVT:RXP2P:-32:15:65 |
| AT+PSEND=66 | +EVT:RXP2P:-33:14:66 |
| AT+PSEND=67 | +EVT:RXP2P:-32:14:67 |
| AT+PSEND=68 | +EVT:RXP2P:-32:14:68 |
| AT+PSEND=69 | +EVT:RXP2P:-32:14:69 |
| AT+PSEND=70 | +EVT:RXP2P:-32:14:70 |
| AT+PSEND=71 | +EVT:RXP2P:-32:14:71 |
| AT+PSEND=72 | +EVT:RXP2P:-32:14:72 |
| AT+PSEND=73 | +EVT:RXP2P:-32:13:73 |
| AT+PSEND=74 | +EVT:RXP2P:-33:13:74 |
| AT+PSEND=75 | +EVT:RXP2P:-32:14:75 |
| AT+PSEND=76 | +EVT:RXP2P:-32:13:76 |
| AT+PSEND=77 | +EVT:RXP2P:-33:13:77 |
| AT+PSEND=78 | +EVT:RXP2P:-32:14:78 |
| AT+PSEND=79 | +EVT:RXP2P:-32:14:79 |
| AT+PSEND=80 | +EVT:RXP2P:-32:13:80 |
| AT+PSEND=81 | +EVT:RXP2P:-32:13:81 |
| AT+PSEND=82 | +EVT:RXP2P:-32:13:82 |
| AT+PSEND=83 | +EVT:RXP2P:-32:13:83 |
| AT+PSEND=84 | +EVT:RXP2P:-32:13:84 |
| AT+PSEND=85 | +EVT:RXP2P:-32:13:85 |
| AT+PSEND=86 | +EVT:RXP2P:-32:13:86 |
| AT+PSEND=87 | +EVT:RXP2P:-33:13:87 |
| AT+PSEND=88 | +EVT:RXP2P:-33:13:88 |
| AT+PSEND=89 | +EVT:RXP2P:-32:14:89 |
| AT+PSEND=90 | +EVT:RXP2P:-32:14:90 |
| AT+PSEND=91 | +EVT:RXP2P:-32:14:91 |
| AT+PSEND=92 | +EVT:RXP2P:-32:13:92 |
| AT+PSEND=93 | +EVT:RXP2P:-33:14:93 |
| AT+PSEND=94 | +EVT:RXP2P:-32:14:94 |
| AT+PSEND=95 | +EVT:RXP2P:-32:14:95 |
| AT+PSEND=96 | +EVT:RXP2P:-32:14:96 |
| AT+PSEND=97 | +EVT:RXP2P:-32:14:97 |
| AT+PSEND=98 | +EVT:RXP2P:-33:14:98 |

---

---

AT+PSEND=99

---

+EVT:RXP2P:-32:15:99

---

#### 4.2.2. *Comprobación del Sistema de Comunicación en Campo*

En esta parte de la comprobación del funcionamiento del sistema de comunicación antes de la implementación final se instaló el sistema de transmisión sobre el sistema de agua potable y el receptor en la ubicación final del nodo central, la distancia entre ambos puntos es de aproximadamente de 1.5 Km. Con una línea de vista libre de obstáculos, en la siguiente figura 115 se muestra la implementación en protoboard del transmisor y al fondo se puede observar el distrito de San Pedro en donde se ubica el receptor LoRa.

**Figura 115**

*Nodo transmisor sobre el reservorio de agua potable*



Se procede nuevamente a enviar la ráfaga de datos enumerados del 0 al 99 para verificar el funcionamiento y la tasa de pérdidas de paquetes cuando los módulos LoRa están separados una distancia de 1.5 Km. finalmente se obtuvo el resultado mostrado en la siguiente tabla.

| Transmisior | Receptor              |
|-------------|-----------------------|
| AT+PSEND=00 | +EVT:RXP2P:-113:-5:00 |
| AT+PSEND=01 | +EVT:RXP2P:-112:-5:01 |
| AT+PSEND=02 | +EVT:RXP2P:-113:-6:02 |
| AT+PSEND=03 | +EVT:RXP2P:-113:-6:03 |
| AT+PSEND=04 | +EVT:RXP2P:-113:-6:04 |
| AT+PSEND=05 | +EVT:RXP2P:-113:-6:05 |
| AT+PSEND=06 | +EVT:RXP2P:-113:-5:06 |
| AT+PSEND=07 | +EVT:RXP2P:-113:-6:07 |
| AT+PSEND=08 | +EVT:RXP2P:-113:-6:08 |
| AT+PSEND=09 | +EVT:RXP2P:-113:-6:09 |
| AT+PSEND=10 | +EVT:RXP2P:-113:-5:10 |
| AT+PSEND=11 | +EVT:RXP2P:-113:-6:11 |
| AT+PSEND=12 | +EVT:RXP2P:-113:-5:12 |
| AT+PSEND=13 | +EVT:RXP2P:-113:-5:13 |
| AT+PSEND=14 | +EVT:RXP2P:-113:-6:14 |
| AT+PSEND=15 | +EVT:RXP2P:-113:-5:15 |
| AT+PSEND=16 | +EVT:RXP2P:-113:-6:16 |
| AT+PSEND=17 | +EVT:RXP2P:-113:-6:17 |
| AT+PSEND=18 | +EVT:RXP2P:-113:-6:18 |
| AT+PSEND=19 | +EVT:RXP2P:-113:-6:19 |
| AT+PSEND=20 | +EVT:RXP2P:-113:-7:20 |
| AT+PSEND=21 | +EVT:RXP2P:-113:-6:21 |
| AT+PSEND=22 | +EVT:RXP2P:-113:-6:22 |
| AT+PSEND=23 | +EVT:RXP2P:-113:-6:23 |
| AT+PSEND=24 | +EVT:RXP2P:-113:-6:24 |
| AT+PSEND=25 | +EVT:RXP2P:-113:-7:25 |
| AT+PSEND=26 | +EVT:RXP2P:-113:-5:26 |
| AT+PSEND=27 | +EVT:RXP2P:-113:-5:27 |
| AT+PSEND=28 | +EVT:RXP2P:-113:-5:28 |
| AT+PSEND=29 | +EVT:RXP2P:-114:-6:29 |
| AT+PSEND=30 | +EVT:RXP2P:-113:-5:30 |
| AT+PSEND=31 | +EVT:RXP2P:-113:-6:31 |
| AT+PSEND=32 | +EVT:RXP2P:-113:-5:32 |
| AT+PSEND=33 | +EVT:RXP2P:-114:-5:33 |
| AT+PSEND=34 | +EVT:RXP2P:-113:-5:34 |
| AT+PSEND=35 | +EVT:RXP2P:-113:-5:35 |
| AT+PSEND=36 | +EVT:RXP2P:-114:-5:36 |
| AT+PSEND=37 | +EVT:RXP2P:-113:-5:37 |
| AT+PSEND=38 | +EVT:RXP2P:-113:-5:38 |
| AT+PSEND=39 | +EVT:RXP2P:-113:-5:39 |
| AT+PSEND=40 | +EVT:RXP2P:-113:-5:40 |
| AT+PSEND=41 | +EVT:RXP2P:-113:-6:41 |
| AT+PSEND=42 | +EVT:RXP2P:-113:-6:42 |
| AT+PSEND=43 | +EVT:RXP2P:-113:-6:43 |
| AT+PSEND=44 | +EVT:RXP2P:-113:-6:44 |
| AT+PSEND=45 | +EVT:RXP2P:-113:-5:45 |
| AT+PSEND=46 | +EVT:RXP2P:-113:-6:46 |
| AT+PSEND=47 | +EVT:RXP2P:-113:-5:47 |
| AT+PSEND=48 | +EVT:RXP2P:-113:-5:48 |

---

|             |                       |
|-------------|-----------------------|
| AT+PSEND=49 | +EVT:RXP2P:-113:-6:49 |
| AT+PSEND=50 | +EVT:RXP2P:-113:-6:50 |
| AT+PSEND=51 | +EVT:RXP2P:-113:-5:51 |
| AT+PSEND=52 | +EVT:RXP2P:-113:-5:52 |
| AT+PSEND=53 | +EVT:RXP2P:-113:-6:53 |
| AT+PSEND=54 | +EVT:RXP2P:-113:-6:54 |
| AT+PSEND=55 | +EVT:RXP2P:-113:-5:55 |
| AT+PSEND=56 | +EVT:RXP2P:-113:-6:56 |
| AT+PSEND=57 | +EVT:RXP2P:-113:-5:57 |
| AT+PSEND=58 | +EVT:RXP2P:-113:-5:58 |
| AT+PSEND=59 | +EVT:RXP2P:-112:-6:59 |
| AT+PSEND=60 | +EVT:RXP2P:-113:-5:60 |
| AT+PSEND=61 | +EVT:RXP2P:-113:-6:61 |
| AT+PSEND=62 | +EVT:RXP2P:-113:-6:62 |
| AT+PSEND=63 | +EVT:RXP2P:-113:-5:63 |
| AT+PSEND=64 | +EVT:RXP2P:-113:-5:64 |
| AT+PSEND=65 | +EVT:RXP2P:-113:-5:65 |
| AT+PSEND=66 | +EVT:RXP2P:-113:-6:66 |
| AT+PSEND=67 | +EVT:RXP2P:-113:-7:67 |
| AT+PSEND=68 | +EVT:RXP2P:-113:-6:68 |
| AT+PSEND=69 | +EVT:RXP2P:-113:-6:69 |
| AT+PSEND=70 | +EVT:RXP2P:-113:-6:70 |
| AT+PSEND=71 | +EVT:RXP2P:-113:-5:71 |
| AT+PSEND=72 | +EVT:RXP2P:-113:-6:72 |
| AT+PSEND=73 | +EVT:RXP2P:-113:-7:73 |
| AT+PSEND=74 | +EVT:RXP2P:-113:-6:74 |
| AT+PSEND=75 | +EVT:RXP2P:-113:-5:75 |
| AT+PSEND=76 | +EVT:RXP2P:-113:-6:76 |
| AT+PSEND=77 | +EVT:RXP2P:-113:-6:77 |
| AT+PSEND=78 | +EVT:RXP2P:-113:-7:78 |
| AT+PSEND=79 | +EVT:RXP2P:-113:-6:79 |
| AT+PSEND=80 | +EVT:RXP2P:-113:-6:80 |
| AT+PSEND=81 | +EVT:RXP2P:-113:-6:81 |
| AT+PSEND=82 | +EVT:RXP2P:-113:-6:82 |
| AT+PSEND=83 | +EVT:RXP2P:-113:-5:83 |
| AT+PSEND=84 | +EVT:RXP2P:-113:-6:84 |
| AT+PSEND=85 | +EVT:RXP2P:-114:-5:85 |
| AT+PSEND=86 | +EVT:RXP2P:-113:-5:86 |
| AT+PSEND=87 | +EVT:RXP2P:-113:-6:87 |
| AT+PSEND=88 | +EVT:RXP2P:-114:-6:88 |
| AT+PSEND=89 | +EVT:RXP2P:-113:-6:89 |
| AT+PSEND=90 | +EVT:RXP2P:-113:-6:90 |
| AT+PSEND=91 | +EVT:RXP2P:-113:-7:91 |
| AT+PSEND=92 | +EVT:RXP2P:-113:-6:92 |
| AT+PSEND=93 | +EVT:RXP2P:-113:-8:93 |
| AT+PSEND=94 | +EVT:RXP2P:-113:-6:94 |
| AT+PSEND=95 | +EVT:RXP2P:-113:-6:95 |
| AT+PSEND=96 | +EVT:RXP2P:-113:-7:96 |
| AT+PSEND=97 | +EVT:RXP2P:-113:-7:97 |
| AT+PSEND=98 | +EVT:RXP2P:-113:-6:98 |

---

---

AT+PSEND=99

---

+EVT:RXP2P:-113:-7:99

---

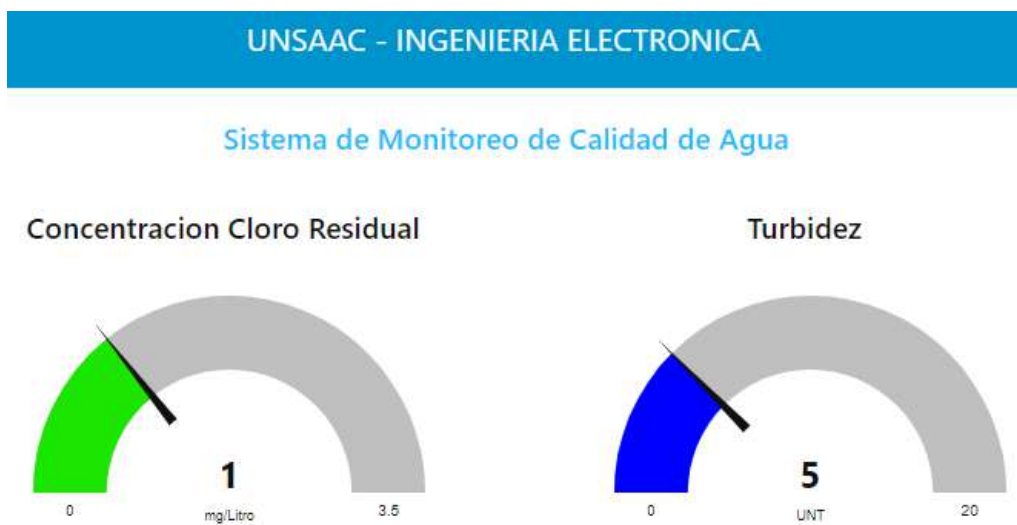
De la tabla anterior se puede observar una pérdida de paquetes del 0 % pudiéndose comprobar la correcta transmisión de la información por medio del sistema de comunicación diseñado. Después de la comprobación del correcto funcionamiento del enlace inalámbrico se procede a instalar el sistema completo (control y comunicación) tal como se muestra en la siguiente figura 116.

**Figura 116**

*Modulo transmisor en campo*



Finalmente tenemos la visualización de grafica de los datos obtenidos por los sensores en Node-RED.

**Figura 117***Entorno grafico sobre Node-RED*

### **Conclusiones**

- Se implementó un sistema de control y monitoreo en tiempo real mediante el cual se regula la concentración de cloro residual y se monitorea las variables que determinan la calidad de agua para consumo humano.
- Mediante el sistema de adquisición de datos se logró realizar un monitoreo remoto de las variables concentración de cloro residual, caudal y nivel de turbidez del agua que ingresa al reservorio.
- Se controló el suministro de la solución hipoclorosa en función de las variables medidas, por lo tanto, se mantuvo los niveles de cloro residual dentro de los límites recomendados por los organismos de salud.
- Mediante la implementación del sistema de comunicación IoT se logró obtener la medición de las variables nivel de turbidez y concentración de cloro residual a diferentes horas del día.
- Se implementó y se validó el funcionamiento del sistema completo de medición, regulación y comunicación.

### **Recomendaciones**

- Es recomendable mantener la temperatura del sistema de cloración constante debido a que la velocidad de variación de la concentración depende de la temperatura del medio ambiente.
- Para mayores tiempos de prueba o una implementación definitiva se recomienda considerar las variaciones del clima de la zona ya que en épocas de lluvia se tiene periodos nublados de hasta 7 días consecutivos, afectando de ese modo la alimentación fotovoltaica del sistema.
- Una alta concentración de hipoclorito de calcio en el tanque de solución madre permite una fácil solidificación de la solución en las manguerillas y en el mecanismo de estrangulamiento cuando la temperatura del medio ambiente disminuye, por lo tanto, es recomendable siempre mantener la concentración de hipoclorito de calcio en valores bajos.
- Existe desinformación por parte de algunos usuarios sobre la desinfección del agua potable mediante el cloro residual argumentando que produce enfermedades como cáncer, lo cual es falso mientras se realice de manera correcta y se mantenga la concentración dentro de los niveles adecuados, por lo tanto, se recomienda mayor concientización de parte de los responsables del área de saneamiento.
- Se recomienda hacer una limpieza periódica del sistema de almacenamiento de agua potable debido a que la concentración de cloro residual decae más rápidamente en entornos de acumulación de sólidos disueltos y presencia de organismos vivos.

## Referencias

- Deitel, P. (2009). *C how to program* (6ª ed.). Pearson.
- Zhu, Y. (2017). *Embedded Systems with ARM Cortex – M Microcontrollers in Assembly Language and C* (3ª ed.). E-Man Press LLC.
- Yiu, J. (2013). *The definitive guide to ARM Cortex-M3 and Cortex-M4 processors* (3ª ed.).
- Dirección General de Salud Ambiental - Ministerio de Salud. (20 711). *Reglamento de la calidad del agua para consumo humano*. MINSA.
- Lora Alliance. (2020). *RP002-1.0.1 LoRaWAN Regional Parameters*. <https://loralliance.org>
- Organización Mundial de la Salud. (2018). *Guías para la calidad del agua de consumo humano: Cuarta edición que incorpora la primera adenda*. OMS.
- Enciso Jauregui, N. A. (2019). *Seguimiento de la concentración de cloro residual en tanque de almacenamiento, red de distribución y tanques residenciales en el municipio de Fortul, departamento de Arauca* (Tesis de grado, Universidad de La Salle). Repositorio Institucional de la Universidad de La Salle.
- Universidad Politécnica Salesiana. (s.f.). *Sistema de telemonitoreo para cloro residual*. <https://dspace.ups.edu.ec/bitstream/123456789/18450/4/UPS-CT008701.pdf>
- STMicroelectronics. (s.f.). *STM32F401xB/C and STM32F401xD/E advanced Arm®-based 32-bit MCUs (Reference Manual)*. STMicroelectronics.
- Vistronica. (s.f.). *Sensor de turbidez y sus aplicaciones*. <https://www.vistronica.com/sensores/sensor-de-turbidez-analogica-para-arduino-detail.html>
- RAK Wireless. (s.f.). *RAK831 Gateway para aplicaciones IoT*. <https://store.rakwireless.com/products/rak831-gateway-module>

### Apéndice A. Hoja de presupuesto

|    | Componente                                | Cantidad | Precio U.    | Precio        | Moneda       |
|----|-------------------------------------------|----------|--------------|---------------|--------------|
| 1  | Microcontrolador STM32F401RE              | 1        | 110          | 110           | Soles        |
| 2  | Microcontrolador ESP32                    | 1        | 35           | 35            | Soles        |
| 3  | Modulo LoRa RAK811                        | 2        | 125          | 250           | Soles        |
| 4  | Sensor de Turbidez                        | 1        | 65           | 65            | Soles        |
| 5  | Batería 12 voltios                        | 1        | 70           | 70            | Soles        |
| 6  | Mini bomba de agua                        | 2        | 10           | 20            | Soles        |
| 7  | Motor Paso a Paso                         | 1        | 15           | 15            | Soles        |
| 8  | Analizador Lógico                         | 1        | 65           | 65            | Soles        |
| 9  | Caja Pequeña de Plástico                  | 2        | 10           | 20            | Soles        |
| 10 | Caja Grande de Plástico                   | 1        | 15           | 15            | Soles        |
| 11 | Recipiente de Plástico                    | 1        | 2            | 2             | Soles        |
| 12 | Protoboard Pequeño                        | 1        | 5            | 5             | Soles        |
| 13 | Cable USB                                 | 3        | 5            | 15            | Soles        |
| 14 | Manguera de Plástico                      | 2        | 2            | 4             | Soles        |
| 15 | Sensor de Caudal                          | 1        | 65           | 65            | Soles        |
| 16 | Conector UPR para Agua                    | 2        | 3            | 6             | Soles        |
| 17 | Juego de Cables para Conexiones           | 1        | 5            | 5             | Soles        |
| 18 | Schmitt Trigger (74LS14)                  | 1        | 2.5          | 2.5           | Soles        |
| 19 | Muestras de Calibracion de S. de Turbidez | 3        | 50           | 150           | Soles        |
| 20 | Driver ULM2003                            | 1        | 2            | 2             | Soles        |
| 21 | Transistor TIP41 NPN                      | 2        | 2            | 4             | Soles        |
| 22 | Transistor TIP42 PNP                      | 1        | 2            | 2             | Soles        |
| 23 | Regulador LM317                           | 1        | 3            | 3             | Soles        |
| 24 | Juego de Resistencias Eléctricas          | 1        | 5            | 5             | Soles        |
| 25 | Router WiFi                               | 1        | 90           | 90            | Soles        |
| 27 | Sensor de Cloro Residual                  | 1        | 1000         | 1000          | Soles        |
| 26 | Adaptador SigBee USB para RAK811          | 1        | 15           | 15            | Soles        |
| 27 | Adaptador Simple para RAK811              | 1        | 5            | 5             | Soles        |
| 28 | Materiales de escritorio/impresiones      | 1        | 100          | 100           | Soles        |
|    |                                           |          | <b>Total</b> | <b>2145.5</b> | <b>Soles</b> |

## Apéndice B. Código de programación del microcontrolador ESP32

En el siguiente código se implementa la programación del ESP32 utilizando el Ide de Arduino , primeramente, se incluyen las librerías necesarias para la comunicación del microcontrolador hacia internet, se define en la macro “SERVER\_IP” la dirección IP del servidor donde se ejecuta una instancia de Node-RED; seguidamente se define las variables para implementar las comunicaciones del ESP32 con el MONITOR SERIAL y el RAK811, ambas comunicaciones a 115200 baudios, todas estas configuraciones se implementan en el bucle `void setup()` .

En la siguiente etapa del código de programación, primeramente, se envía hacia el RAK811 la información ingresada por teclado mediante el monitor serial, las siguientes líneas de código se encargan de extraer los valores de concentración cloro residual y nivel de turbidez enviados desde el nodo remoto en campo hacia el RAK811 implementado en el nodo central, y este a su vez lo transmite hacia el ESP32 mediante la comunicación UART.

Finalmente se implementa en el código una función `void sendPost()` para enviar los datos recibidos hacia la instancia implementada de Node-RED sobre un servidor AWS

```

1  #include <WiFi.h> %CSSC
2  #include <HTTPClient.h>
3
4  #define SERVER_IP "3.144.24.9:1880"
5
6  #ifndef STASSID
7  #define STASSID "TP-Link_5FB0"
8  #define STAPSK  "52002267"
9  #endif
10
11 #define RX 16
12 #define TX 17
13
14 char datorx;
15 String string="";
16 String val_x1;
17 String val_x2;
18 WiFiClient client;
19 HTTPClient http;
20
21
22 void setup() {
```

```

23
24 Serial.begin(115200);
25 Serial2.begin(115200, SERIAL_8N1, RX, TX);
26 while (!Serial) {}
27
28 WiFi.begin(STASSID, STAPSK);
29 Serial.println("Conectando a la red WiFi");
30 while (WiFi.status() != WL_CONNECTED) {
31     delay(500);
32     Serial.print(".");
33 }
34 Serial.println("");
35 Serial.print("Connected! IP address: ");
36 Serial.println(WiFi.localIP());
37 }
38 void loop() {
39
40 if (Serial.available() > 0) {
41     Serial2.write(Serial.read());
42 }
43 if (Serial2.available() > 0) {
44     datorx = Serial2.read();
45     string += datorx;
46 if (datorx == 'A') {
47     int index = string.indexOf(':');
48     int index2 = index + 4;
49     val_x1 = string.substring(index + 1, index2);
50     sendPostA();
51     string = "";
52 }
53
54 if (datorx == 'B') {
55     int index = string.indexOf(':');
56     int index2 = index + 4;
57     val_x2 = string.substring(index + 1, index2);
58     sendPostB();
59     string = "";
60 }
61
62 }
63 }
64 void sendPostA() {
65     Serial.println("[HTTP] Iniciamos la peticion POST");
66     String serverPeticion = "http://" SERVER_IP;
67     serverPeticion += "/sensor";
68     http.begin(client, serverPeticion);
69     http.addHeader("Content-Type", "application/json");
70     String jsonData = "{\"data_cloro\":\"\"";
71     jsonData += val_x1;
72     jsonData += "\"";
73     jsonData += "\"}";
74
75     int httpCode = http.POST(jsonData);
76     if (httpCode > 0) {
77         Serial.printf("[HTTP] POST... code: %d\n", httpCode);
78         if (httpCode == HTTP_CODE_OK) {

```

```

79         const String& payload = http.getString();
80         Serial.println("Respuesta Json:\n<<");
81         Serial.println(payload);
82         Serial.println(">>");
83     }
84     } else {
85         Serial.printf("[GET]      Petición      post      fallo,      error:      %s\n",
86 http.errorToString(httpCode).c_str());
87     }
88     http.end();
89 }
90 void sendPostB() {
91     Serial.println("[HTTP] Iniciamos la petición POST");
92     String serverPetición = "http://" SERVER_IP;
93     serverPetición += "/sensor";
94     http.begin(client, serverPetición);
95     http.addHeader("Content-Type", "application/json");
96     String jsonData = "{\"data_turbidez\":\"\"";
97     jsonData += val_x2;
98     jsonData += "\"";
99     jsonData += "\"}";
100
101     int httpCode = http.POST(jsonData);
102     if (httpCode > 0) {
103         Serial.printf("[HTTP] POST... code: %d\n", httpCode);
104         if (httpCode == HTTP_CODE_OK) {
105             const String& payload = http.getString();
106             Serial.println("Respuesta Json:\n<<");
107             Serial.println(payload);
108             Serial.println(">>");
109         }
110     } else {
111         Serial.printf("[GET]      Petición      post      fallo,      error:      %s\n",
112 http.errorToString(httpCode).c_str());
113     }
114     http.end();
115 }

```

## Apéndice C. Código de programación del microcontrolador STM32F401RE

La implementación del código de programación para este microcontrolador se desarrolló en lenguaje C, utilizando el software **STM32CubeIde** el cual fue desarrollado por el mismo fabricante **STelectronic**; la implementación del código de programación se realizó directamente desde los registros para tener un mejor control y mayor eficiencia del funcionamiento de los periféricos del microcontrolador.

C es un lenguaje de programación estructurado, por lo tanto, la lógica de programación del código sigue una secuencia de estructuras, librerías y funciones que progresivamente son llamadas por una estructura principal denominada `int main(void)`.

En la primera parte del código fuente principal se incluyen las diferentes librerías propias del lenguaje C y también las librerías diseñadas como parte de la lógica de control, seguidamente se definen todos los prototipos de las diferentes funciones las cuales configuran de manera independiente un determinado periférico del microcontrolador, luego se declaran todas las variables utilizadas en código fuente. En la parte final se implementa la estructura principal, la cual de acuerdo una secuencia definida hace el llamado de las diferentes funciones declaradas previamente. También se definió estructuras de cabecera de programa del tipo .h en donde configura el funcionamiento de determinados periféricos que requieren extensas líneas de programación, tales como UART, EXTI, INT\_ANALOG. A continuación, se muestra el código en C.

main.c

```

1  #include <stdint.h>
2  #include <stdio.h>
3  #include "stm32f4xx.h"
4  /* Se incluyen las librerías privada-----*/
5  #include "RCC.h"
6  #include "Delay.h"
7  #include "Config.h"
8  #include "USART.h"
9  #define EXTI_FALLING_IT 1
10 /* Private typedef -----*/
11 /* Private define -----*/
12 #define GPIO_PIN_1 1

```

```

13 #define Kp    4.79 // factor de calibracion de caudal
14
15 /* Private macro -----*/
16 #define USE_SWV                1
17 /* Private variables -----*/
18
19 /* Prototipo de las funciones privadas -----*/
20 void PIN_control_motor(void);
21 void LED_Caudal(void);
22 void Paso_Derecha(void);
23 void Paso_Izquierda(void);
24 void Apagar(void);
25 void Pin_Button(void);
26 void Pin_Bomba_Agua(void);
27 void ADC1_InitSingleMode(void);
28 void ADC1_ChannelConfiguration(void);
29 void RTC_init(void);
30 void Time_Update(void);
31 //Lectura analogica del sensor de cloro
32 uint16_t ADC1_ChannelRead1(uint8_t chx); //Sensor de cloro
33 //Lectura analogica del sensor de turbidez
34 uint16_t ADC1_ChannelRead2(uint8_t chx); //Sensor de turbidez
35 // prototipo de las INTERRUPCIONES
36 static void EXTI1_Config(uint8_t typeIT);
37 void GPIO_EXTI_Callback(uint8_t pin);
38
39 // prototipo del Timer2
40 static void TIMER2_CH3_OC_Config(void);
41 /* Declaracion de las variables usadas-----*/
42 uint8_t txBuffer[200];
43 uint32_t paso = 0;
44 int16_t datoRx =0;
45 uint32_t count;
46 uint32_t caudal;
47 uint32_t len;
48 int16_t frec;
49 uint16_t BufferCloro;
50 uint16_t BufferTurbidez;
51 /* Declaracion de las variables de tiempo-----*/
52 uint32_t hora;
53 uint32_t minuto;
54 uint32_t segundo;
55 uint32_t timeActual=0;
56 uint32_t timeLast=0;
57
58 int tcd;
59 int i=0;
60 int d=10;
61 /* External variables -----*/
62 int main(void)
63 {
64     HSE_Config();
65     PIN_control_motor();
66     LED_Caudal();
67     Delay_Init(SystemCoreClock/1000);
68     EXTI1_Config(EXTI_FALLING_IT);

```

```

69  TIMER2_CH3_OC_Config();
70  USART_Init(USART1, 115200);
71  Pin_Button();
72  Pin_Bomba_Agua();
73  ADC1_InitSingleMode();
74  ADC1_ChannelConfiguration();
75  RTC_init();
76  /* Loop forever */
77  for(;;){
78
79      count =0;
80      delay_ms(1000);
81      caudal = count;
82      printf("Frecuencia->%d\n", (int) count);
83
84      if(caudal>100){
85          datoRx = 15;
86      }
87      else{
88          datoRx =0;
89      }
90
91      if (datoRx != paso){
92          while(datoRx>paso){
93              Paso_Izquierda();
94              paso++;
95          }
96
97          while(datoRx<paso){
98              Paso_Derecha();
99              paso--;
100          }
101          Apagar();
102      }
103
104      Time_Update();
105
106      if (!(GPIOC->IDR & 0b1<<13) | (timeActual - timeLast > 30)){
107
108          timeLast = timeActual;
109          // Encender Bomba1 por 5segundos
110          GPIOC->BSRR |= 1U<<2; //1
111          GPIOC->BSRR &=~ 1U<<18;
112          delay_ms(5000);
113          GPIOC->BSRR &=~ 1U<<2; //0
114          GPIOC->BSRR |= 1U<<18;
115
116          BufferCloro = ADC1_ChannelRead1(8);
117          if(BufferCloro <=9){
118      len = sprintf((char*)txBuffer, "at+send=lorap2p:00%dA\r\n", (uint8_t) (0xFF & BufferCloro));
119          USART_SendData(USART1, txBuffer, len);
120          }
121          else if((BufferCloro >=10) & (BufferCloro <100)){
122      len = sprintf((char*)txBuffer, "at+send=lorap2p:0%dA\r\n", (uint8_t) (0xFF & BufferCloro));
123          USART_SendData(USART1, txBuffer, len);
124          }

```

```

125         else{
126     len = sprintf((char*)txBuffer, "at+send=lorap2p:%dB\r\n", (uint8_t) (0xFF & BufferCloro));
127         USART_SendData(USART1, txBuffer, len);
128         }
129
130         BufferTurbidez = ADC1_ChannelRead2(9);
131
132         if(BufferTurbidez <=9){
133     len = sprintf((char*)txBuffer, "at+send=lorap2p:00%dB\r\n", (uint8_t) (0xFF & BufferTurbidez));
134         USART_SendData(USART1, txBuffer, len);
135         }
136         else if((BufferTurbidez >=10) & (BufferTurbidez <100)){
137     len = sprintf((char*)txBuffer, "at+send=lorap2p:0%dB\r\n", (uint8_t) (0xFF & BufferTurbidez));
138         USART_SendData(USART1, txBuffer, len);
139         }
140         else{
141     len = sprintf((char*)txBuffer, "at+send=lorap2p:%dB\r\n", (uint8_t) (0xFF & BufferTurbidez));
142         USART_SendData(USART1, txBuffer, len);
143         }
144         GPIOC->BSRR |= 1U<<3;    //1
145         GPIOC->BSRR &=~ 1U<<19;
146         delay_ms(5000);
147         GPIOC->BSRR &=~ 1U<<3;    //0
148         GPIOC->BSRR |= 1U<<19;
149     }
150
151 }
152 }
153
154 static void TIMER2_CH3_OC_Config(void){
155     /*configurar el canal*/
156     //PB10-> TIM2_CH3
157     RCC->AHB1ENR |= RCC_AHB1ENR_GPIOBEN;
158     GPIOB->MODER &=~ (GPIO_MODER_MODE10);
159     GPIOB->MODER |= GPIO_MODER_MODE10_1;                //FUNCION ALTERNATIVA
160     GPIOB->OSPEEDR |= GPIO_OSPEEDR_OSPEED10;    //VERY HIGH SPEED
161     GPIOB->PUPDR &=~ (GPIO_PUPDR_PUPD10);        //NO PULL UP/DOWN
162     GPIOB->AFR[1] &=~ GPIO_AFRH_AFSEL10;        //CLEAR
163     GPIOB->AFR[1] |= (1U)<<(10-8)*4;
164     RCC->APB1ENR |= RCC_APB1ENR_TIM2EN;
165     /**
166     * Fpwm = Ftim/(ARR + 1)(PSC + 1)
167     * ARR = Ftim / (PSC +1)Fpwm - 1
168     * ARR = 16MHZ/(15 + 1)1000 - 1
169     * ARR = 1000 - 1
170     */
171     TIM2->PSC = 16 - 1;
172     TIM2->ARR = 1000000 - 1; // Auto Reload Register
173     /*Configurar el timer*/
174     TIM2->CR1 = 0; // Control Register
175     /*Configurar el registro CCMR2      Capture Compare Mode Register CCMR */
176     TIM2->CCMR2 &=~ (TIM_CCMR2_CC3S); //configura el canal como salida
177     TIM2->CCMR2 &=~ (TIM_CCMR2_OC3M);
178     TIM2->CCMR2 |= 3U<<4;                //toggle Mode
179     TIM2->CCER &=~ (TIM_CCER_CC3P);
180     TIM2->CCER |= TIM_CCER_CC3E;        //HABILITAR LA SALIDA DE COMPARACION

```

```

181 TIM2->CCR3 = 499999;
182 TIM2->CR1 |= TIM_CR1_CEN;          //HABILITA EL CONTEO
183
184 }
185
186 void PIN_control_motor(void) {
187
188     /*Habilitamos el CLOCK del bus AHB1*/
189     GPIO_CLOCK_ENABLE(A);
190
191     /*Pin A1*/
192     GPIOX_MODER(MODE_OUT, PA1);
193     GPIOX_PUPDR(MODE_PU_NONE, PA1);
194     GPIOX_OSPEEDR(MODE_SPD_VHIGH, PA1);
195     /*Pin A2*/
196     GPIOX_MODER(MODE_OUT, PA0);
197     GPIOX_PUPDR(MODE_PU_NONE, PA0);
198     GPIOX_OSPEEDR(MODE_SPD_VHIGH, PA0);
199     /*Pin A3*/
200     GPIOX_MODER(MODE_OUT, PA9);
201     GPIOX_PUPDR(MODE_PU_NONE, PA9);
202     GPIOX_OSPEEDR(MODE_SPD_VHIGH, PA9);
203     /*Pin A4*/
204     GPIOX_MODER(MODE_OUT, PA4);
205     GPIOX_PUPDR(MODE_PU_NONE, PA4);
206     GPIOX_OSPEEDR(MODE_SPD_VHIGH, PA4);
207
208 }
209
210 void LED_Caudal(void) {
211     /*User LED indica el funcionamiento del sensor de caudal*/
212     GPIOX_MODER(MODE_OUT, PA5);
213     GPIOX_PUPDR(MODE_PU_NONE, PA5);
214     GPIOX_OSPEEDR(MODE_SPD_VHIGH, PA5);
215 }
216
217 void Pin_Button(void) {
218
219     GPIO_CLOCK_ENABLE(C);
220
221     GPIOX_MODER(MODE_DIGITAL_INPUT, BUTTON);
222     GPIOX_PUPDR(MODE_PU_UP, BUTTON);
223     GPIOX_OSPEEDR(MODE_SPD_VHIGH, BUTTON);
224 }
225
226
227
228 void Pin_Bomba_Agua(void) {
229     /*Pin C2*/
230     GPIOX_MODER(MODE_OUT, PinBomba1);
231     GPIOX_PUPDR(MODE_PU_NONE, PinBomba1);
232     GPIOX_OSPEEDR(MODE_SPD_VHIGH, PinBomba1);
233
234     /*Pin C3*/
235     GPIOX_MODER(MODE_OUT, PinBomba2);
236     GPIOX_PUPDR(MODE_PU_NONE, PinBomba2);

```

```

237  GPIOX_OSPEEDR(MODE_SPD_VHIGH, PinBomba2);
238  }
239
240  void ADC1_InitSingleMode(void) {
241      /*habilitar el reloj*/
242      RCC->APB2ENR |= RCC_APB2ENR_ADC1EN;
243      /*configurar el ADC*/
244      ADC1->COMMON->CCR &=~ADC_CCR_ADCPRE;//PCLK2 / 2 = 8MHZ
245      ADC1->CR1 = 0;
246      ADC1->CR1 |= 0X2UL<<ADC_CR1_RES_Pos;
247
248      ADC1->CR2 = 0;
249
250      /*encender el ADC*/
251      ADC1->CR2 |= ADC_CR2_ADON;
252      return;
253  }
254
255  void ADC1_ChanelConfiguration(void) {
256      /*PB0->CH8 PB1->CH9*/
257      RCC->AHB1ENR |= RCC_AHB1ENR_GPIOBEN;
258
259      //PB0
260      GPIOB->MODER |= GPIO_MODER_MODE0 | GPIO_MODER_MODE1;
261      GPIOB->PUPDR &=~(GPIO_PUPDR_PUPD0);
262      GPIOB->PUPDR &=~(GPIO_PUPDR_PUPD1);
263
264      /*SAMPLING TIME*/
265      ADC1->SMPR2 |= ADC_SMPR2_SMP8_0;//15 Cycles
266      ADC1->SMPR2 |= ADC_SMPR2_SMP9_0;
267      //ADC1->SMPR1 |= ADC_SMPR1_SMP11_1 | ADC_SMPR1_SMP11_0; //56 Cycles
268
269      return;
270  }
271
272  uint16_t ADC1_ChannelRead1(uint8_t chx){
273      /*configurar el SQR1 y SQR3*/
274      ADC1->SQR1 = 0; //Una sola conversion
275      ADC1->SQR3 = chx;//se le asigna el numero de canal
276      /*modo single*/
277      ADC1->CR2 |= ADC_CR2_SWSTART;
278      //esperar que se termine la conversion*/
279      while(!(ADC1->SR & ADC_SR_EOC));
280
281      //se retorna el valor ADC
282      return ADC1->DR;
283  }
284
285  uint16_t ADC1_ChannelRead2(uint8_t chx){
286      /*configurar el SQR1 y SQR3*/
287      ADC1->SQR1 = 0; //Una sola conversion
288      ADC1->SQR3 = chx;//se le asigna el numero de canal
289      /*modo single*/
290      ADC1->CR2 |= ADC_CR2_SWSTART;
291      //esperar que se termine la conversion*/
292      while(!(ADC1->SR & ADC_SR_EOC));

```

```

293
294 //se retorna el valor ADC
295 return ADC1->DR;
296 }
297
298 void Paso_Derecha(void) {
299
300     GPIOA->BSRR &=~ 1U<<1; //0
301     GPIOA->BSRR |= 1U<<17;
302     GPIOA->BSRR &=~ 1U<<0; //0
303     GPIOA->BSRR |= 1U<<16;
304     GPIOA->BSRR |= 1U<<9; //1
305     GPIOA->BSRR &=~ 1U<<25;
306     GPIOA->BSRR |= 1U<<4; //1
307     GPIOA->BSRR &=~ 1U<<20;
308     delay_ms(d);
309     GPIOA->BSRR &=~ 1U<<1; //0
310     GPIOA->BSRR |= 1U<<17;
311     GPIOA->BSRR |= 1U<<0; //1
312     GPIOA->BSRR &=~ 1U<<16;
313     GPIOA->BSRR |= 1U<<9; //1
314     GPIOA->BSRR &=~ 1U<<25;
315     GPIOA->BSRR &=~ 1U<<4; //0
316     GPIOA->BSRR |= 1U<<20;
317     delay_ms(d);
318     GPIOA->BSRR |= 1U<<1; //1
319     GPIOA->BSRR &=~ 1U<<17;
320     GPIOA->BSRR |= 1U<<0; //1
321     GPIOA->BSRR &=~ 1U<<16;
322     GPIOA->BSRR &=~ 1U<<9; //0
323     GPIOA->BSRR |= 1U<<25;
324     GPIOA->BSRR &=~ 1U<<4; //0
325     GPIOA->BSRR |= 1U<<20;
326     delay_ms(d);
327     GPIOA->BSRR |= 1U<<1; //1
328     GPIOA->BSRR &=~ 1U<<17;
329     GPIOA->BSRR &=~ 1U<<0; //0
330     GPIOA->BSRR |= 1U<<16;
331     GPIOA->BSRR &=~ 1U<<9; //0
332     GPIOA->BSRR |= 1U<<25;
333     GPIOA->BSRR |= 1U<<4; //1
334     GPIOA->BSRR &=~ 1U<<20;
335     delay_ms(d);
336 }
337
338 void Paso_Izquierda(void) {
339
340     GPIOA->BSRR |= 1U<<1; //1
341     GPIOA->BSRR &=~ 1U<<17;
342     GPIOA->BSRR |= 1U<<0; //1
343     GPIOA->BSRR &=~ 1U<<16;
344     GPIOA->BSRR &=~ 1U<<9; //0
345     GPIOA->BSRR |= 1U<<25;
346     GPIOA->BSRR &=~ 1U<<4; //0
347     GPIOA->BSRR |= 1U<<20;
348     delay_ms(d);

```

```

349  GPIOA->BSRR &=~ 1U<<1;    //0
350  GPIOA->BSRR |= 1U<<17;
351  GPIOA->BSRR |= 1U<<0;      //1
352  GPIOA->BSRR &=~ 1U<<16;
353  GPIOA->BSRR |= 1U<<9;      //1
354  GPIOA->BSRR &=~ 1U<<25;
355  GPIOA->BSRR &=~ 1U<<4;     //0
356  GPIOA->BSRR |= 1U<<20;
357  delay_ms(d);
358  GPIOA->BSRR &=~ 1U<<1;    //0
359  GPIOA->BSRR |= 1U<<17;
360  GPIOA->BSRR &=~ 1U<<0;     //0
361  GPIOA->BSRR |= 1U<<16;
362  GPIOA->BSRR |= 1U<<9;      //1
363  GPIOA->BSRR &=~ 1U<<25;
364  GPIOA->BSRR |= 1U<<4;      //1
365  GPIOA->BSRR &=~ 1U<<20;
366  delay_ms(d);
367  GPIOA->BSRR |= 1U<<1;      //1
368  GPIOA->BSRR &=~ 1U<<17;
369  GPIOA->BSRR &=~ 1U<<0;     //0
370  GPIOA->BSRR |= 1U<<16;
371  GPIOA->BSRR &=~ 1U<<9;      //0
372  GPIOA->BSRR |= 1U<<25;
373  GPIOA->BSRR |= 1U<<4;      //1
374      GPIOA->BSRR &=~ 1U<<20;
375      delay_ms(d);
376  }
377
378  void Apagar(void) {
379  GPIOA->BSRR &=~ 1U<<1;
380  GPIOA->BSRR |= 1U<<17;
381  GPIOA->BSRR &=~ 1U<<0;
382  GPIOA->BSRR |= 1U<<16;
383  GPIOA->BSRR &=~ 1U<<9;
384  GPIOA->BSRR |= 1U<<25;
385  GPIOA->BSRR &=~ 1U<<4;
386  GPIOA->BSRR |= 1U<<20;
387  }
388
389  void RTC_init(void) {
390
391  // habilitamos el el bus APB1
392  RCC->APB1ENR |= RCC_APB1ENR_PWREN;
393  PWR->CR |= PWR_CR_DBP; // Desabilitamos la proteccion
394                      //para acceder a los registros rtc
395  RCC->BDCR |= RCC_BDCR_RTCSEL; // HSE A 8 M
396  RCC->CFGR |= 0x8<<16U; // DIVIDE ENTRE 8 LA FUENTE
397  RCC->BDCR |= RCC_BDCR_RTCEN; //DE RELOJ PARA LOGRAR 1M
398
399  //deshabilitamos la proteccion mediante las KEYS
400  RTC->WPR = 0xCA;
401  RTC->WPR = 0x53;
402
403
404

```

```

405 RTC->ISR |= RTC_ISR_INIT;
406 while (!(RTC->ISR & RTC_ISR_INITF)); // espera hasta que flag
407                                     //este habilitado
408 RTC->PRER =0;
409 RTC->PRER |= 0x63 << 16;
410 RTC->PRER |= 0x270F;
411
412 RTC->TR = 0<<22 | 1<<20 | 1<<16 | 5<<12 | 1<<8 | 0<<4 | 0<<0;
413 RTC->DR = 20<<20 | 22<<16 | 6<<13 | 0<<12 | 4<<8 | 1<<4 | 6<<0;
414
415 RTC->CR &= ~ RTC_CR_BYPSHAD;
416
417 RTC->ISR &= ~ RTC_ISR_INIT;
418
419 RTC->WPR =0xFF;
420 return;
421
422 }
423
424
425 void Time_Update(void) {
426
427 hora = ((uint32_t) (RTC->TR >>20) & 0b11)*10 + ((uint32_t) (RTC->TR >>16) & 0b1111);
428 minuto = ((uint32_t) (RTC->TR >>12) & 0b111)*10 + ((uint32_t) (RTC->TR >>8) & 0b1111);
429 segundo = ((uint32_t) (RTC->TR >>4) & 0b111)*10 + ((uint32_t) (RTC->TR >>0) & 0b1111);
430
431 timeActual = 60*hora+minuto;
432 return;
433 }
434
435 static void EXTI1_Config(uint8_t typeIT){
436
437 /*enable clock*/
438 RCC->AHB1ENR |= GPIOX_CLOCK(SenCaudal);
439 RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN;
440
441 /*configurar el pin como entrada*/
442 GPIOX_MODER(MODE_DIGITAL_INPUT, SenCaudal);
443 GPIOX_PUPDR(MODE_PU_NONE, SenCaudal);
444
445 /*configurar el borde que dispara la interrupcion*/
446 if(typeIT == EXTI_FALLING_IT)
447     EXTI->FTSR |= 1U<<1;
448 /*habilitar el interrupt mask bit*/
449 EXTI->IMR |= 1U<<1;
450 /*asignar el pin a la linea EXTI*/
451 SYSCFG->EXTICR[0] &= ~ 0xFU<<4;
452 SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI1_PC;
453
454 NVIC_SetPriority(EXTI1_IRQn, 1); //PRIORIDAD 1
455 NVIC_EnableIRQ(EXTI1_IRQn); //HABILITA LA INTERRUPCION EXTERNA
456
457 //clear pending bit
458 EXTI->PR |= 1U<<1;
459 NVIC_ClearPendingIRQ(EXTI1_IRQn);
460 }

```

```

461
462 void GPIO_EXTI_Callback(uint8_t pin){
463     //codigo
464     if(pin == GPIO_PIN_1){
465         count++;
466         GPIOA->ODR ^= 1U<<5;
467     return;
468     }
469 }
470 /*****
471 int __io_putchar(int ch){
472     #if (USE_SWV== 1)
473     ITM_SendChar ( (uint32_t)ch);
474     #endif
475     return ch;
476 }

```

#### Config.h

```

1  #ifndef CONFIG_H_
2  #define CONFIG_H_
3
4  #include "stm32f4xx.h"
5  /*****
6  #define PA1      A , 1
7  #define PA0      A , 0
8  #define PA9      A , 9
9  #define PA4      A , 4
10 #define PA5      A , 5
11 #define SenCaudal C , 1
12 #define PinBomba1 C , 2
13 #define PinBomba2 C , 3
14 #define BUTTON   C , 13
15 #endif /* CONFIG_H_ */

```

#### Delay.h

```

1  #ifndef INC_DELAY_H_
2  #define INC_DELAY_H_
3
4  #include <stdint.h>
5  #include "stm32f4xx.h"
6  #define USE_DELAY_US 0
7
8  /**
9   * @brief Configuración de la SysTick
10  */
11 void Delay_Init(uint32_t ticks);
12
13 #if USE_DELAY_US == 1
14 /**
15  * @brief retardos en us
16  */
17 void delay_us(uint32_t delay);
18 #endif
19 /**

```

```

20     * @brief retardos en ms
21     */
22     void delay_ms(uint32_t delay);
23
24
25     #endif /* INC_DELAY_H_ */

```

#### Delay.c

```

1  #include "Delay.h"
2
3  /*VARIABLES PRIVADAS*/
4  uint32_t uwTick; //incrementar 1us/ms
5
6
7  /**
8   * @brief Retorna el valor de la variable ticks
9   * @param none
10  * @return ticks
11  */
12  uint32_t GetTick(void) {
13
14      return uwTick;
15  }
16
17  /**
18   * @brief Configuración de la SysTick
19   */
20  void Delay_Init(uint32_t ticks){
21      /*deshabilitar la systick*/
22      SysTick->CTRL &=~ SysTick_CTRL_ENABLE_Msk;
23      /*cargar el valor de reload al registro LOAD*/
24      SysTick->LOAD = ticks - 1; //2^24 - SystemCoreClock * delay/1000
25      /*selecciona la fuente de reloj*/
26      SysTick->CTRL |= 1u<<2;
27      /*elegir la fuente de reloj Habilitar el conteo*/
28      /*cambiar la prioridad*/
29      NVIC_SetPriority(SysTick_IRQn,7); //prioridad de 7
30      /*habilitar la IT*/
31      SysTick->CTRL |= 1U<<1; //se habilita el bit de interrupcion
32      SysTick->CTRL |= 1; //SYSCLK
33
34      return;
35  }
36
37
38
39  /**
40   * @brief retardos en us
41   */
42  #if USE_DELAY_US == 1
43  void delay_us(uint32_t delay){
44      int32_t tickstart = GetTick();
45      uint32_t wait = delay;
46      while ((GetTick() - tickstart) < wait); //retardo en us
47
48      return;

```

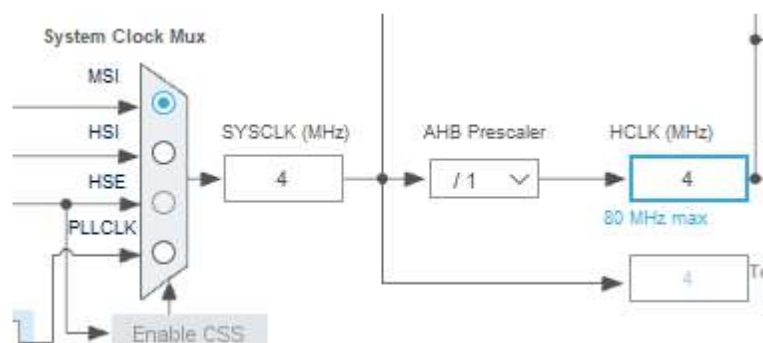
```

49  }
50  #endif
51  /**
52   * @brief retardos en ms
53   */
54  void delay_ms(uint32_t delay){
55  #if USE_DELAY_US == 1
56      for(uint32_t i = 0; i<delay; i++){
57          delay_us(1000);                //1ms de retardo
58      }
59  #else
60      int32_t tickstart = GetTick();
61      uint32_t wait = delay;
62      while ((GetTick() - tickstart) < wait); //retardo en ms
63  #endif
64  }
65  }

```

Para establecer la velocidad de operación del microcontrolador se configuro el archivo llamado **Reloj Clock Configuration (RCC)** en donde se definieron funciones para cada una de las tres fuentes de reloj principales. La función `flash_cofig(void)` configura la fuente HSI como valor del SysClock del sistema, el cual tiene un valor por defecto de 16Mhz. La función `HSE_Config(void)` configura como fuente de reloj del SysClock una fuente externa en modo de bypass, en este microcontrolador se puede añadir a la tarjeta un oscilador de cristal o utilizar como fuente de reloj externo la salida del microcontrolador encargado de la depuración y la grabación el cual tiene un valor de 8Mhz y ya se encuentra integrado en la tarjeta de desarrollo del microcontrolador.

En la siguiente figura se muestra el diagrama simplificado de las fuentes de reloj y el multiplexor encargado de seleccionar la fuente más adecuada.



Finalmente se implementa la función `PLL_Config(void)` la cual selecciona como fuente de reloj la salida del módulo PLL, pudiéndose generar oscilaciones de hasta 84Mhz como máximo. Las siguientes líneas de código muestran la implementación de la lógica de programación.

RCC.h

```

1  #ifndef RCC_H_
2  #define RCC_H_
3
4  #include "stm32f4xx.h"
5
6  /**
7   * Configuración de la flash
8   */
9  void flash_cofig(void);
10 /**
11  * Selecciona el HSE como sysclk
12  */
13 void HSE_Config(void);
14 /**
15  * Configurar el PLL a 84MHz
16  */
17 void PLL_Config(void);
18
19 #endif /* RCC_H_ */

```

RCC.c

```

1  #include "RCC.h"
2
3  void flash_cofig(void) {
4      FLASH->ACR |= 1<<9;
5      FLASH->ACR |= 1<<10;
6      FLASH->ACR |= 1<<8;
7      //SE HABILITAN ALGUNOS CLOCK
8      RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN;
9      RCC->APB1ENR |= RCC_APB1ENR_PWREN;
10 }
11
12 void HSE_Config(void) {
13     RCC->CR |= RCC_CR_HSEON | RCC_CR_HSEBYP; //se habilita el HSE en modo BYPASS
14     while(!(RCC->CR & RCC_CR_HSERDY)); //se espera que el HSE este listo
15     RCC->CFGR = RCC_CFGR_SW_HSE; //Se selecciona el HSE Como fuente sysclk
16     while(!(RCC->CFGR & RCC_CFGR_SWS_HSE)); //se espera que se haga el cambio
17     SystemCoreClockUpdate();
18 }
19
20
21 /**
22  * configura el PLL a 84MHz

```

```

23     */
24 void PLL_Config(void) {
25     HSE_Config();
26     RCC->CR &=~ (1U<<24);
27     while(RCC->CR & 1U<<25);
28     //1. PLL Fuente de Reloj
29     RCC->PLLCFGR |= RCC_PLLCFGR_PLLSRC_HSE; //selecciona HSE CLOCK SOURCE
30     //2. configure division factor (M, N, P, Q)
31     /**
32      * f(VCO clock) = f(PLL clock input) Ã— (PLLN / PLLM) (192 and 432 MHz)
33      * f(PLL general clock output) = f(VCO clock) / PLLP          <= 84MHz
34      */
35     //M
36     RCC->PLLCFGR &=~ RCC_PLLCFGR_PLLM;
37     RCC->PLLCFGR |= 4U; //VCO frecuencia de entrada (1MHz to 2MHz)
38     //N
39     RCC->PLLCFGR &=~ RCC_PLLCFGR_PLLN;
40     RCC->PLLCFGR |= 168U<<6; //168MHz
41     //P
42     RCC->PLLCFGR &=~ RCC_PLLCFGR_PLLP;
43     RCC->PLLCFGR |= 1U<<16; //84MHz
44     //3. enable PLL
45     RCC->CR |= RCC_CR_PLLON;
46     while(!(RCC->CR & RCC_CR_PLLRDY));
47     /*configurar AHBx, APBx prescaler*/
48     RCC->CFGR = 0x00000000; //reset
49     RCC->CFGR |= 0b100<<10; //100: AHB clock dividido por 2
50     /*configurar flash latency*/
51     FLASH->ACR &=~ FLASH_ACR_LATENCY;
52     FLASH->ACR |= FLASH_ACR_LATENCY_2WS;
53     /*seleccionar el system clock source*/
54     RCC->CFGR |= RCC_CFGR_SW_PLL;
55     while(!(RCC->CFGR & RCC_CFGR_SWS_PLL));
56     SystemCoreClockUpdate();
57     return;
58 }

```

#### USART.h

```

1  #ifndef INC_USART_H_
2  #define INC_USART_H_
3
4  #include "stm32f4xx.h"
5  #include "Config.h"
6  #include <stdlib.h>
7
8
9  #define OVERSAMPLING16          0
10 #define OVERSAMPLING8          1
11
12 typedef struct
13 {
14     USART_TypeDef *pUSARTx;

```

```

15     uint8_t *pTxBuffer;
16     uint8_t *pRxBuffer;
17     uint32_t TxLen;
18     uint32_t RxLen;
19     uint8_t TxBusyState;
20     uint8_t RxBusyState;
21 }USART_Handle_t;
22
23 /*prototipo de funciones*/
24 /**
25  * @brief configura el USARTx como:
26  * 1-bit stop
27  * sin paridad
28  * 8-bit de datos
29  */
30 void USART_Init(USART_TypeDef * USARTx,uint32_t baud);
31
32 /**
33  * @brief enviar N data por el USART
34  */
35 void USART_SendData(USART_TypeDef * USARTx,uint8_t *pTxBuffer, uint32_t Size);
36 /**
37  * @brief enviar N data por el USART
38  */
39 void USART_ReceiveData(USART_TypeDef * USARTx,uint8_t *pRxBuffer, uint32_t Size);
40
41
42
43 #endif /* INC_USART_H_ */

```

#### USART.c

```

1  #include "USART.h"
2
3  /**
4   * @brief configura los pines PA9 y P10
5   */
6  static void USART1_PinoutConfig(void) {
7      /*habilitar el reloj*/
8      RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
9      /*CONFIGURAMOS PA9 - PA10*/
10     GPIOA->MODER &=~ (GPIO_MODER_MODE10 | GPIO_MODER_MODE9);
11     GPIOA->MODER |= GPIO_MODER_MODE10_1 | GPIO_MODER_MODE9_1;
12     //Funcion alternativa
13     GPIOA->AFR[1] &=~ (GPIO_AFRH_AFSEL10 | GPIO_AFRH_AFSEL9);
14     GPIOA->AFR[1] |= 7U<<GPIO_AFRH_AFSEL10_Pos;//PA10 -> USART1_RX
15     GPIOA->AFR[1] |= 7U<<GPIO_AFRH_AFSEL9_Pos;//PA9 -> USART1_TX
16     GPIOA->OSPEEDR |= GPIO_OSPEEDR_OSPEED10 | GPIO_OSPEEDR_OSPEED9;
17     GPIOA->PUPDR |= GPIO_PUPDR_PUPD10 | GPIO_PUPDR_PUPD9; //PULL UP
18     return;
19 }
20
21 /**
22  * @configura el USARTx como:
23  * 1-bit stop

```

```

24 * sin paridad
25 * 8-bit de datos
26 */
27 void USART_Init(USART_TypeDef * USARTx, uint32_t baud) {
28     /*habilitar el reloj*/
29     if(USARTx == USART1){
30         /*habilitar el reloj*/
31         RCC->APB2ENR |= RCC_APB2ENR_USART1EN;
32         /*configurar los pines (TX y RX)*/
33         USART1_PinoutConfig();
34
35     }
36     /*CONFIGURAR LOS PARETOS DE LA COMUNICACION*/
37     USARTx->CR1 &=~ (USART_CR1_M); //8-BIT DE DATOS
38     USARTx->CR1 &=~ (USART_CR1_OVER8); //OVERSAMPLING 16
39     USARTx->BRR = UART_BRR_SAMPLING16(SystemCoreClock, baud);
40     USARTx->CR1 |= USART_CR1_TE | USART_CR1_RE; //habilita la rx y tx
41     USARTx->CR2 &=~ (USART_CR2_STOP); //1bit de stop
42     USARTx->CR1 &=~ (USART_CR1_PCE); //sin paridad
43     /*configuracion de interrupciones*/
44
45     /*habilitar el USART*/
46     USARTx->CR1 |= USART_CR1_UE; //Habilitacion del USART
47     return;
48 }
49
50
51 /**
52  * @brief enviar N data por el USART
53  */
54 void USART_SendData(USART_TypeDef * USARTx, uint8_t *pTxBuffer, uint32_t Size){
55     uint16_t *pdata;
56
57     for(uint32_t i = 0; i < Size; i++){
58         //verificar el flag TXE
59         while(!(USARTx->SR & USART_SR_TXE));
60
61         //verificar si se ve transmitir en modo de 8bit o 9bit
62         if(USARTx->CR1 & USART_CR1_M){
63             //enviar el dato
64             pdata = (uint16_t*) pTxBuffer;
65             USARTx->DR = (*pdata & (uint16_t)0x1FF);
66             //Verificar la paridad
67             if(!(USARTx->CR1 & USART_CR1_PCE )){
68                 pTxBuffer++;
69                 pTxBuffer++;
70             }
71             else{
72                 pTxBuffer++;
73             }
74         }
75         else{ //modo de 8bit
76             USARTx->DR = (*pTxBuffer & (uint8_t)0xFF);
77             pTxBuffer++;
78         }
79     }

```

```

80      /*esperamos que TC se ponga a 1*/
81      while(!(USARTx->SR & USART_SR_TC));
82
83      return;
84 }
85
86 /**
87  * @brief enviar N data por el USART
88  */
89 void USART_ReceiveData(USART_TypeDef * USARTx, uint8_t *pRxBuffer, uint32_t Size){
90     for(uint32_t i= 0; i<Size;i++){
91         //verificar el flag RXNE
92         while(!(USARTx->SR & USART_SR_RXNE));
93         //verificar si se ve transmitir en modo de 8bit o 9bit
94         if(USARTx->CR1 & USART_CR1_M){
95
96             /*VERIFICAR LA PARIDAD*/
97             if(!(USARTx->CR1 & USART_CR1_PCE )){ //sin paridad
98                 *((uint16_t*) pRxBuffer) = USARTx->DR & (uint16_t) 0x1FF;
99                 pRxBuffer++;
100                pRxBuffer++;
101            }
102            else{ //paridad
103                *pRxBuffer = USARTx->DR & (uint8_t) 0xFF;
104                pRxBuffer++;
105            }
106        }
107        else{
108            /*VERIFICAR LA PARIDAD*/
109            if(!(USARTx->CR1 & USART_CR1_PCE)){
110                *pRxBuffer = USARTx->DR & (uint8_t) 0xFF;
111            }
112            else{
113                *pRxBuffer = USARTx->DR & (uint8_t) 0x7F;
114            }
115            pRxBuffer++;
116        }
117    }
118    return;
119 }

```

El software STM32CubeIde crea por defecto el archivo stm32f4xx\_it.c en donde se introduce la lógica de programación cuando se genera un llamado de interrupción de cualquier periférico del microcontrolador, para el desarrollo de este proyecto de tesis se utilizó la interrupción generada por el periférico SysTick para generar bases de tiempo por hardware y una interrupción generada de manera externa por el sensor de caudal. En la siguiente secuencia de código se muestra en la línea 130 que la variable uwTick incrementa su conteo cada vez que

se genera una interrupción, y la línea de código 142 muestra la implementación de la función de interrupción externa, la cual a su vez hace un llamado de otra función GPIO\_EXTI\_Callback(1) la cual se encuentra implementada en el código fuente principal main.c

#### stm32f4xx\_it.c

```

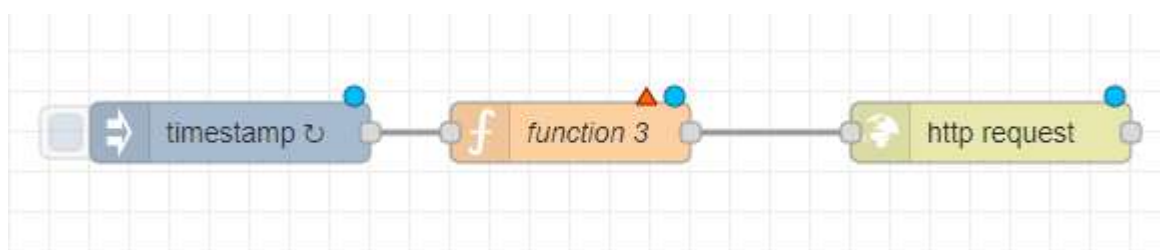
127 /**
128  * @brief LINEAS DE CODIGO OMITIDAS.
129  */
130 void SysTick_Handler(void)
131 {
132     uwTick++;
133 }
134
135 /*****
136  * STM32F4xx Peripheral Interrupt Handlers
137  * Add here the Interrupt Handlers for the used peripherals.
138  * For the available peripheral interrupt handler names,
139  * please refer to the startup file (startup_stm32f4xx.s).
140  *****/
141
142 void EXTI1_IRQHandler(void) {
143
144     if(EXTI->PR & 1u<<1) {
145         EXTI->PR |= 1u<<1;
146         /*codigo*/
147         GPIO_EXTI_Callback(1);
148     }
149
150     return;
151 }

```

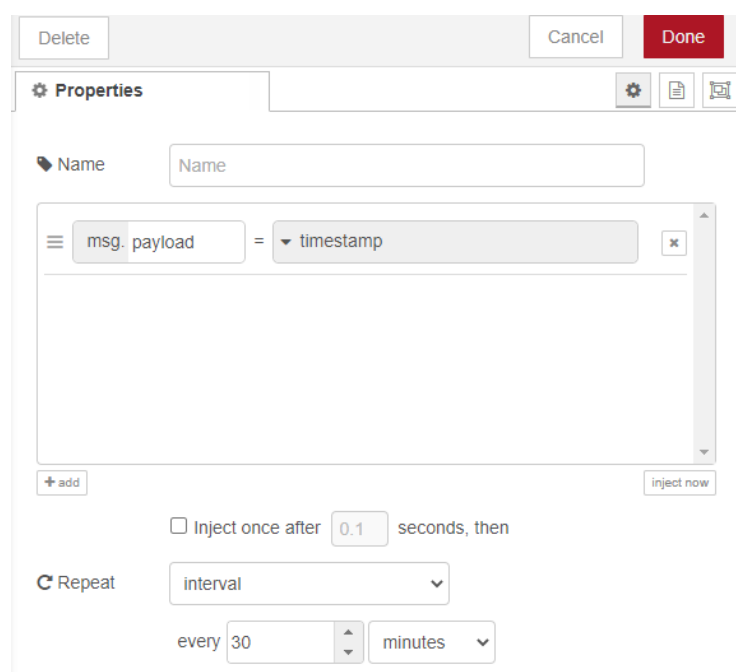
## Apéndice C. Implementación de Node-RED y el almacenamiento de los valores obtenidos en una hoja de cálculo en Google Drive.

Los valores obtenidos de las mediciones realizadas por los sensores deberán ser enviados hacia Node-RED para una visualización grafica en tiempo real. Estos valores también serán almacenados en una hoja de cálculo de Excel almacenado en una cuenta de Google Drive, para tener una estadística de los valores medidos y poder visualizar sus variaciones a diferentes horas del día.

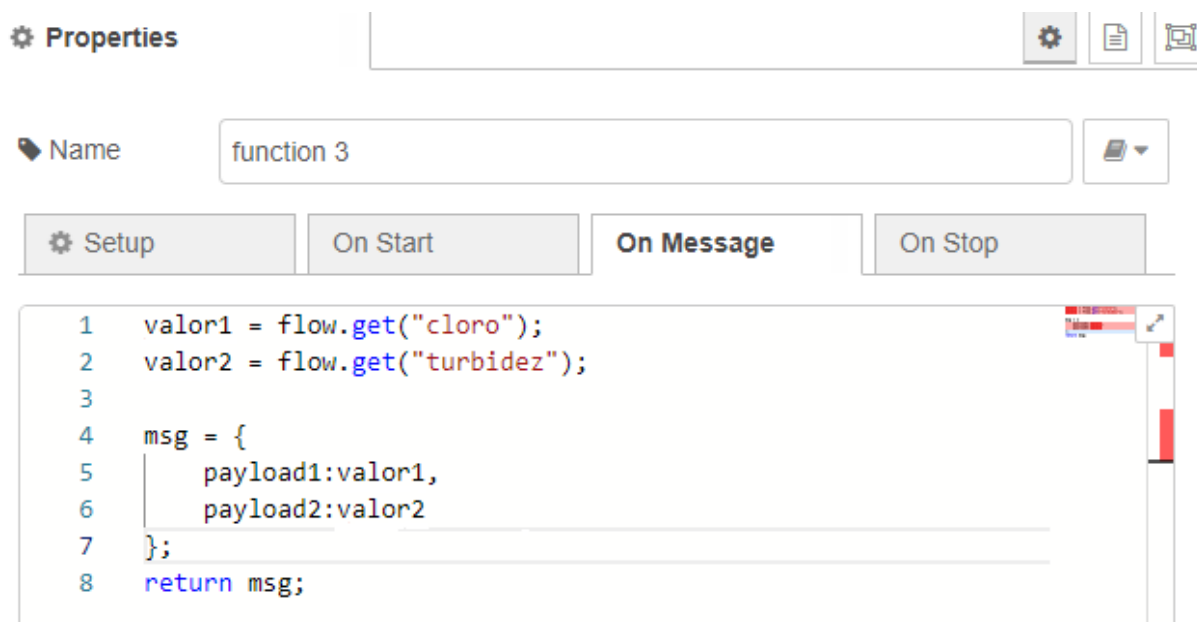
Para lograr este objetivo se añade a la secuencia de programación de la hoja de Node-RED los siguientes nodos mostrados en la figura.



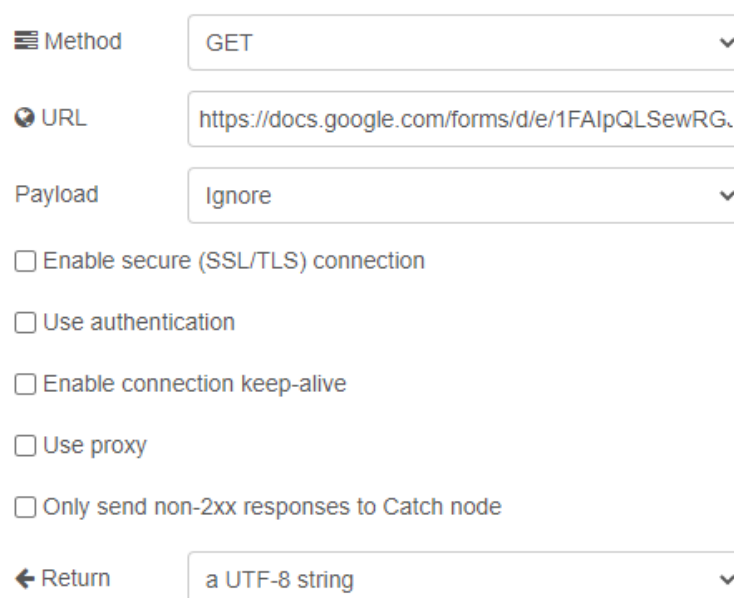
En donde el bloque *timestamp* establece la fecha y hora de la muestra recibida y el periodo de tiempo de envío de información hacia la hoja de Excel en 30 min. La siguiente figura muestra su configuración.



El bloque **function 3** obtiene los valores recibidos y los almacena en variables independientes para su respectivo almacenamiento en la hoja de cálculo Excel. La siguiente figura muestra su configuración interna.



Finalmente, en el último bloque **http request** se configura la dirección URL hacia donde se enviarán los valores del cloro residual y nivel de turbidez. La siguiente figura se muestra su configuración.



Hoja de cloro ☆ 📄 ☁

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda [Última modificación hace unos segundos](#)

100% € % ,0 .00 123 Predetermi... 10 B I A

|   | A                   | B                    | C                       | D | E | F |
|---|---------------------|----------------------|-------------------------|---|---|---|
| 1 | Marca temporal      | Valor Cloro Residual | Valor Nivel de Turbidez |   |   |   |
| 2 | 18/07/2022 10:50:49 | 1,11                 | 0                       |   |   |   |
| 3 | 18/07/2022 11:20:49 | 1,1                  | 0,1                     |   |   |   |
| 4 | 18/07/2022 11:50:49 | 1,01                 | 0                       |   |   |   |
| 5 | 18/07/2022 12:20:49 | 0,95                 | 0                       |   |   |   |
| 6 | 18/07/2022 12:50:49 | 0,93                 | 0                       |   |   |   |
| 7 | 18/07/2022 13:20:49 | 0,94                 | 0                       |   |   |   |
| 8 | 18/07/2022 13:50:49 | 0,95                 | 0                       |   |   |   |

## Apéndice E. Sensor de cloro residual

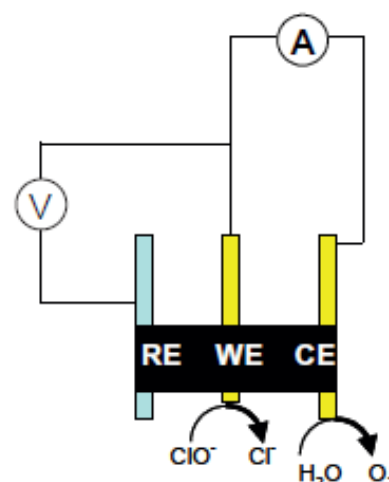
El análisis amperométrico consiste en la medida de la intensidad de corriente generada en la reacción de reducción u oxidación de un analito al aplicar un potencial eléctrico adecuado.

En el caso del análisis de cloro libre

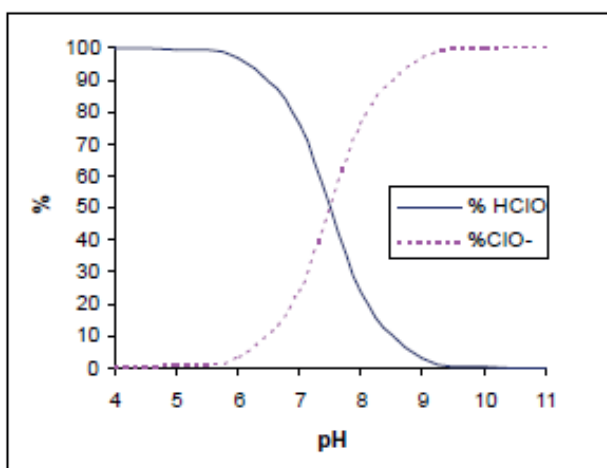


Como se puede deducir de la reacción anterior, la intensidad de corriente es proporcional a la cantidad de ácido hipocloroso presente en la solución.

La reducción del cloro tiene lugar en el electrodo de trabajo de oro (W), en el que se está aplicando el potencial adecuado a partir de la lectura del electrodo de referencia de Ag/AgCl (R). El circuito eléctrico para la lectura de la intensidad se cierra con un electrodo auxiliar (contraelectrodo) de oro (C). Finalmente, como que las intensidades de corriente generadas son muy bajas (del orden de nanoamperios), con el fin de mantener una señal lo más estable posible se incluye un cuarto electrodo, también de oro, que actúa como toma de tierra para eliminar cualquier corriente residual que pueda hallarse presente en el agua.



Conviene tener en cuenta que el ácido hipocloroso es un ácido débil y como tal, la distribución de sus especies depende del pH del medio.



Al potencial de trabajo del sensor amperométrico no sólo da respuesta el ácido hipocloroso, sino que también el hipoclorito reacciona. Por este motivo es recomendable realizar una compensación de la respuesta del sensor en función del valor de pH en el medio dentro del intervalo de pH 6.5 a 9.0. Fuera de estos intervalos de pH existen reacciones parásitas en la superficie del electrodo que impiden la corrección de las lecturas generadas.

## Apéndice F. Sensor de turbidez

Turbidity\_sensor\_SKU\_SEN0169-DFRobot

### Introduction

(<https://www.dfrobot.com/product-1394.html>)

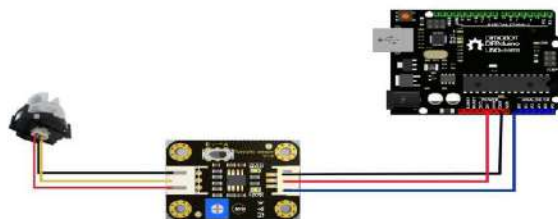
The **gravity arduino turbidity sensor** (<https://www.dfrobot.com/product-1394.html>) detects water quality by measuring the levels of turbidity, or the opaqueness. It uses light to detect suspended particles in water by measuring the light transmittance and scattering rate, which changes with the amount of total suspended solids (TSS) in water. As the TSS increases, the liquid turbidity level increases. Turbidity sensors are used to measure water quality in rivers and streams, wastewater and effluent measurements, control instrumentation for settling ponds, sediment transport research and laboratory measurements. This liquid sensor (<https://www.dfrobot.com/category-68.html>) provides analog and digital signal output modes. The threshold is adjustable when in digital signal mode. You can select the mode according to your MCU.

**NOTE:** The top of probe is not waterproof.

### Specification

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time : <500ms
- Insulation Resistance: 100M (Min)
- Output Method:
  - Analog output: 0-4.5V
  - Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5°C~90°C
- Storage Temperature: -10°C~90°C
- Weight: 30g
- Adapter Dimensions: 38mm\*28mm\*10mm/1.5inches \*1.1inches\*0.4inches

### Connection Diagram



#### Interface Description:

1. "D/A" Output Signal Switch
  - i. "A": Analog Signal Output, the output value will decrease when in liquids with a high turbidity
  - ii. "D": Digital Signal Output, high and low levels, which can be adjusted by the threshold potentiometer
2. Threshold Potentiometer: you can change the trigger condition by adjusting the threshold potentiometer in digital signal mode.

### Examples

Here are two examples: Example 1 uses Analog output mode Example 2 uses Digital output mode

#### Example 1

```
void setup() {
  Serial.begin(9600); //Baud rate: 9600
}
void loop() {
  int sensorValue = analogRead(A0); // read the input on analog pin 0:
  float voltage = sensorValue * (5.0 / 1024.0); // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  Serial.println(voltage); // print out the value you read:
  delay(500);
}
```

#### Example 2

```
int ledPin = 13; // Connect an LED on pin 13, or use the onboard one.
int sensor_in = 2; // Connect turbidity sensor to Digital Pin 2

void setup(){
  pinMode(ledPin, OUTPUT); // Set ledPin to output mode
  pinMode(sensor_in, INPUT); //Set the turbidity sensor pin to input mode
}

void loop(){
  if(digitalRead(sensor_in)==LOW){ //read sensor signal
    digitalWrite(ledPin, HIGH); // if sensor is LOW, then turn on
  }
  else{
    digitalWrite(ledPin, LOW); // if sensor is HIGH, then turn off the led
  }
}
```

This is a reference chart for the mapping from the output voltage to the NTU according to different temperature. e.g. If you leave the sensor in the pure water, that is NTU < 0.5, it should output "4.1±0.3V" when temperature is 10~50°C.



## Apéndice G. Sensor de caudal

### Product Description

FS400A G1 DN25 1inch 1-60L/min 3.5-24V Water Flow Sensor

Waterproof, heat resistance, pressure resistance, cold resistance

Easy to install

Frequency:  $F=4.8 * Q$  (L / Min)

error:  $\pm 2\%$

voltage: 3.5-24VDC, current can not exceed 10mA, 450 output pulses/liters,

$F=\text{constant} * \text{units of flow (L / min)} * \text{time (seconds)}$ .

Flow range: 1-60L/min

Maximum current: 15 mA (DC 5V)

Working voltage range: DC 5~24 V

Load capacity: 10 mA (DC 5V)

Operating Temp: -20~70, Operating humidity: 35%~90%RH

Allow compression: Water pressure >1.20Mpa

Insulation resistance: >100M OHM

Storage Temperature: -25~+80

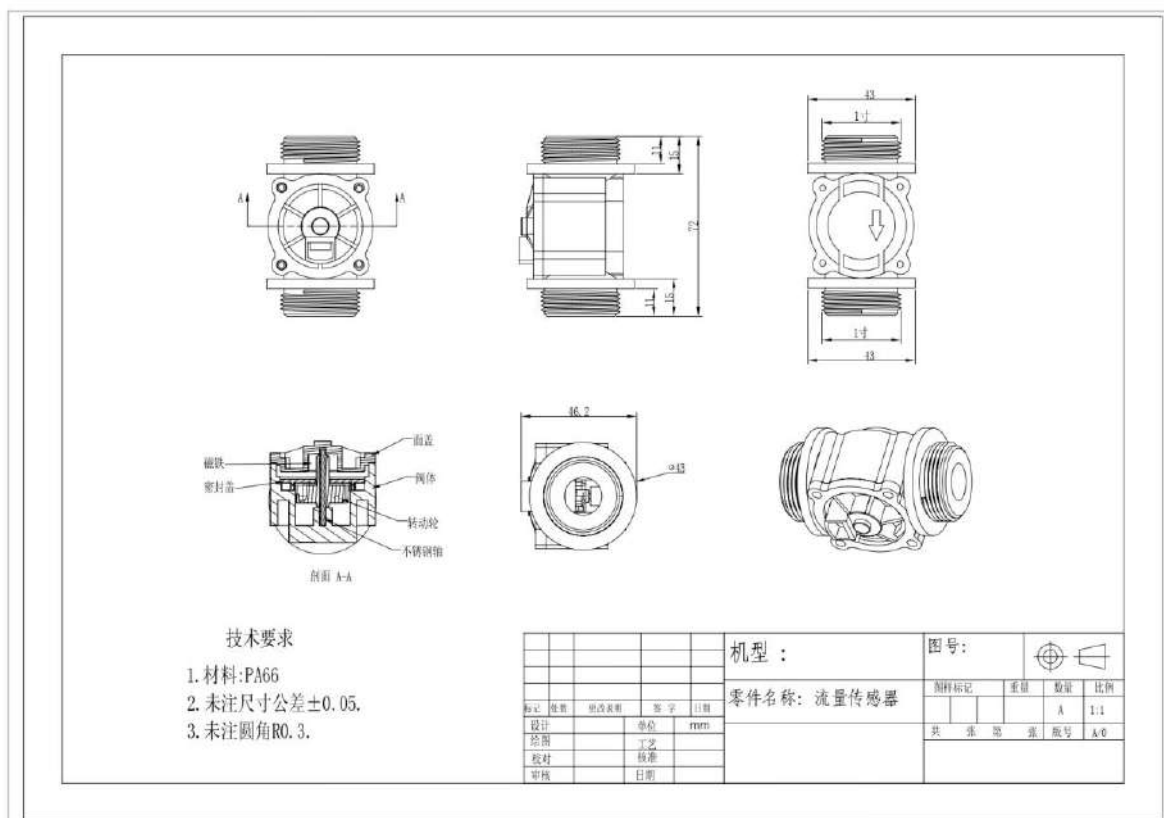
Storage humidity: 25%~95%RH

Output Waveform: Square Wave, output pulse signal

ROHS Compliant

Material: ABS Plastic

Color: Black



## Apéndice H. Diagramas esquemáticos de los dispositivos electrónicos utilizados.

### Microcontrolador ESP32

