



UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Carrera de Electrónica y Telecomunicaciones

Diseño e implementación de un registrador de datos para la adquisición de datos pluviométricos

*Trabajo de titulación previo a la
obtención del título de Ingeniero en
Electrónica y Telecomunicaciones.*

Autor:

Kevin Samuel Piedra Orellana
C.I: 010463317-7
kevinspo94@hotmail.com

Director:

Ing. Darwin Fabián Astudillo Salinas, PhD
C.I:010390703-6

Codirector:

Ing. Luis Geovany Lupercio Novillo, Mgst
C.I: 010236100-3

**Cuenca - Ecuador
7 de octubre de 2020**



Resumen

El Programa para el Manejo del Agua y del Suelo (PROMAS) es un departamento de investigación de la Universidad de Cuenca. Este departamento requiere monitorear ciertos parámetros hidro-meteorológicos utilizando estaciones meteorológicas (alrededor de 130) distribuidas en un área de interés, entre los parámetros se encuentra la cantidad de lluvia. Las mediciones son usadas con fines de investigación en diferentes proyectos [1]. Los proyectos contribuyen al manejo sostenible de los recursos agua y suelo, mediante la generación de conocimiento científico, la capacitación y la extensión.

Las estaciones cuentan con un registrador de datos. El registrador se encarga de la recolección de los parámetros hidro-meteorológicos. Estos datos requieren ser transmitidos/copiados a los servidores del PROMAS. Actualmente, el personal del PROMAS se desplaza hacia las estaciones para descargar los datos *in-situ* almacenados en cada estación. El acceso a estas estaciones resulta complicado debido a que éstas están distribuidas en un área geográficamente extensa y una buena parte de ellas en lugares de difícil acceso[1].

El desarrollo de prototipos para transmitir la información recolectada en *quasi*-tiempo real es parte del proyecto "Aplicación de Tecnologías Inalámbricas al Monitoreo Climatológico en la Cuenca del Río Paute". La tecnología usada para lograr este objetivo es Long Range (LoRa) y el protocolo Long Range Wide Area Network (LoRaWAN)[2]. Entre los prototipos desarrollados está el registrador de datos pluviométricos.

En este trabajo, se presenta el diseño e implementación de un registrador de datos pluviométricos en Printed Circuit Board (PCB) compacto utilizando componentes de montaje superficial con tecnología LoRa. Esta implementación incorpora la adquisición, el almacenamiento y la transmisión alámbrica e inalámbrica de los datos. LoRa es una de las tecnologías de redes de largo alcance y bajo consumo Low Power Wide Area Network (LPWAN) para la aplicación en proyectos de Internet of Things (IoT) y se caracteriza por su sensibilidad, costo accesible y simplicidad de implementación. Para la comunicación de extremo a extremo este se adhiere a el protocolo LoRaWAN, su arquitectura esta integrada por nodos distribuidos en toda el área de interés, y de un dispositivo que es un *gateway* entre los nodos y el servidor de red. El registrador de datos tiene como función la descarga de datos *in-situ* usando tecnología Wireless Fidelity (WiFi), para esto se utiliza un *gateway* que es conectado de manera alámbrica. La arquitectura del sistema es adaptada a 3,3 V en su alimentación, para reducir el consumo energético. Como resultado, de acuerdo a pruebas realizadas, el dispositivo desarrollado presentó un funcionamiento óptimo en la adquisición, transmisión y almacenamiento de datos, así como también un bajo consumo de energía.

Palabras clave : LoRaWAN. LoRa. Nodo. PCB. Comunicación. Prototipo. Transceptor. Protocolo. WiFi.



Abstract

PROMAS is a research department of the University of Cuenca. This department requires monitoring certain hydro-meteorological parameters using meteorological stations (around 130) distributed in an area of interest, among the parameters is the amount of rain. Measurements are used for research purposes in different projects [1]. The projects contribute to the sustainable management of water and soil resources, through the generation of scientific knowledge, training and extension.

The stations have a data logger. The data logger is in charge of collecting the hydro-meteorological parameters. These data require to be transmitted / copied to the **PROMAS** servers. Currently, **PROMAS** personnel travel to the stations to download the *in-situ* data stored at each station. Access to these stations is complicated because they are distributed over a geographically wide area and most of them in places that are difficult to access [1].

The development of prototypes to transmit the collected information in *quasi*-real time is part of the project “Aplicación de Tecnologías Inalámbricas al Monitoreo Climatológico en la Cuenca del Río Paute”. The technology used to achieve this objective is **LoRa** and the **LoRaWAN** protocol [2]. The data logger is one of the prototypes developed.

In this work, the design and implementation of a compact **PCB** pluviometric data logger using surface mount components with **LoRa** technology is presented. This implementation incorporates the acquisition, storage, and wired and wireless transmission of data. **LoRa** is one of the **LPWAN** low consumption and long range network technologies for application in **IoT** projects and is characterized by its sensitivity, affordable cost and simplicity of implementation. For end-to-end communication it adheres to the **LoRaWAN** protocol, its architecture is made up of nodes distributed throughout the area of interest, and a device that is a *gateway* between the nodes and the network server. The data logger has the function of downloading data *in-situ* using **WiFi** technology, for this a *gateway* is used that is wired. The architecture of the system is adapted to 3,3 V in its power supply, to reduce energy consumption. As a result, according to tests carried out, the developed device presented an optimal operation in the acquisition, transmission and storage of data, as well as low energy consumption.

Palabras clave : **LoRaWAN. LoRa. Nodo. PCB. Comunicación. Prototipo. Transceptor. Protocolo. WiFi.**



Índice general

| | |
|---|-------------|
| Resumen | I |
| Abstract | II |
| Índice general | III |
| Índice de figuras | VI |
| Índice de tablas | IX |
| Cláusula de Propiedad Intelectual | X |
| Cláusula de licencia y autorización para publicación en el Repositorio Institucional | XI |
| Certifico | XII |
| Certifico | XIII |
| Dedicatoria | XIV |
| Agradecimientos | XV |
| Abreviaciones y acrónimos | XVI |
| 1. Introducción | 1 |
| 1.1. Identificación del problema | 1 |
| 1.2. Justificación | 2 |
| 1.3. Alcance | 3 |
| 1.4. Objetivos | 3 |
| 1.4.1. Objetivo general | 3 |
| 1.4.2. Objetivos específicos | 3 |
| 2. Marco teórico y estado del arte | 5 |
| 2.1. Redes de sensores inalámbricos | 5 |
| 2.2. Redes de cobertura amplia y baja potencia LPWAN | 6 |
| 2.3. Long range wide area network (LoRaWAN) | 7 |
| 2.3.1. Seguridad de LoRaWAN: Activación de un nodo final | 9 |
| 2.4. Estaciones pluviométricas | 10 |



| | |
|---|-----------|
| 2.5. Protocolos de comunicación | 10 |
| 3. Trabajos Relacionados | 13 |
| 4. Diseño e Implementación | 17 |
| 4.1. Análisis de requerimientos | 17 |
| 4.2. Descripción de los dispositivos utilizados | 18 |
| 4.2.1. Microcontrolador ATmega 1284p | 18 |
| 4.2.2. Transceptor Ra02 LoRa SX1278 | 19 |
| 4.2.3. Sensores | 19 |
| 4.2.4. Reloj en tiempo real (RTC) | 21 |
| 4.2.5. Memoria Externa | 21 |
| 4.3. Comunicación entre dispositivos | 22 |
| 4.4. Aspecto final del esquemático del dispositivo | 23 |
| 4.5. Características de la PCB | 23 |
| 4.6. Aplicación del Protocolo LoRaWAN | 25 |
| 4.6.1. Nodo final LoRa | 26 |
| 4.6.2. LoRa Gateway | 27 |
| 4.6.2.1. Reenviador de paquetes | 28 |
| 4.6.3. The Things Network | 28 |
| 4.7. Almacenamiento de datos | 29 |
| 4.7.1. Estructura de archivo para almacenamiento de datos | 30 |
| 4.8. Descarga de datos <i>in-situ</i> | 30 |
| 4.8.1. Nodo | 30 |
| 4.8.2. Wifi Gateway | 31 |
| 4.8.3. Aplicación Móvil | 31 |
| 4.9. Contenedor de Dispositivos | 33 |
| 4.9.1. Nodo | 33 |
| 4.9.2. WiFi gateway | 34 |
| 5. Resultados | 36 |
| 5.1. PCB | 36 |
| 5.2. Pruebas de Laboratorio | 38 |
| 5.2.1. Transmisión | 38 |
| 5.2.2. Descarga de datos en <i>in-situ</i> | 39 |
| 5.2.3. Configuración de parámetros del dispositivo | 41 |
| 5.3. Pruebas de consumo energético | 46 |
| 6. Conclusiones y Recomendaciones | 48 |
| 6.1. Conclusiones | 48 |
| 6.2. Recomendaciones | 49 |
| 6.3. Trabajos futuros | 49 |



| | |
|---|-----------|
| A. Especificaciones Técnicas de dispositivos | 50 |
| A.1. Microcontrolador ATmega1284p | 50 |
| A.2. Transceptor Ra02 SX1278 | 51 |
| A.2.1. SX1278 | 52 |
| A.3. Sensor DHT11 | 53 |
| A.4. RTC DS3231 | 54 |
| A.5. Memoria externa W25Q128JV | 55 |
| A.6. Modulo WiFi Kit 8 | 55 |
| A.7. Pluviómetro | 56 |
| B. Configuración y programación de dispositivos | 58 |
| B.1. Configuración del registrador de datos | 58 |
| B.2. Configuración del software | 61 |
| B.2.1. Instalación de librerías | 63 |
| B.3. Configuración de LoRaWAN | 67 |
| B.3.1. Nodo | 68 |
| B.3.2. Gateway | 69 |
| B.3.3. TTN | 70 |
| B.4. Configuración descarga datos in-situ | 74 |
| B.5. Código fuente generado para los dispositivos | 79 |
| C. Montaje del prototipo | 80 |
| C.1. Montaje del prototipo en protoboard | 80 |
| C.2. Montaje del registrador de datos para pruebas de laboratorio | 81 |
| D. Diseño esquemático final del dispositivo | 85 |
| Bibliografía | 87 |



Índice de figuras

| | |
|--|----|
| 1.1. Diagrama de bloques del diseño del prototipo[4]. | 2 |
| 2.1. Esquema de una WSN[10]. | 6 |
| 2.2. Velocidad de datos requerida contra capacidad de rango de las tecnologías de comunicación de radio.[11] | 6 |
| 2.3. Ventajas de Sigfox, LoRa y Narrowband-Internet of Things (NB-IoT) en términos de factores de IoT.[11] | 7 |
| 2.4. Arquitectura de la red LoRaWAN[15] | 8 |
| 2.5. Diagrama de Serial Peripheral Interface (SPI) síncrono.[25] | 11 |
| 2.6. Diagrama de conexión de múltiples esclavos independientes en un bus SPI.[25] | 12 |
| 2.7. Diagrama de conexión de múltiples esclavos en cascada.[25] | 12 |
| 3.1. Descripción general del sistema.[5] | 14 |
| 3.2. Gráfico comparativo de la medición de la intensidad de la lluvia entre un registrador de datos comercial y el prototipo.[5] | 15 |
| 3.3. Diagrama de dispersión.[5] | 15 |
| 4.1. Diagrama de bloques del diseño del prototipo.[4] | 18 |
| 4.2. Microcontrolador ATmega 1284p. | 18 |
| 4.3. Transceptor Ra02 LoRa SX1278. | 19 |
| 4.4. Sensor de temperatura y humedad DTH11. | 20 |
| 4.5. Pluviómetro de la estación Davis 6162 Wireless Vantage Pro2 Plus. | 20 |
| 4.6. Conexión del pluviómetro en el prototipo. Se muestra los pines de salidas y el filtro implementado previo a la adquisición de la señal[27]. | 21 |
| 4.7. RTC DS3231. | 21 |
| 4.8. Memoria flash W25Q128JV. | 22 |
| 4.9. Diagrama de bloques del diseño del prototipo. | 23 |
| 4.10. Diseño PCB de la cara superior del dispositivo. | 24 |
| 4.11. Diseño PCB de la cara inferior del dispositivo. | 24 |
| 4.12. Diseño PCB final del dispositivo. | 25 |
| 4.13. Arquitectura de la red LoRaWAN. | 25 |
| 4.14. Nodo final- Registrador de datos Pluviométricos. | 26 |
| 4.15. Proceso de agregación de nodos. | 27 |
| 4.16. Raspberry Pi junto al Ra02 SX1278. | 27 |
| 4.17. Interfaz de la consola de The Things Network (TTN). | 29 |



| | |
|--|----|
| 4.18. Ventana para desarrollar el decodificador en TTN | 29 |
| 4.19. Estructura de trama para el almacenamiento en el archivo del sensor de lluvia. | 30 |
| 4.20. Estructura de trama de datos para el almacenamiento en el archivo de estado. | 30 |
| 4.21. Arquitectura de la red para la descarga de datos <i>in-situ</i> | 30 |
| 4.22. Registrador de datos. | 31 |
| 4.23. Módulo Wifi Kit 8 | 31 |
| 4.24. Aplicación situada en sistema operativo Android. | 32 |
| 4.25. Interfaz de la aplicación móvil. | 33 |
| 4.26. Contenedor del Nodo | 34 |
| 4.27. Contenedor del WiFi gateway | 35 |
| | |
| 5.1. PCB del registrador de datos pluviométrico | 37 |
| 5.2. Visualización de datos en la aplicación del servidor TTN. | 38 |
| 5.3. Visualización de datos recolectados por el sensor en la aplicación del servidor TTN. | 39 |
| 5.4. Visualización de datos del estado del dispositivo en la aplicación del servidor TTN. | 39 |
| 5.5. Descarga de datos del sensor pluviométrico. | 40 |
| 5.6. Descarga de datos de estado del dispositivo. | 41 |
| 5.7. Ventana de configuración de parámetros del nodo. | 42 |
| 5.8. Verificación de la conexión del nodo en la aplicación del servidor TTN. | 43 |
| 5.9. Verificación de los paquetes de datos en la aplicación del servidor TTN. | 43 |
| 5.10. Descarga de datos del dispositivo aplicando la configuración de parámetros(1 y 3 minutos). | 44 |
| 5.11. Ventana de configuración de parámetros del nodo. | 45 |
| 5.12. Verificación de la conexión del nodo en la aplicación del servidor TTN. | 46 |
| | |
| A.1. Diagrama de pines del microcontrolador ATmega1284[39]. | 51 |
| A.2. Diagrama de pines del Ra02 SX1278 [40]. | 51 |
| A.3. Diagrama de bloques general de un transceptor SX1276-78 [41]. | 52 |
| A.4. Diagrama de pines del SX1278 [32]. | 53 |
| A.5. Sensor DHT11 .[42] | 54 |
| A.6. Circuito típico de operación del RTC DS3231[43]. | 55 |
| A.7. Diagrama de pines de la memoria externa[44]. | 55 |
| A.8. Modulo Wifi Kit 8 [45]. | 56 |
| A.9. Pluviómetro ONSET [46]. | 57 |
| | |
| B.1. Configuración para Gestor de URLs Adicionales de tarjetas. | 59 |
| B.2. Instalación de tarjetas de MightyCore. | 59 |
| B.3. Interconexión entre <i>master</i> y <i>slave</i> | 60 |
| B.4. Interconexión entre <i>master</i> y <i>slave</i> | 60 |
| B.5. Selección de características para el ATmega1284P. | 61 |
| B.6. Quemado del bootloader completado. | 61 |
| B.7. Diagrama de flujo del funcionamiento del software del registrador de datos pluviométricos. | 63 |
| B.8. Ventana de opciones para instalar librerías. | 64 |
| B.9. Ventana de instalación de librería RTCLib. | 64 |
| B.10. Ventana de instalación de librería DTH. | 65 |



| | |
|--|----|
| B.11. Ventana de instalación de librería SerialFlash. | 65 |
| B.12. Configuración para Gestor de URLs Adicionales de tarjetas. | 66 |
| B.13. Ventana de opciones para instalar Tarjetas. | 66 |
| B.14. Ventana de instalación de librería ESP8266. | 67 |
| B.15. Ventana de instalación de librería Lmic. | 67 |
| B.16. Configuración de claves para unirse a la red LoRaWAN | 68 |
| B.17. Función de mapeo de pines para el módulo LoRa | 68 |
| B.18. Definir la frecuencia central para que el nodo del sensor. | 68 |
| B.19. Salida de monitor en serie del nodo. | 69 |
| B.20. Salida de consola para el reenviador de paquetes de un solo canal. | 69 |
| B.21. Ventana para el registro del gateway. | 70 |
| B.22. Ventana para el registro del gateway. | 70 |
| B.23. Ventana de información del gateway. | 71 |
| B.24. Ventana para el registro del dispositivo. | 71 |
| B.25. Formulario de registro del dispositivo. | 72 |
| B.26. Establecer el método de activación Activation by Personalization (ABP) | 72 |
| B.27. Ventana de información del nodo. | 73 |
| B.28. Función del decodificador. | 73 |
| B.29. Datos entrantes de estado del dispositivo capturados por la aplicación. | 74 |
| B.30. Datos entrantes del sensor pluviométrico del dispositivo capturados por la aplicación. | 74 |
| B.31. Configuración del tamaño y tiempo de las muestras. | 75 |
| B.32. Configuración de la comunicación serial. | 75 |
| B.33. Configuración de parámetros del gateway WiFi | 76 |
| B.34. Configuración de la red WiFi en el dispositivo móvil. | 76 |
| B.35. Aplicación situada en el menú del dispositivo móvil. | 77 |
| B.36. Interfaz de la aplicación móvil con conexión exitosa hacia el <i>gateway</i> | 78 |
| B.37. Interfaz de aplicación móvil sin conexión al <i>gateway</i> | 78 |
| | |
| C.1. Montaje del circuito en <i>protoboard</i> sin sensor pluviométrico. | 80 |
| C.2. Montaje del circuito en <i>protoboard</i> con sensor pluviométrico. | 81 |
| C.3. Pruebas de funcionamiento del circuito en <i>protoboard</i> | 81 |
| C.4. Nodo final instalado en su contenedor. | 82 |
| C.5. Gateway LoRa en funcionamiento. | 82 |
| C.6. Conexión del nodo y el sensor pluviométrico. | 83 |
| C.7. Montaje final para pruebas de funcionamiento del nodo | 83 |
| C.8. Conexión del <i>gateway WiFi</i> al nodo para la descarga de datos <i>in-situ</i> | 84 |
| | |
| D.1. Diseño del esquemático. | 86 |



Índice de tablas

| | |
|--|----|
| 4.1. Nomenclatura utilizada para la comunicación ATmega1284p y SX1278. | 22 |
| 4.2. Nomenclatura utilizada para la comunicación entre ATmega1284 y W25Q128JV. | 22 |
| 4.3. Nomenclatura utilizada para la comunicación entre ATmega1284 y RTC DS3231. | 23 |
| 4.4. Nomenclatura utilizada para la comunicación entre Raspberry Pi y Ra02 SX1278. | 28 |
| 5.1. Estimación del consumo de energía del registrador de datos. | 47 |



Cláusula de Propiedad Intelectual

Yo, Kevin Samuel Piedra Orellana, autor del trabajo de titulación “Diseño e implementación de un registrador de datos para la adquisición de datos pluviométricos”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 7 de octubre de 2020

A handwritten signature in black ink, reading "Kevin Piedra Orellana", written over a horizontal line.

Kevin Samuel Piedra Orellana
010463317-7



Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Kevin Samuel Piedra Orellana en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación “Diseño e implementación de un registrador de datos para la adquisición de datos pluviométricos”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 7 de octubre de 2020

Kevin Samuel Piedra Orellana

010463317-7



Certifico

Que el presente proyecto de tesis: Diseño e implementación de un registrador de datos para la adquisición de datos pluviométricos, fue dirigido y revisado por mi persona.

A handwritten signature in blue ink, consisting of stylized, overlapping loops and a long horizontal stroke extending to the right.

Ing. Darwin Fabián Astudillo Salinas, PhD
Director



Certifico

Que el presente proyecto de tesis: Diseño e implementación de un registrador de datos para la adquisición de datos pluviométricos, fue dirigido y revisado por mi persona.

A handwritten signature in blue ink, consisting of several loops and a horizontal line at the bottom.

Ing. Luis Geovany Lupercio Novillo, Mgst
Co-director



Dedicatoria

Este trabajo se lo dedico a Dios, por todas las bendiciones y fuerzas que me da día tras día.

A mis padres, que han sido el motor durante toda mi vida, siempre con su amor y apoyo incondicional, sin ellos no hubiese tenido las fuerzas necesarias para la culminación de esta etapa de mi vida.

A mi hermano, que siempre me ha dado su apoyo incondicional y motivación en todo momento, además de guiarme siempre por un buen camino.

A toda mi familia y amigos, que siempre han estado en las buenas y en las malas brindándome su apoyo y cariño.

Kevin Samuel Piedra Orellana



Agradecimientos

El camino para llegar a culminar esta etapa de mi vida, fue muy difícil de lograrlo, pero al final se pudo cumplir, esto gracias a los esfuerzos realizados por cada uno de ustedes. Por eso, les agradezco de todo corazón por todo su esfuerzo puesto en este proceso. El afecto que les tengo es infinito y lo voy a llevar guardado durante toda la vida.

A todos los profesores por sembrar su conocimiento y sabiduría en cada etapa de nuestra vida universitaria, pero en especial al Ing. Fabián Astudillo, PhD y al Ing. Geovany, MsC por el apoyo, motivación y ayuda en el desarrollo de esta tesis.

A mis familiares, en especial a mis padres y hermano que siempre me brindaron su apoyo incondicional en todo momento, siendo el pilar fundamental en mi vida.

A mis amigos, por hacer que esta etapa de mi vida sea la mejor, siempre contando con su apoyo y teniendo experiencias inolvidables.

Al Ing. Sebastián Placencia por su asesoría y ayuda técnica.

Kevin Samuel Piedra Orellana



Abreviaciones y Acrónimos

ABP Activation by Personalization. 9, 26, 29, 33, 68, 72

AES Advanced Encryption Standard. 9

AppSKey Application Session Key. 9, 10, 26, 44, 68, 72

BW Bandwidth. 7

CF Carrier Frequency. 7

CMOS Complementary Metal Oxide Semiconductor. 12

CPU Central Processing Unit. 50

CR Coding Rate. 7, 38

CRC Cyclic Redundancy Check. 52

CSS Chirp Spread Spectrum modulation. 7

DevAddr Device Address. 9, 26, 44, 68, 72

DR Data Rate. 38

EMA Estación Meteorológica Automática. 10

GPRS General Packet Radio Service. 5, 14

GPS Global Positioning System. 13

HTTP Hypertext Transfer Protocol. 31

I2C Inter-Integrated Circuit. 10, 12, 21–23, 62

iOS iPhone Operating System. 49

IoT Internet of Things. 6, 7, 16

IP Internet Protocol. 8, 28

IP65 Ingress Protection 65. 34

IP67 Ingress Protection 67. 3, 30, 34

ISM Industrial, Scientific and Medical. 14, 16

Li-Po Litio-Polímero. 31

LoRa Long Range. 2, 3, 5–7, 13, 14, 16, 18, 19, 22, 23, 25–28, 38, 46, 51, 62, 68–70, 81, 82

LoRaWAN Long Range Wide Area Network. 2–5, 7–9, 13, 15–18, 25–28, 33, 38, 48, 49, 61, 62, 67–69

LPWAN Low Power Wide Area Network. 3, 5–7

MAC Media Access Control. 8, 9

MCU Microcontroller Unit. 58



- MIPS** Microprocessor without Interlocked Pipeline Stages. 50
- MISO** Master Input Slave Output. 11, 12, 19, 49
- MOSI** Master Output Slave Input. 11, 12, 19
- MPPT** Maximum Power Point Tracker. 14
- NB-IoT** Narrowband-Internet of Things. 6, 7
- NwkSKey** Network Session Key. 9, 10, 26, 44, 68, 72
- OTAA** Over The Air Activation. 9
- OTP** One Time Password. 55
- PCB** Printed Circuit Board. 2, 3, 17, 23–26, 34, 36, 37, 49, 80
- PROMAS** Programa para el Manejo del Agua y del Suelo. 1–4, 10, 46, 48
- RF** Radio Frecuencia. 2, 3, 19, 23, 34
- RISC** Reduced Instruction Set Computer. 50
- RSSI** Received Signal Strength Indicator. 38
- RTC** Real-Time Clock. 13, 18, 21, 23, 54, 61, 62, 64
- RTOS** Real-Time Operating System. 13
- SCK** Serial Clock. 11, 19
- SCL** Serial Clock. 11
- SD** Secure Digital. 13
- SDA** Serial Data. 11
- SF** Spreading Factor. 7, 16
- SNR** Signal to Noise Ratio. 7, 38
- SPI** Serial Peripheral Interface. 3, 11, 12, 19, 22, 23, 27, 49, 58, 59, 61, 62, 65, 68
- SS** Slave Select. 11, 19
- TP** Transmission Power. 7
- TTL** Transistor Transistor Logic. 12
- TTN** The Things Network. 28, 29, 38, 39, 41–46, 48, 68–71, 73
- UART** Universal Asynchronous Receiver Transmitter. 30, 35, 62, 75
- UDP** User Datagram Protocol. 8
- UV** Radiación Ultravioleta. 10
- WiFi** Wireless Fidelity. 3, 13, 25, 30, 31, 34, 35, 48, 65, 74–76, 81, 84
- WM Bus** Wireless Meter Bus. 15
- WSN** Wireless Sensor Network. 5, 14, 15



Introducción

Este capítulo presenta la identificación del problema (Sección 1.1), justificación (Sección 1.2), alcance (Sección 1.3) y los objetivos del presente proyecto (Sección 1.4).

1.1. Identificación del problema

El **PROMAS** es un departamento de investigación de la Universidad de Cuenca. Este departamento tiene como misión contribuir al “manejo sostenible de los recursos agua y suelo, mediante la generación de conocimiento científico, la capacitación y la extensión; aplicando nuevas tecnologías y adaptando metodologías que permitan dar soluciones integrales, en equilibrio ambiental y con responsabilidad social”. El **PROMAS** requiere monitorear ciertos parámetros utilizando estaciones meteorológicas (alrededor de 130) distribuidas en un área de interés, geográficamente extensa, que comprende desde el sector del CAJAS hasta la provincia del Cañar. Las estaciones miden varios parámetros hidro-meteorológicos, entre ellos la cantidad de lluvia. Las mediciones son usadas con fines de investigación en diferentes proyectos [1].

Las estaciones cuentan con un registrador de datos. Este realiza la adquisición de datos de los parámetros hidro-meteorológicos. La adquisición de datos consiste en tomar la información del fenómeno físico a través de un dispositivo (transductor). Este dispositivo permite convertir las variables del fenómeno físico (cantidad de lluvia por metro cuadrado) en señales eléctricas (señales analógicas), para luego ser tratadas por equipos electrónicos que las convertirán en señales digitales [3]. Estas señales digitales debidamente codificadas constituyen los datos que el registrador se encargará de almacenar, gestionar y transmitir hasta la oficina central. Los datos pluviométricos brindarán la posibilidad de disponer de información para evaluar crecidas, calcular balances de agua de los ríos, etc.

Los datos recolectados por los registradores de datos requieren ser transmitidos/copiados a los servidores del **PROMAS**. Se tienen dos opciones para la copia/extracción de los datos: i) Transmisión manual, en donde una persona visita cada estación, copia los datos desde los registradores de datos, y los lleva al **PROMAS** utilizando cualquier medio de almacenamiento. ii) Transmisión inalámbrica, en la cual la transmisión se realiza a través de medios de comunicación no guiados (microonda, satélite, etc)[1].

Actualmente, la recolección de datos de las estaciones pluviométricas se la realiza *in-situ* para la mayoría de estaciones, personal del **PROMAS** se desplaza a las estaciones para copiar los datos almacenados en cada

estación. La periodicidad de visita a las estaciones es de entre 30 a 45 días. El acceso a estas estaciones resulta complicado debido a que éstas están distribuidas en un área geográficamente extensa y una buena parte de ellas en lugares de difícil acceso[1].

1.2. Justificación

El proyecto “Aplicación de Tecnologías Inalámbricas al Monitoreo Climatológico en la Cuenca del Río Paute” ha venido desarrollando acciones para permitir que las estaciones pertenecientes al PROMAS transmitan la información recolectada en *quasi*-tiempo real. El proyecto contempla que los datos de las estaciones sean transmitidos cada 5 minutos. La tecnología usada para lograr este objetivo es LoRa y el protocolo LoRaWAN[2]. Desde marzo del presente año (2020) se cuenta con la implementación del primer prototipo en el sector de loma larga, Cuenca.

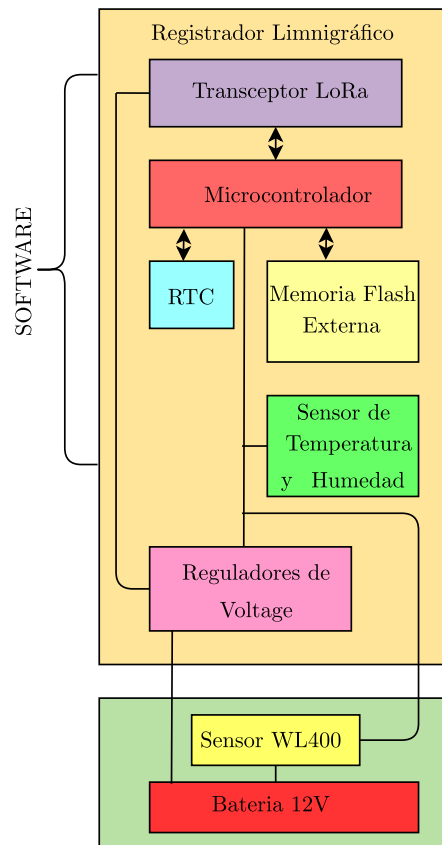


Figura 1.1: Diagrama de bloques del diseño del prototipo[4].

Dentro de las actividades del proyecto antes mencionado, se realizaron varios prototipos para diferentes tipos de estaciones. Entre estos prototipos está el registrador de datos pluviométricos. El artículo “*Rainfall intensity datalogger system. LoPy4-based design and implementation*” detalla la implementación de este registrador de datos [5]. En la tesis “Diseño e implementación de un registrador de datos para la adquisición de información limnigráfica” se construyó un registrador de datos limnigráficos [4]. El objetivo de este trabajo fue experimentar la factibilidad de implementar el circuito Radio Frecuencia (RF) usando los servicios de construcción y ensamblaje de PCBs disponibles localmente, con el fin de abaratar los costos en el producto final. Si bien en el trabajo se



logra el objetivo, se menciona que se tuvieron varias complicaciones a la hora de implementarlo. En nuestro medio no hay empresas que provean servicios de soldado de componentes de montaje superficial automático. El montaje de los componentes fue a mano y se produjeron varias imprecisiones a la hora de soldarlos, como por ejemplo en la unión de dos pines por una pequeña cantidad de suelda. En el diseño se ocupó un controlador de carga con salida de 5V; además se contemplaba el uso de un *stepdown*; sin embargo, debido a la interferencia con la parte RF se usó un regulador de 3.3V que es menos eficiente que el *stepdown* (como se muestra en la Figura 1.1). Con esta experiencia, la autora recomendó usar dispositivos LoRa con los componentes RF integrados. Además, la autora identificó mejoras que se pueden implementar en futuras versiones; entre ellas está el reducir el consumo de energía, y la descarga de datos *in-situ*. La implementación de estas recomendaciones son parte de los objetivos del proyecto.

1.3. Alcance

En el presente proyecto se propone realizar un prototipo de registrador de datos en PCB compacto utilizando componentes de montaje superficial complementado con un módulo LoRa SX1278 embebido, optimizando el tamaño, consumo energético, transmisión de largo alcance (al menos 1 Km [6]) y descarga de datos en *quasi*-tiempo real (5 min según el requerimiento del PROMAS). El registrador de datos tendrá como función la descarga de datos *in situ* usando tecnología WiFi, para esto se utilizará un *gateway* que será conectado de manera alámbrica usando una conexión serial SPI con el conector serial tipo Ingress Protection 67 (IP67). La otra función es la transmisión remota de información usando el protocolo LoRaWAN; el módulo a utilizar es el RF SX1278; este está diseñado para aplicaciones de baja potencia y de red de área amplia LPWAN, este módulo tiene la etapa RF integrada y encapsulada, por lo que es más fácil de soldar, y los costos van a ser menores [7]. En el diseño se contempla el microcontrolador ATmega1284. Para reducir el consumo energético se va a cambiar la arquitectura del sistema. En el registrador de datos se alimenta directamente a 3.3V con una batería externa o controlador de carga. Este trabajo contribuye al objetivo 1, del proyecto “Aplicación de tecnologías inalámbricas al sistema de predicción de caudales en la cuenca del río Tomebamba” [1], ganador del XV concurso de proyectos de la DIUC.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar e implementar un dispositivo registrador de datos compacto usando componentes de montaje superficial para la adquisición de datos pluviométricos y su transmisión en *quasi*-tiempo real usando tecnología LoRa.

1.4.2. Objetivos específicos

El presente trabajo tiene los siguientes objetivos específicos:

- Diseñar y construir una placa PCB con componentes de montaje superficial para el registrador de datos pluviométricos.
- Diseñar y construir una placa PCB con componentes de montaje superficial para el dispositivo *gateway* de descarga *in situ*.



- Implementar el protocolo [LoRaWAN](#) para el microcontrolador ATMEL.
- Implementar el software necesario en el *gateway* para la descarga de datos *in situ*.
- Implementar la aplicación para móviles Android para la descarga de datos *in situ*.
- Realizar pruebas de adquisición de datos y transmisión en *quasi*-tiempo real con el dispositivo registrador de datos prototipo hacia el servidor del [PROMAS](#).
- Realizar pruebas con el dispositivo *gateway* para la descarga de datos *in-situ*.



Marco teórico y estado del arte

En este capítulo se presenta una revisión teórica de los conceptos relacionados con el desarrollo del trabajo. Se inicia describiendo los aspectos generales de las [Wireless Sensor Network \(WSN\)](#) (Sección 2.1) y [LPWAN](#) (Sección 2.2). Luego se menciona de manera más detallada las características y funcionamiento de la tecnología [LoRa](#) y de su protocolo [LoRaWAN](#) (Sección 2.3). Después, se aborda a las estaciones pluviométricas y al sensor que mide la cantidad de lluvia en un determinado tiempo (Sección 2.4). Y finalmente, se describe la comunicación a usarse dentro del dispositivo entre los diferentes componentes (Sección 2.5).

2.1. Redes de sensores inalámbricos

Las [WSN](#) se basan en dispositivos de bajo costo y bajo consumo. Estos dispositivos son conocidos como nodos. Están diseñados para obtener información de su entorno y transmitirla a través de enlaces inalámbricos hasta un nodo central de coordinación llamado *gateway* o nodo *sink*. El *gateway* recepta la información sensada por los nodos y la retransmite usando otro tipo de tecnología, por ejemplo [General Packet Radio Service \(GPRS\)](#), hacia un servidor generalmente ubicado en una red externa[8].

Los nodos son capaces de formar una red en la que es posible la retransmisión de mensajes para llegar desde los nodos más lejanos hasta el *gateway*. Las [WSN](#) se caracterizan por tener capacidades de auto-diagnóstico, auto-configuración, auto-organización y reparación [9]. La red de sensores usualmente está compuesta por: nodo sensor, *gateway*, servidor de red y usuario (aplicación), ver Figura 2.1.

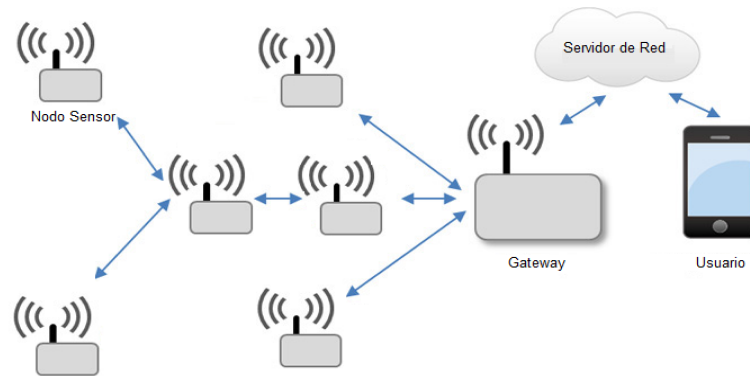


Figura 2.1: Esquema de una WSN[10].

2.2. Redes de cobertura amplia y baja potencia LPWAN

Las **LPWAN** son una tecnología que permite una comunicación de bajo consumo, largo alcance y bajo costo. Su alcance teórico es de 40 km en zonas rurales y de 5 km en zonas urbanas [11]. Dicha tecnología permite aplicaciones **IoT** en donde se requiere la transmisión de pequeñas cantidades de datos a gran distancia. La Figura 2.2 muestra la comparación de la tasa de datos en relación al rango de cobertura de las tecnologías existentes en la actualidad.

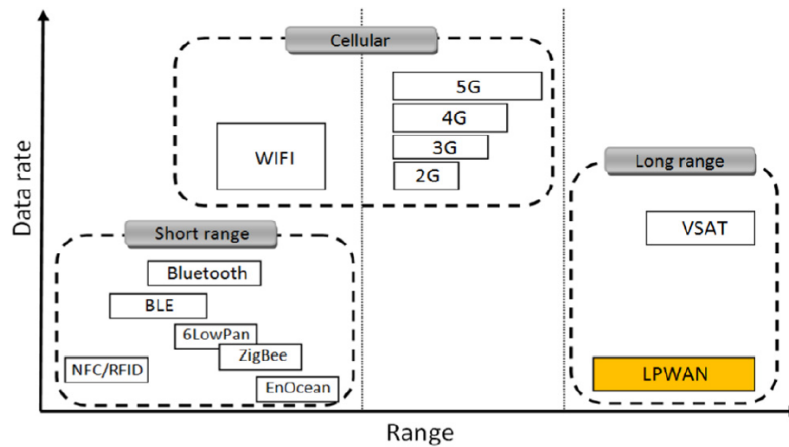


Figura 2.2: Velocidad de datos requerida contra capacidad de rango de las tecnologías de comunicación de radio.[11]

Las soluciones **LPWAN** ofrecen un gran rango de transmisión gracias a un diseño de capa física que permite una alta sensibilidad del receptor. La sensibilidad típica de un receptor **LPWAN** es de alrededor de -150 dBm. La desventaja de estas conexiones de largo alcance es la baja tasa de datos, la cual oscila generalmente entre algunos cientos y unos pocos miles de bits por segundo, significativamente más baja que las velocidades de bits admitidas por las tecnologías de corto alcance (250 kb/s en ZigBee y 1 – 2 Mb/s en Bluetooth)[11]. Tecnologías **LPWAN** como Sigfox, **LoRa** y **NB-IoT** han surgido tanto en frecuencias licenciada como en no licenciadas. La Figura 2.3 presenta las ventajas en factores de **IoT** como la escalabilidad, costo, cobertura, rango, latencia, longitud de carga útil, calidad de servicio, despliegue y consumo de energía, de las tecnologías Sigfox, **LoRa** y **NB-IoT**.

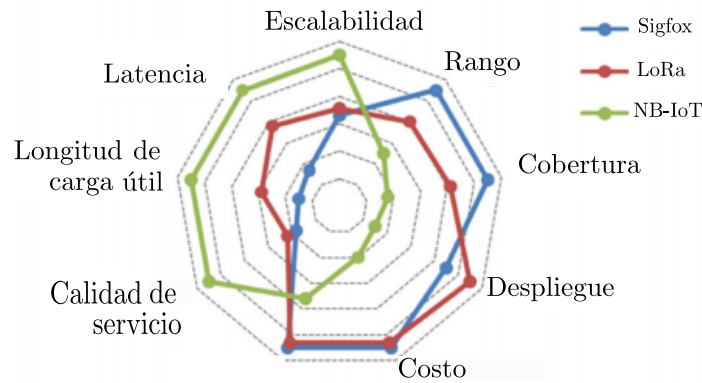


Figura 2.3: Ventajas de Sigfox, LoRa y NB-IoT en términos de factores de IoT.[11]

La tecnología seleccionada para el presente trabajo es LoRa. La gran sensibilidad de esta tecnología se alcanza gracias a la modulación LoRa. LoRa fue desarrollada por Semtech; es una técnica de modulación de capa física basada en Chirp Spread Spectrum modulation (CSS) con corrección de errores hacia adelante. Esto hace que las señales sean robustas ante interferencias y ruido de canal[12]. LoRa utiliza tres parámetros que afectan principalmente a la robustez y la tasa de bits: el Spreading Factor (SF), el Bandwidth (BW) y el Coding Rate (CR).

Un equipo de transmisión LoRa proporciona cinco parámetros de configuración: Transmission Power (TP), Carrier Frequency (CF), SF, BW y CR. La combinación de todos estos parámetros determinan el rango de transmisión y el consumo de energía. A continuación se describe cada uno de estos parámetros:

- **Potencia de transmisión (transmission power, TP):** Se puede ajustar de 2 dBm a 20 dBm en pasos de 1 dB.
- **Frecuencia portadora (carrier frequency, CF):** Varía entre 137 MHz y 525 MHz para el dispositivo SX1278.
- **Factor de ensanchamiento (spreading factor, SF):** Se conoce como SF a la relación entre la tasa de símbolos y la velocidad de chips. Mientras más alto el SF el Signal to Noise Ratio (SNR) aumenta; por lo tanto el alcance aumenta. Un aumento de SF reduce la tasa de transmisión por lo que la duración de la transmisión crece al igual que el consumo energético. El SF varía de 6 a 12.
- **Ancho de Banda (bandwidth, BW):** Un mayor BW proporciona una mayor tasa de datos con una sensibilidad menor. Hay que tomar en cuenta que los datos se envían a una velocidad de chip igual al ancho de banda. Una red LoRa opera con anchos de banda de 500 kHz, 250 kHz o 125 kHz.
- **Tasa de codificación (code rate, CR):** Es la tasa de error hacia adelante que protege contra ráfagas de interferencia, mientras más alta sea la CR, existe mayor protección; sin embargo, aumenta el tiempo en el aire. Se puede configurar en 4/5, 4/6, 4/7 o 4/8.

2.3. Long range wide area network (LoRaWAN)

LoRaWAN es un protocolo que utiliza LoRa, que incluye funciones MAC de estándar abierto destinada a la creación de redes LPWAN. Este protocolo posee comunicación bidireccional de forma inalámbrica con seguridad de extremo a extremo. Tiene influencia directa sobre varios parámetros de rendimiento, que incluyen: la vida útil de la batería de un nodo, la capacidad y seguridad de la red[13].

La arquitectura de red **LoRaWAN** usa una topología en estrella, donde los nodos finales solo pueden comunicarse con el *gateway* y no directamente entre sí. Múltiples *gateways* están conectados a un servidor de red central. Los *gateways* solo son responsables de reenviar paquetes de datos sin procesar desde los nodos finales hacia el servidor de red encapsulándolos en paquetes **User Datagram Protocol (UDP)/Internet Protocol (IP)**. El servidor de red es responsable de enviar paquetes de enlace descendente y comandos **Media Access Control (MAC)** hacia dispositivos finales, si es necesario. Además, la comunicación termina en los servidores de aplicaciones. Se pueden conectar varias capas de aplicación a un único servidor de red [14]. La Figura 2.4 muestra una visión general de la arquitectura de una red **LoRaWAN**.

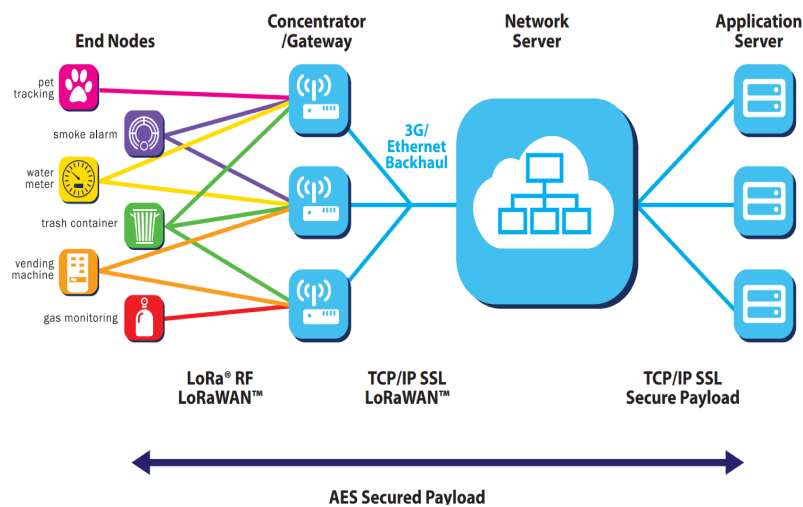


Figura 2.4: Arquitectura de la red **LoRaWAN** [15]

Los componentes que forman parte de la red son:

- **Nodos finales:** Adquieren los datos meteorológicos, empaquetan y transmiten las variables meteorológicas.
- **Gateway:** Permite la comunicación entre los nodos y el servidor.
- **Servidor:** Administra la seguridad y la recepción de datos.
- **Servidor de aplicaciones:** Decodifica y visualiza los datos.

LoRaWAN administra la tasa de datos y la salida de radio frecuencia para cada dispositivo final. A este proceso se lo conoce como un esquema de tasa de datos adaptable. Gracias a la adaptación de la tasa de datos se maximiza la duración de la batería de los nodos finales y la capacidad general de la red.

LoRaWAN utiliza tres clases de **MAC** para dispositivos finales, con el propósito de optimizar la latencia de comunicación de enlace descendente de la red frente a la duración de la batería, y acoplarse a los perfiles requeridos por las aplicaciones finales [16]. El estándar **LoRaWAN** define las siguientes clases: La Clase A (Dispositivos finales bidireccionales), Clase B (Dispositivos finales bidireccionales con ranuras de recepción programadas) y Clase C (Dispositivos finales bidireccionales con ranuras de recepción máxima). La clase que va a ser utilizada en este trabajo es la Clase A, la cual tiene las siguientes características:

- Tiene menor consumo de energía que las clases B y C.
- Se basa en el protocolo *ALOHA*,
- El nodo final planifica la transmisión.



- Está diseñada para aplicaciones que sólo requieren comunicación de enlace ascendente hacia el servidor, una ventana de enlace descendente se abre poco después de que el dispositivo final haya realizado una transmisión.
- Las comunicaciones de enlace descendente desde el servidor tienen que esperar hasta el siguiente enlace ascendente programado.

2.3.1. Seguridad de LoRaWAN: Activación de un nodo final

LoRaWAN incluye dos niveles de seguridad uno para la red y otro para la aplicación. En la capa de seguridad de la aplicación se garantiza que el operador no tenga acceso a los datos de la aplicación del usuario final. El algoritmo de cifrado utilizado es [Advanced Encryption Standard \(AES\)-128bits](#) [17]. Después de la activación del nodo final se almacena la siguiente información:

- *DevAddr*: Dirección lógica (equivalente a una dirección IP) que se utilizará para toda comunicación posterior con la red e identifica el nodo final dentro de la red actual.
- *NwkSKey*: Clave de cifrado entre el *gateway* y el nodo final para cifrar y descifrar el campo de carga útil de mensajes de datos solo para *MAC*, además se usa para garantizar la integridad de los datos.
- *AppSKey*: Clave de cifrado entre el dispositivo y el operador (a través de la aplicación), utilizada para cifrar y descifrar el campo de carga útil de los mensajes de datos específicos de la aplicación.
- *ApplicationEUI*: Es un ID de aplicación global que identifica de forma única al proveedor de la aplicación.
- *DeviceEUI*: Es un ID de dispositivo global que identifica de forma única al nodo final.

El proceso de activación de un nodo se lleva a cabo a través de un mecanismo de autenticación mutua entre el nodo y el servidor de red. Esto se realiza mediante el siguiente procedimiento:

- El nodo y el servidor comparten una llave [Application Session Key \(AppSKey\)](#).
- El nodo utiliza la [AppSKey](#) para calcular el código de integridad de mensaje, para hacer una petición de conexión a la red.
- El servidor comprueba la autenticidad del mensaje verificando la [AppSKey](#).
- El servidor responde con un mensaje de aceptación de cifrado con la misma [AppSKey](#), de esta forma el nodo puede comprobar la autenticidad del mensaje.

LoRaWAN establece un mecanismo denominado registro y activación mediante el cual se autoriza a un nodo a unirse a una red. Hay dos formas de realizar el proceso de registro y activación de los nodos: [Over The Air Activation \(OTAA\)](#) y [ABP](#). El método de activación usado en el presente trabajo por sus características es el [ABP](#)[18].

El modo [ABP](#) es el modo más sencillo de conexión, vincula directamente un dispositivo final a una red específica, al eludir el procedimiento de aceptación. La activación por personalización significa que [Device Address \(DevAddr\)](#) y las dos claves de sesión [Network Session Key \(NwkSKey\)](#) y [AppSKey](#) se almacenan directamente en el nodo final, entonces este se encuentra listo con la información necesaria para formar parte de una red [LoRaWAN](#) específica[19]. La conexión se realiza de la siguiente manera:

1. El dispositivo envía datos hacia al *gateway*.
2. El *gateway* válida que los datos corresponda a la sesión.
3. Si la sesión es correcta se procesan los datos, sino se rechazan.

Entre las características más importantes de este modo, tenemos las siguientes:



- Este procedimiento añade el dispositivo directamente a la red sin la necesidad de los mensajes de solicitud y aceptación de la red.
- Esta forma de activación requiere que se almacene de manera manual las claves la DevAddr y dos claves de sesión **NwkSKey** y **AppSKey** antes de activar el nodo.
- En cada dispositivo se tiene, **NwkSKey** y **AppSKey** únicas con el propósito de no comprometer la seguridad de la comunicación de otros dispositivos.
- La desventaja es que al encontrarse la llave de encriptación en el dispositivo pudiera ser extraída y clonada por un atacante[19].

2.4. Estaciones pluviométricas

Una **Estación Meteorológica Automática (EMA)** es una “estación meteorológica en la que se realizan y se transmiten observaciones automáticamente”[20]. Es decir, no se requiere de una persona para tomar medidas y registrar los datos. Existen dos tipos: en tiempo real y fuera de línea. Las del primer tipo proporcionan datos de lo que ocurre en el momento. Por otro lado, las del segundo tipo registran los datos en dispositivos internos o externos y deben ser enviados o recolectados bajo petición. Se recomienda que las observaciones se realicen siempre a la misma hora del día con el fin de mantener la homogeneidad de los datos [21].

Una **EMA** está compuesta por sensores, un registrador de datos y equipos periféricos: un sistema de energía, un reloj de tiempo real, dispositivos de conexión y visualización, entre otros instrumentos. Los sensores pueden ser analógicos o digitales. Los sensores analógicos básicamente transforman la información en niveles continuos de potencial eléctrico, los digitales codifican la información mediante dos niveles (alto y bajo) y los sensores “inteligentes” cuentan con un propio microcontrolador que procesa y transmite la información al procesador principal.

El **PROMAS** tiene estaciones meteorológicas comerciales con registradores de datos. La estación consta de una consola que permite la descarga de datos y la interacción con el computador mediante el software *Weather-Link*. La estación tiene incluidos los siguientes sensores: pluviómetro, anemómetro, sensor de temperatura y humedad, sensor de radiación solar y sensor de **Radiación Ultravioleta (UV)**. Para el presente trabajo se usará un sensor pluviométrico, el cual ocupa un balancín que activa un interruptor de lengüeta magnético cada vez que cambia de posición al volcarse el agua (*tip*), la resolución es de $0,2mm$ [22].

El sensor pluviométrico realiza la adquisición de variables pluviométricas, que son registradas continuamente en el transcurso de tiempo y en un lugar determinado. Estas variables son de tipo mecánico o electrónico. Las variables pluviométricas permiten que se den alertas de tempestades o inundaciones, de manera que se pueda actuar a tiempo para reducir su impacto [23]. Entre las características de las variables pluviométricas tenemos: el intervalo de actualización (20 a 24 seg), la resolución (0,1 mm), el rango (0 a 1999,9 mm/hr) y la tolerancia (1 mm/hr), que son medidos de forma directa por la estación.

2.5. Protocolos de comunicación

Existen diferentes formas para establecer la comunicación entre dispositivos que ayudan a simplificar e incorporar nuevos módulos o tarjetas integradas. Los protocolos de comunicación más comunes en estos tipos de módulos se detallan a continuación:

- **Comunicación I2C:** Es un protocolo síncrono de comunicación serial, en la que se diferencian dos elementos básicos, un maestro y un esclavo. El protocolo **Inter-Integrated Circuit (I2C)** utiliza sólo dos

vías de comunicación. El **Serial Data (SDA)**, que es la vía de comunicación entre el maestro y esclavo para enviarse información, y el **Serial Clock (SCL)**, que corresponde a la vía por donde viaja la señal de reloj.

- **Comunicación serial:** Es una de las más utilizadas por su simplicidad, confiabilidad y gran acogida por parte de fabricantes para la adaptación de sus dispositivos y módulos con microprocesadores y computadores. Los pines de comunicación son: TXD (Datos Transmitidos) y RXD (Datos Recibidos). Para este protocolo es necesario especificar la velocidad de transferencia de datos en baudios, por ejemplo, a 9600 bps[24].
- **Comunicación SPI:** Para utilizar el protocolo de comunicación **SPI** se ocupa los pines:
 - **Master Output Slave Input (MOSI):** Línea utilizada para transferir los datos del maestro hacia el esclavo.
 - **Master Input Slave Output (MISO):** Línea utilizada para transferir los datos del esclavo hacia el maestro.
 - **Serial Clock (SCK):** Línea encargada de enviar la señal de reloj para sincronizar los dispositivos.
 - **Slave Select (SS):** Línea generada por el maestro y permite seleccionar el dispositivo esclavo para iniciar la comunicación.

El microcontrolador permite conectar varios módulos o circuitos integrados. El protocolo de comunicación **SPI** tiene una arquitectura de tipo maestro-esclavo. El dispositivo maestro (*master*) puede iniciar la comunicación con uno o varios dispositivos esclavos (*slave*), y enviar o recibir datos de ellos. Los dispositivos esclavos no pueden iniciar la comunicación, ni intercambiar datos entre ellos directamente [25]. La comunicación **SPI** de tipo maestro-esclavo tiene tres configuraciones que son:

- **SPI bus síncrono:** El dispositivo maestro proporciona una señal de reloj, que mantiene a todos los dispositivos sincronizados. Esto reduce la complejidad del sistema frente a los sistemas asíncronos.



Figura 2.5: Diagrama de **SPI** síncrono.[25]

- **Conexión de múltiples esclavos independientes en un bus SPI:** En el cual se necesita de hardware adicional o salidas digitales dedicadas y controladas por el dispositivo maestro para seleccionar el dispositivo esclavo conectado al bus.

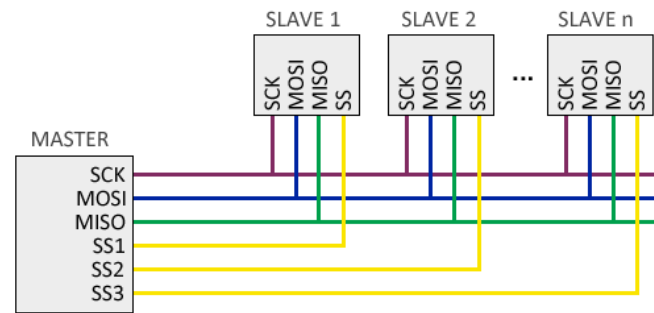


Figura 2.6: Diagrama de conexión de múltiples esclavos independientes en un bus SPI.[25]

- **SPI conexión en cascada:** En la cual el maestro genera una señal que habilita a todos los dispositivos a la vez y las salidas de datos de los esclavos **MISO** se conecta en cascada con la entrada de datos **MOSI** del siguiente esclavo, así el maestro enviará los datos sólo al primer esclavo y recibirá respuesta del último esclavo [25].

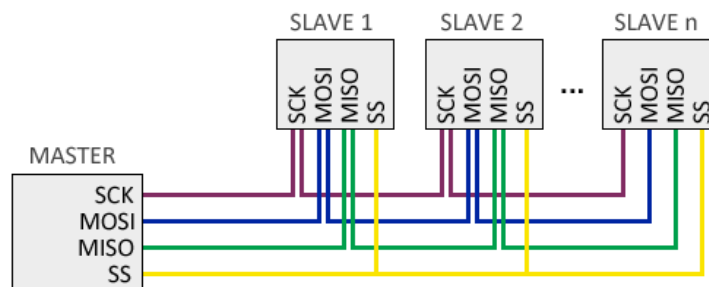


Figura 2.7: Diagrama de conexión de múltiples esclavos en cascada.[25]

- **Protocolo 1-Wire:** El protocolo *1-Wire* es un sistema de bus de comunicación del dispositivo diseñado por Dallas Semiconductor Corp. que proporciona datos de baja velocidad, señalización y alimentación a través de un solo canal cableado. El bus de 1 cable es un sistema digital basado en voltaje que puede funcionar con dos contactos, datos y tierra, para la comunicación bidireccional *semidúplex*. A diferencia de otros sistemas de comunicación en serie, como **I2C** o **SPI**, los dispositivos de 1 cable están diseñados para su uso en un entorno de red de área local con bajas velocidades de datos. Tanto el maestro como los esclavos están configurados como transeptores que permiten que los datos secuenciales de bits fluyan en cualquier dirección, pero solo en una dirección a la vez; el maestro inicia y controla todos los dispositivos. Los datos son secuenciales en bytes y secuenciales en bits con datos leídos y la primera señal de bit menos significativo escrita se transfiere en intervalos de tiempo. No se requiere reloj del sistema; cada parte de 1 cable se sincroniza automáticamente mediante un oscilador interno sincronizado con el borde descendente del maestro. El protocolo *1-Wire* utiliza niveles lógicos **Complementary Metal Oxide Semiconductor (CMOS)/Transistor Transistor Logic (TTL)** convencionales (máximo 0.8 V para “cero” lógico y un mínimo de 2.2 V para “uno” lógico) con operación especificada en un rango de voltaje de suministro de 2.8 V a 6 V[26].



Trabajos Relacionados

En este capítulo se presentan los principales trabajos de investigación sobre el diseño de registradores de datos en diversas áreas. Un resumen de trabajos importantes referentes a la experiencia en proyectos con la modulación [LoRa](#) y el protocolo [LoRaWAN](#).

En el trabajo propuesto por [4], se incluye el diseño y desarrollo de un prototipo de estación limnigráfica con la capacidad de recopilar, almacenar y transmitir la información con un área de cobertura de al menos 1 km. El dispositivo consta de un microcontrolador y un transceptor SX1276 de bajo consumo de energía y largo alcance [LoRa](#), lo que permite configurarlo como una red de área amplia. Además, para el registro de fecha y hora, usa un [Real-Time Clock \(RTC\)](#), así como un sensor de temperatura y humedad para controlar las condiciones de funcionamiento del dispositivo. El prototipo tiene reguladores de voltaje de salida a 3,3 V y 5 V. En las pruebas de alcance se observó la superioridad del uso de dispositivos con tecnología [LoRa](#) con respecto a las demás tecnologías (hasta 5 veces mayor alcance que [WiFi](#) o [Zigbee](#)).

En [27] se realizó el diseño e implementación de un dispositivo prototipo para adquirir y transmitir automáticamente los datos de una estación hidro-meteorológica. El dispositivo usa transductores de una estación *Davis 6162 Wireless Vantage Pro2 Plus*. Para el manejo de la adquisición, el almacenamiento y la transmisión de los datos se ocupó un microcontrolador con un sistema operativo de tiempo real ([Real-Time Operating System \(RTOS\)](#)). Los datos son almacenados en una memoria [Secure Digital \(SD\)](#) y transmitidos por medio de un módulo XBee. Los datos relacionados al tiempo, la fecha y hora se obtiene de un [RTC](#) sincronizado con un [Global Positioning System \(GPS\)](#). Además, el prototipo admite configurar los periodos de muestreo y transmisión. Luego de probar y validar satisfactoriamente el circuito, se procedió a enviar un diseño con componentes en su mayoría de montaje superficial para su manufactura en una empresa especializada local (Cuenca). Se logró que el dispositivo sea de bajo consumo y de un tamaño reducido, debido a que se usaron componentes de montaje superficial en la mayoría de los elementos. Finalmente, se realizaron las pruebas de funcionamiento, donde se verificó la adquisición, almacenamiento y transmisión de los datos de manera correcta. Las observaciones reportadas por el prototipo son comparables a las de la estación comercial de referencia. La correlación entre los datos de referencia y la estación de prueba es alta.

En el documento [5], se propone un sistema para la adquisición de datos de la intensidad de lluvia basado en la placa de desarrollo *LoPy4* de la compañía Pycom. El sistema propuesto integra la adquisición, el almacenamiento y la transmisión de datos de intensidad de lluvia. La estación está diseñada para funcionar con el sensor de

lluvia de cualquier fabricante. El sistema propuesto consta de tres subsistemas: el subsistema de la fuente de alimentación, el subsistema del registrador de datos y el subsistema del sensor. El sensor de lluvia utilizado a lo largo de esta investigación es el sensor pluviométrico de la marca Davis. La transmisión de datos es realizada por las estaciones que operan como nodos de red, que son parte de una red inalámbrica de sensores **WSN**. La **WSN** se basa en la tecnología **LoRa**. Los datos se transmiten desde los nodos de estación a los nodos *gateway*. El nodo *gateway* reenvía la información al centro de datos. Los nodos de red utilizan el módulo **SX1272** integrado en la placa de desarrollo *LoPy4*, estos módulos funcionan en la banda **Industrial, Scientific and Medical (ISM)** de 915 MHz. El sistema integrado del nodo *gateway* utiliza una placa *LoPy4* e integra un módulo **GPRS** para permitir la comunicación con el servidor de datos. El diagrama de bloque del registrador de datos se muestra la Figura 3.1.

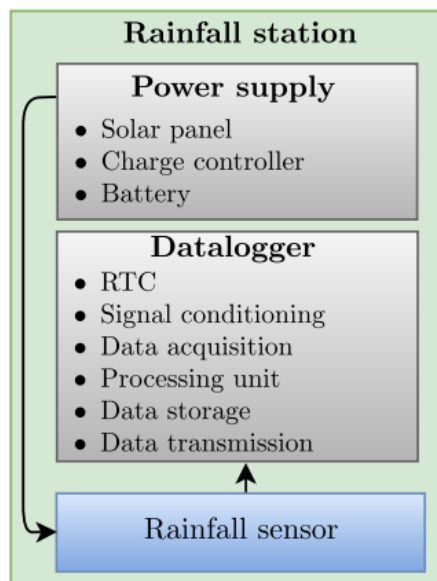


Figura 3.1: Descripción general del sistema.[5]

El propósito del sistema de suministro de energía es ser autosuficiente y lograr una autonomía de al menos 7 días. El sistema tiene un panel solar de 50 W, un controlador de carga basado en **Maximum Power Point Tracker (MPPT)** para aprovechar toda la energía disponible del panel solar y una batería de 12 V - 12 Ah.

La estación desarrollada ha sido comparada con una estación comercial. Las pruebas se realizaron durante 16 días, de los cuales se registró la precipitación en 8 días (ver Figura 3.2).

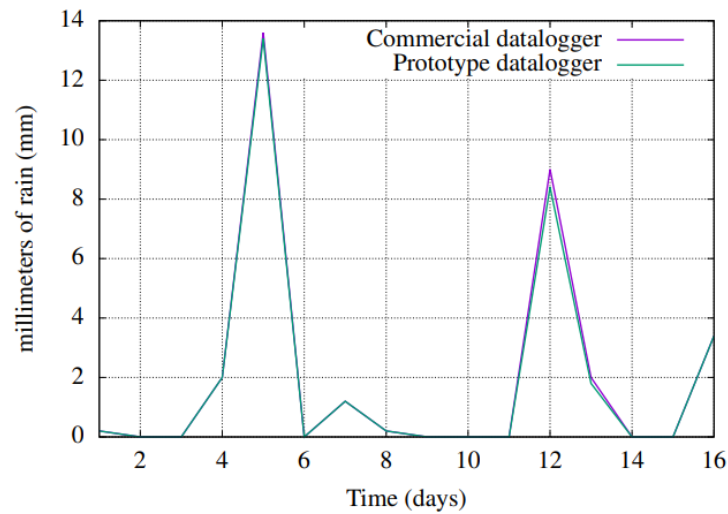


Figura 3.2: Gráfico comparativo de la medición de la intensidad de la lluvia entre un registrador de datos comercial y el prototipo.[5]

Los datos entre la estación comercial y el sistema propuesto son estadísticamente similares, lo que significa que existe una alta correlación. El coeficiente de correlación resultante es $r = 1,0$ (ver Figura 3.3).

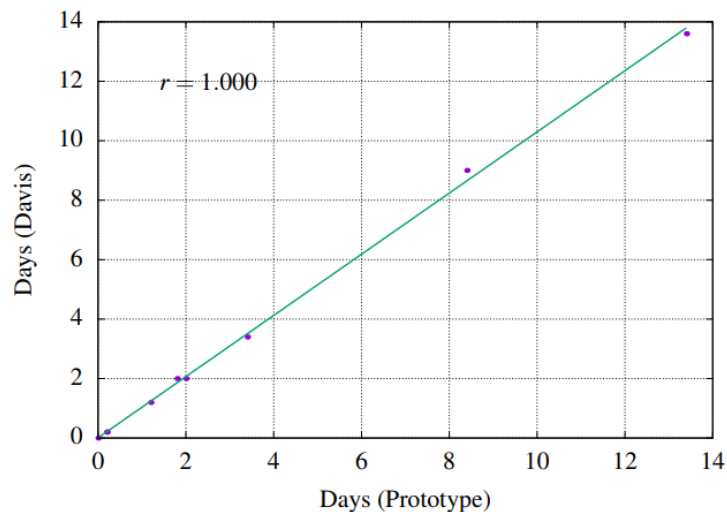


Figura 3.3: Diagrama de dispersión.[5]

En el documento [28] se realiza la comparación de dos soluciones diferentes para la conectividad de los medidores inteligentes con respecto a la duración de las baterías. Uno es el tradicional y bien establecido **Wireless Meter Bus (WM Bus)** y el otro es la solución **LoRaWAN**. Para ambas soluciones, la comparación se realiza utilizando una red real y realizando pruebas de tensión en un medidor específico. Donde se demuestra mediante una medición real, que los medidores que usan **LoRaWAN** tienen una vida útil mucho más larga que los medidores que usan **WM Bus**.

En [29] se muestra la implementación práctica de una **WSN** diseñada para actualizar sistemas ciberfísicos que están dedicados a aplicaciones de monitoreo de salud estructural en el dominio de la construcción. La red consiste en una red de malla formada por nodos sensores inalámbricos **LoRaWAN** sin batería que recopilan



datos físicos y nodos de comunicación que interconectan los nodos sensores con el mundo digital a través de Internet. Estos nodos de detección realizan mediciones de temperatura y humedad relativa y transmiten los datos medidos de forma inalámbrica a largas distancias utilizando la tecnología [LoRa](#) y el protocolo [LoRaWAN](#). Los datos de temperatura y humedad relativa se miden mediante dos nodos de detección sin batería alimentados de forma inalámbrica por la misma fuente de alimentación de RF dedicada que opera en la banda de frecuencia [ISM](#) de 868 MHz.

En [\[30\]](#), la apicultura de precisión se basa en la recopilación constante de datos de colonias de abejas, donde los datos recopilados deben transferirse a un centro de datos remoto para su posterior análisis. La solución tecnológica [LoRaWAN](#), permitió que los dispositivos de bajo consumo de energía se comunicaran con aplicaciones conectadas a Internet a través de conexiones inalámbricas de largo alcance durante muchos años con una sola batería. Tres colonias de abejas fueron equipadas con sensores de temperatura habilitados para [LoRaWAN](#). Las mediciones de los sensores se transmitieron a los nodos *gateway* y servidores de la red [LoRaWAN](#) y se proporcionó acceso a los datos recopilados a través de la aplicación web del portal Lattelecom. La solución permite el monitoreo y teledetección de colonias de abejas y apicultura. La implementación apoya a la gestión de la salud y la productividad de las abejas.

En el estudio [\[31\]](#), se explora la aplicación de la tecnología [LoRa](#) para aplicaciones [IoT](#) de *e-Health*. Se hace el uso del protocolo [LoRaWAN](#) para la transmisión de datos de salud en sistemas de atención médica (o datos críticos de salud en entornos de desastre). Un análisis del banco de pruebas [LoRaWAN](#) real fue realizado para diferentes valores de [SF](#) y diferentes tamaños de carga útil de datos. La temperatura corporal o la presión sanguínea pueden codificarse en pequeñas cargas útiles y pueden notificarse a los centros de salud con una tasa de muestreo muy pequeña (horas o días).



Diseño e Implementación

En este capítulo se presentan las metodologías a aplicar, los aspectos técnicos de los dispositivos y su configuración. Primero, se define el análisis de los requerimientos y descripción de los dispositivos utilizados, para el diseño y desarrollo del circuito electrónico (Sección 4.1 y Sección 4.2). Lo siguiente que se presenta es la comunicación entre el microcontrolador con los dispositivos electrónicos (Sección 4.3) y el diseño final del esquemático y PCB (Sección 4.4 y Sección 4.5). Además, el desarrollo del protocolo LoRaWAN para la placa de circuito impreso final y el almacenamiento de los datos (Sección 4.6 y Sección 4.7). Por último, se presenta la implementación del *gateway* para la descarga de datos *in situ* (Sección 4.8) y el contenedor de dispositivos (Sección 4.9).

4.1. Análisis de requerimientos

En la Figura 4.1, se presenta el diagrama de bloques del diseño del prototipo. El sistema prototipo requiere un bajo consumo de energía, un alcance de al menos 1 km, la adquisición de la señal del sensor pluviométrico, el almacenamiento y transmisión en *quasi*-tiempo real. Además, se obtiene información sobre el estado del equipo usando los parámetros de temperatura, humedad, y nivel de carga de la batería, con su fecha y hora.

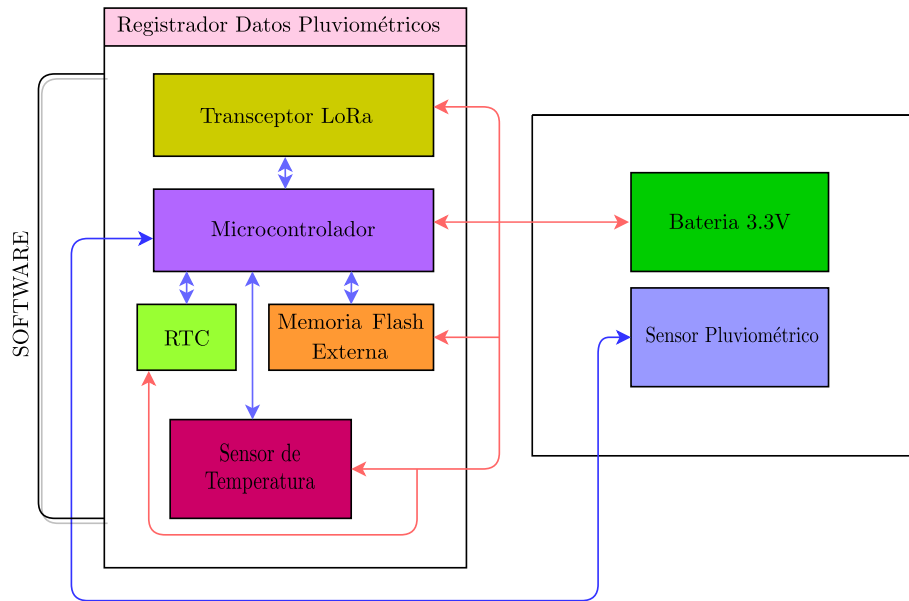


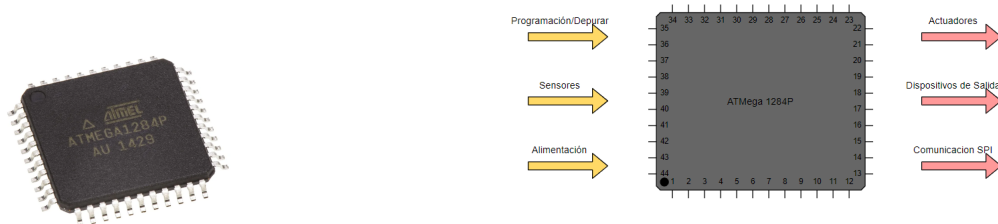
Figura 4.1: Diagrama de bloques del diseño del prototipo.[4]

4.2. Descripción de los dispositivos utilizados

El prototipo es desarrollado para cumplir las funciones de medición, transmisión y almacenamiento de varias variables en determinados intervalos de tiempo. Para eso se procedió a ocupar componentes electrónicos con la tecnología **LoRa**, además del transceptor se usa un microcontrolador de bajo consumo de energía para el control y la lógica programable del transceptor **LoRa**, memoria flash, sensor de temperatura y humedad, **RTC** y el pluviómetro.

4.2.1. Microcontrolador ATmega 1284p

Para el diseño del circuito se usó el microcontrolador ATmega1284p de montaje superficial (ver Figura 4.2a). Este microcontrolador fue seleccionado por aspectos importantes como el uso de memoria programable, bajo consumo energético, bajo costo, compatibilidad con los sensores y otros dispositivos utilizados, además de la adaptación con las librerías **LoRaWAN**. El microcontrolador ATmega1284, se encarga de adquirir los datos que son leídos por los sensores en especial del sensor pluviométrico, control del transceptor **LoRa**, de la memoria externa para el almacenamiento de datos y otros dispositivos.



(a) Dispositivo para montaje superficial.

(b) Diagrama de bloques del funcionamiento del Microcontrolador en el prototipo.

Figura 4.2: Microcontrolador ATmega 1284p.

El microcontrolador y las conexiones con el resto de componentes se encuentran expuestas en la Figura D.1, la cual representa el esquema eléctrico del prototipo. El microcontrolador requiere de un oscilador el cual determina la frecuencia y velocidad de operación, por lo que el fabricante sugiere el uso de un oscilador externo para obtener un mejor funcionamiento. En la Figura 4.2b, se observa el diagrama de bloque del microcontrolador y su funcionalidad en el registrador de datos.

4.2.2. Transceptor Ra02 LoRa SX1278

Para el diseño se decidió usar transceptores LoRa, específicamente el Ra02 LoRa SX1278 que se muestra en la Figura 4.3a. Este viene encapsulado para montaje superficial y directamente con un pin de conexión RF para la antena. Entre sus características están: bajo consumo energético, frecuencia y corriente configurable. La frecuencia de operación es de 433 MHz. Su sensibilidad es mayor a -148dBm y contiene un amplificador de potencia de $+20\text{dBm}$. El voltaje de alimentación es de 1,8 a 3,7 V. La corriente en modo recepción para los módulos SX1278 es de 9,9 mA. La corriente en modo transmisión puede ser configurada desde 10 mA a 240 mA, siendo 100 mA la corriente sugerida por el fabricante [32].

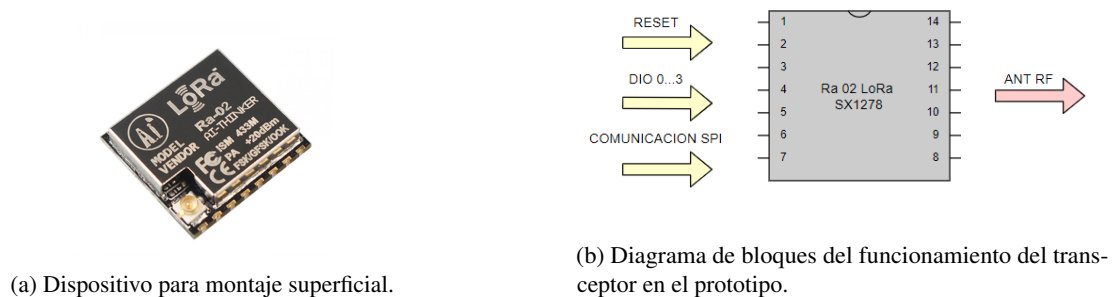


Figura 4.3: Transceptor Ra02 LoRa SX1278.

El diagrama de bloque que se muestra en la Figura 4.3b detalla la funcionalidad del transceptor en el diseño del circuito. El transceptor se encarga de transmitir los datos leídos por los sensores y otros dispositivos conectados al microcontrolador ATmega1284. La transmisión se logra con la conexión de la antena RF y tres entradas:

- Comunicación SPI
- Pin de entrada (RESET) utilizado al inicio o al arrancar el transceptor
- Pines de entrada/salida, configurables por software (DIO 0-5).

Para la comunicación SPI se realizó la conexión de los pines MISO, MOSI, SCK y SS entre ambos integrados ATmega1284p y SX1278.

4.2.3. Sensores

Los sensores miden los cambios en su entorno o detectan eventos y los envían al microcontrolador para su posterior procesamiento. Específicamente, utilizaremos los siguientes sensores con dispositivos finales LoRa. La adquisición de datos de la estación se realiza usando sensores. Los datos de los sensores deben ser mediciones de alta precisión para la recolección del microcontrolador mediante niveles eléctricos. El registrador de datos cuenta

con un sensor magnético *reed-switch* para las variables de precipitación y un sensor digital para la temperatura y humedad. A continuación se detalla cada tipo de sensor.

- **Sensor de temperatura y humedad:** La recolección de datos de temperatura y humedad estará a cargo del módulo DHT11 (Figura 4.4). Este sensor transmite la información mediante el protocolo *1-Wire*. Cumple la función de medir la temperatura y la humedad del ambiente.

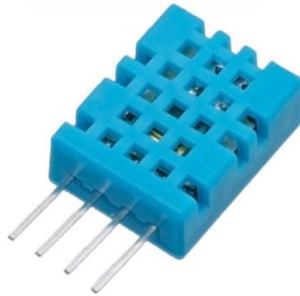


Figura 4.4: Sensor de temperatura y humedad DHT11.

El sensor DHT11 cuenta con la señal digital calibrada, que asegura la alta estabilidad y fiabilidad a lo largo del tiempo. El sensor tiene rangos de medición de humedad desde 20 % hasta 90 % y de temperatura de 0°C a 50°C.

- **Pluviómetro:** El registrador de datos contiene un pluviómetro que trabaja mediante un sistema de balancín para medir la cantidad de lluvia. Por medio del colector (Figura 4.5a) el agua fluye hacia los depósitos del balancín (Figura 4.5b) donde, tan pronto se llena el depósito, este desciende y se vacía, mientras la función del otro extremo es captar el agua de la lluvia y así continuamente. El balancín tiene un imán y en la base se localiza un interruptor *reed-switch* normalmente abierto. Con cada cambio de posición del balancín, el imán pasa sobre el interruptor, lo cual provoca que este se cierre y se vuelva a abrir. De este modo, cada pulso generado es equivalente a 0,2 mm de agua.



(a) Colector de agua de lluvia[33].



(b) Balancín del pluviómetro[33].

Figura 4.5: Pluviómetro de la estación Davis 6162 Wireless Vantage Pro2 Plus.

El microcontrolador es el encargado de realizar la detección de los pulsos generados por el interruptor. Cuando ocurre la conmutación del interruptor se puede tener la presencia de rebote o ruido de tensión eléctrica, por lo que pueden ser detectados como pulsos falsos de conmutación por el microcontrolador. Para solucionar este problema de ruido, se añade un filtro RC para suavizar el ruido producido por los cambios de estado del interruptor. El tiempo de carga y descarga de un capacitor (el cual debe ser mayor o igual a la duración del ruido por rebotes) está dado por la ecuación $\tau = RC$, donde se

despeja la capacitancia C , y se reemplaza el tiempo de carga y descarga del capacitor por $\tau \geq 1$ ms, y la resistencia con un valor comercial de $R = 10$ k Ω , se encuentra $C = 0,1$ μ F. En la Figura 4.6 se muestra la implementación del filtro y las conexiones realizadas para el pluviómetro [27].

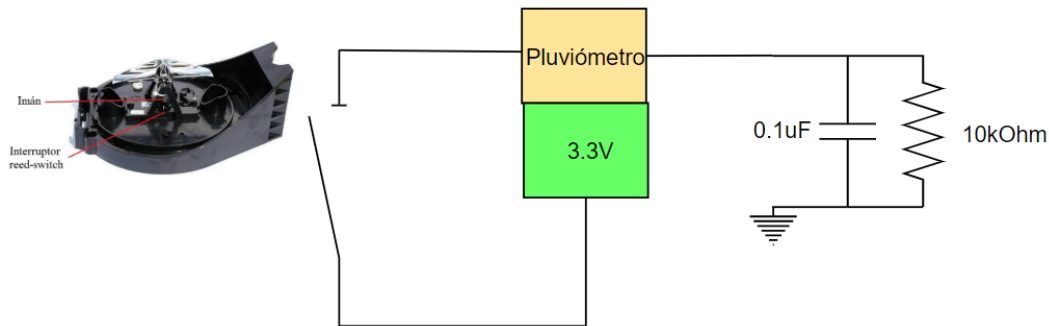
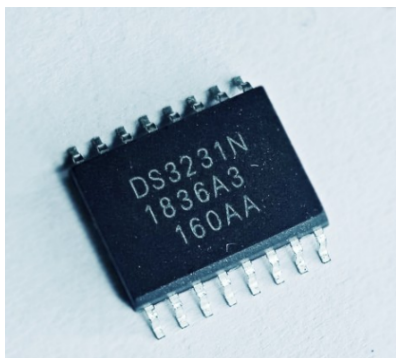


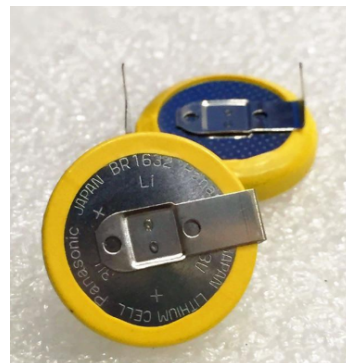
Figura 4.6: Conexión del pluviómetro en el prototipo. Se muestra los pines de salidas y el filtro implementado previo a la adquisición de la señal[27].

4.2.4. Reloj en tiempo real (RTC)

Para obtener los datos de la fecha y la hora se seleccionó el RTC DS3231 que se muestra en la Figura 4.7a. El DS3231 tiene una alta precisión en la medición del tiempo. La comunicación con el microcontrolador se la realiza a través del bus I2C. El voltaje de operación del DS3231 es de 2.3 a 5.5V[34].



(a) Dispositivo para montaje superficial.



(b) Batería Panasonic CR2032.

Figura 4.7: RTC DS3231.

Para evitar problemas como mantener la hora y fecha cuando se retira la alimentación, se usa la batería Panasonic CR2032 de 3 V no recargable (Figura 4.7b). Una vez que se retira la alimentación por VCC, entra en funcionamiento inmediatamente la batería, haciendo que no haya necesidad de configurar la hora y la fecha nuevamente. Para la sincronización de datos del tiempo se hace el uso de la librería RTCLib.

4.2.5. Memoria Externa

Para el almacenamiento de datos recolectados por el registrador de datos se utiliza la memoria flash W25Q128JV (Figura 4.8), la cual proporciona soluciones de almacenamiento para sistemas con espacio, pines y potencia limitados. Su espacio de almacenamiento es de 128 Mbits. El voltaje de alimentación está entre 2,7 V y

3,6 V, con un bajo consumo de corriente $1 \mu\text{A}$ para el apagado. La comunicación con el microcontrolador se la realiza con la **SPI** estándar.



Figura 4.8: Memoria flash W25Q128JV.

4.3. Comunicación entre dispositivos

El microcontrolador ATmega1284p utiliza el protocolo de comunicación **SPI** de múltiple esclavos independientes descrito en la Sección 2.5, que establece la comunicación serial síncrono *full duplex* con la memoria externa W25Q128JV así como con el Ra02 LoRa SX1278. El dispositivo que se configura como maestro es el ATmega1284 mientras que la memoria W25Q128JV y el Ra02 SX1278 van a comportarse como esclavos. Las conexiones de estos elementos se detallan en las Tablas 4.1 y 4.2.

Tabla 4.1: Nomenclatura utilizada para la comunicación ATmega1284p y SX1278.

| Descripción | Ra02 SX1278 | ATmega 1284p |
|------------------|-------------|--------------|
| COMUNICACIÓN SPI | SCK | 7 |
| COMUNICACIÓN SPI | MISO | 6 |
| COMUNICACIÓN SPI | MOSI | 5 |
| COMUNICACIÓN SPI | NSS | 4 |
| CONTROL | NRESET | 1 |
| CONTROL | DIO0 | 13 |
| CONTROL | DIO3 | 22 |
| ALIMENTACIÓN | 3.3V | 3.3V |
| ALIMENTACIÓN | GND | GND |

Tabla 4.2: Nomenclatura utilizada para la comunicación entre ATmega1284 y W25Q128JV.

| Descripción | W25Q128JV | ATmega 1284p |
|------------------|-----------|--------------|
| COMUNICACIÓN SPI | SCK | 7 |
| COMUNICACIÓN SPI | MISO | 6 |
| COMUNICACIÓN SPI | MOSI | 5 |
| COMUNICACIÓN SPI | NSS | 12 |
| ALIMENTACIÓN | 3.3V | 3.3V |
| ALIMENTACIÓN | GND | GND |

El protocolo de comunicación **I2C** es el que enlaza al microcontrolador ATmega1284p con el Reloj a tiempo

real, la conexión de los pines se encuentran definidos en la Tabla 4.3. Además, las conexiones de la comunicación SPI y I2C se muestran en la Figura D.1.

Tabla 4.3: Nomenclatura utilizada para la comunicación entre ATM1284 y RTC DS3231.

| Descripción | RTC DS3231 | ATmega 1284p |
|------------------|------------|--------------|
| COMUNICACIÓN I2C | SCL | 16 |
| COMUNICACIÓN I2C | SDA | 17 |
| ALIMENTACIÓN | 3.3V | 3.3V |
| ALIMENTACIÓN | GND | GND |

4.4. Aspecto final del esquemático del dispositivo

El diagrama de bloques de la estación pluviométrica se encuentra en la Figura 4.9 el cual contiene los bloques para su funcionamiento. Para empezar en la parte izquierda del diagrama se encuentra el módulo transceptor LoRa SX1278 Ra-02, es el encargado de la transmisión hacia la antena RF. En el siguiente bloque tenemos el microcontrolador ATmega 1284p, su función es el control de comunicación con los dispositivos en este caso con la memoria flash externa, el RTC, transceptor LoRa y los sensores. Por último, para futuras conexiones de sensores o dispositivos se añadieron pines de control analógicos/digitales.

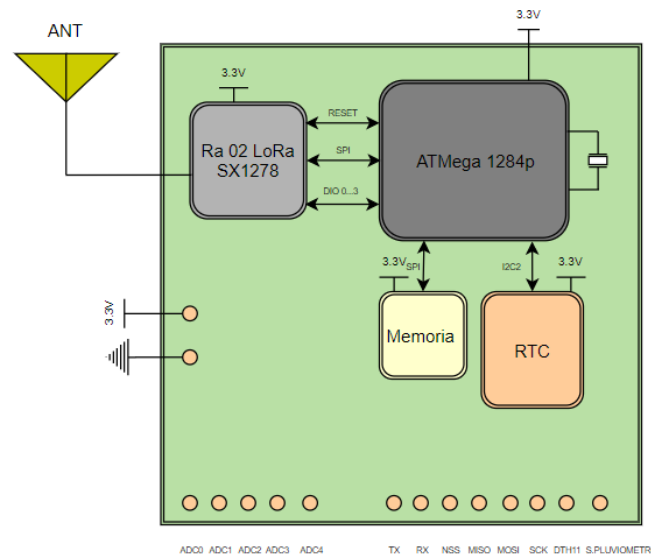


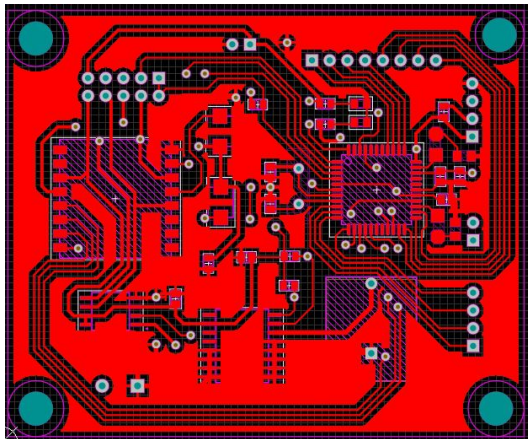
Figura 4.9: Diagrama de bloques del diseño del prototipo.

La Figura D.1 presenta el esquemático con todos los componentes y sus conexiones correspondientes.

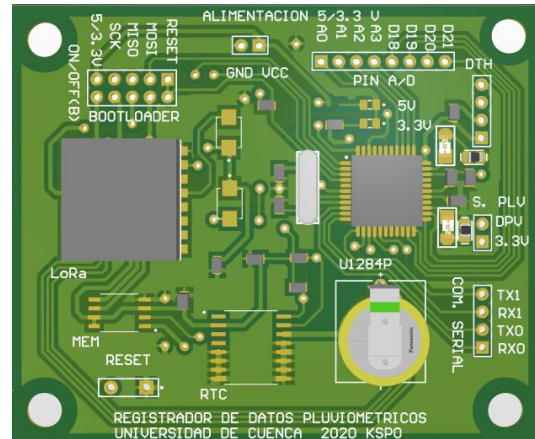
4.5. Características de la PCB

Las dimensiones del PCB desarrollado son de 75,86 × 63,294 mm, las pistas de conexión tienen un grosor de 0,35 mm y las de alimentación de 0,5 mm, como regla de diseño cuenta con un *clearance* de 0,25 mm ya que debido a la cercanía de las pistas puede generar un acople indeseado (*crossstalk*). Los componentes se agrupan

según las señales a procesar esto se hace para reducir la distancia de las pistas y los efectos parásitos resistivos e inductivos. Los componentes usados son de montaje superficial. En las Figuras 4.10a y 4.10b se muestra la cara superior en 2D y 3D respectivamente, en la cual consta de todos los elementos mencionados en la Sección 4.2, pistas, vías y los pines para la comunicación y alimentación externa.



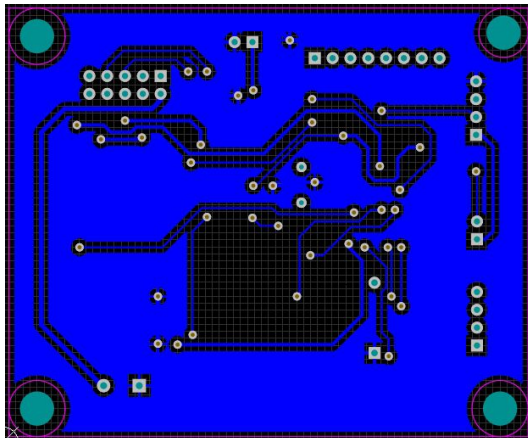
(a) Diseño en 2D.



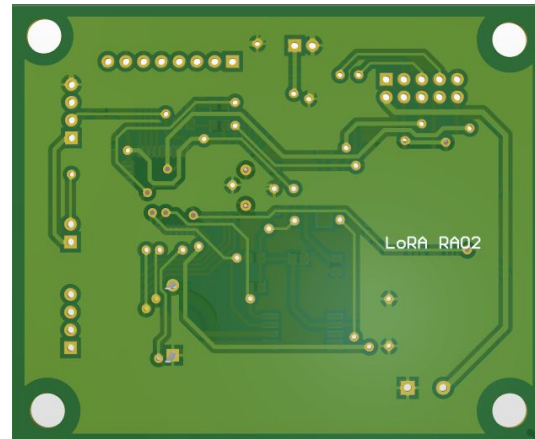
(b) Diseño en 3D.

Figura 4.10: Diseño PCB de la cara superior del dispositivo.

En las Figuras 4.11a y 4.11b se muestra la cara inferior en 2D y 3D respectivamente, en la cual consta de las pistas y vías que conectan con la cara superior de la PCB.



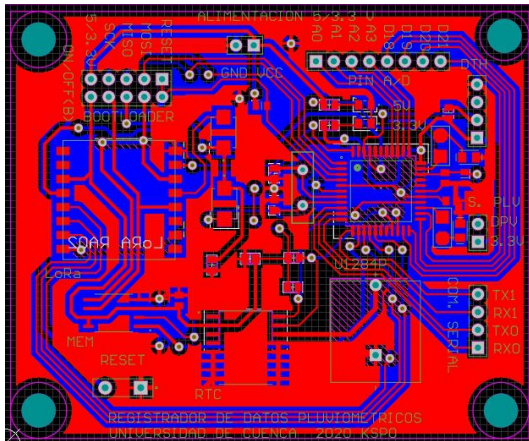
(a) Diseño en 2D.



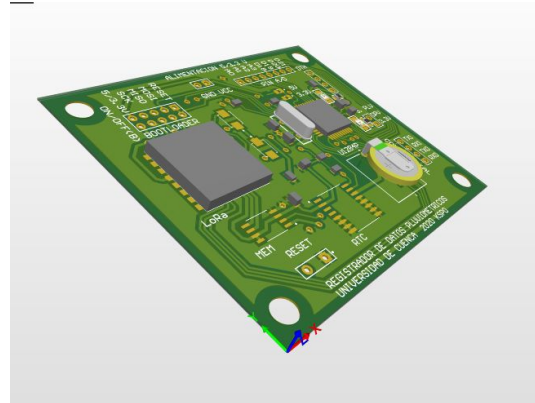
(b) Diseño en 3D.

Figura 4.11: Diseño PCB de la cara inferior del dispositivo.

En las Figuras 4.12a y 4.12b se muestra el diseño PCB del prototipo final en 2D y 3D respectivamente, en el modelo 2D se encuentran visibles todas las capas disponibles, mientras que en el modelo 3D se muestra el diseño PCB en el cual fue añadido el *screen* laminado para la identificación de los pines de comunicación, alimentación y control.



(a) Diseño en 2D.



(b) Diseño en 3D.

Figura 4.12: Diseño PCB final del dispositivo.

4.6. Aplicación del Protocolo LoRaWAN

La red de área amplia de largo alcance **LoRaWAN** es el protocolo de comunicación y la arquitectura del sistema para la red, mientras que la capa física **LoRa** habilita el enlace de comunicación. **LoRa** y **LoRaWAN** proporcionan una comunicación de largo alcance (mayor que el celular, cobertura interior profunda, topología en estrella), duración máxima de la batería, usos múltiples y bajo costo (infraestructura mínima, nodos finales de bajo costo, software abierto).

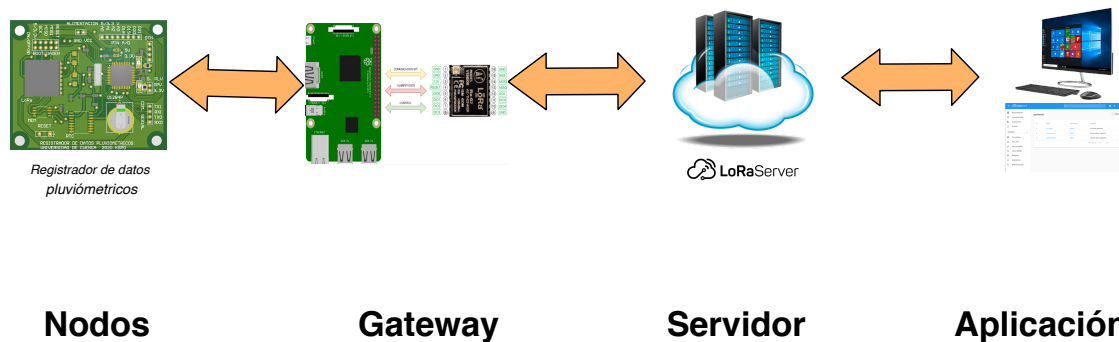


Figura 4.13: Arquitectura de la red LoRaWAN.

LoRaWAN consta de nodos finales (dispositivos finales), *gateways* (puertas de enlace), un servidor de red y servidores de aplicaciones (Figura 4.13). En una red **LoRaWAN**, un nodo sensor final adquiere los datos meteorológicos, que son transmitidos y recibidos por múltiples *gateways*. Una vez que se reciben los datos, cada puerta de enlace reenviará el paquete recibido al servidor de red a través de la red celular, Ethernet, **WiFi** o satélite. El software que se ejecuta en la puerta de enlace es responsable de reenviar cualquier paquete de datos entrantes al servidor de red. El servidor de red envía y recibe mensajes **LoRaWAN** hacia y desde dispositivos, que se comunica con servidores de aplicaciones ascendentes. El servidor de aplicaciones es el destino de los datos de la aplicación del dispositivo enviados como carga útil en los mensajes de **LoRaWAN**[35].

4.6.1. Nodo final LoRa

Los nodos finales están físicamente fijos o móviles, transmiten datos del sensor a un *gateway* LoRa. Se puede construir un nodo final combinando varias opciones de *hardware* y *software*. La Figura 4.14 muestra una implementación del prototipo PCB del nodo final. Un nodo final LoRa consta de un transceptor LoRa, un microcontrolador, un sensor, una unidad de fuente de alimentación, memoria externa y una antena. Un nodo final LoRa puede recibir datos a través del enlace descendente desde el LoRa gateway.

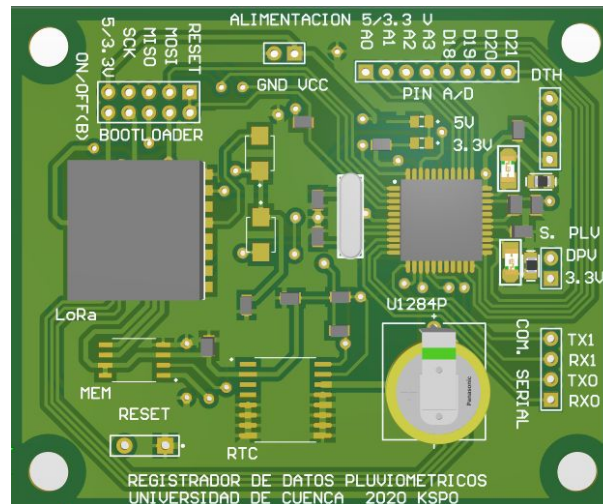


Figura 4.14: Nodo final- Registrador de datos Pluviométricos.

Para la configuración de la red LoRaWAN se seleccionó el método de activación ABP. El uso de este método requiere definir los siguientes valores:

- Dirección del nodo (*DevAddr*).
- Clave de sección de aplicación (*AppSKey*).
- Clave de sección de red (*NwkSKey*).

Los valores *DevAddr*, *AppSKey* y *NwkSKey* se configuran en el microcontrolador ATmega 1284p y en el servidor LoRa para el proceso de autenticación. La comunicación entre un nodo y el servidor, se da según el proceso que se muestra en la Figura 4.15.

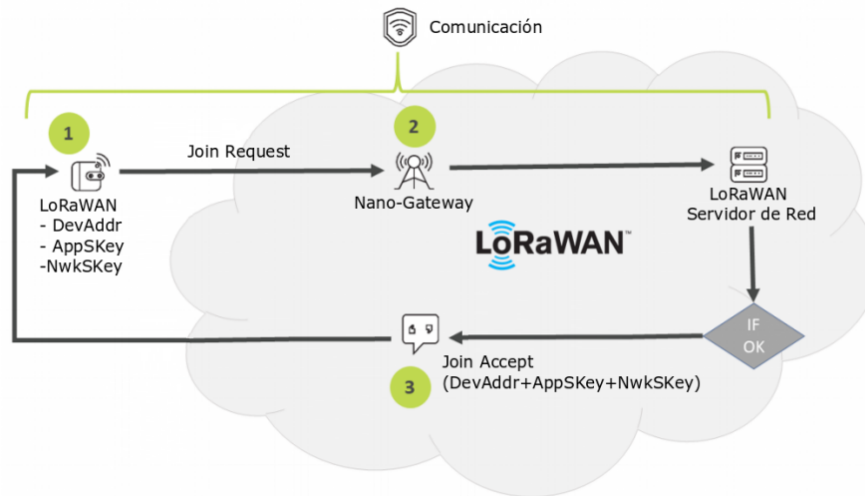


Figura 4.15: Proceso de agregación de nodos.

Para inicializar el nodo en modo **LoRaWAN** se definen la dirección, las claves, tipo de activación y la frecuencia que opera el transceptor **LoRa**, en este caso 433 Mhz.

4.6.2. LoRa Gateway

Los *gateways* **LoRa** deberían tener más potencia de procesamiento que los nodos finales. Por esto, se procedió a implementar un *gateway* **LoRa** con Raspberry Pi. Un *gateway* **LoRa** consta de un transceptor de radio, un microprocesador, una fuente de alimentación y una antena. El Raspberry Pi funciona con reenviadores de paquetes detallados en la Sección 4.6.2.1.

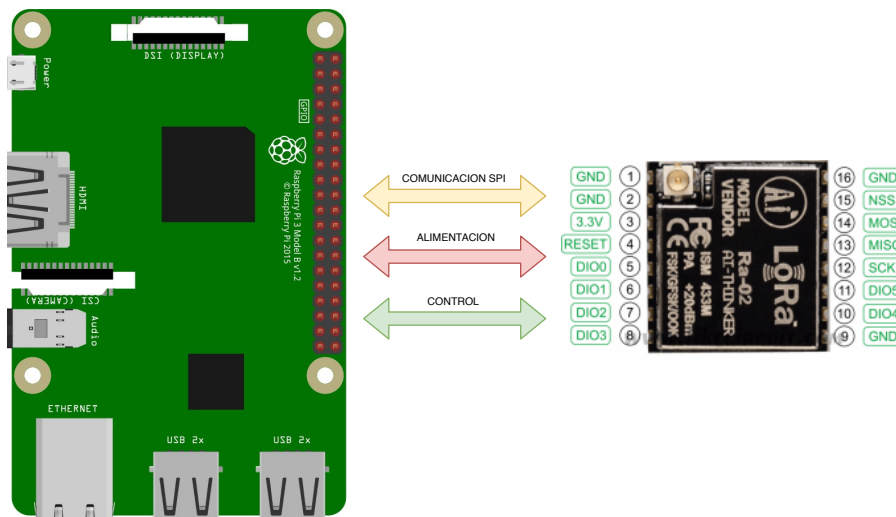


Figura 4.16: Raspberry Pi junto al Ra02 SX1278.

Toda la comunicación se manejará a través de **SPI** entre el transceptor de radio y el Raspberry Pi. Los pines **SPI** del transceptor de radio admiten los niveles lógicos de 3,3 V. En la Tabla 4.4 se muestra la conexión de los pines del transceptor de radio con el microcontrolador Raspberry Pi por medio de comunicación **SPI**.



Tabla 4.4: Nomenclatura utilizada para la comunicación entre Raspberry Pi y Ra02 SX1278.

| Descripción | Ra02 SX1278 | Raspberry Pi |
|------------------|-------------|--------------|
| COMUNICACIÓN SPI | SCK | 23 |
| COMUNICACIÓN SPI | MISO | 21 |
| COMUNICACIÓN SPI | MOSI | 19 |
| COMUNICACIÓN SPI | NSS | 22 |
| ALIMENTACIÓN | 3.3V | 1 |
| ALIMENTACIÓN | GND | 6 |
| CONTROL | RESET | 11 |
| CONTROL | D0 | 7 |

4.6.2.1. Reenviador de paquetes

Un reenviador de paquetes es el *software* que se ejecuta en *gateway* LoRa. Permite que el concentrador LoRa transmita y reciba paquetes LoRa para enlaces ascendentes y enlaces descendentes desde nodos finales a servidores de red y desde servidores de red a nodos finales. Por lo general, un reenviador de paquetes tiene las siguientes funciones:

- Reenviar los paquetes LoRa recibidos por el concentrador (módulo LoRa) al servidor de red a través del enlace IP.
- Emitir paquetes LoRa que envía el servidor de red.

Para la implementación de este trabajo se procedió a ocupar el reenviador de paquetes *Raspberry Pi single-channel packet forwarder* que se encuentra disponible en el repositorio *Github* [36]. Para el reenvío de paquetes del *gateway* se usan dos tecnologías de transmisión. La tecnología LoRa que sirve para la comunicación con el nodo final, y la red de retorno IP (*Backhaul IP stack*) que sirve para la conexión a la red Ethernet con el servidor.

4.6.3. The Things Network

TTN desarrolla una implementación de un servidor de red mundial LoRaWAN. Esta red permite conectar los gateways, registrar dispositivos y enviar / recibir datos en *quasi*-tiempo real. La ventaja de TTN es que se puede usar la cobertura existente de manera gratuita en cualquier momento y lugar. En la Figura 4.17 se observa la interfaz de la consola de TTN, que tiene disponible las aplicaciones y los *gateways* que se requieran registrar.

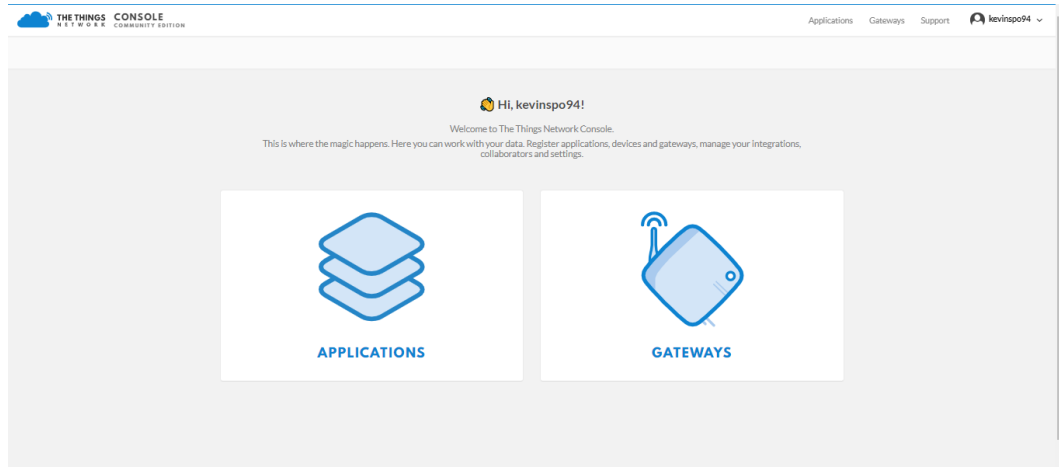


Figura 4.17: Interfaz de la consola de TTN.

Para la visualización de los datos de las aplicaciones en TTN, los *payloads* tienen que ser decodificados, para eso la aplicación tiene un sección para desarrollar el decodificador en el lenguaje *Javascript*, que permita reconocer si los paquetes de datos son del sensor pluviométrico o del estado del dispositivo. En la Figura 4.18 se muestra la pantalla, donde se implementa el decodificador.

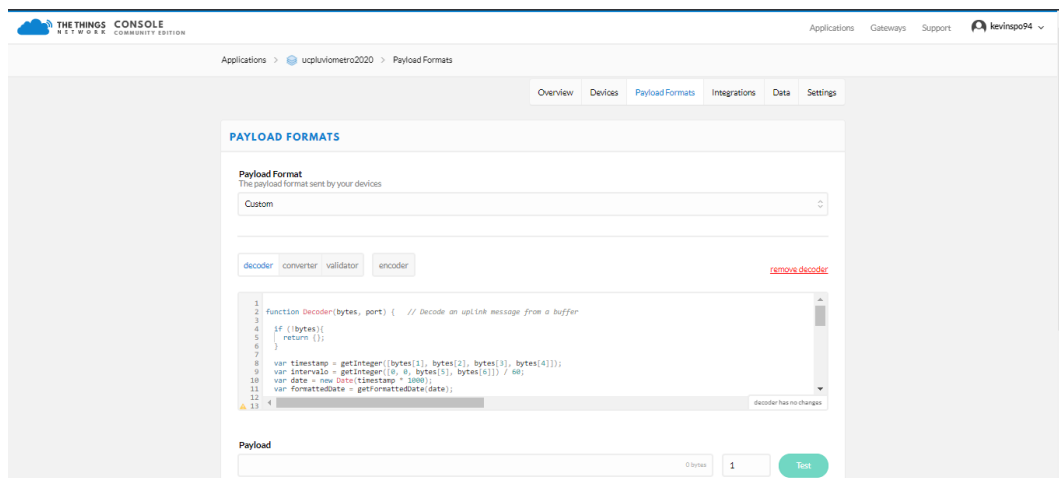


Figura 4.18: Ventana para desarrollar el decodificador en TTN

4.7. Almacenamiento de datos

La información del dispositivo se almacena en dos tipos de archivos: de configuración y de datos. El archivo de configuración `rain.cnf`, contiene la información de configuración de la estación. En este archivo se escribe, el intervalo de tiempo de cada periodo en el que los datos se van a almacenar en memoria y enviar hacia el servidor. Además, la configuración de las credenciales del enlace ABP entre el nodo y la aplicación del servidor.

Para el almacenamiento de los datos recolectados por el dispositivo, se crean dos tipos de archivos. El primer archivo almacena datos del sensor de lluvia; su prefijo es (`sensor-[fecha].dat`). El segundo archivo almacena información de estado sobre el nivel de batería, temperatura y humedad del ambiente en el que se encuentra el nodo; su prefijo es (`estado-[fecha].dat`). Los registros del sensor se almacenan de manera predeterminada cada 5

minutos (o cómo se configura en el archivo `rain.conf`) y el estado se almacena cada 60 minutos. A continuación, se detalla la estructura de cada uno de los archivos.

4.7.1. Estructura de archivo para almacenamiento de datos

Definimos dos tipos de registros: registro de datos (Figura 4.19) y registro de estado (Figura 4.20). El registro de datos tiene información adquirida por el sensor de lluvia; estos datos se almacenan cada 5 minutos, el tamaño cada registro es de 64 bits. El registro de estado almacena información sobre la condición interna del registrador de datos. La información de estado se almacena cada hora y el tamaño de cada registro es de 72 bits.

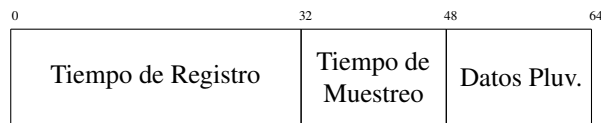


Figura 4.19: Estructura de trama para el almacenamiento en el archivo del sensor de lluvia.

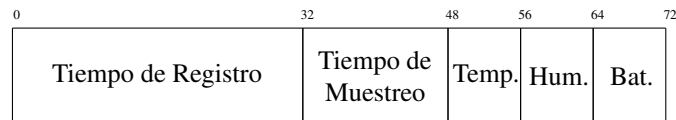


Figura 4.20: Estructura de trama de datos para el almacenamiento en el archivo de estado.

4.8. Descarga de datos *in-situ*

En la descarga de datos intervienen tres dispositivos: un nodo sensor, un *gateway* y un dispositivo móvil (Figura 4.21). El *gateway* se conecta a los nodos sensores utilizando comunicación serial **Universal Asynchronous Receiver Transmitter (UART)**. Entre el dispositivo móvil y el *gateway* se usa **WiFi**.

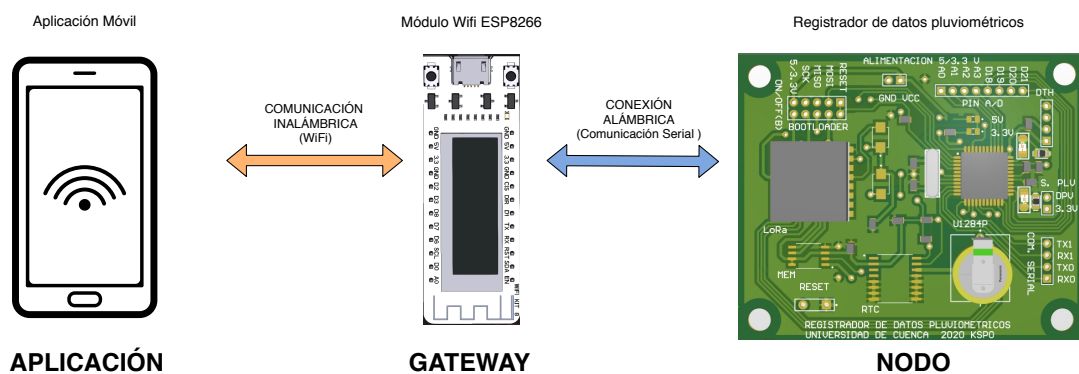


Figura 4.21: Arquitectura de la red para la descarga de datos *in-situ*.

4.8.1. Nodo

El nodo, es decir el registrador de datos, es el encargado de responder las solicitudes enviadas desde la aplicación móvil por medio del *gateway*. La conexión alámbrica que proviene del *gateway* es conectada al puerto de red RJ45 con protección **IP67** integrado en el nodo, que hace uso de la comunicación serial **UART**.

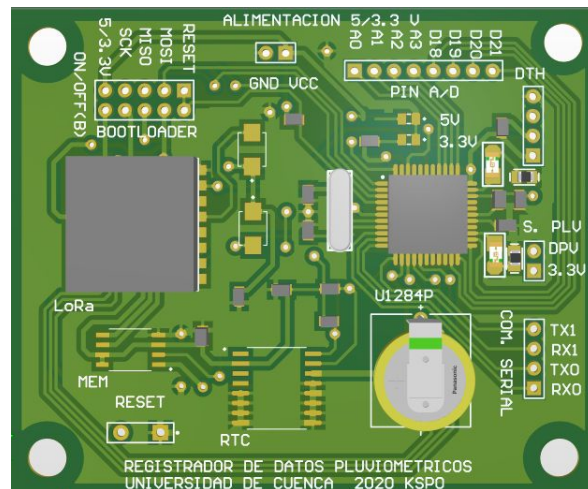


Figura 4.22: Registrador de datos.

4.8.2. Wifi Gateway

El módulo que va a actuar como *gateway* es el *Wifi Kit 8*. Este es un módulo ESP8266 (Figura 4.23), que incluye un sistema de administración de batería [Litio-Polímero \(Li-Po\)](#) y una pantalla OLED de 0.9. Para establecer la comunicación entre el nodo y el dispositivo móvil, se desarrolló el *software* que permite procesar las peticiones realizadas y el reenvío de los paquetes hacia la aplicación móvil, mediante el protocolo [Hypertext Transfer Protocol \(HTTP\)](#).

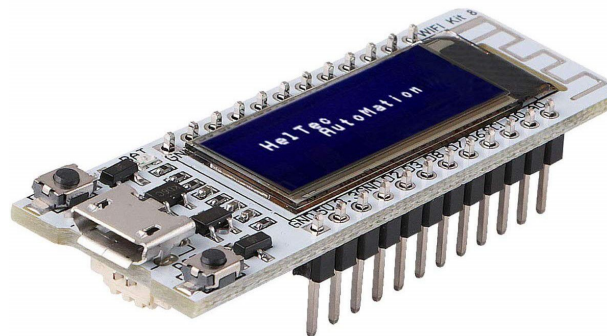


Figura 4.23: Módulo Wifi Kit 8

El módulo [WiFi](#) es alimentado con una batería [Li-Po](#) de 3,7 V, además se integra un cable Ethernet para tener acceso al nodo. Para unirse a la red inalámbrica, se configura el nombre y la contraseña, que son ingresados desde el dispositivo móvil.

4.8.3. Aplicación Móvil

Con el objetivo de visualizar y decodificar los datos adquiridos y almacenados por el nodo, se implementa una aplicación móvil. La aplicación es nativa para dispositivos con sistema operativo *Android*, desarrollada en el lenguaje *JAVA* (Figura 4.24).

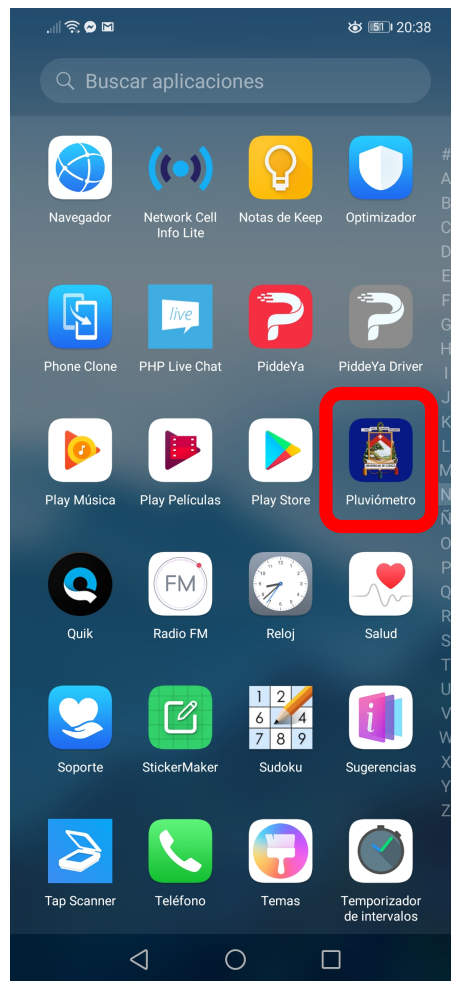


Figura 4.24: Aplicación situada en sistema operativo Android.

La aplicación móvil, permite al usuario buscar los archivos almacenados en memoria (Sección 4.7), mediante dos campos de búsqueda: fecha y tipo de archivo. Si el archivo existe, se visualizan los datos en pantalla. Además se dispone de la función para la descarga del archivo en formato (.csv), la interfaz de la aplicación móvil se muestra en la Figura 4.25.



Figura 4.25: Interfaz de la aplicación móvil.

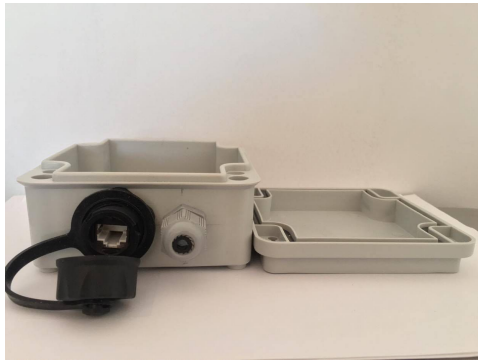
Adicionalmente, la aplicación tiene la opción de configuración de parámetros, la cual permite cambiar los intervalos de tiempo de cada registro y cambiar las credenciales de autenticación de [ABP](#) a la aplicación del servidor [LoRaWAN](#).

4.9. Contenedor de Dispositivos

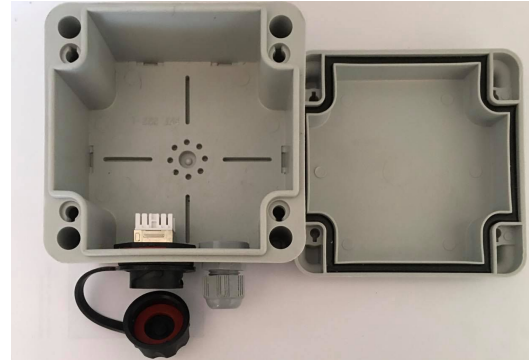
4.9.1. Nodo

El dispositivo está destinado a ser usado a la intemperie, por lo que se busca evitar el contacto con agentes externos y medioambientales, tales como la lluvia, el polvo o la humedad, ya que pueden provocar un mal funcionamiento o reducir el tiempo de vida útil. Se buscó en el mercado un contenedor que sea capaz de proteger al circuito del ambiente externo, se eligió ocupar una caja KemaKeur con medidas $10 \times 10 \times 5$ cm y con el código

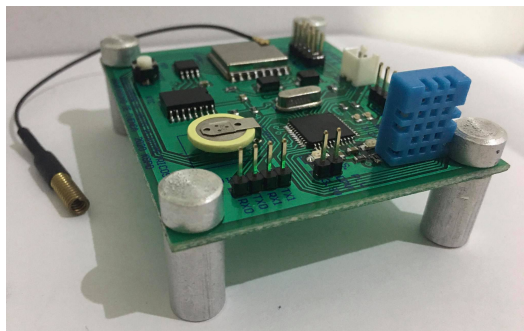
Ingress Protection 65 (IP65), que refleja el grado de protección [37]. De esta forma, los únicos componentes expuestos al ambiente son los conectores y la antena RF, de tal manera que se procedió a la colocación de una prensa estopa plástica PG7 y un puerto de red RJ45 con protección IP67, como se muestra en las Figuras 4.26a y 4.26b.



(a) Contenedor del registrador de datos(vista frontal).



(b) Contenedor del registrador de datos(vista superior).



(c) Registrador de datos con tornillos colocados.



(d) Registrador de datos integrado en el contenedor.

Figura 4.26: Contenedor del Nodo

El tamaño del PCB del prototipo fue en base al espacio interior del contenedor. Para fijar y sujetar la placa dentro del contenedor, se realizaron perforaciones en sus cuatro esquinas para la incorporación de tornillos. En las Figuras 4.26c y 4.26d se observan la instalación de la placa dentro del contenedor.

4.9.2. WiFi gateway

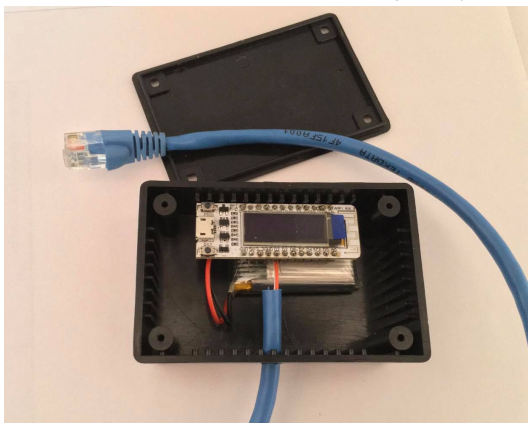
Para realizar la descarga de datos *in-situ*, se requiere la movilización del *gateway* WiFi hacia los nodos sensores, por lo que se coloca en un contenedor adquirido en el mercado local, con medidas $8 \times 5 \times 3$ cm y donde se realizan dos agujeros. El primer agujero es usado para la salida del cable de datos Ethernet. El segundo agujero es ocupado para la alimentación y programación del dispositivo (Figura 4.27a y Figura 4.27b).



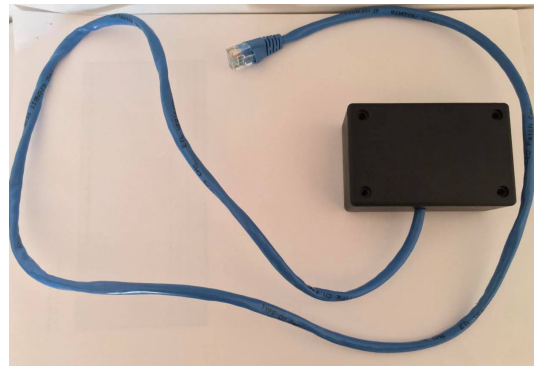
(a) Partes del contenedor del WiFi gateway.



(b) Unión de partes del contenedor WiFi.



(c) Integración del WiFi gateway en el contenedor.



(d) Wifi gateway integrado en el contenedor.

Figura 4.27: Contenedor del WiFi gateway

Para la fijación del *gateway* con su batería dentro del contenedor, se hace uso de silicona, además se sueldan los cables del cable Ethernet a los puertos de la comunicación serial(UART) del módulo WiFi, su integración se muestra en las Figuras 4.27c y 4.27d.

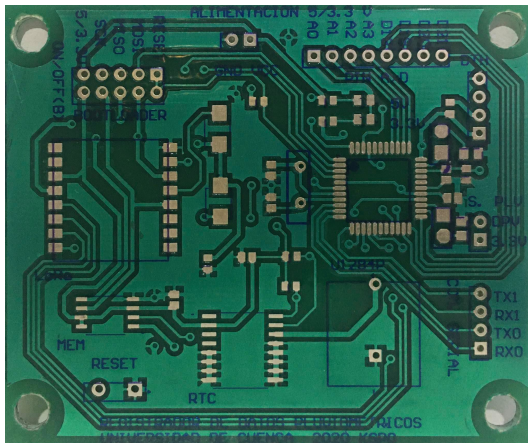


Resultados

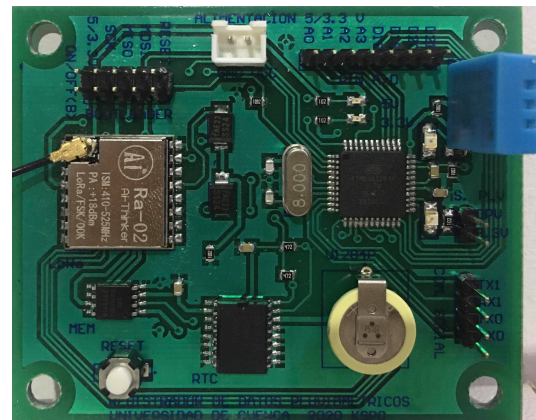
En este capítulo se describen los resultados obtenidos durante el diseño, construcción y pruebas del prototipo. Se detallan el funcionamiento final y los resultados obtenidos en varias pruebas relacionadas con el almacenamiento, la transmisión, y el consumo energético del sistema.

5.1. PCB

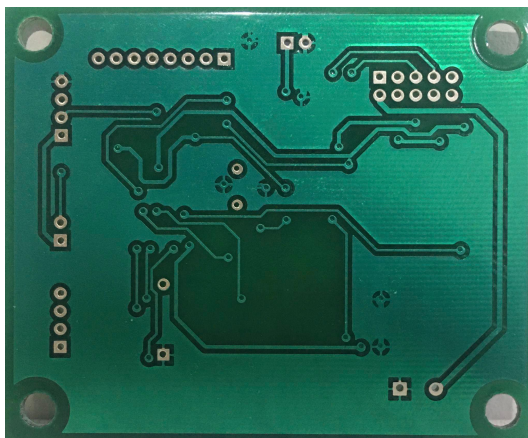
El diseño presentado en la Sección 4.5, se envió a la manufactura en una empresa local especializada. En la Figura 5.1a se muestra la vista superior del diseño final del PCB sin los componentes soldados y en la Figura 5.1b se muestra el PCB con los componentes soldados. En las Figuras 5.1c y 5.1d se muestra la vista inferior del PCB, en la primera figura sin los componentes soldados y en la segunda con los componentes soldados.



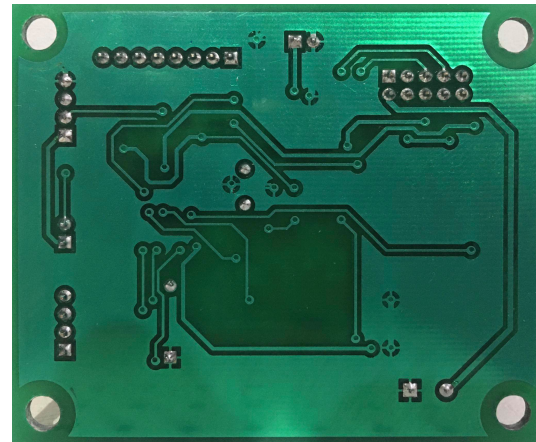
(a) Aspecto final del PCB sin soldar componentes (vista superior).



(b) Aspecto final del PCB con componentes de montaje superficial (vista superior).



(c) Aspecto final del PCB sin soldar componentes (vista inferior).



(d) Aspecto final del PCB con componentes de montaje superficial (vista inferior).

Figura 5.1: PCB del registrador de datos pluviométrico

5.2. Pruebas de Laboratorio

5.2.1. Transmisión

Para la transmisión de datos hacia el servidor **LoRaWAN**, se utilizaron los dispositivos: Nodo, *gateway* **LoRa** y un ordenador (detallados en la Sección 4.6). El nodo transmite los datos hacia el *gateway* y este reenvía los paquetes hacia el servidor **LoRaWAN** de **TTN**. Los paquetes del sensor son recibidos cada 5 minutos y los datos del estado del nodo cada 60 minutos. Los datos en recepción son visualizados en la aplicación del servidor **LoRaWAN** de **TTN** (Figura 5.2).

| time | counter | port | payload | fecha | intervalo |
|----------|---------|------|--|------------------------------|----------------------|
| 24-59:01 | 54 | 2 | payload: 00 5F 16 3CDE012C000000000000 | fecha: "2020-07-23 00:54:01" | Intervalo: 5 minutos |
| 24-54:28 | 53 | 2 | payload: 00 5F 16 38AA012C000600000000 | fecha: "2020-07-23 00:54:28" | Intervalo: 5 minutos |
| 24-53:56 | 52 | 2 | payload: 01 5F 16 38AA0E3000B302000064 | fecha: "2020-07-23 00:53:56" | Intervalo: 5 minutos |
| 24-48:54 | 51 | 2 | payload: 00 5F 16 3A7E012C001400000000 | fecha: "2020-07-23 00:48:54" | Intervalo: 5 minutos |
| 24-43:52 | 50 | 2 | payload: 00 5F 16 3952012C000800000000 | fecha: "2020-07-23 00:43:52" | Intervalo: 5 minutos |
| 24-38:50 | 49 | 2 | payload: 00 5F 16 3927012C000000000000 | fecha: "2020-07-23 00:38:50" | Intervalo: 5 minutos |
| 24-33:48 | 48 | 2 | payload: 00 5F 16 36FB012C000200000000 | fecha: "2020-07-23 00:33:48" | Intervalo: 5 minutos |
| 24-28:46 | 47 | 2 | payload: 00 5F 16 35CF012C000600000000 | fecha: "2020-07-23 00:28:46" | Intervalo: 5 minutos |
| 24-23:44 | 46 | 2 | payload: 00 5F 16 34A3012C001C00000000 | fecha: "2020-07-23 00:23:44" | Intervalo: 5 minutos |
| 24-18:41 | 45 | 2 | payload: 00 5F 16 3377012C000400000000 | fecha: "2020-07-23 00:18:41" | Intervalo: 5 minutos |
| 24-13:39 | 44 | 2 | payload: 00 5F 16 324B012C000C00000000 | fecha: "2020-07-23 00:13:39" | Intervalo: 5 minutos |

Figura 5.2: Visualización de datos en la aplicación del servidor **TTN**.

El detalle de un paquete con información recolectada por un sensor se muestra en la Figura 5.3, y la información del estado del nodo se muestra en la Figura 5.4. En el paquete del sensor se muestran: la fecha, el intervalo de transmisión y la cantidad de lluvia. En el paquete del estado se encuentran los datos de la batería, la fecha y hora, la temperatura y humedad, y el intervalo de transmisión de cada paquete. Además, todos los paquetes tienen metadatos con información de la conexión **LoRaWAN** como: la frecuencia, **Data Rate (DR)**, **CR**, **SNR**, **Received Signal Strength Indicator (RSSI)** y el tiempo en el que llegó el paquete.

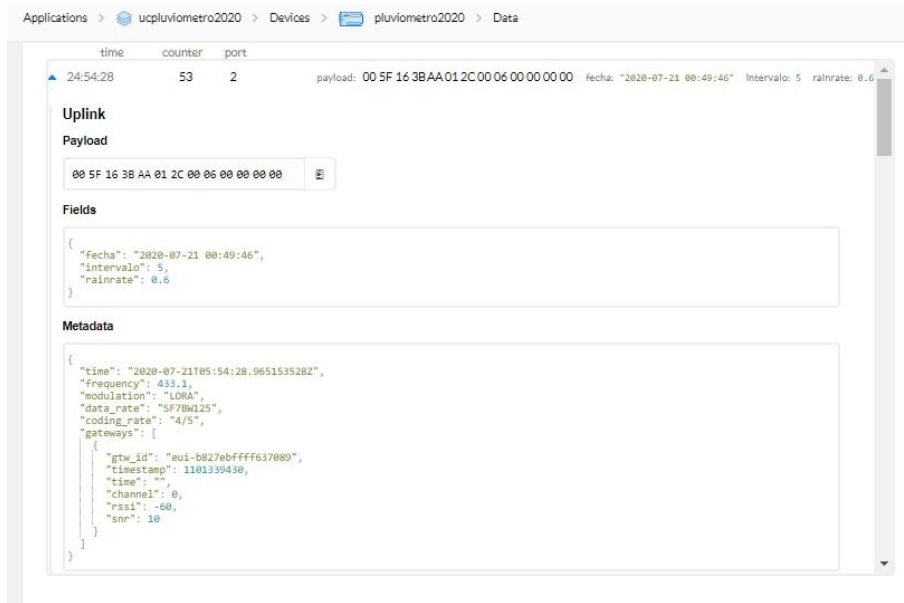


Figura 5.3: Visualización de datos recolectados por el sensor en la aplicación del servidor TTN.

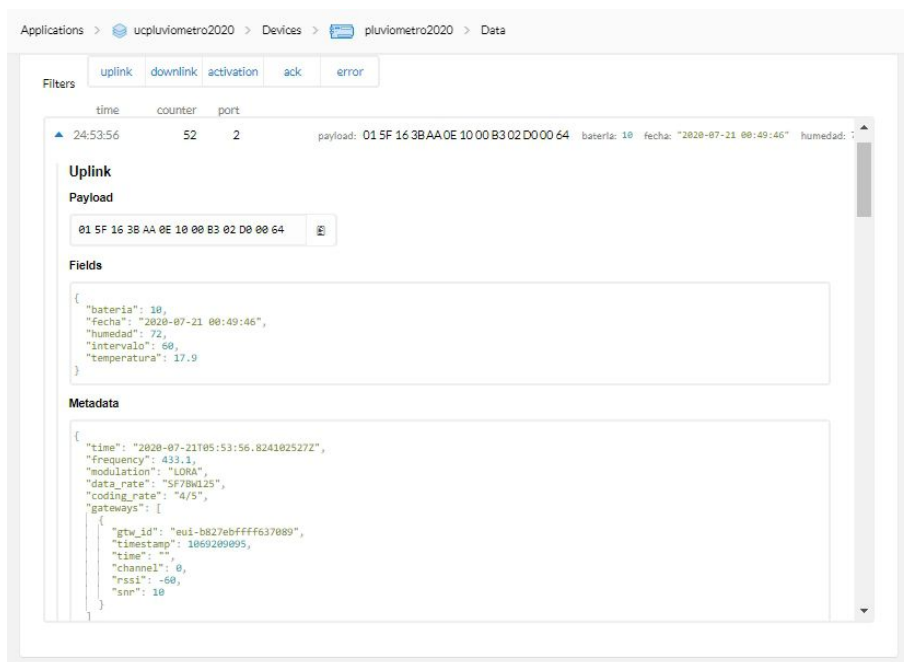


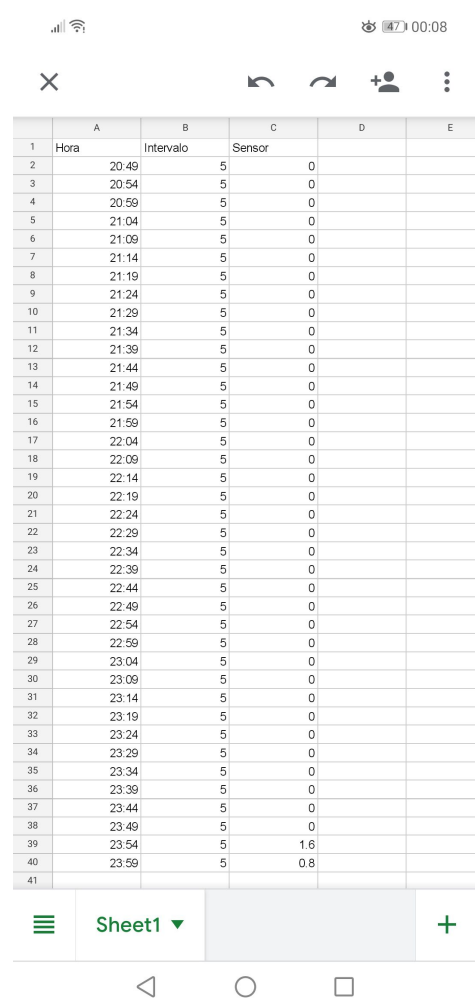
Figura 5.4: Visualización de datos del estado del dispositivo en la aplicación del servidor TTN.

5.2.2. Descarga de datos en *in-situ*

Para descargar los datos guardados en el nodo sensor, se desarrolló una aplicación móvil. Los nombres de los archivos almacenados en el nodo corresponden a la fecha de registro de datos. Cada archivo contiene las muestras registradas del sensor pluviométrico y del estado del dispositivo. La aplicación permite descargar los datos en el dispositivo móvil en formato (.csv). En las Figuras 5.5a y 5.5b se muestra un ejemplo de los datos del sensor pluviométrico visualizados en la aplicación móvil y descargados en formato (.csv).



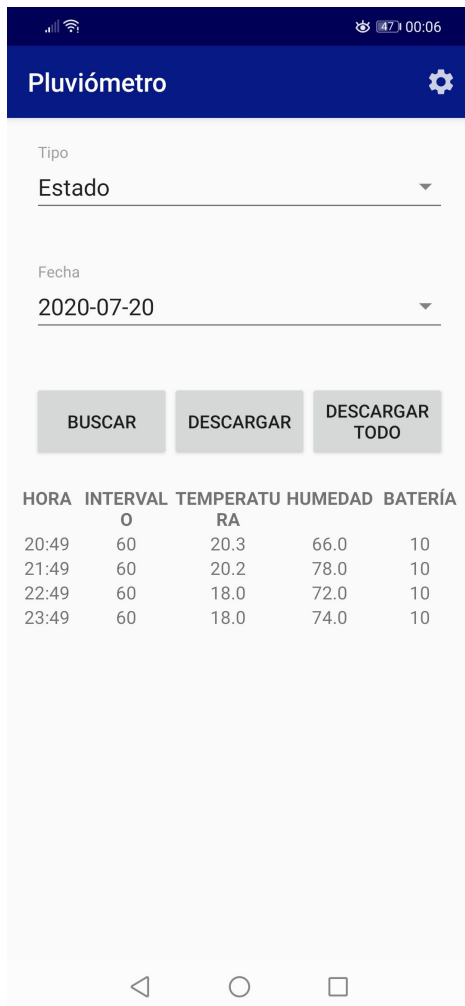
(a) Visualización de datos del sensor en la aplicación móvil.



(b) Visualización de datos del sensor almacenados en el dispositivo móvil en un archivo (.csv).

Figura 5.5: Descarga de datos del sensor pluviométrico.

En las Figuras 5.6a y 5.6b se muestran los datos del estado del dispositivo en la aplicación móvil y descargados en formato (.csv).



(a) Visualización de datos del estado del nodo en la aplicación móvil.



(b) Visualización de datos del estado del nodo almacenados en el dispositivo móvil en un archivo (.csv).

Figura 5.6: Descarga de datos de estado del dispositivo.

Teniendo un registro de la información con la tasa de muestreo designada a cada paquete: sensor 5 minutos y de estado del dispositivo 60 minutos (Sección 4.7), el dispositivo escribe 2520 bytes cada día en la memoria. De esta manera, con la memoria flash de 16 MB de capacidad, y el dispositivo funcionando de forma continua, el periodo de registro de datos es de $16777216 \text{ Bytes} / 2520 \text{ Bytes} = 6657$ días, o $6657 / 365 = 18,23$ años.

5.2.3. Configuración de parámetros del dispositivo

La aplicación móvil además de la descarga de datos, tiene la función de configuración de parámetros. Para verificar el funcionamiento se ingresaron los parámetros de tiempo de muestreo de cada uno de los paquetes y las credenciales para enlazarse a la aplicación del servidor TTN. En la Figura 5.7 se observa como se registraron los datos en la ventana de la aplicación móvil.

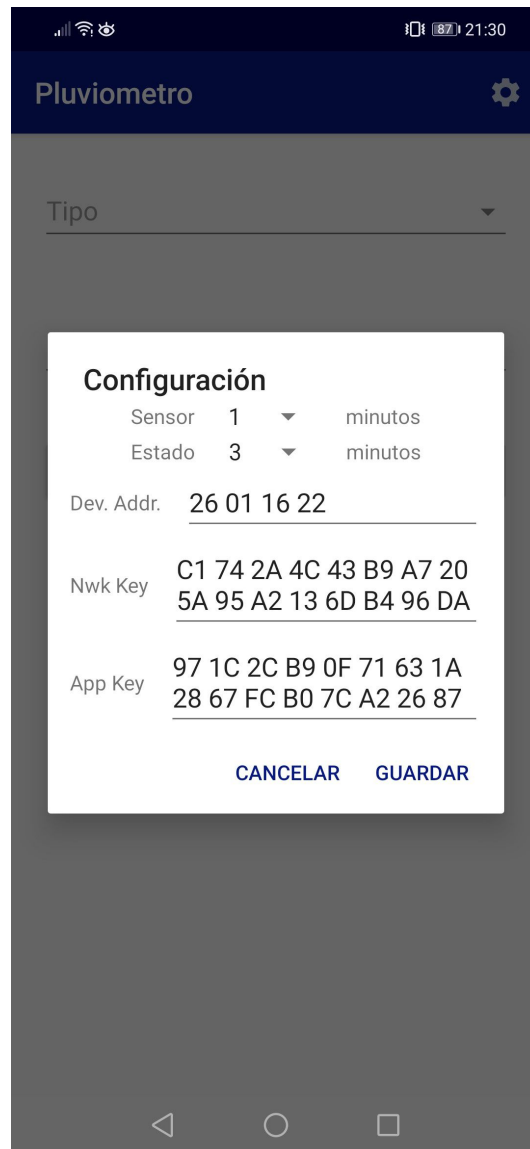


Figura 5.7: Ventana de configuración de parámetros del nodo.

En la Figura 5.8 se muestra como la conexión realizada fue exitosa entre el nodo y la aplicación del servidor TTN, las claves ingresadas desde el móvil son las mismas que la aplicación del servidor.

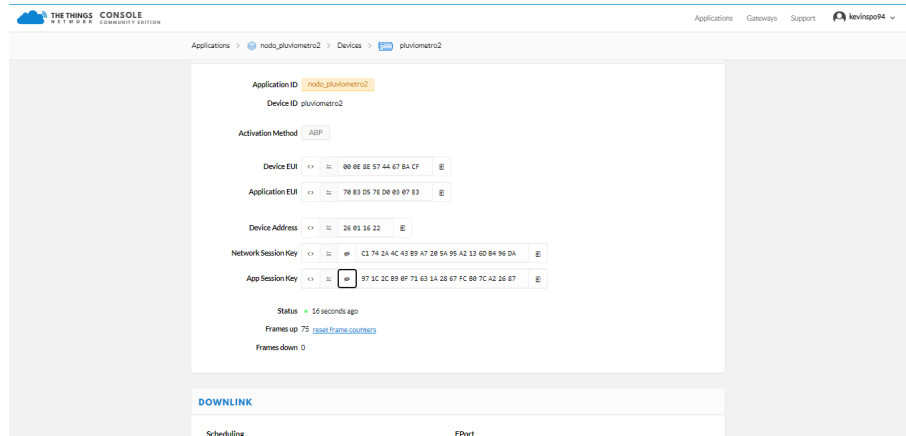


Figura 5.8: Verificación de la conexión del nodo en la aplicación del servidor TTN.

En la Figura 5.9, se muestran los datos que llegaron a la aplicación del servidor, donde el sensor llega cada minuto y el estado cada 3 minutos tal como se muestra en la configuración de la aplicación móvil.

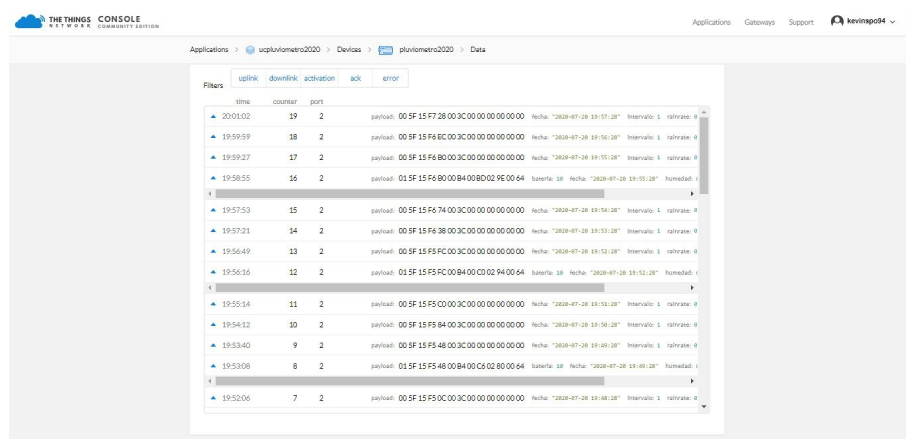


Figura 5.9: Verificación de los paquetes de datos en la aplicación del servidor TTN.

En las Figuras 5.10a y 5.10b se muestran los datos visualizados en el aplicación móvil que fueron almacenados en la memoria coincidiendo con los datos configurados.

| HORA | INTERVALO | SENSOR |
|-------|-----------|--------|
| 19:43 | 1 | 0.0 |
| 19:44 | 1 | 0.0 |
| 19:45 | 1 | 0.0 |
| 19:46 | 1 | 0.0 |
| 19:47 | 1 | 0.0 |
| 19:48 | 1 | 0.0 |
| 19:49 | 1 | 0.0 |
| 19:50 | 1 | 0.0 |
| 19:51 | 1 | 0.0 |
| 19:52 | 1 | 0.0 |
| 19:53 | 1 | 0.0 |
| 19:54 | 1 | 0.0 |
| 19:55 | 1 | 0.0 |
| 19:56 | 1 | 0.0 |
| 19:57 | 1 | 0.0 |

(a) Visualización de datos del sensor en la aplicación móvil .

| HORA | INTERVALO | TEMPERATU O | HUMEDAD | BATERÍA |
|-------|-----------|----------------|---------|---------|
| 19:43 | 3 | 20.9 | 61.0 | 10 |
| 19:46 | 3 | 20.8 | 74.0 | 10 |
| 19:49 | 3 | 19.8 | 64.0 | 10 |
| 19:52 | 3 | 19.2 | 66.0 | 10 |
| 19:55 | 3 | 18.9 | 67.0 | 10 |

(b) Visualización de datos del estado del nodo en la aplicación móvil.

Figura 5.10: Descarga de datos del dispositivo aplicando la configuración de parámetros(1 y 3 minutos).

Desde la aplicación móvil se pueden cambiar parámetros como la frecuencia con la que se envían los datos recolectados y las credenciales para la conexión a otra aplicación del servidor TTN. En la Figura 5.11 se observa como se ingresan los nuevos parámetros en la configuración, los paquetes de datos del sensor cada 2 minutos y de estado del nodo cada 5 minutos. Además, de las nuevas credenciales [DevAddr](#), [NwkSKey](#) y [AppSKey](#),

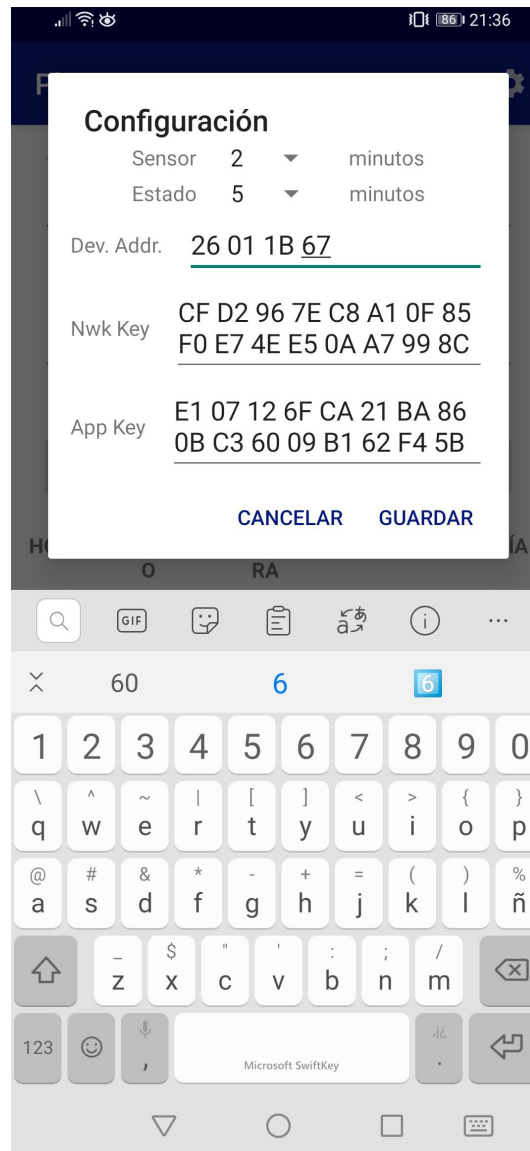


Figura 5.11: Ventana de configuración de parámetros del nodo.

En la Figura 5.12 muestra los datos y claves del nodo en la aplicación del servidor TTN, donde se muestra la conexión exitosa con las claves ingresadas desde el móvil.

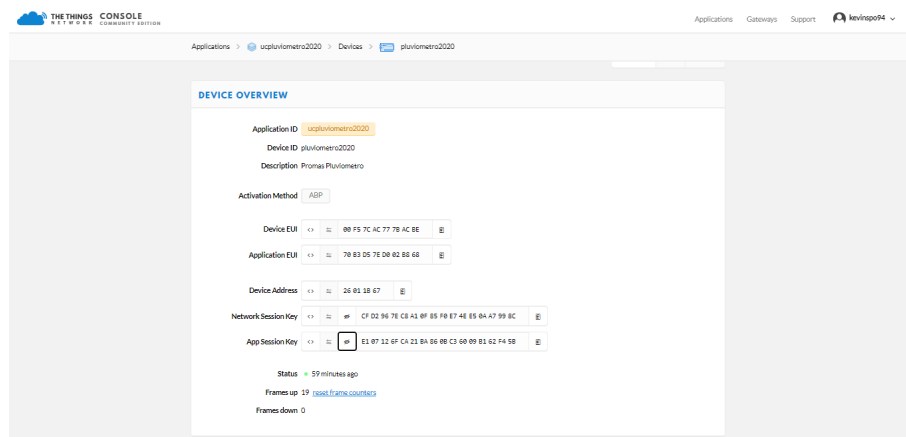


Figura 5.12: Verificación de la conexión del nodo en la aplicación del servidor TTN.

5.3. Pruebas de consumo energético

El dispositivo desarrollado está destinado a funcionar en ubicaciones donde generalmente no se tiene alimentación desde la red eléctrica, por lo que es importante conocer su consumo energético. El nodo fue alimentado con 3,3 V, el cual es el voltaje óptimo de funcionamiento.

El análisis del consumo se realiza utilizando la metodología Nguyen [38]. Los tiempos activos por componente (ciclo de trabajo) se utilizan para calcular el consumo total. El período de tiempo del ciclo de trabajo es de 5 minutos. Este tiempo lo establecen los requisitos de las aplicaciones del PROMAS. Durante este período, la información se transmite al centro de datos.

Para el análisis, el registrador de datos se divide en tres partes: adquisición y control, el sensor y la transmisión. El proceso de adquisición del sensor de lluvia y el control de los dispositivos permanece activo todo el tiempo. Por lo tanto, el dispositivo involucrado en este proceso tiene un ciclo de trabajo del 100 %. El consumo del sensor de lluvia es 0,034 mA para el valor máximo medido. La precipitación máxima en el área a partir de los datos registrados por PROMAS es de 22 mm en una hora (entre los años 2015 y 2017), con este valor el sensor consume 0,088 mA [5].

La transmisión por medio del transceptor LoRa está configurado con un factor de dispersión de 7 y un ancho de banda de 125 KHz. El ciclo de trabajo del transceptor es de 0,062 % si se considera un tiempo de transmisión de 186 ms. La Tabla 5.1 presenta el consumo promedio de cada componente del registrador de datos y el consumo general, que es de 28,510 mW.



Tabla 5.1: Estimación del consumo de energía del registrador de datos.

| Dispositivos | Modo | Consumo de Corriente (mA) | Ciclo de Trabajo (%) | Corriente Promedio (mA) | Potencia (mW) ($I \times 3,3V$) |
|-------------------------------------|--------|---------------------------|----------------------|-------------------------|--------------------------------------|
| Adquisición y Control | Activo | 8,624 | 100,00 | 8,624 | 28,459 |
| | Reposo | 0,000 | 0,00 | | |
| Sensor | Activo | 0,088 | 1,29 | 0,001 | 0,003 |
| | Reposo | 0,000 | 98,71 | | |
| Transmisión | Activo | 23,896 | 0,06 | 0,014 | 0,048 |
| | Reposo | 0,000 | 99,38 | | |
| Consumo de energía promedio general | | | | | 28,510 |

En las pruebas de laboratorio realizadas, se usó una fuente de alimentación variable continua (0 – 20) V, la cual fue configurada a 3,3 V, voltaje requerido para el funcionamiento del registrador de datos. El consumo de corriente fue medido con un amperímetro conectado en serie entre el pin positivo de la fuente y el pin de entrada (V_{in}) del registrador de datos. En el proyecto de investigación “Aplicación de tecnologías inalámbricas al sistema de predicción de caudales en la cuenca del río Tomebamba” se usan baterías de ácido sólido de 7200 mAh-12 V. Por lo que en este proyecto se usará el mismo tipo de batería integrando un controlador de carga a 3,3 V en pruebas de campo, esta batería contiene una carga mínima del 25 %, el tiempo teóricamente para el próximo ciclo de recarga es $0,75 \times 7200 \text{ mAh} \times 12 \text{ V} / 28,510 \text{ mW} = 2272,886 \text{ horas} = 94 \text{ días}$ aproximadamente.



Conclusiones y Recomendaciones

En este capítulo se presentan las conclusiones finales del trabajo, las recomendaciones que se deben aplicar y las propuestas para el desarrollo de trabajos futuros.

6.1. Conclusiones

El diseño del registrador de datos se probó en un *protoboard* con componentes de inserción *through hole*. Luego de realizar las pruebas y validar el circuito con éxito, se procedió a enviar el diseño con componentes de montaje superficial a una empresa local para su manufactura. De acuerdo a los objetivos planteados, el registrador de datos utiliza un microcontrolador y un transceptor de bajo consumo (compacto en montaje superficial), además de la integración del sensor pluviométrico.

En primera instancia, se pensó en desarrollar un diseño PCB para el *gateway* **WiFi**, pero debido a su alto costo de implementación y manufactura en la empresa local, se procedió a realizar la compra del mismo, por su alta disponibilidad en el mercado y bajo costo.

La implementación del protocolo **LoRaWAN** permitió la transmisión de los datos pluviométricos remotamente hacia el servidor **LoRaWAN**, con la ventaja principal de brindar al usuario el acceso a la información en cualquier momento y lugar, debido a que el acceso *in-situ* a las estaciones meteorológicas resulta complicado, por su distribución en áreas geográficamente extensas y de difícil acceso.

El desarrollo de la aplicación para dispositivos móviles Android se realizó con herramientas de código abierto, ésta tuvo un funcionamiento correcto en cada una de las pruebas realizadas. La aplicación móvil permite descargar los datos, y configurar los parámetros del nodo para la transmisión y almacenamiento.

Las pruebas que se realizaron para la transmisión de datos hacia el servidor **LoRaWAN** fueron satisfactorias. Debido a la emergencia sanitaria no se pudieron realizar las pruebas de campo y envío de datos hacia el servidor del **PROMAS**, por esta razón se usó otro servidor **LoRaWAN**, en este caso **TTN**, el cual permitió observar los datos en su interfaz web.

El *gateway* **WiFi** es móvil y se puede llevar hacia cualquier nodo evitando incluir un módulo **WiFi** en cada nodo. Para la validación del funcionamiento de la descarga de datos *in-situ*, se realizó la conexión del *gateway* **WiFi** al nodo. Los datos pluviométricos fueron adquiridos y descargados correctamente en la aplicación móvil, validando así los requerimientos solicitados por el **PROMAS**.



6.2. Recomendaciones

- En el diseño del circuito se debe separar la parte del *bootloader* con la del circuito completo, debido a que al subir los programas se energiza con 5 V, y cuando el circuito se pone en funcionamiento a 3,3 V. Esto ocasiona problemas con la comunicación **SPI**, daños en los dispositivos conectados por cambios de tensión, interferencia con los esclavos, además de fallos al momento de subir los programas en el microcontrolador.
- En el diseño se recomienda colocar diodos en los puertos **MISO** de cada dispositivo. Estos diodos se deben colocar cuando se conectan varios dispositivos esclavos en el bus **SPI** del microcontrolador. Se deben usar para que no existan interferencias con el microcontrolador y cause fallos en el funcionamiento del dispositivo.
- Se debe colocar un filtro a la entrada analógica del microcontrolador que realiza la lectura de los datos del sensor pluviométrico, debido al ruido que puede generar el balancín del sensor.
- Se debe configurar correctamente todos los parámetros de la red **LoRaWAN** en especial la frecuencia y credenciales en cada uno de los dispositivos.

6.3. Trabajos futuros

En esta sección, se explica cómo podría ser mejorado el registrador de datos pluviométricos y futuras aplicaciones.

- El registrador de datos pluviométricos es alimentado con 3,3 V por lo que es necesario desarrollar un controlador de carga, así se puede integrar una batería para la alimentación con carga solar.
- La adición de varios sensores podrían ser integrados, ya que el registrador de datos cuenta con pines analógicos y digitales extras. En este caso, se podría usar el dispositivo para otras aplicaciones.
- Debido a un alto porcentaje de usuarios de dispositivos móviles con sistema operativo **iPhone Operating System (iOS)**, se hace necesario desarrollar una aplicación móvil para dichos dispositivos.
- En caso de no tener fuente con salida a 3,3 V, en el diseño de la **PCB** se debe añadir un divisor de tensión en la parte de alimentación de la memoria flash, debido a que las baterías existentes en el mercado son de 3,7 V y el voltaje máximo permitido de la memoria es de 3,6 V.



Especificaciones Técnicas de dispositivos

A.1. Microcontrolador ATmega1284p

El ATmega1284p es un microcontrolador de baja potencia basado en la arquitectura [Reduced Instruction Set Computer \(RISC\)](#) de AVR. Al ejecutar instrucciones en un solo ciclo de reloj, el ATmega1284p logra rendimientos cercanos a 1 [Microprocessor without Interlocked Pipeline Stages \(MIPS\)](#) por MHz. Esto permite al diseñador del sistema optimizar el consumo de energía del dispositivo en comparación con la velocidad de procesamiento. Combina una [Central Processing Unit \(CPU\) RISC](#) de 8 bits con un *flash* auto-programable en el sistema de un chip monolítico [39]. En la Figura A.1 se muestra el diagrama de pines de este microcontrolador. El ATmega1284p ofrece las siguientes características:

- Rango de Voltaje de Operación (V): 1,8 – 5,5.
- Rango de temperatura (°C): -40 a 85.
- Consumo de energía a 1 MHz, 1,8 V, 25 °C.
- Modo activo: 0,3 mA.
- Modo de apagado: 0,1 μ A.
- Modo de ahorro de energía: 0,6 μ A (incluido 32 kHz RTC).
- 128 Kbytes de flash programable en el sistema con capacidades de lectura mientras se escribe.
- EEPROM de 4 Kbytes.
- SRAM de 16 Kbytes.
- 32 registros de trabajo de propósito general, contador de tiempo real (RTC).
- Tres Temporizador / Contadores flexibles con modos de comparación.
- PWM.
- ADC de 8 bits y 8 canales.

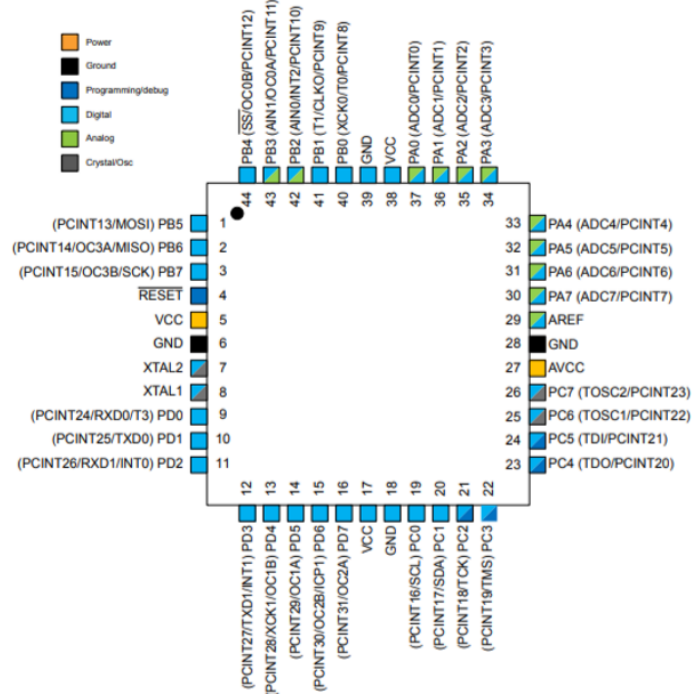


Figura A.1: Diagrama de pines del microcontrolador ATmega1284[39].

A.2. Transceptor Ra02 SX1278

Ra-02 es un dispositivo de montaje superficial que contiene un módulo LoRa SX1278 (Ver Figura A.2).



Figura A.2: Diagrama de pines del Ra02 SX1278 [40].

A.2.1. SX1278

El SX1278 es un transceptor de largo alcance (137 MHz a 525 MHz) y baja potencia[41]. Este módulo tiene las siguientes características:

- LoRaTM Modem 168 dB presupuesto de enlace máximo.
- +20 dBm - 100 mW de salida de RF constante vs.
- Suministro de VPA de alta eficiencia de +14 dBm.
- Velocidad de bits programable de hasta 300 kbps.
- Alta sensibilidad: hasta -148 dBm.
- Parte delantera a prueba de balas: IIP3 = -11 dBm.
- Baja corriente RX de 9,9 mA, retención de registro de 200 nA.
- Sintetizador totalmente integrado con una resolución de 61 Hz.
- Modulación FSK, GFSK, MSK, GMSK, LoRa y OOK.
- Sincronizador de bits incorporado para recuperación de reloj.
- Detección de preámbulos.
- Rango dinámico de 127 dB RSSI.
- Motor de paquetes de hasta 256 bytes con [Cyclic Redundancy Check \(CRC\)](#).

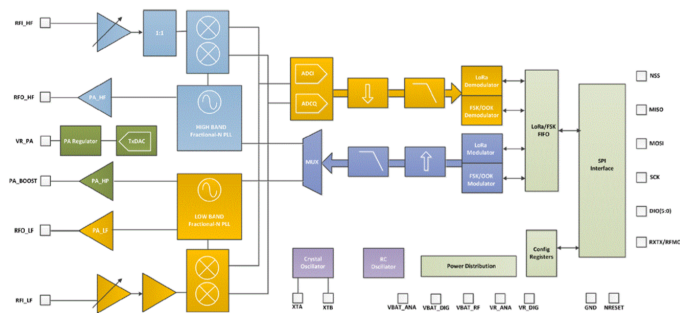


Figura A.3: Diagrama de bloques general de un transceptor SX1278-78 [41].

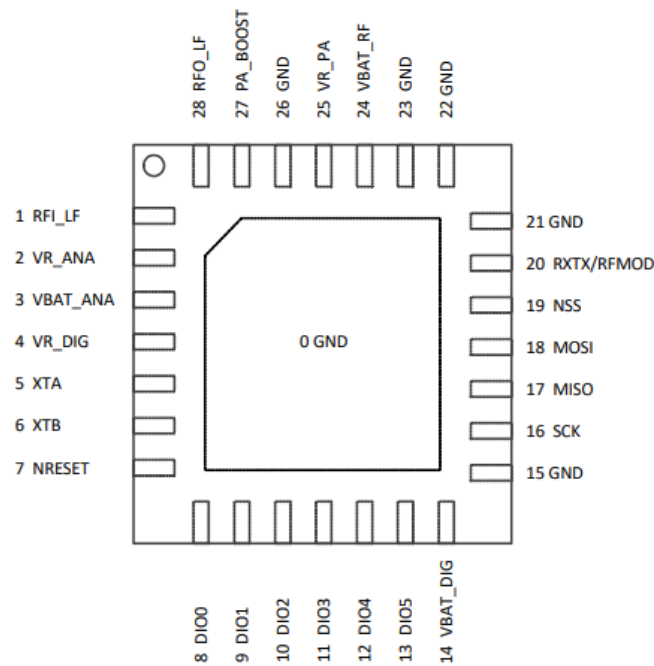


Figura A.4: Diagrama de pines del SX1278 [32].

A.3. Sensor DHT11

El DHT11 es un sensor de temperatura y humedad digital de bajo costo (Ver Figura A.5). Las características principales del sensor se detalla a continuación:

- Alimentación: 3 V a 5 V.
- Rango de medición de temperatura: 0 a 50°C.
- Precisión de medición de temperatura: +/- 2,0°C.
- Resolución Temperatura: 0,1 °C.
- Rango de medición de humedad: 20 a 90 RH.
- Precisión de medición de humedad: 4 RH.
- Resolución Humedad: 1 RH.

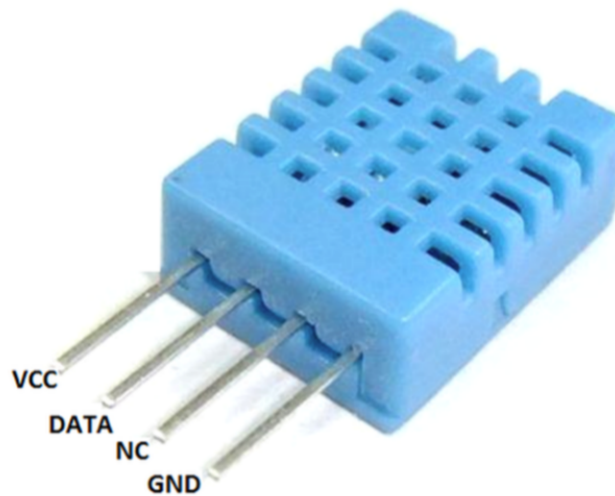


Figura A.5: Sensor DHT11 .[42]

A.4. RTC DS3231

El **RTC** es un módulo de alta precisión que administra las funciones de adquisición de tiempo en un entorno real (Ver Figura A.6), este cuenta con las siguientes características:

- El reloj de tiempo real cuenta los segundos, los minutos, las horas, la fecha del mes, el mes, el día de la semana y el año, con una compensación de año bisiesto válida hasta 2100.
- Precisión ± 2 ppm de 0.
- 0°C a $+40^{\circ}\text{C}$.
- Precisión $\pm 3,5$ ppm de -40 .
- RST Output.
- Resolución Humedad: 1 RH.
- Dos alarmas de hora del día.
- Señal de salida de onda cuadrada programable.
- La interfaz serial simple se conecta a la mayoría de Microcontroladores.
- Interfaz I2C rápida (400 kHz).
- Entrada de batería de respaldo para el cronometraje continuo.
- La operación de bajo consumo de energía prolonga el tiempo de ejecución de la batería de respaldo.
- Operación 3,3 V a 5 V.
- Rangos de temperatura de funcionamiento: comercial (0°C a $+70^{\circ}\text{C}$) e industrial (-40°C a $+85^{\circ}\text{C}$).

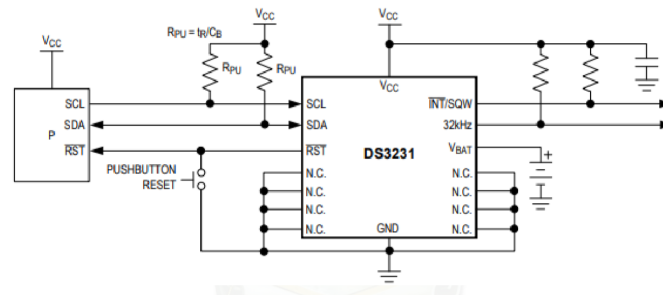


Figura A.6: Circuito típico de operación del RTC DS3231[43].

A.5. Memoria externa W25Q128JV

- Capacidad: 128 M-bit / 16 M-byte.
- SPI estándar: CLK, / CS, DI, DO.
- SPI dual: CLK, / CS, IO0, IO1.
- Quad SPI: CLK, / CS, IO0, IO1, IO2, IO3.
- Restablecimiento de Software y Hardware.
- Flash Serial de Alto Rendimiento.
- Velocidad de transferencia de datos continua de 66 MB/S.
- Retención de datos de más de 20 años.
- Lectura continua y eficiente.
- Baja potencia, amplio rango de temperatura.
- Suministro individual de 2,7 a 3,6 V.
- $-40\text{ }^{\circ}\text{C}$ a $+85\text{ }^{\circ}\text{C}$ rango de operación.
- Arquitectura flexible con sectores de 4 KB.
- Funciones de seguridad avanzadas.
- Software y hardware de protección contra escritura.
- Bloqueo de la fuente de alimentación.
- Registros de seguridad de 3×256 bytes con bloqueos **One Time Password (OTP)**.
- Bits de registro de estado volátiles y no volátiles.

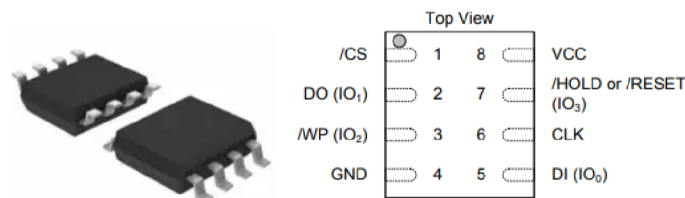


Figura A.7: Diagrama de pines de la memoria externa[44].

A.6. Modulo WiFi Kit 8

- Tamaño: $50,8 \times 18,2 \times 6,69$ mm.

- Interfaz: Micro USB.
- Programación disponible para entornos Arduino y NodeMCU.
- Flash: 32 Mbyte.
- Voltaje de funcionamiento: 3,3V 7V.
- Chip maestro: ESP8266.
- OLED: 0,91 pulgadas/128 × 32.
- Chip adaptador USB: CP2102.
- Temperatura de funcionamiento: -40 a 90 °C.
- Modo de soporte: softAp, estación, softAp + estación, WIFI-Direct.
- Potencia de salida: 19,5 dBm @ 11b, 16,5 dBm @ 0,39 oz, 15,5 dBm @ 11n.
- Recursos de hardware: UART x 2; SPI x 1; I2C x 1; I2S x 1; 10 – bits ADC; 17 general GPIO.

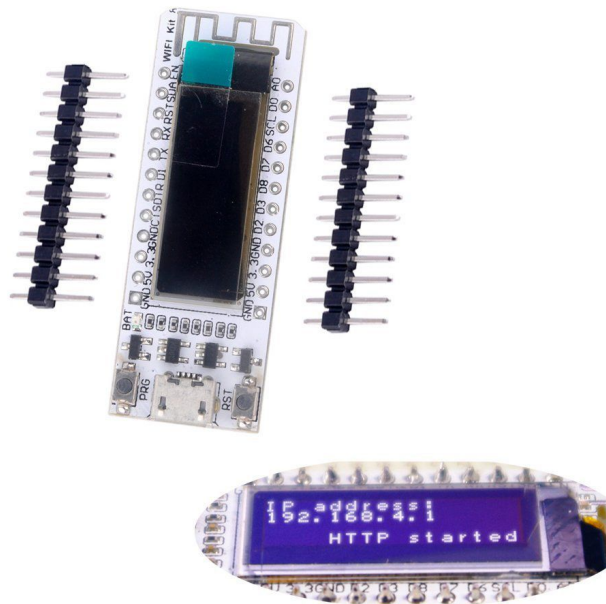


Figura A.8: Modulo Wifi Kit 8 [45].

A.7. Pluviómetro

- Tasa máxima de lluvia: 12,7 cm (5 pulg.) por hora.
- Precisión de calibración: $\pm 1,0\%$ (hasta 1 pulgada / hora para el RG3 o 20 mm / hora para el RG3-M).
- Resolución: 0,01 pulg. (RG3) o 0,2 mm (RG3-M).
- Rango de temperatura de funcionamiento: 0 ° a + 50 ° C (+ 32 ° a + 122 ° F).
- Clasificación ambiental: Resistente a la intemperie.
- Carcasa: cucharón de aluminio de 15,24 cm (6 pulg.).
- Mecanismo de cucharón basculante: eje de acero inoxidable con cojinetes de latón.
- Dimensiones: 25,72 cm de altura x 15,24 cm de diámetro (10,125 x 6 pulg.); Orificio de recepción de 15,39 cm (6,06 pulg.).



- Peso: 1, 2 kg (2, 5 lb).



Figura A.9: Pluviómetro ONSET [46].



Configuración y programación de dispositivos

En este apéndice, se presenta un breve resumen de la configuración y programación de los dispositivos.

B.1. Configuración del registrador de datos

Para el uso del registrador de datos, se requiere que el microcontrolador contenga un gestor de arranque (*bootloader*). Éste es un software alojado en la memoria del **Microcontroller Unit (MCU)** que permite programar el prototipo por medio del puerto serie. El *bootloader* de Arduino verifica el arranque del **MCU**, comprueba si se está intentando programar, si es así, el programa es grabado en la memoria, caso contrario, ejecuta el programa que fue grabado últimamente. Los fabricantes ocasionalmente envían sus dispositivos sin cargar el *bootloader*. Por lo tanto, éste se puede grabar usando una placa de Arduino, la cual actúa como programador. El Arduino se configura como *master*, y el registrador de datos que se programa como *slave*. La comunicación entre el Arduino y el registrador de datos es **SPI**. La comunicación entre el ordenador y el Arduino se realiza utilizando el puerto serie. Lo primero que se configura es el *Arduino*, se comienza cargando el *Sketch "ArduinoISP"* que se encuentra en la librería de ejemplos de Arduino, lo que hace que el Arduino se convierta en un programador **SPI**. Luego hay que seguir las siguientes instrucciones:

1. Ir a **Archivo > Preferencias**; e ingresar en el **Gestor de URLs adicionales de tarjetas** el siguiente enlace https://mcdude.github.io/MightyCore/package_MCUdude_MightyCore_index.json, tal como se muestra en la Figura B.1, esto servirá para que el IDE Arduino reconozca el **MCU ATmega 1284p**.

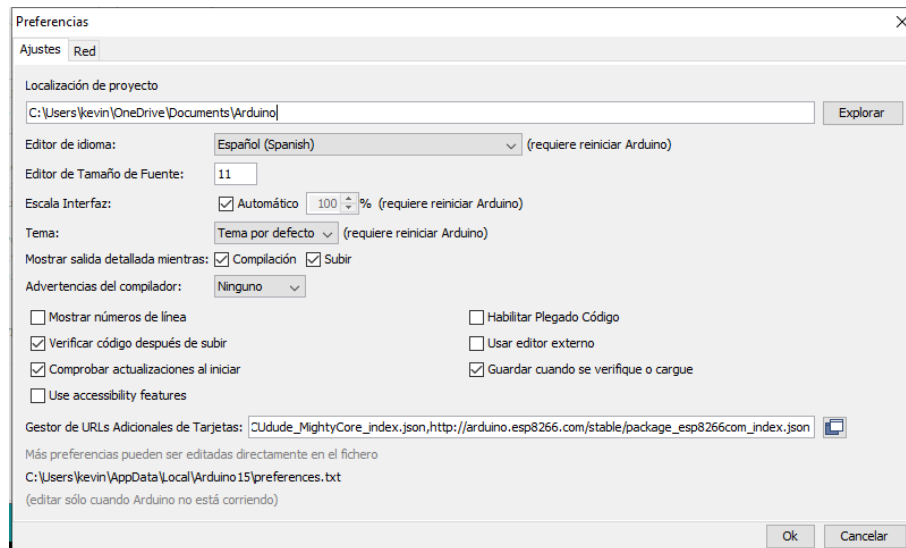


Figura B.1: Configuración para Gestor de URLs Adicionales de tarjetas.

2. Dirigirse a **Herramientas >Placa >Gestor de tarjetas**, buscar el paquete de *MightyCore* e instalarlo, como se muestra en la Figura B.2. Este paquete contiene la tarjeta ATmega1284/p.
3. Desconectar el Arduino del ordenador.
4. Conectar el Arduino y el registrador de datos.
5. Conectar los pines del bus **SPI**, el **VCC** y el **GND** entre ambas placas, como se indica en las Figuras B.3 y B.4.

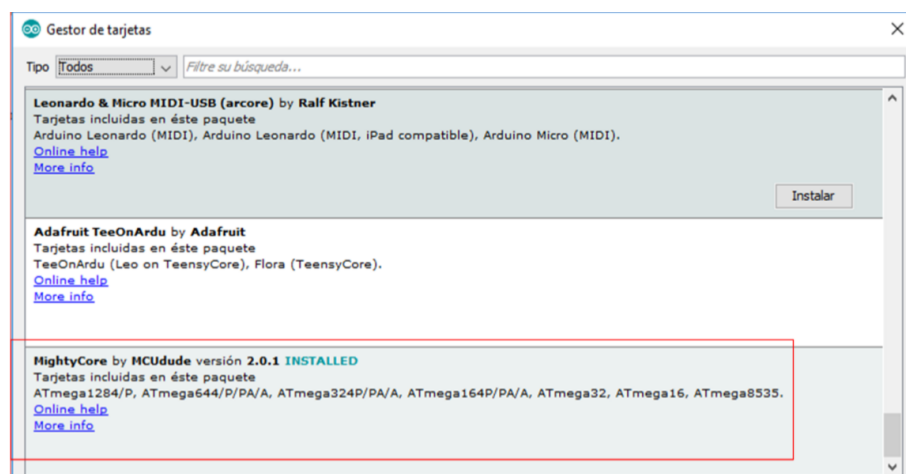


Figura B.2: Instalación de tarjetas de MightyCore.

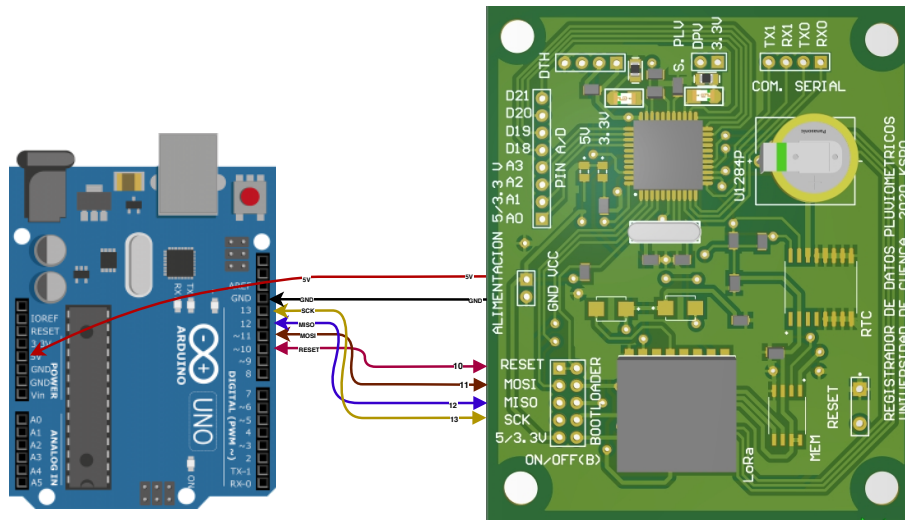


Figura B.3: Interconexión entre *master* y *slave*.

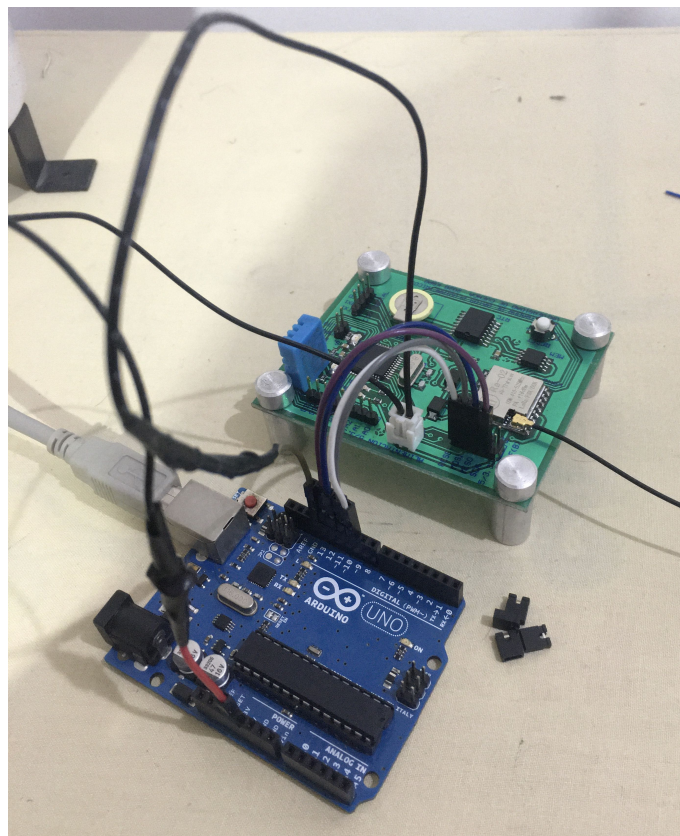


Figura B.4: Interconexión entre *master* y *slave*.

6. En **Herramientas** del IDE Arduino, seleccionar la placa “ATmega1284p”, clock “8 MHz external”, programador “Arduino as ISP (MightyCore)”, las demás características se las mantiene con los valores por defecto, como se muestra en la Figura B.5.

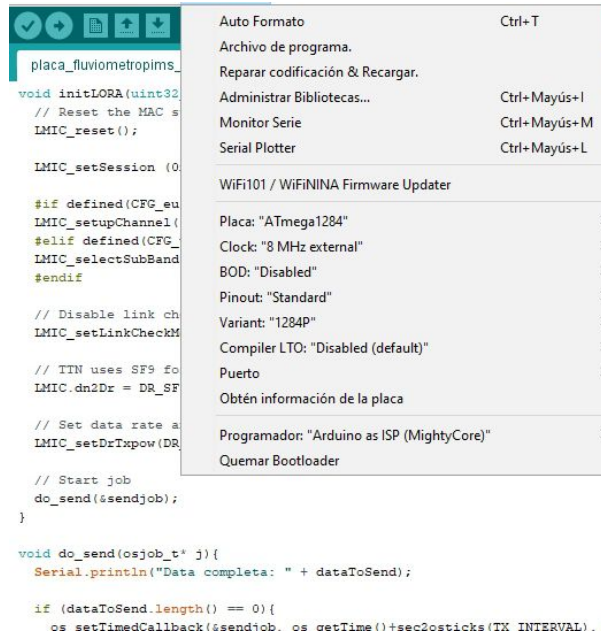


Figura B.5: Selección de características para el ATmega1284P.

7. Seleccionar la opción **Quemar Bootloader**, si todo está conectado correctamente se indicará un mensaje diciendo que el *bootloader* fue subido con éxito como se indica en la Figura B.6.

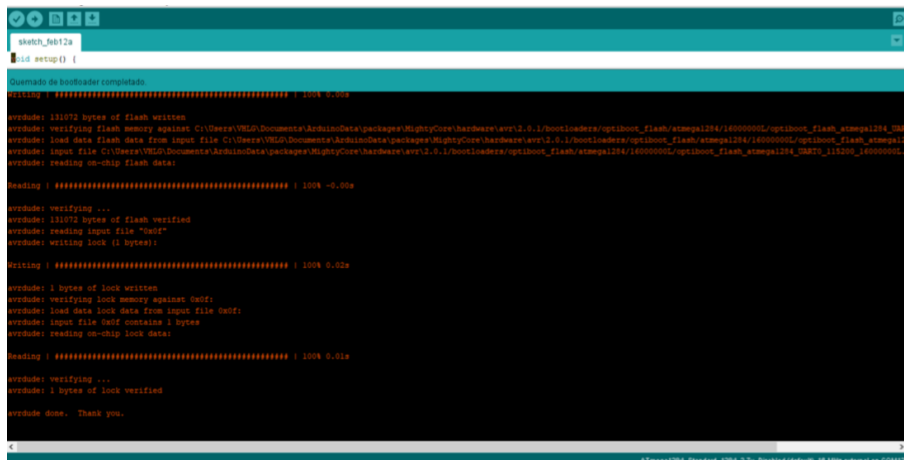


Figura B.6: Quemado del bootloader completado.

Después de este proceso, el registrador de datos está listo para ser programado y utilizado [4].

B.2. Configuración del software

La Figura B.7, presenta un diagrama de flujo que describe la estructura del programa desarrollado para el microcontrolador, donde se definen las tareas que éste debe ejecutar. A continuación, se detallan estas tareas:

- Las librerías de los diferentes componentes utilizados son verificadas, así como: el sensor de temperatura y humedad, el módulo **RTC**, la memoria externa, la comunicación **SPI** y la librería **LoRaWAN** del módulo



LoRa.

- Se establece la comunicación **I2C** y **SPI** entre los componentes y el microcontrolador, mientras se escriben los registros para el transceptor SX1278, la adquisición del tiempo por el módulo **RTC** y de los sensores.
- Se adquiere los datos de los dispositivos y los prepara para ser transmitirlos y almacenarlos en la memoria externa, estableciendo los tiempos de muestreo y tiempos de activación de cada uno de ellos.
- Se establecen los intervalos de tiempo para la adquisición de datos de los sensores, para ser transmitidos y almacenados en la memoria externa.
- Los datos adquiridos por el microcontrolador son sincronizados, almacenados en memoria, enviados al transceptor SX1278 y al puerto serial **UART**, para la transmisión remota **LoRaWAN** y la descarga de datos *in-situ* respectivamente.

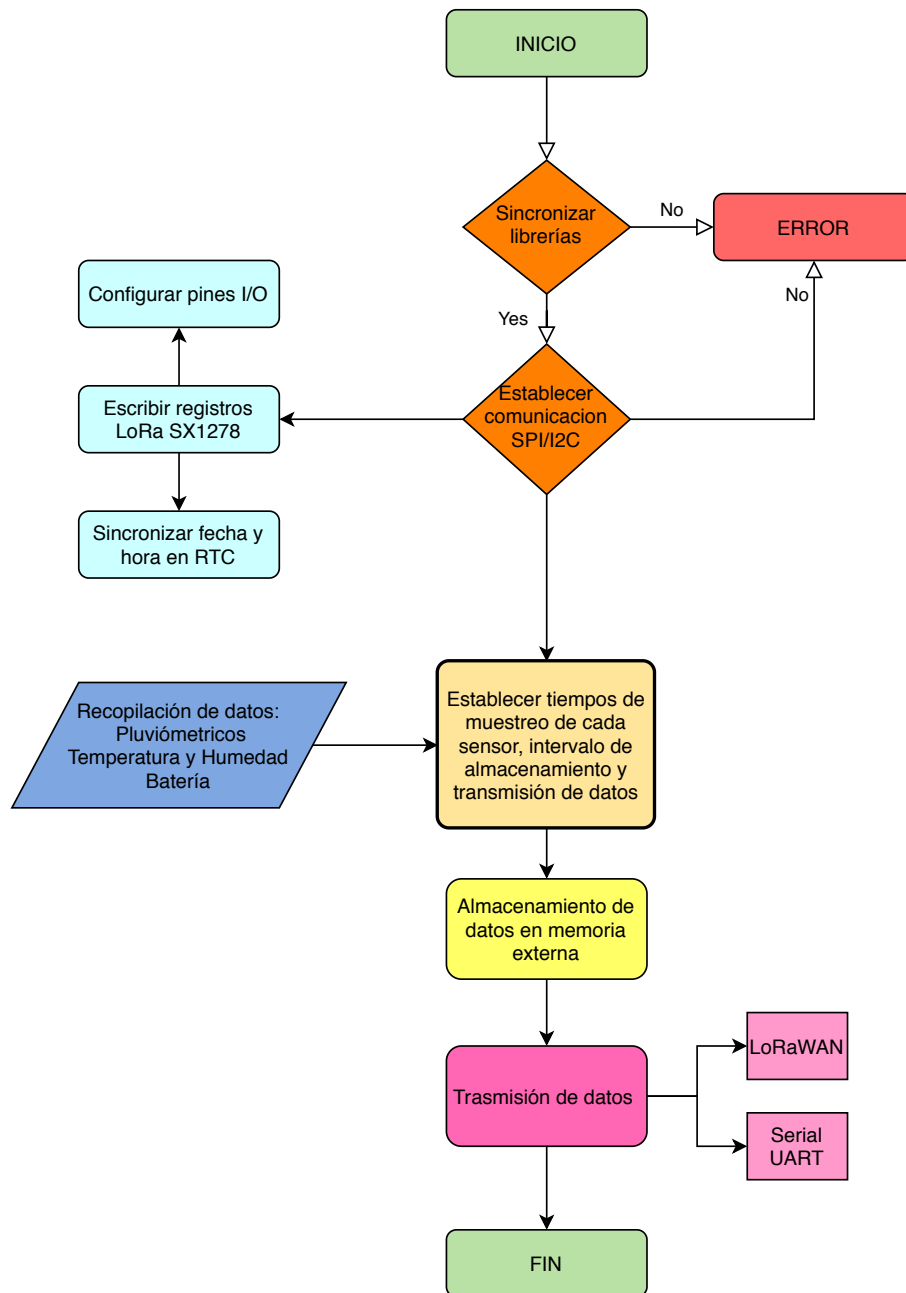


Figura B.7: Diagrama de flujo del funcionamiento del software del registrador de datos pluviométricos.

B.2.1. Instalación de librerías

Las librerías requeridas por los dispositivos, se las instala usando el gestor de librerías del IDE Arduino. Para lo cual hay que ir a **Programa>Incluir Librería>Gestionar Librerías** (Ver Figura B.8).

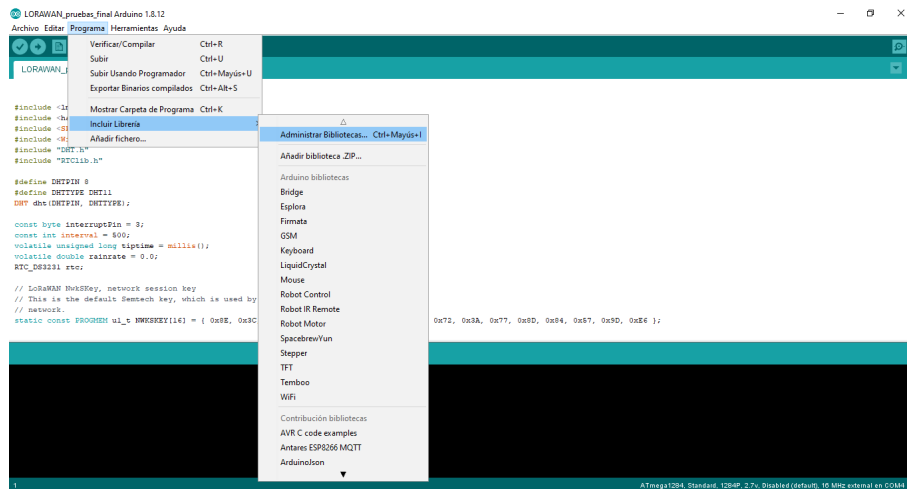


Figura B.8: Ventana de opciones para instalar librerías.

A continuación, se muestra la instalación de las librerías requeridas para el funcionamiento del registrador de datos.

RTCLib

La librería RTCLib permite programar el RTC (DS3231) en el entorno de Arduino. Para instalar la librería hay que acceder a **Gestionar Librerías**, buscar en la ventana *RTCLib*, seleccionar *RTCLib by Adafruit* y presionar **Instalar** (Figura B.10).

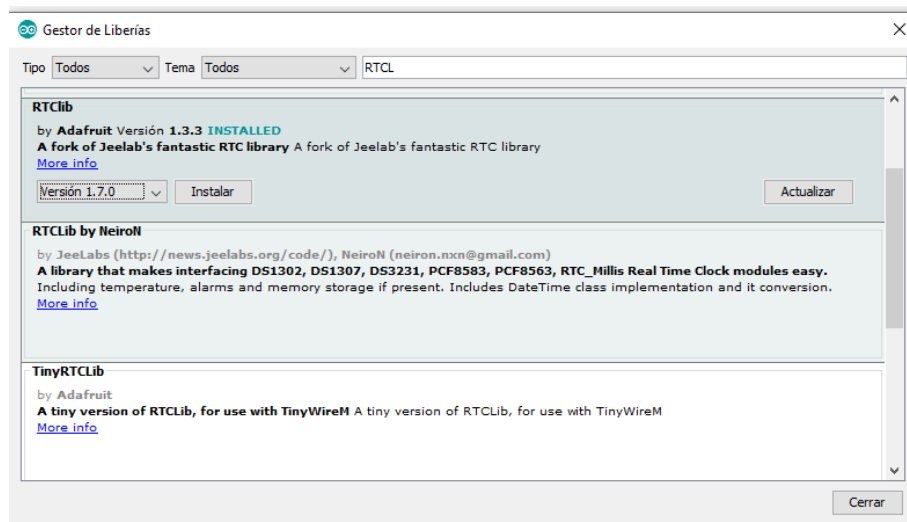


Figura B.9: Ventana de instalación de librería RTCLib.

DHT

La librería DHT permite obtener los valores de temperatura y humedad del sensor DHT11. Para instalar la librería hay que acceder a **Gestionar Librerías**, buscar en la ventana *DHT*, seleccionar *DHT sensor library by Adafruit* y presionar **Instalar** (Figura B.10).

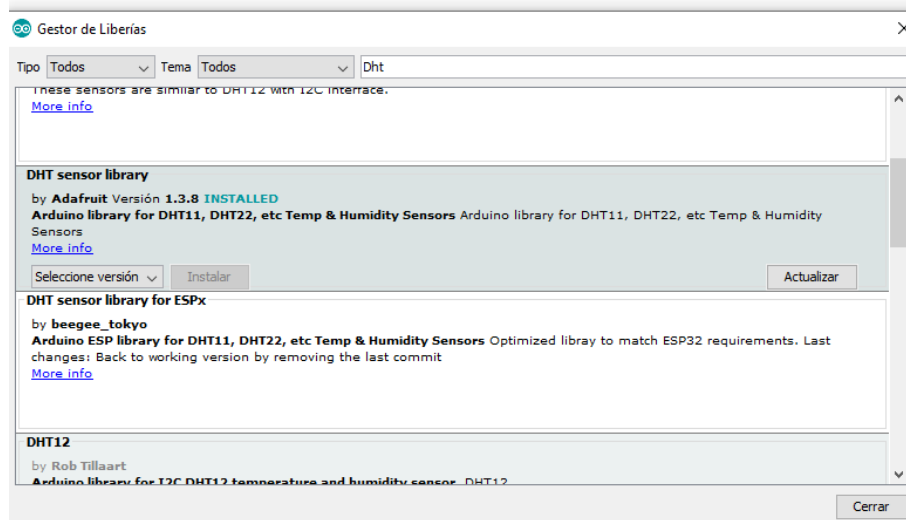


Figura B.10: Ventana de instalación de librería DTH.

SerialFlash

SerialFlash proporciona acceso de baja latencia y alto rendimiento a la memoria Flash **SPI** con una interfaz similar a un sistema de archivos para usar con placas compatibles con IDE Arduino. Para instalar la librería hay que acceder a **Gestionar Librerías**, buscar en la ventana *SerialFlash*, seleccionar *SerialFlash by Paul Stoffregen* y presionar **Instalar** (Figura B.11).

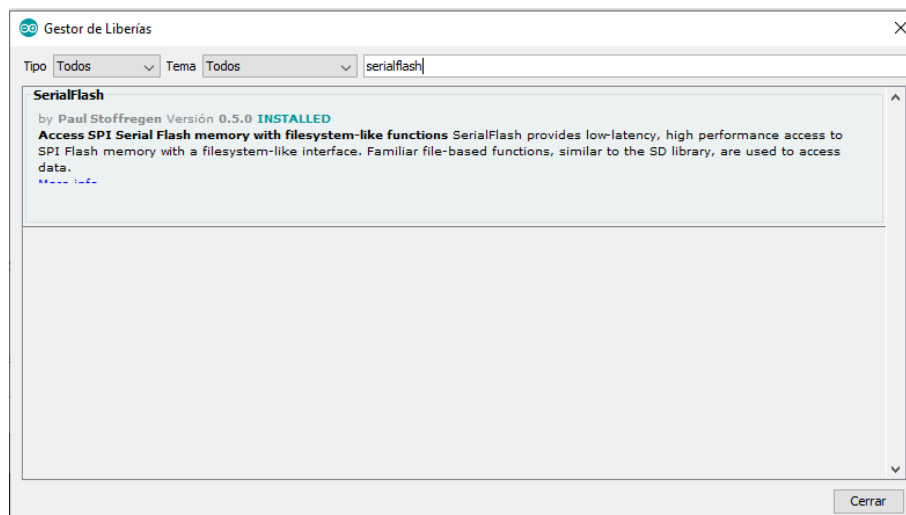


Figura B.11: Ventana de instalación de librería SerialFlash.

Heltec ESP866 Dev-Boards

Para desarrollar el software en el *gateway WiFi*, se debe instalar la tarjeta ESP8266 y la librería de la misma. Para instalar la tarjeta en el entorno de IDE Arduino, se dirige a **Preferencias** y en el campo URLs de **Board Manager** adicionales, se ingresa el siguiente link: http://arduino.esp8266.com/stable/package_esp8266com_index.json (Figura B.12). Después, se abre el **Gestor de tarjetas** y se

busca ESP8266, una vez que se lista, se selecciona la opción **Instalar** (Figura B.13 y B.14).

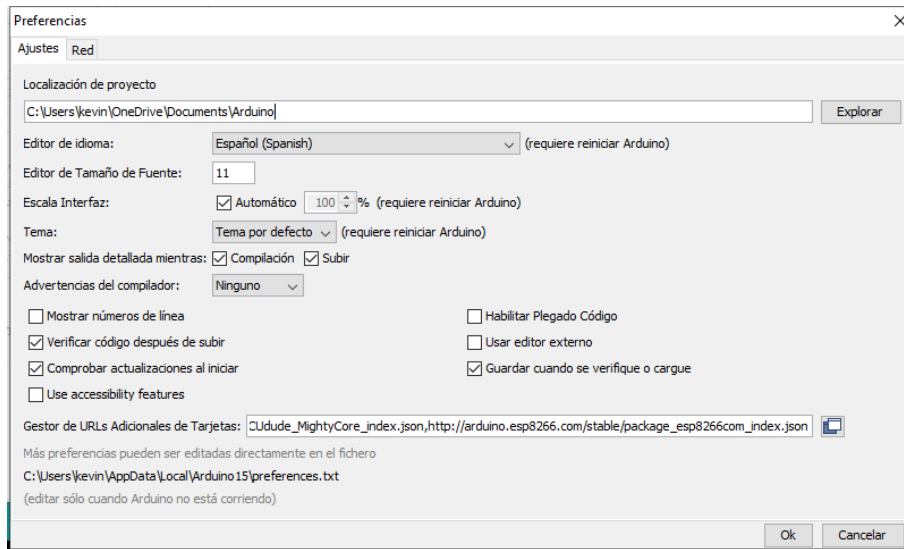


Figura B.12: Configuración para Gestor de URLs Adicionales de tarjetas.

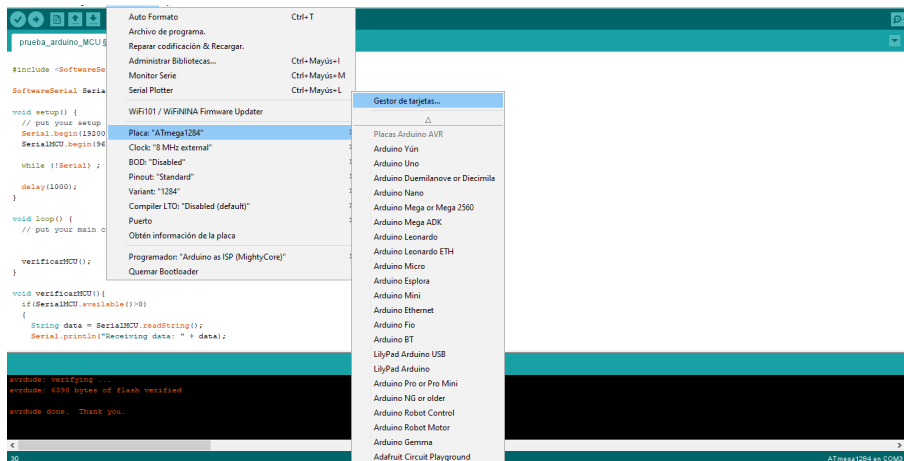


Figura B.13: Ventana de opciones para instalar Tarjetas.

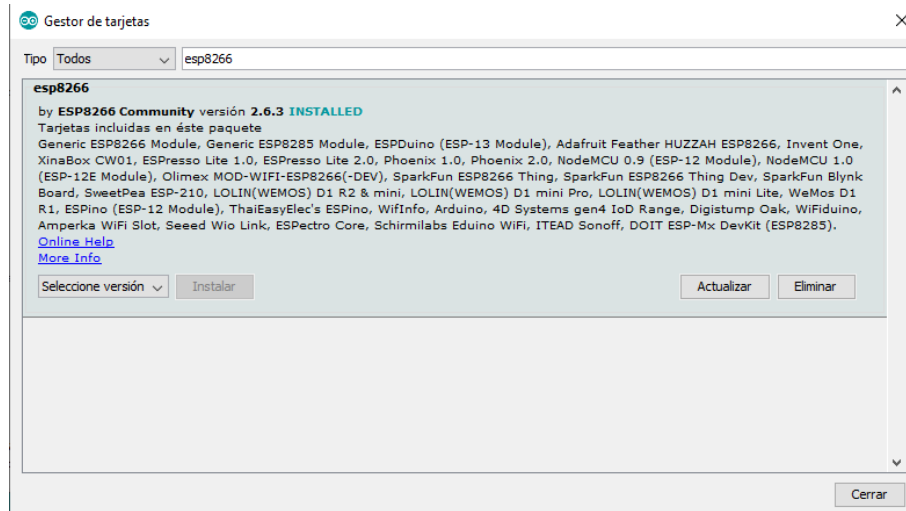


Figura B.14: Ventana de instalación de librería ESP8266.

Lmic

La biblioteca LMIC proporciona una implementación bastante completa de [LoRaWAN](#). Tiene las siguientes funciones: el envío de paquetes de enlace ascendente, teniendo en cuenta el ciclo de trabajo; cifrado y comprobación de integridad de mensajes; recepción de paquetes de enlace descendente en la ventana RX2; frecuencias personalizadas y configuraciones de velocidad de datos[47]. Para instalar la librería hay que acceder a **Gestionar Librerías**, buscar en la ventana *lmic*, seleccionar *MCCI LoRaWAN LMIC Library by IBM* y presionar **Instalar** (Figura B.15).

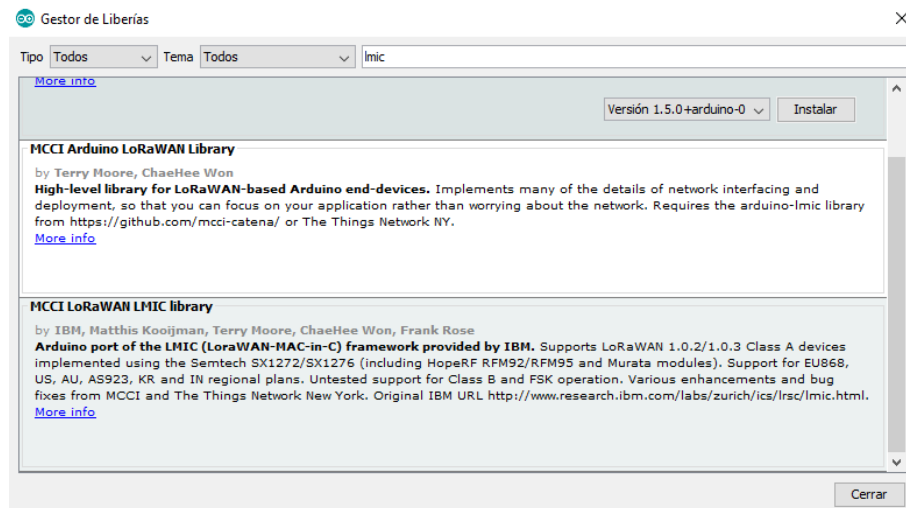


Figura B.15: Ventana de instalación de librería Lmic.

B.3. Configuración de LoRaWAN

En esta sección, se presenta la configuración que se realiza en el nodo, *gateway* y en la aplicación del servidor [LoRaWAN](#).

B.3.1. Nodo

Una vez instaladas las librerías se configura el nodo, se abre el programa desarrollado en el IDE de Arduino, en el cual se modifican los siguientes campos: las claves de unión **ABP**, la asignación de pines y la frecuencia de operación del transceptor **LoRa**. En la Figura B.16) se observa el ingreso de las claves generadas por la aplicación de **TTN** (**NwkSKey**, **AppSKey** y **DevAddr**).

```
//LORAWAN DATA
static uint8_t NWSKEY[16] = { 0xCF, 0xD2, 0x84, 0x7E, 0xC9, 0xA1, 0xF7, 0x88, 0xF0, 0x27, 0x4E, 0x58, 0x9A, 0xA7, 0x99, 0x8C };
static uint8_t APPSKEY[16] = { 0x81, 0x07, 0x12, 0x8F, 0xC3, 0x21, 0xA8, 0x84, 0x8B, 0xC3, 0x6F, 0x09, 0xA1, 0x42, 0x94, 0x8B };
static uint32_t DEVADDR = 0x2E011867;

static osjob_t sendjob;
const unsigned TX_INTERVAL = 30;
String dataToSend = "";

// Pin mapping
const IMC_pinmap IMC_pins = {
  .nss = 4,
  .rxtx = IMC_UNUSED_PIN,
  .rst = 1,
  .dio = {13, 14, 15},
};

void os_getArtBui (ul_s* buif) { }
void os_getLwBui (ul_s* buif) { }
void os_getDevKey (ul_s* buif) { }

void onEvent (ev_s ev) {
  Serial.println(getTime());
  Serial.println("");
  switch(ev) {
    case EV_SCAN_TIMEOUT:
      Serial.println(F("EV_SCAN_TIMEOUT"));
      break;
  }
}
```

Figura B.16: Configuración de claves para unirse a la red **LoRaWAN**.

En la función de asignación de pines (Figura B.17), se ingresan los números de pines de control y comunicación **SPI** del módulo **LoRa** conectados al microcontrolador.

```
//LORAWAN DATA
static uint8_t NWSKEY[16] = { 0xCF, 0xD2, 0x84, 0x7E, 0xC9, 0xA1, 0xF7, 0x88, 0xF0, 0x27, 0x4E, 0x58, 0x9A, 0xA7, 0x99, 0x8C };
static uint8_t APPSKEY[16] = { 0x81, 0x07, 0x12, 0x8F, 0xC3, 0x21, 0xA8, 0x84, 0x8B, 0xC3, 0x6F, 0x09, 0xA1, 0x42, 0x94, 0x8B };
static uint32_t DEVADDR = 0x2E011867;

static osjob_t sendjob;
const unsigned TX_INTERVAL = 30;
String dataToSend = "";

// Pin mapping
const IMC_pinmap IMC_pins = {
  .nss = 4,
  .rxtx = IMC_UNUSED_PIN,
  .rst = 1,
  .dio = {13, 14, 15},
};

void os_getArtBui (ul_s* buif) { }
void os_getLwBui (ul_s* buif) { }
void os_getDevKey (ul_s* buif) { }

void onEvent (ev_s ev) {
  Serial.println(getTime());
  Serial.println("");
  switch(ev) {
    case EV_SCAN_TIMEOUT:
      Serial.println(F("EV_SCAN_TIMEOUT"));
      break;
  }
}
```

Figura B.17: Función de mapeo de pines para el módulo **LoRa**.

En la función `setup()`, se define la frecuencia de operación del transceptor **LoRa** integrado en el nodo. La frecuencia central debe coincidir con la frecuencia central que se definió en la consola **TTN** para el *gateway*. En este caso, la frecuencia central es de 433,1 MHz (Figura B.18),

```
void initLoRa(uint32_t daddr, uint8_t nwkkey, uint8_t appkey){
  // Reset the MAC state. Session and pending data transfers will be discarded.
  LMIC_reset();

  LMIC_setSession (0x1, daddr, nwkkey, appkey);

  #if defined(CPU_eu868)
  LMIC_setupChannel(0, 433100000, DR_RANGE_G40(DR_SF12, DR_SF7), BAND_HILL1); // g-band
  #elif defined(CPU_us915)
  LMIC_setupChannel(0, 915000000, DR_RANGE_G40(DR_SF12, DR_SF7), BAND_HILL1); // g-band
  #endif

  // Disable link check validation
  LMIC_setLinkCheckMode(0);

  // TTN uses SF9 for its RX2 window.
  LMIC.dnDdr = DR_SF9;

  // Set data rate and transmit power for uplink (note: txpow seems to be ignored by the library)
  LMIC_setDrTxpow(DR_SF9, 14);

  // Start job
  do_send(&sendjob);
}

void do_send(osjob_t* j){
  Serial.println("Data completa: " + dataToSend);

  if (dataToSend.length() == 0){
    os_setTimeCallback(&sendjob, os_getTime()+secToClocks(TX_INTERVAL), do_send);
  }
}
```

Figura B.18: Definir la frecuencia central para que el nodo del sensor.



Una vez, configurados todos los parámetros, se carga el programa al nodo. Para verificar el funcionamiento, se abre el monitor serial para revisar el envío de datos (Figura B.19).

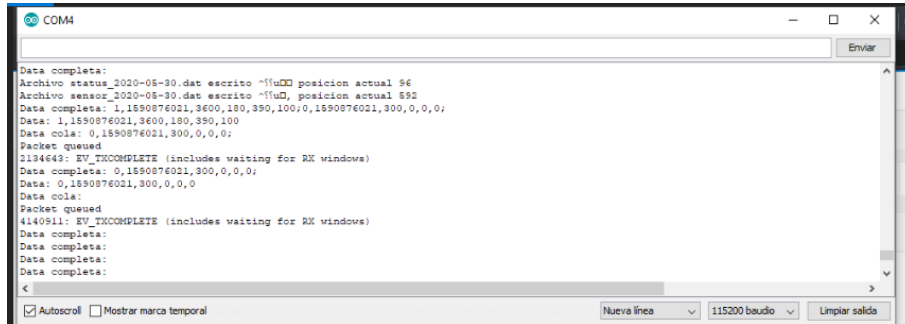


Figura B.19: Salida de monitor en serie del nodo.

B.3.2. Gateway

Una vez instalado el hardware, para el reenvío de datos, se instala el software del *gateway LoRaWAN* de un solo canal que se encuentra en el repositorio de *GitHub*¹. A continuación, se presentan los pasos para instalar este software en la Raspberry Pi:

1. Ingresar al terminal del Raspberry Pi e instalar el Git, usando el comando `git clone`.
2. Abrir el archivo `main.cpp` en el directorio `single_chan_pkt_fwd`, donde se configura la frecuencia central del módulo transceptor *LoRa*, en este caso 433 MHz.
3. Definir la dirección del enrutador de *TTN* en el campo `SERVER1`, donde se ingresa la dirección IP 52.169.76.203. 9.
4. Guardar el archivo y compilar el código con el comando `make`.

La Figura B.20 muestra el estado del transceptor *LoRa* como “*SX1278 detected, starting*”. Si se recibe el mensaje de estado “*Transceptor not detected*”, se debe verificar nuevamente las conexiones de cableado entre el módulo *LoRa* y la Raspberry Pi. También se mostrará la ID del *gateway*, la cual será la ID que se ingresa en la configuración del *gateway* en la red *TTN*. La ID del *gateway* autogenerada para este proyecto es b8:27:eb:ff:ff:63:70:89.

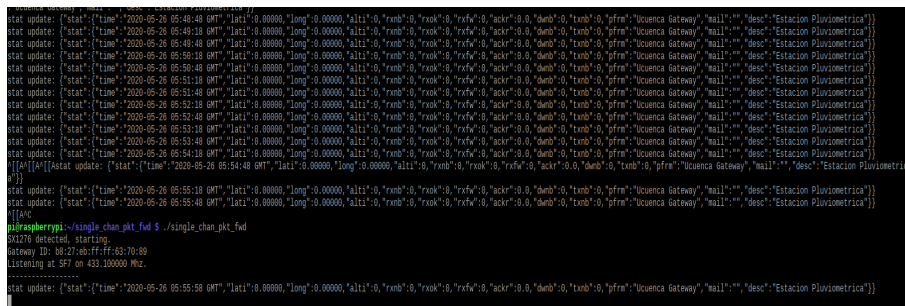


Figura B.20: Salida de consola para el reenviador de paquetes de un solo canal.

¹https://github.com/tftelkamp/single_chan_pkt_fwd

B.3.3. TTN

Para iniciar con el uso de TTN, se debe crear una cuenta y registrarse en (<https://www.thethingsnetwork.org/>). A continuación, se presentan los pasos para registrar el *gateway* y crear la aplicación para el registro del nodo.

Una vez creada la cuenta en TTN, dirigirse a la consola en <https://console.thethingsnetwork.org/>. En la página de inicio de la consola, seleccionamos la opción *Gateways*, en donde se oprime “registrar gateway” (Figura B.22).

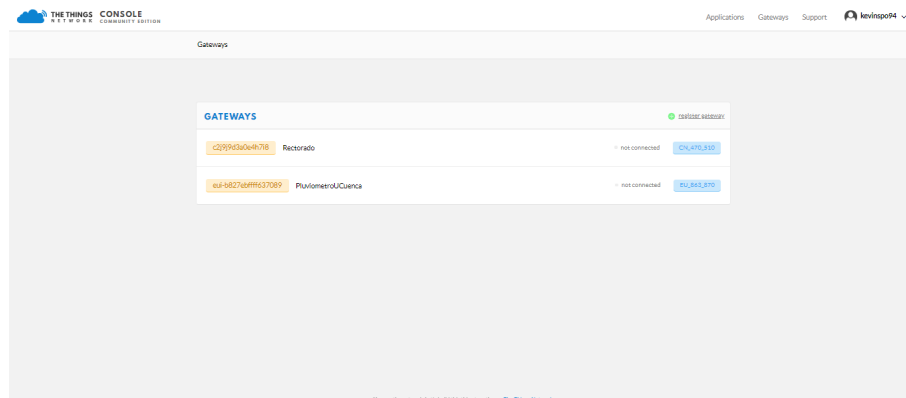


Figura B.21: Ventana para el registro del gateway.

En la página de registro, se llena el formulario con la siguiente información :

- **ID del gateway:** Es un identificador único. Se usa la ID generada por el reenviador de paquetes (revisar Subsección B.3.2).
- **Plan de frecuencia:** Se elige un plan de frecuencia para usar con el módulo tranceptor de radio LoRa de la puerta de enlace.
- **Enrutador:** Se elige un enrutador de la lista desplegable para conectar el *gateway* a la red TTN.
- **Ubicación:** Se hace clic en el mapa para marcar la ubicación del *gateway*.

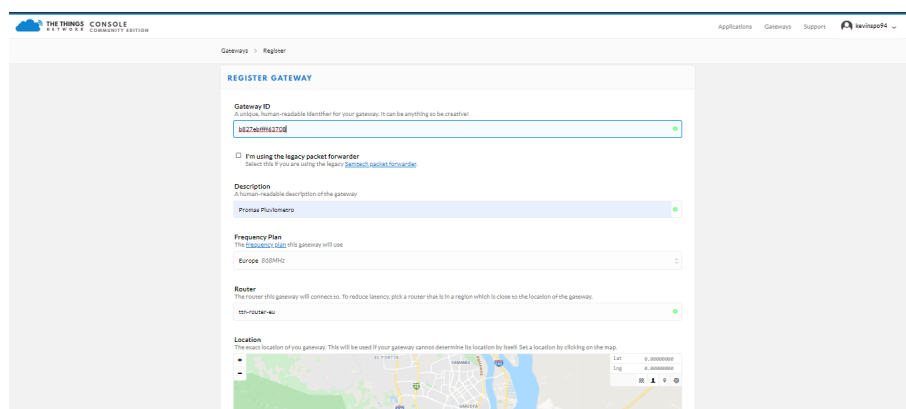


Figura B.22: Ventana para el registro del gateway.

Una vez llenado el formulario, se selecciona *Registrar Gateway* para completar el registro. Para observar el

estado de conexión del *gateway*, se ingresa en la ventana de información (Figura B.23). El estado debe estar en “conectado” si el software de reenvío de paquetes se está ejecutando correctamente en la Raspberry Pi.

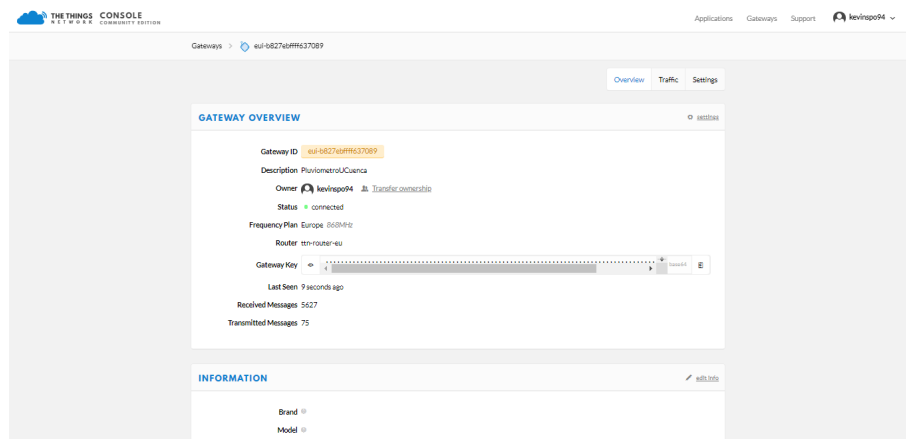


Figura B.23: Ventana de información del gateway.

Registrar el nodo

Para registrar el nodo en la aplicación de TTN, se ingresa en *Aplicaciones* en la consola de TTN, hacer clic en el enlace **Agregar Aplicación**, en la sección **Agregar aplicación**, se completa el formulario con los siguientes datos (Figura B.24).

- **ID de la aplicación:** Se ingresa el identificador único de su aplicación en la red.
- **Descripción:** Esta es una descripción de la nueva aplicación.
- **Registro del controlador:** Se selecciona el *router* en el que desea registrar esta aplicación.

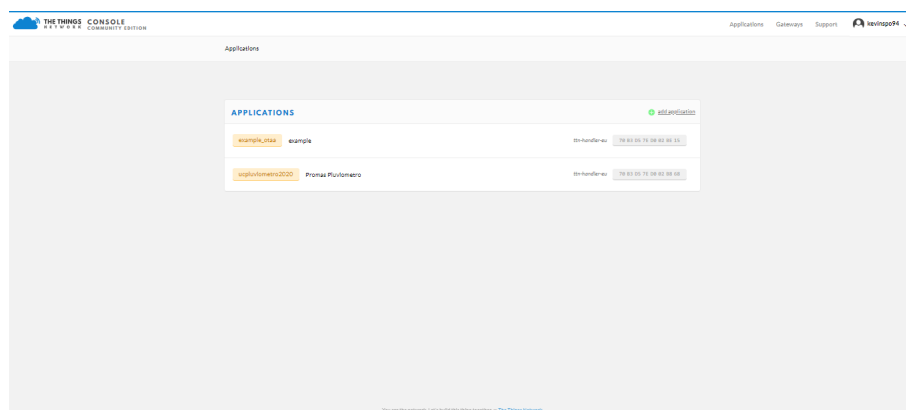


Figura B.24: Ventana para el registro del dispositivo.

Al presionar en el botón **Agregar aplicación** se crea la aplicación. Una vez en la ventana de la aplicación, se busca la sección **Dispositivos**, donde se selecciona **Registrar dispositivo** (Figura B.25) y se procede a llenar el formulario con los siguientes datos:

- **ID del dispositivo:** Es el identificador único para el dispositivo en esta aplicación.

- **Dispositivo EUI:** El dispositivo EUI es el identificador único para este dispositivo en la red. El dispositivo EUI debe constar de exactamente 8 bytes.
- **AppKey:** La clave de la aplicación se utilizará para asegurar la comunicación entre su dispositivo y la red. Esta es una clave autogenerada.
- **Aplicación EUI:** Este es un identificador auto-generado.

The screenshot shows the 'ADD APPLICATION' form in the The Things Console. The form has four main sections: 'Application ID' with the value 'nodo_01/vomero', 'Description' with 'Pruebas Plomero', 'Application EUI' with a default value 'EUI issued by The Things Network', and 'Handler registration' with 'tts-handler-eu'. A green 'Add Application' button is located at the bottom right of the form.

Figura B.25: Formulario de registro del dispositivo.

Una vez registrado el dispositivo, se ingresa en configuración, se selecciona **ABP** como método de activación. Esto generará la dirección del dispositivo, la clave de sesión de red y la clave de sesión de la aplicación (Figura B.26).

The screenshot shows the 'SETTINGS' page for a device in the The Things Console. The 'Activation Method' is set to 'ABP'. The 'Device Address' is '2d 01 05 07', 'Network Session Key' is '0f 22 f4 74 c8 2c 2f 23 f2 48 22 c4 67 89 8c', and 'App Session Key' is '81 07 12 0f c4 21 84 84 08 c3 60 0f 81 62 f4 98'. A note at the bottom states: 'Having a Network Session Key and App Session Key is required to use the LoRaWAN ABP protocol.'

Figura B.26: Establecer el método de activación **ABP**.

En la ventana de descripción general del dispositivo (Figura B.27), se puede observar la **DevAddr**, la **AppSKey** y la **NwksKey** de la aplicación, los cuales se ingresan en el software desarrollado para el nodo. Además, del estado de conexión y los paquetes recibidos.

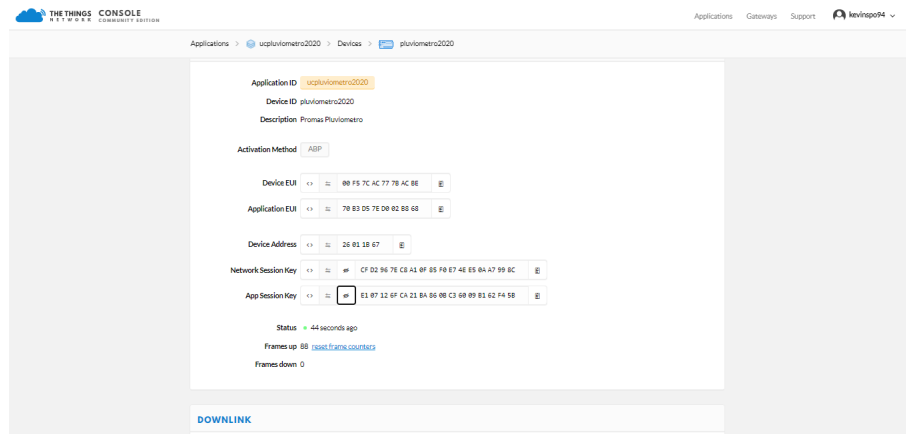


Figura B.27: Ventana de información del nodo.

Decodificador de payload

En la página de la aplicación, se ingresa en *Payload Formats* y se selecciona la pestaña *Decodificador*. Luego se reemplaza la función *JavaScript* existente con la función desarrollada para visualizar los datos del sensor y del estado del dispositivo (Figura B.27).

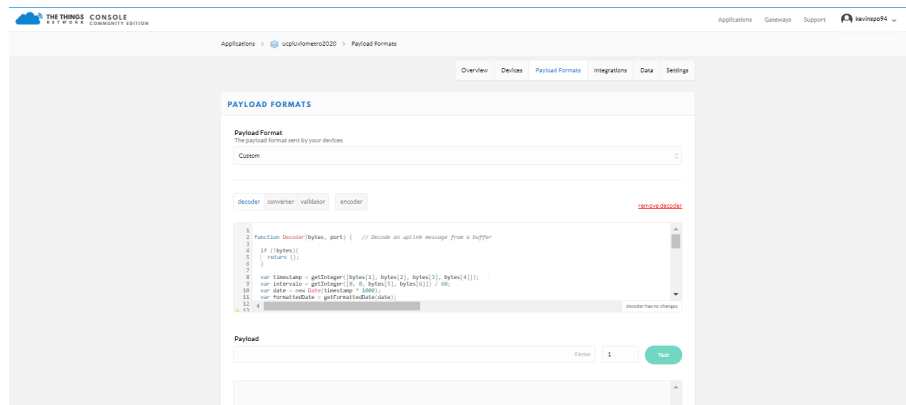


Figura B.28: Función del decodificador.

Para ver los datos de los *payloads* decodificados, se ingresa en la consola de **TTN**, en la sección **Datos de la aplicación**, se puede ver todos los paquetes de datos entrantes (Figura B.29 y B.30).

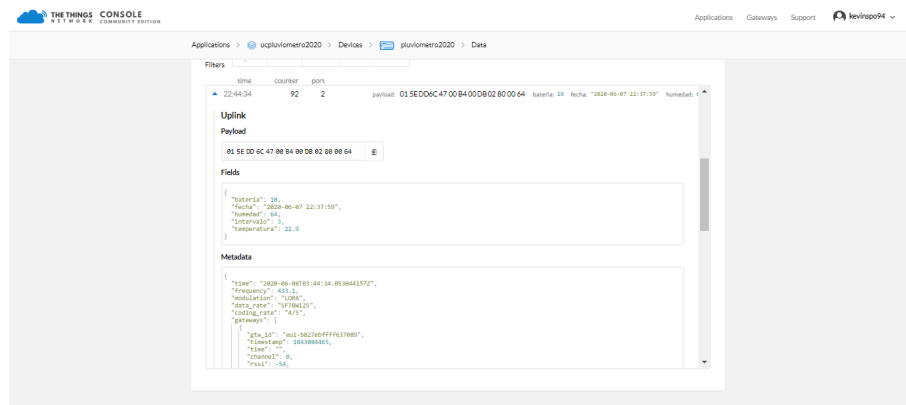


Figura B.29: Datos entrantes de estado del dispositivo capturados por la aplicación.

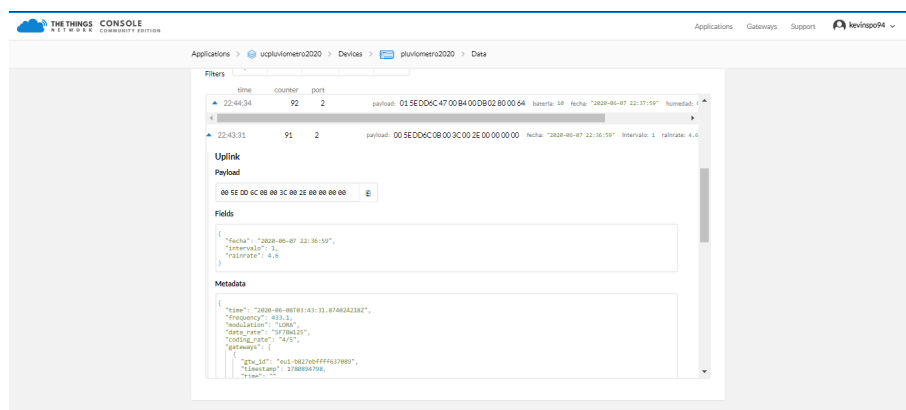


Figura B.30: Datos entrantes del sensor pluviométrico del dispositivo capturados por la aplicación.

B.4. Configuración descarga datos in-situ

A continuación, se presenta la configuración realizada en el nodo, *gateway WiFi* y la aplicación móvil, para la descarga de datos.

Configuración del Nodo

En el nodo se configura el puerto serial por donde se enviarán los datos hacia el *gateway WiFi*, el tamaño de los archivos, los tiempos de toma de muestras y la velocidad del puerto serial seleccionado (Figura B.31 y B.32).

```
Archivo Editar Programa Herramientas Ayuda
placa_fluviometropims_FINAL
#include <RTClib.h>
#include <SerialFlash.h>
#include <SPI.h>
#include <Wire.h>
#include <lmic.h>
#include <hal/hal.h>
#include <DHT.h>

#define SENSOR_FILE_SIZE 4096
#define STATUS_FILE_SIZE 1024
#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
//SoftwareSerial Serial1(3,4);
const int FlashChipSelect = 12;
int intervaloEstado = 100;
int intervaloSensor = 60;
unsigned long lastEstado = 0;
unsigned long lastSensor = 0;

//Datos para sensor
int contra = 0; //Variable para guardar el conteo de los pulsos
int pin = A6;
int value = 1;
float rainRate = 0.0;
```

Figura B.31: Configuración del tamaño y tiempo de las muestras.

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial1.begin(9600);

  #ifdef VCC_ENABLE
    // For Pinoccio Scout boards
    pinMode(VCC_ENABLE, OUTPUT);
    digitalWrite(VCC_ENABLE, HIGH);
    delay(1000);
  #endif

  pinMode(pin, INPUT);

  // wait for Arduino Serial Monitor
  while (!Serial);

  delay(1000);

  if (!SerialFlash.begin(FlashChipSelect)) {
    Serial.println("Unable to access SPI Flash chip");
  }

  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
}
```

Figura B.32: Configuración de la comunicación serial.

Configuración del Wifi Gateway

En el *gateway* WiFi, se configura los pines de la comunicación UART para realizar la comunicación serial con el nodo, la velocidad de la comunicación serial, las credenciales de la red y el puerto del servidor (Figura B.33).



```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <SoftwareSerial.h>

// Replace with your network credentials
const char* ssid = "Arduino Connection";
const char* password = "ucuenca2020";

ESP8266WebServer server(80);
SoftwareSerial SerialArduino(D2, D3);

void setup() {
  delay(1000);
  Serial.begin(9600);
  SerialArduino.begin(9600);

  WiFi.softAP(ssid, password);

  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);

  //Manejar raiz o verificar conexión
  server.on("/", handleRoot);

  // Ruteo para '/files' para poder obtener el listado de archivos disponibles en memoria
  server.on("/files", []() {
```

Figura B.33: Configuración de parámetros del gateway WiFi.

Configuración de la Aplicación Móvil

Para acceder a la comunicación con el *gateway* WiFi, se configura la conexión a la red WiFi en el dispositivo móvil (Figura B.34).



Figura B.34: Configuración de la red WiFi en el dispositivo móvil.

Una vez conectado al *gateway*, se procede a buscar la aplicación llamada *Pluviómetro* en el menú del móvil (Figura B.35)

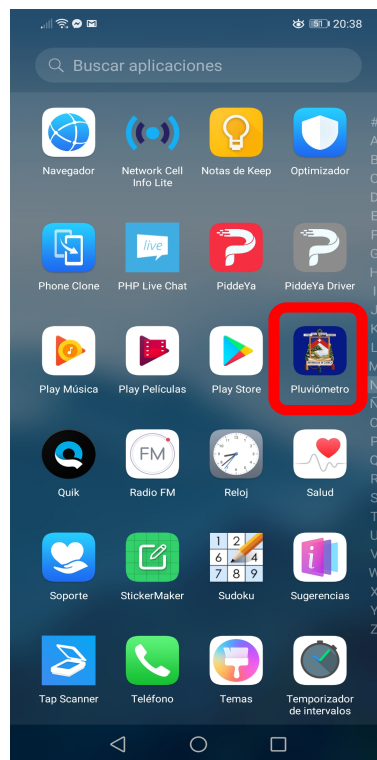


Figura B.35: Aplicación situada en el menú del dispositivo móvil.

Se ingresa en la aplicación, y si la conexión fue exitosa se tendrá las funciones de la aplicación habilitadas y listas para usar (Figura B.36), en caso de que no exista la conexión con el *gateway* mostrará el mensaje “*No existe conexión con el dispositivo*” (Figura B.37).

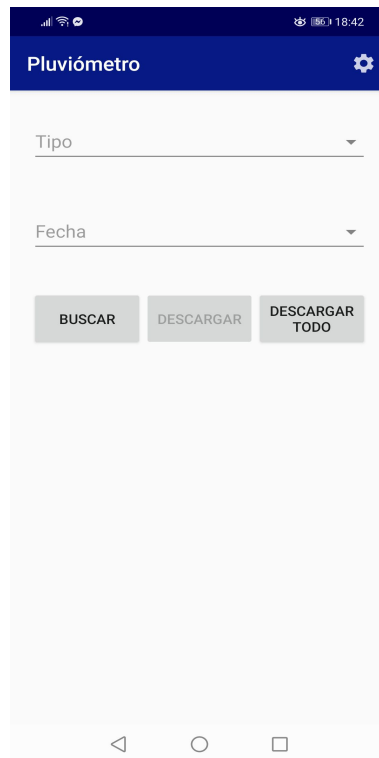


Figura B.36: Interfaz de la aplicación móvil con conexión exitosa hacia el *gateway*.



Figura B.37: Interfaz de aplicación móvil sin conexión al *gateway*.



B.5. Código fuente generado para los dispositivos

Los códigos desarrollados para cada dispositivo se encuentra en el repositorio de Github [\[48\]](#).



Montaje del prototipo

En este apéndice, se presenta el montaje del prototipo en *protoboard* y del registrador de datos para realizar las pruebas de funcionamiento.

C.1. Montaje del prototipo en protoboard

El montaje del prototipo en el *protoboard*, se usó para realizar las pruebas de funcionamiento del diseño final antes de proceder a imprimir el **PCB** y soldar los componentes de montaje superficial.

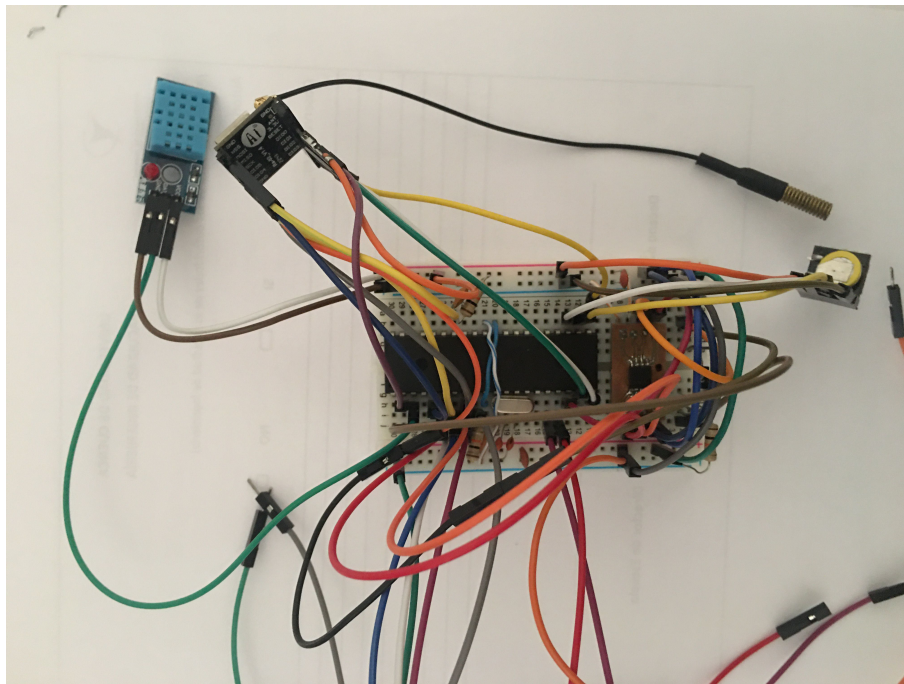


Figura C.1: Montaje del circuito en *protoboard* sin sensor pluviométrico.



Figura C.2: Montaje del circuito en *protoboard* con sensor pluviométrico.

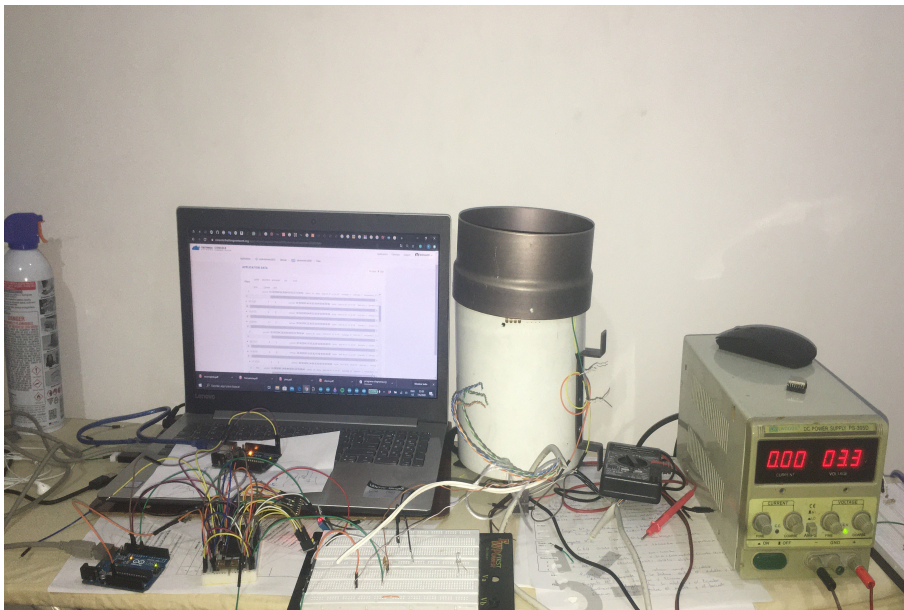


Figura C.3: Pruebas de funcionamiento del circuito en *protoboard*.

C.2. Montaje del registrador de datos para pruebas de laboratorio

En esta sección, se presenta el montaje del registrador de datos para realizar las pruebas de funcionamiento en laboratorio, en las que se incluye: la fuente de alimentación, el sensor pluviométrico, el *gateway* LoRa y WiFi.



Figura C.4: Nodo final instalado en su contenedor.



Figura C.5: Gateway LoRa en funcionamiento.



Figura C.6: Conexión del nodo y el sensor pluviométrico.



Figura C.7: Montaje final para pruebas de funcionamiento del nodo .



Figura C.8: Conexión del *gateway* WiFi al nodo para la descarga de datos *in-situ*.



Diseño esquemático final del dispositivo



Registrador de Datos Pluviométricos

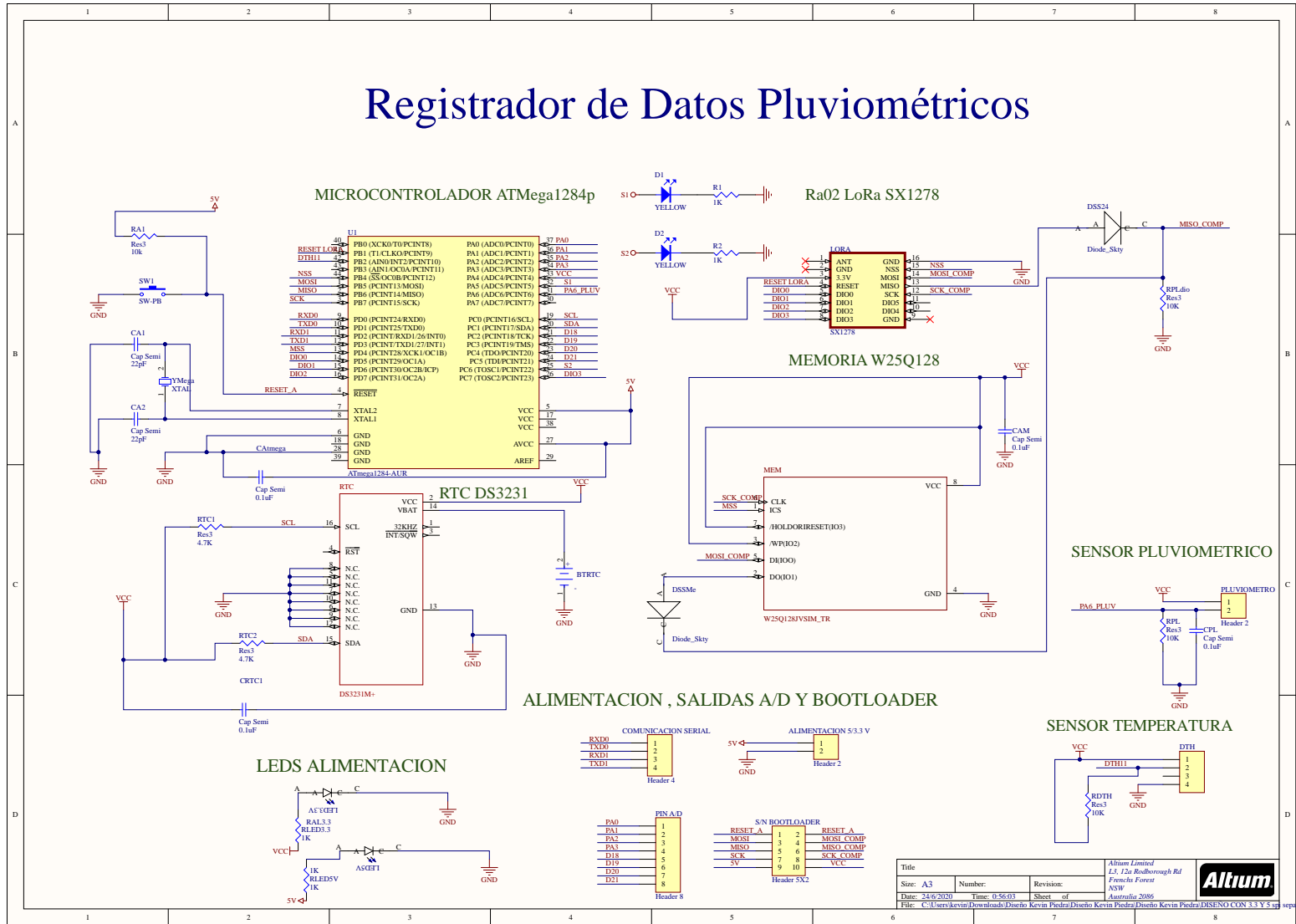


Figura D.1: Diseño del esquemático.



Bibliografía

- [1] A. V. Fabián Astudillo, “Aplicación de tecnologías inalámbricas al sistema de predicción de caudales en la cuenca del río Tomebamba, pp. 1–10” , 2016.
- [2] F. Astudillo-Salinas y G. Barros-Gavilanes, “Aplicación de tecnologías inalámbricas al monitoreo climatológico en la cuenca del río paute,” *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2014*, num. September, pp. 19–23, 2014.
- [3] *Introducción a la hidrometeorología*. [En línea]. Disponible: http://www.hidromet.com.pa/educacion_hidrologico.php
- [4] A. C. López Ochoa, “Diseño e implementación de un registrador de datos para la adquisición de información limnigráfica” , 2019. [En línea]. Disponible: <http://dspace.ucuenca.edu.ec/handle/123456789/32561>
- [5] S. Placencia, D. Astudillo, A. Vázquez-Rodas, I. Minchala, y J. Guaman, “Rainfall intensity datalogger system. lopy4-based design and implementation,” 11 2019, pp. 45–50.
- [6] S. Daud, T. S. Yang, M. A. Romli, Z. A. Ahmad, N. Mahrom, y R. A. A. Raof, “Performance evaluation of low cost LoRa modules in IoT applications,” *IOP Conference Series: Materials Science and Engineering*, vol. 318, p. 012053, mar 2018. [En línea]. Disponible: <https://doi.org/10.1088%2F1757-899x%2F318%2F1%2F012053>
- [7] S. Tech, “Semtech SX1276, 100 MHz to 1020 MHz Low Power Long Range Transceiver.” . [En línea]. Disponible: <https://www.semtech.com/products/wireless-rf/lora-transceivers/SX1276>
- [8] M. C. V. Ian F. Akyildiz, “Wireless Sensor Network.” , 2010. [En línea]. Disponible: <https://www.yuden.co.jp/ut/solutions/iot/>
- [9] C. A. M. Borda, *Diseño de una red inalámbrica para la gestión y control de un sistema integral de sistemas de riego* , 2017. [En línea]. Disponible: <http://repository.udistrital.edu.co/bitstream/11349/5880/1/PinedaCusbaLeonardo2017.pdf>
- [10] T. Yuden, “Taiyo Yuden’s IoT (Internet of Things) sensor nodes to data display” . [En línea]. Disponible: <https://www.yuden.co.jp/ut/solutions/iot/>
- [11] M. Centenaro, L. Vangelista, A. Zanella, y M. Zorzi, “Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios,” *IEEE Wireless Communications*, vol. 23, num. 5, pp. 60–67, October 2016.



- [12] L. Vangelista, “Frequency shift chirp modulation: The lora modulation,” *IEEE Signal Processing Letters*, vol. 24, num. 12, pp. 1818–1821, Dec 2017.
- [13] M. Knight y B. Seeber, “Decoding lora: Realizing a modern lpwan with sdr,” *Proceedings of the GNU Radio Conference*, vol. 1, num. 1, 2016. [En línea]. Disponible: <https://pubs.gnuradio.org/index.php/grcon/article/view/8>
- [14] J. Haxhibeqiri, E. De Poorter, I. Moerman, y J. Hoebeke, “A survey of lorawan for iot: From technology to application,” *Sensors*, vol. 18, p. 3995, 11 2018.
- [15] L. A. TM, “*The LoRaWAN™ Specifications*,” , 2019. [En línea]. Disponible: <https://lora-alliance.org/about-lorawan>
- [16] J. F. P. Santos, *Evaluación del rango de transmisión de LoRa para redes de sensores inalámbricos con LoRaWAN en cuencas hidrográficas en función del tamaño de carga útil, altura del gateway y potencia de transmisión* . [En línea]. Disponible: <http://dspace.ucuenca.edu.ec/handle/123456789/31506>
- [17] I. O. Monfort, “*Estudio de la Arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa*” , 2017. [En línea]. Disponible: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/64365/6/iordonezTFM0617mem%C3%B2ria.pdf>
- [18] E. Engineering, *Lora Key Concepts*. [En línea]. Disponible: https://docs.exploratory.engineering/lora/lora_key_concepts/
- [19] C. Y. Burbano Ordoñez, “*Implementación de una red de sensores inalámbricos LPWAN mediante módulos LoRa para el monitoreo de la calidad del agua en 2 ríos*” . [En línea]. Disponible: <http://hdl.handle.net/11349/6433>
- [20] O. M. Meteorológica, “*Guía de prácticas climatológicas (omm-n100)*” , 2011. [En línea]. Disponible: http://www.wmo.int/pages/prog/wcp/ccl/guide/documents/wmo_100_es.pdf
- [21] —, “*Guía de instrumentos y métodos de observación meteorológicos (omm-n° 8)*” , 2017. [En línea]. Disponible: https://library.wmo.int/doc_num.php?explnum_id=3664
- [22] D. Instruments, “*Instrumentos Meteorologicos.*” . [En línea]. Disponible: <https://www.davisinstruments.com/product/rain-collector-with-flat-base-for-vantage-pro2/>
- [23] M. Aref y A. Sikora, “Free space range measurements with Semtech LoRa technology,” *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2014*, num. September, pp. 19–23, 2014.
- [24] D. R. M. Garcés, *Diseño e implementación de un dispositivo para la adquisición de datos meteorológicos y conectividad IP que se integre con la red de datos de SENAGUA en la cuenca del río Santa Bárbara* , 2016. [En línea]. Disponible: <http://dspace.ucuenca.edu.ec/handle/123456789/25961>
- [25] L. Llamas, “*El Bus SPI en Arduino*” , 2016. [En línea]. Disponible: <https://www.luisllamas.es/arduino-spi/>
- [26] V. Pamadi y B. G. Nickerson, *Getting Started With 1-Wire Bus Devices.*, 2015. [En línea]. Disponible: <https://www.cs.unb.ca/tech-reports/documents/TR15-235.pdf>



- [27] J. J. V. P. Marco Paúl Gutama Morocho, “Prototipo de un dispositivo de adquisición, almacenamiento y transmisión de datos meteorológicos usando los transductores de la estación DAVIS 6162 Wireless Vantage Pro2 Plus.”, 2017. [En línea]. Disponible: <http://dspace.ucuenca.edu.ec/handle/123456789/28600>
- [28] P. P. L. Vangelista, L. Dell’Anna, “A battery lifetime comparison between lorawan and wireless mbus smart meters,” in *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, June 2019, pp. 1–3.
- [29] G. Loubet, A. Takacs, E. Gardner, A. De Luca, F. Udrea, y D. Dragomirescu, “Lorawan battery-free wireless sensors network designed for structural health monitoring in the construction domain,” *Sensors*, vol. 19, num. 7, 2019. [En línea]. Disponible: <https://www.mdpi.com/1424-8220/19/7/1510>
- [30] A. Zacepins, J. Jelinskis, A. Kvišis, M. Dzenis, V. Komasilovs, y O. Komasilova, “Application of lorawan technology in precision beekeeping.” D. Kovačević, Ed. East Sarajevo, Bosnia-Herzegovina: University of East Sarajevo, Faculty of Agriculture, 2018, pp. 1759–1765.
- [31] M. T. Buyukakkaslar, M. A. Erturk, M. A. Aydin, y L. Vollero, “Lorawan as an e-health communication technology,” in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, July 2017, pp. 310–313.
- [32] Semtech, “SEMTECH LoRa SX1276/77/78/79. (Datasheet)” , 2019. [En línea]. Disponible: <https://www.semtech.com/uploads/documents/DS{ }SX1276-7-8-9{ }W{ }APP{ }V6.pdf>
- [33] D. Instruments, “Base et augets pour pluviomètre Vantage Pro2 - 7852.804 - Davis Instruments.” , 2012. [En línea]. Disponible: <http://www.meteo-shopping.fr/Station-meteo/Base-et-augets-pour-pluviometre-Vantage-Pro2-7852804-Davis-Instruments>
- [34] M. Integrated, “Real time clock DS3231 datasheet.” . [En línea]. Disponible: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>
- [35] P. Seneviratne, *Beginning LoRa Radio Networks with Arduino: Build Long Range, Low Power Wireless IoT Networks*, 01 2019.
- [36] “Raspberry Pi single-channel packet forwarder ” . [En línea]. Disponible: https://github.com/tftelkamp/single_chan_pkt_fwd
- [37] N. Enclosures, “IP65 Enclosures. [En línea]. Disponible: <https://www.nemaenclosures.com/enclosure-ratings/ip-enclosures/ip65-enclosures.html>
- [38] T. Nguyen, “Energy efficient wireless sensor network and low power consumption station design for an urban water level monitoring system,” in *2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, 2016, pp. 252–256.
- [39] Microchip, ““ATmega1284,” Tech. Rep.” . [En línea]. Disponible: www.microchip.com/wwwproducts/en/ATmega1284
- [40] A. Thinker, ““Ra0 product specification.” . [En línea]. Disponible: http://wiki.ai-thinker.com/_media/lora/docs/c048ps01a1_ra-02_product_specification_v1.1.pdf
- [41] Semtech, “SEMTECH LoRa SX1276/77/78/79. (Datasheet),” Tech. Rep. [En línea]. Disponible: <https://www.semtech.com/uploads/documents/DS{ }SX1276-7-8-9{ }W{ }APP{ }V6.pdf>



- [42] D. R. UK, “*DHT11 Humidity & Temperature Sensor*,” *Tech. Rep.* [En línea]. Disponible: <https://www.micro4you.com/files/sensor/DHT11.pdf>
- [43] M. Integrated, “*Real time clock DS3231 datasheet*,” *Tech. Rep.* [En línea]. Disponible: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>
- [44] Winbond, *W25Q128JV serial flash memory*, *Tech. Rep.* [En línea]. Disponible: <https://www.winbond.com/resource-files/w25q128jvspirevc11162016.pdf>
- [45] Heltec, *WiFi kit 8*. [En línea]. Disponible: <https://heltec.org/project/wifi-kit-8/>
- [46] Onsetcomp, *HOBO Rain Gauge Data Logger*. [En línea]. Disponible: <https://www.onsetcomp.com/products/data-loggers/rg3/>
- [47] M. Kooijman, *Lmic Library*. [En línea]. Disponible: <https://github.com/matthijskooijman/arduino-lmic>
- [48] K. S. Piedra, *Pluviometro UC*, 2020. [En línea]. Disponible: <https://github.com/dastudillo1983>