



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Red de Sensores Distribuidos para Estaciones de Monitoreo

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Juan Martín Chiale, Diego Matías Echeverría, Gonzalo
Zarazola

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRICISTA.

TUTORES

Nicolás Pérez Universidad de la República
Pablo Monzón Universidad de la República

TRIBUNAL

Leonardo Steinfeld Universidad de la República
Mariana del Castillo Universidad de la República
Mauricio González Universidad de la República

Montevideo
jueves 26 noviembre, 2020

Red de Sensores Distribuidos para Estaciones de Monitoreo, Juan Martín Chiale,
Diego Matías Echeverría, Gonzalo Zarazola.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Contiene un total de 138 páginas.
Compilada el jueves 26 noviembre, 2020.
<http://iie.fing.edu.uy/>

El único modo de lograr un gran trabajo es amar lo que haces.

STEVE JOBS

Esta página ha sido intencionalmente dejada en blanco.

Agradecimientos

A nuestra familia y amigos que estuvieron siempre a lo largo de este proceso.

A nuestros tutores Pablo Monzón y Nicolás Pérez que nos guiaron en el transcurso del proyecto.

A Mebilor S.A por prestarnos sus instalaciones y equipos para el desarrollo del módulo y realizar pruebas del sistema.

Resumen

Continuando con la línea de trabajo del Departamento de Sistemas y Control del Instituto de Ingeniería Eléctrica del desarrollo de sistemas modulares de adquisición y medición autónomos, se implementó un sistema capaz de conectar de forma inalámbrica una red de sensores distribuidos.

El sistema permite cubrir grandes extensiones geográficas llegando a lugares remotos del país. Para ello se desarrollaron tres tipos de módulos. Un módulo de medición de parámetros ambientales, un módulo de comunicación inalámbrica para desplegar una red en estrella con los sensores del sistema y un módulo que permite el acceso a internet. Para la interconexión de los diferentes módulos desarrollados se utiliza un bus de datos y alimentación que se ha estandarizado en el departamento.

EL módulo de comunicación inalámbrico desarrollado es de largo alcance, con tecnología LoRa. Estos módulos son los encargados de interconectar toda la red de sensores.

El módulo con acceso a internet permite la comunicación entre la red de sensores y el usuario. La red de sensores es configurada desde un servidor WEB que se implementó durante el proyecto. Por lo tanto de forma remota se puede monitorear toda la red, conocer el estado de la misma y desplegar los datos obtenidos.

Se probó el sistema desplegando la red de sensores a nivel de campo, a una exigencia grande por una semana por lo que se considera exitoso el proyecto.

Tabla de contenidos

Agradecimientos	III
Resumen	IV
Glosario	XII
1. Introducción	1
1.1. Objetivos y alcance	1
1.2. Antecedentes	2
1.3. Requerimientos	2
1.4. Resumen de los capítulos	3
2. Descripción del sistema	5
2.1. Descripción de los componentes del sistema	6
2.2. Funcionamiento global del sistema	6
2.3. Ejemplos de posibles aplicaciones	8
3. Servidor	11
3.1. Implementación del Servidor	11
3.2. Transferencia de datos	11
3.2.1. Método GET	12
3.3. Servidor Web y base de datos	12
3.3.1. Estructura Base de datos	12
3.3.1.1. Configuración	13
3.3.1.2. Nodo	13
3.4. Descripción aplicación Web	13
3.4.1. Scripts de manejo de datos	14
3.4.1.1. Consulta.php	14
3.4.1.2. Proc.php	15
4. Módulo GPRS	17
4.1. Características que debe cumplir el módulo	17
4.2. Tecnología de comunicación	17
4.3. Hardware del Módulo	19
4.3.1. Modem de comunicación SIM900	19
4.3.1.1. Comandos AT	20

Tabla de contenidos

4.3.2. Arduino Nano	21
4.4. Firmware del módulo	21
4.4.1. Funcionamiento	21
4.4.2. Máquinas de Estado	22
4.4.3. Librerías auxiliares	24
5. Módulo LoRa	27
5.1. Características que debe cumplir el módulo	27
5.2. Tecnología de comunicación inalámbrica	27
5.2.1. Zigbee	27
5.2.2. LoRa	28
5.3. Hardware principal del módulo	29
5.3.1. Módulo de Comunicación LoRa - RN2903	29
5.3.2. Microcontrolador - dsPIC33	30
5.3.3. Diseño del PCB	31
5.3.3.1. Placa ReSeDEM v1.0	32
5.3.3.2. Placa ReSeDEM v2.0	33
5.3.3.3. Placa ReSeDEM v3.0 - Versión final	33
5.4. Firmware de los Módulos	34
5.4.1. Funcionamiento Nodo Principal	34
5.4.2. Funcionamiento Nodo N	35
5.4.3. Maquinas de estado	35
6. Módulo Sensor	39
6.1. Características que debe cumplir el módulo	39
6.2. Hardware del Módulo	39
6.2.1. Microcontrolador	39
6.2.2. Sensores	39
6.2.2.1. Sensor de Temperatura y Humedad Ambiente - DT11	40
6.2.2.2. Sensor Humedad Suelo - FC-28	40
6.3. Firmware del módulo	41
6.3.1. Funcionamiento	41
6.3.2. Máquinas de estado	42
7. Arquitectura de la Red LoRa	43
7.1. Topologías de red posibles para implementar	43
7.2. Elección de la topología de red para el sistema	44
7.3. Arquitectura de la Red LoRa	45
8. Interfaces y protocolos de comunicación	47
8.1. Interfaces utilizadas	47
8.1.1. UART	47
8.1.2. RS232	48
8.1.3. I2C	49
8.2. Intercambio de mensajes	50
8.2.1. Servidor- Nodo principal	50

Tabla de contenidos

8.2.2.	Módulo GPRS - Módulo LoRa	52
8.2.3.	Módulo Lora - Bus I2C	52
8.2.4.	Estado y Manejo de errores del sistema	53
8.2.5.	Ejemplo de un intercambio de mensajes extremo a extremo	55
9.	Diseño final	57
9.1.	Componentes para el ensamblado	57
9.2.	Ensamblado final	58
9.2.1.	Modulo Lora, Sensor, GPRS	58
10.	Pruebas del Sistema	61
10.1.	Pruebas de Alcance	61
10.1.1.	Primera prueba	61
10.1.1.1.	Resultados	62
10.1.2.	Prueba con el sistema final	62
10.1.2.1.	Resultados	64
10.2.	Pruebas de Consumo	64
10.2.1.	Módulo LoRa Principal + Módulo GPRS	64
10.2.1.1.	Resultados	65
10.2.2.	Módulo LoRa Secundario	66
10.2.2.1.	Resultados	66
10.2.3.	Módulo Sensor	67
10.3.	Pruebas de Desempeño del Sistema	67
10.3.1.	Tasa de fallas	68
10.3.1.1.	Resultados	68
11.	Trabajos Futuros	71
11.1.	Nodo Principal	71
11.2.	Nuevas funcionalidades del Servidor	71
11.3.	Seguridad en el sistema	72
11.3.1.	Transferencia de datos Servidor-Nodo Principal	72
11.3.2.	Comunicación entre Nodos LoRa	72
11.4.	Narrow band	72
11.5.	Modo Bajo Consumo	72
12.	Conclusiones	75
12.1.	Evaluación de resultados	75
12.2.	Aprendizajes	75
	Apéndices	79
A.	Comunicación LoRa	79
A.1.	Modulación de espectro ensanchado	79
A.2.	Parámetros de la comunicación LoRa	80
A.2.1.	Spread Factor, Bandwidth, Code Rate	80
A.2.2.	Relación entre los parámetros, el bitrate y la sensibilidad	82

Tabla de contenidos

B. PCB, lista de componentes y esquemático del Módulo LoRa	83
C. Comandos GPRS	87
D. Módulo Alimentación - Baterías	89
E. Manual Técnico	103
Referencias	121
Índice de tablas	121
Índice de figuras	122

Glosario

- ADC** Analogic digital converter - Convertidor analógico digital. 31, 40
- APN** Access Point Name - Nombre del punto de acceso. 21
- ASCII** American Standard Code for Information Interchange — Código Estándar Estadounidense para el Intercambio de Información. 51
- AT** Attention Comands - Comandos de atención. 19–21, 25, 29, 87, 121
- CDMA** Code Division Multiple Access - Acceso múltiple por división de código. 79
- CNC** Computer Numerical Control - Control numérico de computadoras. 32
- CR** Code Rate - Ratio del código. 28, 36, 72, 82
- CSS** Cascading Style Sheets - Hojas de estilo en cascada. 12, 14
- Css** Chirp Spread Sprectrum - Modulación chirp de espectro expandido. 79
- DB** Database - Base de datos. 13, 15
- EEPROM** Electrically Erasable Programmable Read-Only - ROM programable y borrrable eléctricamente. 22
- FEC** Foward Error Correction - Corrección de errores hacia adelante. 81
- FTP** File Transfer Protocol - Protocolo de transferencia de archivos. 87
- GND** Ground - Tensión de referencia. 49
- GPRS** General Packet Radio Service - Servicio global de paquetes por radio. 17, 18, 21, 72
- GSM** Global System for Mobile - Sistema Global para las Comunicaciones. 17, 18, 29
- HTML** HyperText Markup Language - Lenguaje de marcas de hipertexto. 12, 14

Glosario

HTTP Hypertext Transfer Protocol - Protocolo de transferencia de hipertexto. 11, 12, 14, 21, 23, 24, 72, 87

HTTPS Hypertext Transfer Protocol Secure - Protocolo seguro de transferencia de hipertexto. 72

I2C Inter-Integrated Circuit - Circuito Inter-Integrado. 2, 5, 7, 27, 31, 35, 39, 41, 49, 52, 53

IEEE Institute of Electrical and Electronics Engineers - Instituto de Ingeniería Eléctrica y Electrónica. 27

IIE Instituto de Ingeniería eléctrica. 9

IoT Internet of Things - Internet de las cosas. 1, 11, 18, 27, 28

IP Internet Protocol - Protocolo de internet. 2, 11, 20, 21, 23, 72

ISP Internet Service Provider - Proveedor de Servicio de Internet. 18, 19, 23, 72, 123

LAMP Linux-Apache-MySQL-Php. 12

LoRa Long Range - Largo alcance. 6, 27–29, 31, 32, 34–37, 51

LTE Long Term Evolution - Evolución a largo plazo. 18

MIT Massachusetts Institute of Technology - Instituto tecnológico de Massachus. 25

Modem Modulator/Demodulator - Modulador/Demodulador. 19, 23, 24

MYSQL Open System Interconnection - Sistema de interconexión abierto. 12, 13

OSI Open System Interconnection - Sistema de interconexión abierto. 11

PCB Printed Circuit Board - Placa de circuito impreso. 3, 33, 71

PHP Hypertext Preprocessor - Preprocesador de hipertexto. 12

PWM Pulse Width Modulation- Modulación por ancho de pulsos. 31

QOS Quality of Service - Calidad de servicio. 87

SCL Serial Clock - Reloj serial. 49

SDA Serial Data - Datos serial. 49

SF Spread Factor - Factor de expansión. 28, 36, 72

SMS Short Message Service - Servicio de mensaje Corto. 20, 21, 71, 87

- SQL** Structured Query Language - Lenguaje de consulta estructurada. 12
- SRAM** Static Random Access Memory - Memoria estática de acceso aleatorio. 22, 31
- SS** Spread Spectrum - Espectro expandido. 28, 79
- TCP** Transmission Control Protocol - Protocolo de control de transmisión. 11, 20, 87
- TTL** Transistor-Transistor Logic - Lógica transistor a transistor. 48
- UART** Universal Asynchronous Receiver-Transmitter - Transmisor-Receptor Asíncrono Universal. 19, 27, 31, 47, 49, 52
- UDP** User Datagram Protocol - Protocolo de usuario por datagramas. 87
- UMTS** Universal Mobile Telecommunications System - Sistema global para las comunicaciones móviles. 17, 18
- URL** Uniform Resource Locator - Localizador de recursos uniforme. 14, 21

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 1

Introducción

Con el avance de la tecnología, los nuevos conceptos de big-data e IoT, cada vez son más los dispositivos que intercambian información entre sí. Utilizando dispositivos que compartan sus estados, sus mediciones y procesando la mayor cantidad de información posible, se puede predecir comportamientos o tomar mejores decisiones en la industria, o producción agrícola, por ejemplo.

ReSeDEM propone una solución para interconectar diferentes dispositivos de recolección de medidas ambientales en diferentes puntos geográficos con una gestión centralizada que permita la visualización de la información recolectada desde un solo lugar.

1.1. Objetivos y alcance

Los objetivos generales del proyecto son los siguientes:

1. Diseñar e implementar un módulo de comunicación inalámbrico de largo alcance, tratando de minimizar el consumo de energía.
2. Estudiar diferentes topologías de red e implementar una arquitectura para una red de nodos de medición utilizando el módulo anterior.
3. Implementar una comunicación remota de un nodo central con un servidor a través de una tecnología de red de acceso inalámbrica que permita ajustar parámetros de adquisición de medidas de cualquier nodo en la red y ver el estado de los mismos.

En cuanto al alcance, este proyecto cumple con los siguientes puntos acordados en la planificación del mismo.

1. Desarrollo de firmware, hardware y housing del módulo de comunicación de fácil integración a los desarrollos del Instituto.
2. Establecer una red de al menos tres nodos con posibilidad de anexar más de forma sencilla.

Capítulo 1. Introducción

3. Desarrollo de un módulo central (firmware, hardware y housing) y servidor que se comuniquen entre sí para el manejo y configuración de periféricos y análisis de datos.
4. Implementar una Interfaz Gráfica para Usuarios que permita configurar el dispositivo y realizar el análisis de los datos sensados.

1.2. Antecedentes

El Departamento de Sistemas y Control del Instituto de Ingeniería Eléctrica ha estado trabajando en el desarrollo de sistemas modulares de adquisición y medición autónomos. Cada módulo que componen estos sistemas debe respetar un estándar de conexión propuesto por el departamento de manera que sea fácil la integración. Los módulos pueden ser de alimentación, de adquisición de medidas, de transmisión de datos, etc. Existen un proyectos de fin de carrera que ha trabajado sobre esta base:

- Sarabee: Sistema autónomo remoto de adquisición. [2]

Este sistema se compone de distintos módulos con las especificaciones del Departamento.

Otros grupos del IIE han desarrollados líneas de investigación respecto a redes de sensores que han servido como base para el desarrollo del sistema como lo son:

- Agrovisión: Red de sensores inalámbricos para la previsión y tratamiento de plagas en cultivos frutales [3].
- RSitrust: Red de Sensores inalámbricos para monitoreo de condiciones micro climáticas en cultivos de cítricos. [4]

1.3. Requerimientos

El proyecto se puede separar esencialmente en dos partes, el desarrollo de los módulos y por otro lado la implementación del servidor y la red de sensores.

- Módulos: Deberán ser independientes, estar montados en gabinetes individuales y compartir la comunicación y alimentación a través de un mismo bus de datos/alimentación (+12V, +5V, +3V, GND, SDA(I2C), SCL(I2C)). En la tabla 1.1 se especifica el número de pin y su función.

Además, deberán ser previstos para funcionar a la intemperie, por lo que el código de protección que deben cumplir es el IP 65.

- Módulo Principal: Será capaz de comunicarse con los otros módulos del sistema y con el servidor mediante de un módem inalámbrico a través la red celular. Este módulo será el encargado de controlar los periféricos que componen el sistema.

1.4. Resumen de los capítulos

Pin	Función
1	12V
2	5V
3	0V
4	SDA
5	SCL
6	3.3V
7	No utilizado

Tabla 1.1: Correspondencia Pin - Función del Bus de datos/alimentación

- **Módulo de Comunicación:** implementará la comunicación entre nodos. Se trata de un PCB diseñado por el equipo de proyecto que logrará comunicar 2 nodos a una distancia mínima de 1 km de enlace visto. Se establecerá una red de sensores de al menos 3 nodos que puedan comunicarse con el módulo principal
- **Autonomía:** Cada nodo deberá tener una autonomía mínima de una semana.
- **Servidor:** Se podrá acceder desde cualquier punto a través de Internet para visualizar los datos relevados por cualquier nodo. Deberá ser capaz de enviar comandos para configurar parámetros de los ensayos y dar de alta nuevos nodos.

1.4. Resumen de los capítulos

■ Capítulo 1 - Introducción

En este primer capítulo se describe el proyecto, los objetivos, alcance y requisitos de funcionamiento del sistema. También se indican los antecedentes del mismo.

■ Capítulo 2 - Descripción del Sistema

En el segundo capítulo se describe el sistema desarrollado. Se definen y describen las partes que lo componen, se presenta el funcionamiento global del sistema y posibles aplicaciones.

■ Capítulo 3 - Servidor

En el tercer capítulo se describe como se implementa el servidor, las características y los protocolos de transferencia de datos.

■ Capítulo 4 - Módulo GPRS

En el cuarto capítulo se describe el módulo que se utiliza para la comunicación entre el servidor y la red de sensores. También se describe el firmware implementado.

Capítulo 1. Introducción

- Capítulo 5 - Módulo LoRa

En este capítulo se presenta el módulo diseñado para la comunicación inalámbrica entre sensores. Se describe el protocolo elegido y el firmware implementado.

- Capítulo 6 - Módulo Sensor

En este capítulo detallan las principales características de los sensores que se van a utilizar y su firmware.

- Capítulo 7 - Arquitectura de la Red LoRa

En este capítulo se presenta la arquitectura de red inalámbrica elegida para la comunicación entre sensores de la red.

- Capítulo 8 - Interfaces y protocolos de comunicación

En este capítulo se describen los intercambios de mensajes implementados para la correcta comunicación entre todas las partes involucradas.

- Capítulo 9 - Diseño final

En este capítulo se presenta el hardware final de todos los componentes del sistema para su implementación en campo.

- Capítulo 10 - Pruebas del Sistema

En este capítulo se presentan las pruebas realizadas para la validación del sistema y el análisis de los resultados obtenidos.

- Capítulo 11 - Posibles líneas de investigación y desarrollos

En este capítulo se presentan posibles líneas de investigación a futuro e implementaciones que mejorarían el desempeño del sistema

- Capítulo 12 - Conclusiones

En este último capítulo se evalúan los resultados del proyecto.

- Apéndices

En los Apéndices se brinda información teórica complementaria, un Manual de Usuario y Técnico.

Capítulo 2

Descripción del sistema

El sistema esta formado por una red de nodos distribuidos geográficamente, cada nodo se compone de diferentes módulos que cumplen distintas funciones, por ejemplo tomar medidas ambientales.

Los nodos se comunican de forma inalámbrica con un nodo maestro, el cual llamaremos *Nodo Principal*. Este nodo es el encargado de recibir las consultas del servidor y de recabar los datos de todos los sensores solicitados ya sea de su nodo o de los nodos secundarios, los cuales llamaremos *Nodos N*. En cada nodo, la comunicación con los sensores se hará mediante I2C, ya que es el protocolo utilizado por el grupo de medidas de la facultad.

Cada sensor, el cual llamaremos *Módulo Sensor*, está identificado con una dirección. Esta es una dirección I2C. Por lo tanto el Servidor envía las direcciones de los sensores que deben realizar alguna acción.

En la siguiente figura 2.1 se muestra una representación básica de la comunicación Sensor-Servidor.

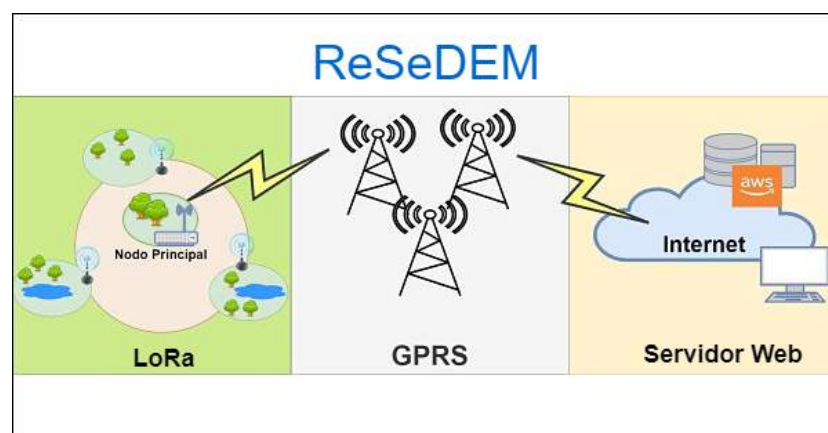


Figura 2.1: Descripción de la red

Capítulo 2. Descripción del sistema

2.1. Descripción de los componentes del sistema

A continuación se describen los principales componentes del sistema y su función:

Servidor - Es el encargado de guardar registro de los datos recabados por los sensores. Es la interfaz entre el usuario y los sensores. Desde el mismo se puede configurar los sensores, tiempo de muestreo y solicitar datos de forma independiente a los mismos.

Nodo Principal - Nodo intermediario entre el servidor y los sensores de toda la red. Está conformado por un Módulo GPRS, un Módulo LoRa y Módulos Sensores. Recibe las peticiones del Servidor y consulta a quien corresponda, ya sea sensores de su Nodo o de otros Nodos de la red (Nodos N).

Nodos N - Son los nodos secundarios, a diferencia del Nodo Principal, no tienen comunicación con el Servidor de forma directa, responden a pedido del Nodo Principal. Están conformados por un Módulo LoRa y Módulos Sensores.

Módulo LoRa - Nexos entre el Nodo principal y los Nodos N. Realiza las consultas a los sensores cuando corresponda. Es el módulo que maneja el Nodo. En el Nodo Principal, además, es quien administra la Red LoRa.

Nota: Al Módulo LoRa Principal, se lo llamará también LoRa Principal, mientras que a los Módulos LoRa de los nodos N se los llamará LoRa Secundarios.

Módulo GPRS - Se encarga de establecer la comunicación con el Servidor de forma bidireccional.

Módulo Sensor - Son el extremo de la red, responden solamente cuando son solicitados enviando medidas de lo sensado. Se comunican mediante I2C con el Módulo LoRa del nodo al cual pertenecen.

Red LoRa - Conjunto de Módulos LoRa que se comunican mediante ondas de radio frecuencia utilizando el protocolo LoRa.

En la figura 2.2 se observa un esquema del sistema y sus partes constitutivas.

2.2. Funcionamiento global del sistema

A continuación se presenta una secuencia general del funcionamiento del sistema. En los siguientes capítulos, se describe más detalladamente cómo se realiza la secuencia y los intercambios de mensajes.

Secuencia:

2.2. Funcionamiento global del sistema

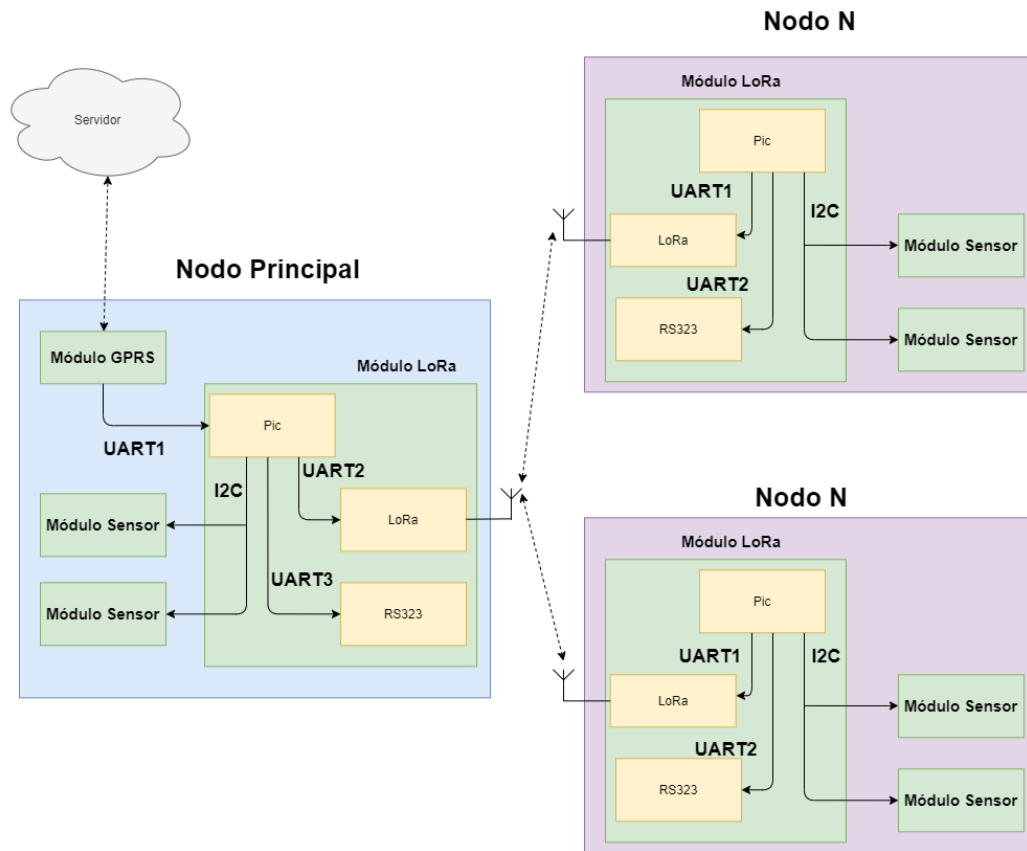


Figura 2.2: Esquema de la arquitectura del sistema

1. El Módulo GPRS consulta al servidor por las acciones que hay que realizar en la red, por ejemplo, solicitar medida a un sensor determinado
2. El Servidor responde con acciones que debe realizar la red de sensores (agregar nodos, cambiar parámetros, solicitar medidas)
3. El Módulo GPRS envía los datos recibidos al Módulo LoRa Principal
4. El Módulo LoRa Principal consulta por su bus de datos y alimentación y por la Red LoRa la solicitud. Todos los módulos LoRa de los Nodos N envían por su bus de datos la consulta y esperan respuesta.
5. Solo responderá el Módulo Sensor solicitado. Para la respuesta existen 2 posibilidades:
 - Si la solicitud corresponde algún componente del Nodo Principal, la respuesta viene en el bus de datos I2C del Nodo
 - Si la solicitud corresponde a algún componente de un Nodo N, este módulo lo enviará al Nodo Principal mediante Módulo LoRa
6. El Módulo LoRa Principal envía la respuesta al Módulo GPRS

Capítulo 2. Descripción del sistema

7. El Módulo GPRS envía los datos al Servidor y en la respuesta recibe en cuantos minutos será la hora de la próxima consulta
8. El Módulo GPRS va a un modo de ahorro de energía hasta la hora de la próxima solicitud

Siguiendo la secuencia antes mencionada, en la figura 2.3 se muestra el funcionamiento del sistema en el caso de que el Servidor solicita dato al sensor S_4 perteneciente a un Nodo N.

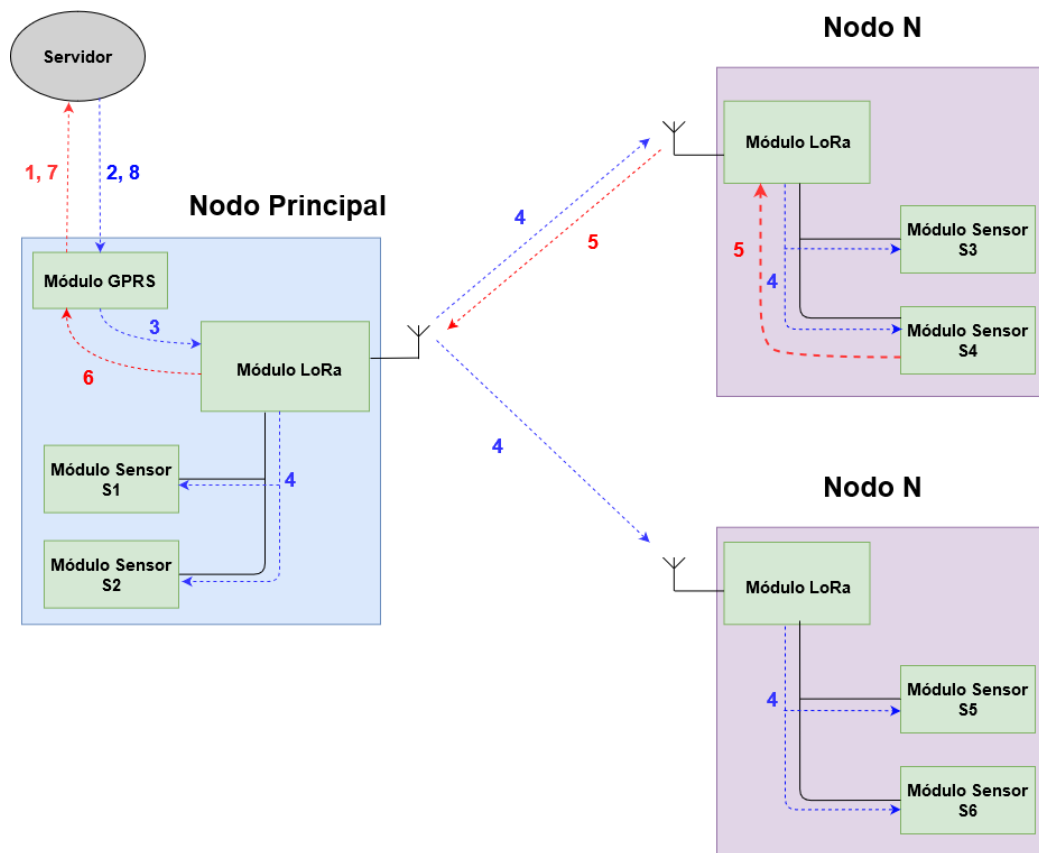


Figura 2.3: Ejemplo de funcionamiento del sistema

2.3. Ejemplos de posibles aplicaciones

El sistema es pensado de forma muy flexible, para que conectando en los extremos de la red un sensor de cualquier tipo, la medida de este sea capaz de llegar al servidor. Solo se debe mantener el firmware propuesto, introduciendo las funciones de medida/datos a transmitir específicas.

2.3. Ejemplos de posibles aplicaciones

A efectos prácticos, se decide instanciar el equipo de medición con sensores de variables medioambientales por ser de interés del Grupo de Medidas e Instrumentación del IIE.

A continuación se presentan una serie de aplicaciones que se pueden implementar:

- **Prevención de incendios forestales:** tomando medidas de temperatura, humedad y velocidad del viento en una gran extensión se podría detectar o predecir focos de incendios.
- **Invernaderos:** Se puede monitorizar la temperatura, humedad y luz e implementar un sistema remoto de control
- **Calidad de agua sobre un río:** Existe un proyecto de facultad de Química junto con el Departamento donde se monitorea el nivel de fósforo y PH del agua. Todo el sistema de bombeo, medición y procesamiento de lo medido puede ser considerado como un Módulo Sensor a los efectos de este proyecto. Pudiéndose implementar varios de estos equipos a unos km de distancia para tener varios puntos de medida.
- **Ciudades:** Monitoreos en diferentes puntos de la ciudad obteniendo los diferentes niveles de ruido o coordinando los semáforos en base a los niveles de tráfico.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 3

Servidor

3.1. Implementación del Servidor

La red de sensores cuenta con un servidor web para el almacenamiento y despliegue de los datos recolectados por la red de sensores y a su vez la configuración de los mismos. Actualmente existen diversos servicios y librerías que facilitan la implementación de este tipo de aplicaciones, pero una de las ideas principales de este proyecto fue poder realizar, aprender y desarrollar todos los aspectos claves para el funcionamiento de la red. Por este motivo se buscaron las herramientas que permitieran el mayor control sobre el desarrollo de la aplicación web.

3.2. Transferencia de datos

Para la transferencia de datos desde el Nodo Principal al Servidor se buscó un protocolo que fuera confiable de modo de poder asegurar que todos los datos lleguen correctamente sin tener pérdidas. Investigando los diferentes protocolos existentes de internet e IoT se decidió optar por HTTP. Éste protocolo es el utilizado para las transferencias de información en la World Wide Web (WWW) y se cuenta con mucha documentación, tanto para implementar en el servidor como para desarrollar en lenguaje C en nodos de la red. HTTP es un protocolo de capa de aplicación del modelo OSI, sobre TCP/IP, cumpliendo con los requisitos que se tenían previstos para la red de sensores. TCP brinda la confiabilidad en la transmisión de datos gracias a su mecanismo de reconocimiento y retransmisión de los datos sobre internet.

HTTP basa su arquitectura de forma petición-respuesta entre el servidor web y un cliente; para nuestra red el cliente es el nodo principal el cual realiza consultas al servidor web.

Capítulo 3. Servidor

3.2.1. Método GET

HTTP define una serie predefinida de métodos de petición, El método GET solicita una representación del recurso especificado al servidor. Para la red, el Nodo Principal hace peticiones GET a los distintos recursos del Servidor obteniendo las respuestas sobre las acciones a tomar. Cuando el Servidor recibe una petición GET ejecuta scripts programados en PHP para devolver la respuesta. más adelante se detalla el funcionamiento de estos scripts.

3.3. Servidor Web y base de datos

El Servidor Web debe ser accesible a través de internet, por lo que se contrató el servicio de hosting de Amazon, Amazon Web Service (AWS). Sobre este servidor se desarrollaron los programas encargados de implementar el servidor web con HTTP, la base de datos de la red y la interfaz gráfica para los usuarios.

Se estudiaron diferentes formas de desarrollar la base de datos y la interfaz gráfica, eligiéndose el conjunto MYSQL, PHP, HTML, CSS y JavaScript.

- **MYSQL:** es un administrador de bases de datos basado en el popular lenguaje SQL para la gestión de sistemas de bases de datos relacionadas. Cuenta con mucha documentación.
- **PHP:** es un lenguaje de programación del lado servidor que permite realizar páginas dinámicas y la interacción con las bases de datos.
- **HTML y CSS:** son lenguajes de programación para el diseño de páginas web, HTML se basa en etiquetas para dar estructura y contenido mientras que CSS se encarga de darle estilo a la página web, combinando ambos lenguajes se desarrolla la interfaz gráfica para los usuarios.
- **JavaScript:** es un lenguaje de programación utilizado principalmente del lado del cliente para mejorar las interfaces de usuarios y páginas web dinámicas.

En el servidor de Amazon se instala el paquete LAMP (Linux, Apache, MYSQL, PHP) que contiene los programas necesarios para poder utilizar los lenguajes mencionados anteriormente y además proporciona un servidor web basado en HTTP mediante el programa conocido como Apache.

De esta forma se implementa un servidor web que cuenta con una interfaz gráfica, permite acceder a las bases de datos con la información recibida de parte del Nodo Principal y a su vez permite enviarle comandos para las distintas funciones de la red de sensores.

3.3.1. Estructura Base de datos

Como se menciona anteriormente, toda la información de la red se debe guardar en bases de datos (DB), desde los datos de mediciones de los Sensores hasta los

3.4. Descripción aplicación Web

parámetros de configuración de los elementos del sistema. La figura 3.1 muestra la estructura de como se almacena la información.

En MYSQL se tiene una base de datos (DB) para las configuraciones generales de la red llamada configuraciones y otra DB por cada nodo de la red.

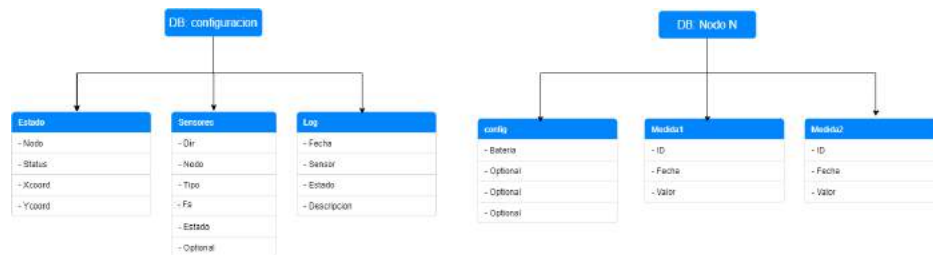


Figura 3.1: Estructura de la base de datos

3.3.1.1. Configuración

En esta DB se almacena la información general de la red, como lo son los nodos y sensores que la componen. Dentro de “configuración” hay una tabla “Estado” donde se guardan los nodos, coordenadas geográficas y estado de los nodos en la red. Luego en la tabla “Sensores” se almacena todo lo referido a los sensores de cada nodo, por ejemplo, a que nodo corresponde el sensor, que medida toma y que estado tiene. Por último se tiene la tabla “log” donde se guardan todos los mensajes y eventos de la red.

3.3.1.2. Nodo

Por cada nodo se crea una DB para almacenar los datos medidos y la configuración de dicho nodo. Además por cada sensor que tenga el nodo se crea una tabla donde almacenar las medidas realizadas, por cada medida se guarda el valor y el tiempo en cuál se midió.

3.4. Descripción aplicación Web

La interfaz Web muestra información general de la red y una barra lateral donde se encuentran las diferentes funciones que se pueden realizar:

- Agregar/Quitar nodos
- Agregar/Editar/Quitar sensores
- Información de cada nodo
- Monitor del sistema en tiempo real
- Log de estados

Capítulo 3. Servidor

El diseño de la web esta basado en Bootstrap, una librería muy popular para el diseño de sitios web. esto incluye diferentes estilos de colores, botones, y barras de herramientas para utilizar con CSS y HTML.

3.4.1. Scripts de manejo de datos

El Servidor ejecuta dos scripts principales para el manejo de datos y comunicación con el Nodo principal.

- Consulta.php
- Proc.php

Estos scripts cada vez que son solicitados por el Nodo Principal a través del método GET de HTTP informan sobre que acciones debe tomar sobre la Red. Luego que el nodo principal realice todas las acciones devuelve los resultados otra vez a otra URL del servidor para que sean almacenados en las distintas bases de datos y luego se puedan ver en la Web del Servidor.

3.4.1.1. Consulta.php

Este script cada vez que es ejecutado por el Nodo Principal realizando un GET a la URL: *resedem.bitnamiapp.com/resedem/consulta.php* devuelve un mensaje con las acciones que debe tomar el Nodo principal. Este mensaje contiene la dirección de sensores que necesitan datos y si hay Sensores para agregar o eliminar. En el Capítulo 8 se detalla el formato del este mensaje. La figura 3.2 muestra las funciones que realiza el script cada vez que se ejecuta.

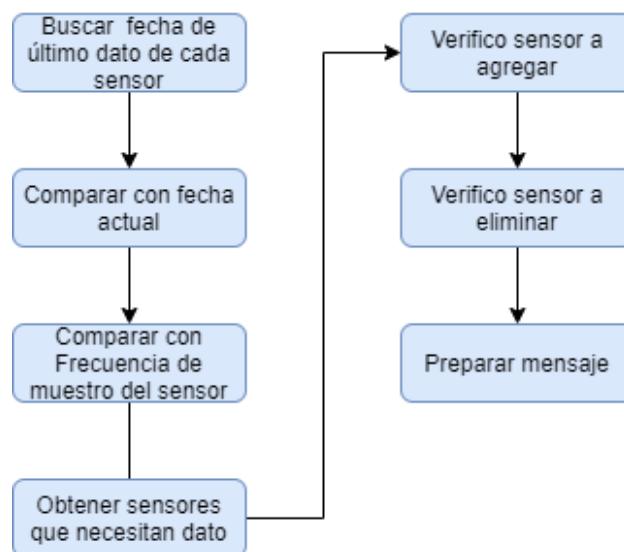


Figura 3.2: Script consulta.php

3.4. Descripción aplicación Web

Como se muestra en la figura, para cada sensor se toma la fecha de la última muestra, se la compara con la fecha actual (fecha al momento de ejecutar el script y/m/d hh:mm:ss) y luego esta diferencia se la compara con la frecuencia de muestro (fs) del Sensor para ver si necesita dato o no. Por último se busca si hay Sensores para agregar y/o eliminar.

3.4.1.2. Proc.php

Este script se ejecuta para guardar datos en el servidor, el Nodo principal realiza un GET a la URL *resedem.bitnamiapp.com/resedem/proc.php?valor = mensaje*, donde *mensaje* son los datos que se desean guardar. El formato de los mensajes se puede ver en detalle en el Capítulo 8.

Se compone de direcciones de sensores más un dato medido o un estado del Sensor. El script se encarga de guardar el la tabla de la DB que corresponde los mensajes recibidos.

Luego de guardar todos los datos se devuelve al Nodo Principal cuando se debe realizar la próxima comunicación, este valor se obtiene a partir de la frecuencia de muestreo de los diferentes sensores. La figura 3.3 resume el comportamiento del script.

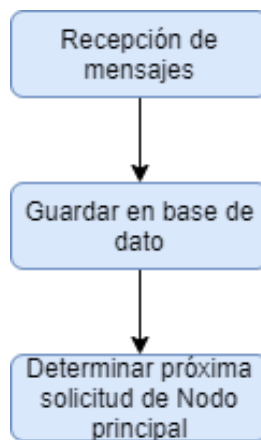


Figura 3.3: Script proc.php

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 4

Módulo GPRS

4.1. Características que debe cumplir el módulo

Las principales características que debe cumplir el módulo son:

- Servir de nexo entre la Red de sensores y el servidor
- Tener cobertura nacional
- Tener un consumo reducido de energía
- Ser transparente en la comunicación. Es decir, lo que recibe del Servidor debe enviarlo directamente al Módulo LoRa principal. De igual forma lo que recibe del Módulo LoRa enviarlo directamente al Servidor. Por lo tanto el manejo de datos es mínimo, dedicado principalmente a establecer la comunicación.

4.2. Tecnología de comunicación

En la actualidad existe una gran variedad de tecnologías que permiten acceso a internet y por ende permiten enviar datos a servidores.

El estudio para la elección de la tecnología se centro en 3 de ellas. Estas fueron en principio seleccionadas para comparar debido a su popularidad y por la gran trascendencia que están teniendo en la actualidad.

Tecnologías a analizar:

- GSM (2G)
- UMTS (3G)
- NARROWBAND IoT (NB-IoT)

Las tecnologías GSM, en su extensión GPRS para datos, y su evolución, UMTS, son pensadas para la comunicación celular. GSM es una de las tecnologías preferidas históricamente para el acceso a internet para la transición de datos de forma

Capítulo 4. Módulo GPRS

inalámbrica.

En cambio NB-IOT es una tecnología desarrollada específicamente para el IoT. Es más reciente en el tiempo, pero que se está imponiendo rápidamente.

Características de velocidad y consumo

En general las tecnologías para la comunicación de los dispositivos de IoT se caracterizan por utilizar un ancho de banda muy chico, lo que implica velocidades lentas de transmisión de datos. Como ventaja tienen un consumo de energía en transmisión mucho menor que las tecnologías celulares. Tal es el caso de NB-IOT.

En la tabla 4.1 se presentan las características teóricas de velocidad y consumo para cada tecnología.

Tecnología	Velocidad subida (kbps)	Consumo medio Tx (mA)
GPRS	20	170
UMTS	384	500
NB-IoT	250	110

Tabla 4.1: Tecnologías para acceso a internet

Analizando el consumo medio en transmisión, donde el consumo es mayor, se aprecia la ventaja de las tecnologías IoT, respecto a las tradicionales.

Si bien, se pensaría que elegir la tecnología IoT sería la mejor opción, la gran desventaja al inicio del proyecto era la cobertura nacional de la misma.

Cobertura

Se estudió la cobertura que proporcionan los principales ISP (Internet Service Provider) de Uruguay, observando que el despliegue en GSM y UMTS es muy grande. Específicamente en GSM, Antel, tiene una cobertura casi total. Y en menor medida de UMTS.

La tecnología NB-IoT utiliza el despliegue de la red de LTE, lo cual representa una gran ventaja. En el sitio web de Antel, se puede observar como dicha cobertura esta a la altura de GSM. Por lo tanto ésta sería la tecnología ideal para una red de sensores que se quiera desplegar en cualquier parte del Uruguay.

Al momento de realizar el análisis de cual tecnología utilizar, NB-IOT no tenia un despliegue más allá de Montevideo por lo que por un tema de cobertura se descartó dicha opción.

4.3. Hardware del Módulo

En la figura 4.1 se muestra la cobertura del ISP Antel, para las diferentes tecnologías. Se muestra el despliegue al inicio del proyecto y al final.

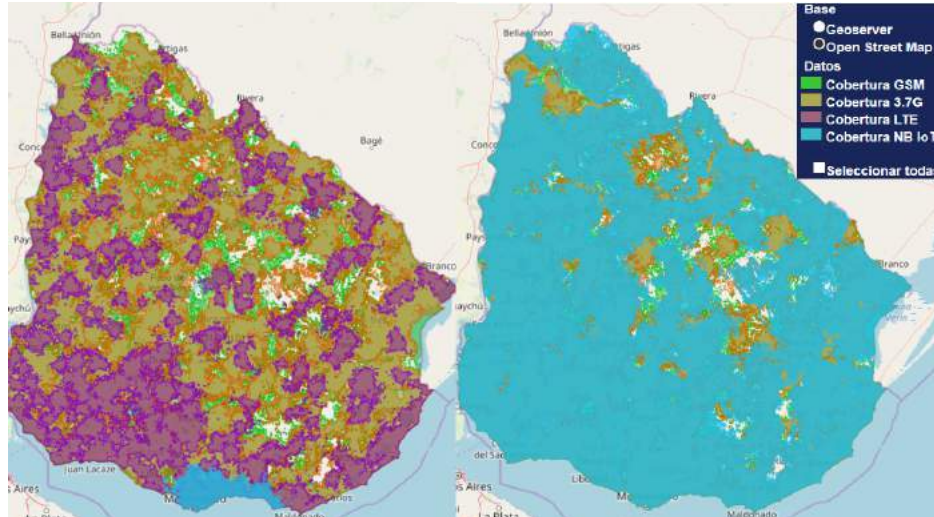


Figura 4.1: Cobertura ISP ANTEL - Inicio proyecto (Izq), Actualidad (Der) [10]

4.3. Hardware del Módulo

Para cumplir con las características funcionales descritas en 4.1 se optó por el siguiente hardware:

- Placa de comunicación SIM900 de Seedstudio, en su versión SHIELD V3.0 para Arduino. Ver figura 4.2a
- Arduino Nano V3.0. Ver figura 4.2b

4.3.1. Modem de comunicación SIM900

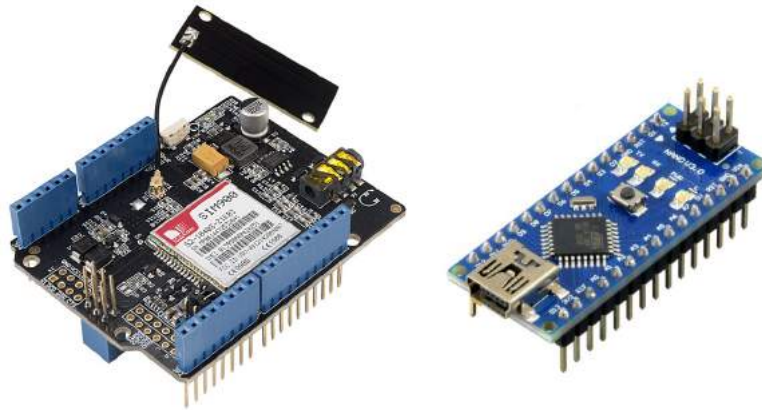
El Modem de comunicación elegido fue el SIM900. Se optó por comprar en su versión de shield para Arduino del fabricante Seedstudio ya que es el microcontrolador elegido para el módulo. Ver sección 4.3.2.

Las principales características del Shield del SIM900 [6] son:

- Chip elaborado por SIMCOM [5]
- Cuatribanda: 850/900/1800/1900 MHz
- Funciona con comandos AT
- Se comunica con el arduino mediante serial UART (pines 7, 8 y 9)
- Tiene modo de bajo consumo, 15mA

Capítulo 4. Módulo GPRS

- En TX, tiene picos de hasta 2A
- Incorpora pila TCP/IP



(a) Shield GSM [6]

(b) Arduino Nano

Figura 4.2: Hardware Módulo GPRS [11]

4.3.1.1. Comandos AT

Los comandos Hayes AT son comandos para operar Modems elaborado por la compañía Hayes Communications y posteriormente estandarizado por la ITU¹ en el standard V.250.

Estos permiten la interacción entre el usuario y el Módem de forma sencilla. Con ellos se puede obtener parámetros de la red, estado del módem, realizar llamadas, enviar SMS, y establecer una conexión a Internet, entre otros.

Se tienen 3 tipos de comandos AT:

1. **Básicos** - Realizan acciones - $AT + < x > < n >$; donde $< x >$ es el comando y $< n >$ el argumento. Por ejemplo una llamada “ATD+59898699974”.
2. **Parámetros** - Modifican registros del Modem- $ATS < n > = < m >$; donde $< n >$ es el número de registro S y $< n >$ el valor a asignar
3. **Ampliados** - Comandos para lectura escritura, prueba y ejecución:
 - a) **Prueba** - Devuelve los parámetros modificables y los rangos aceptables de los mismos - $AT + < x > = ?$; donde $< x >$ es el comando
 - b) **Lectura** - Devuelve el valor del comando - $AT + < x > ?$; donde $< x >$ es el valor del comando que se quiere conocer

¹Unión Internacional de Telecomunicaciones

4.4. Firmware del módulo

- c) **Escritura** - Configura un comando - $AT+ < x > = < \dots >$; donde $< x >$ es el valor del comando que se quiere configurar
- d) **Ejecución** - Ejecuta un proceso con parámetros no variables - $AT+ < x >$ donde $< x >$ es el valor del comando a ejecutar

Específicamente el estudio se centró en lograr establecer la comunicación con un servidor web mediante HTTP. En la Tabla 4.2 se describen los comandos AT (ampliados) más importantes para lograr establecer la conexión mencionada.

Comando AT	Descripción	Principales parámetros
AT+SAPBR	Establece conexión de datos	GPRS, APN, IP
AT+HTTPINIT	Inicia conexión HTTP	-
AT+HTTPPARA	Establece los parametros de conexión HTTP,	URL, Datos a enviar
AT+HTTPACTION	Realiza el GET al servidor	GET, POST
AT+HTTPREAD	Lee respuesta del GET	Respuesta servidor
AT+HTTPTERM	Finaliza conexión HTTP	-

Tabla 4.2: Comandos AT para conexión a servidor WEB [7]

En el Apéndice C se describen otros comandos AT que pueden ser de utilidad a la hora de establecer una conexión diferente con el proveedor de internet, agregar prestaciones a la conexión, y funcionalidades extras como pueden SMS, por ejemplo.

4.3.2. Arduino Nano

Se optó por utilizar Arduino debido a que ya se tenía conocimiento del mismo y también debido a la gran cantidad de documentación existente por ser de código abierto. En la Tabla 4.3 se muestran las principales características de los módulos analizados.

Más específicamente se elige Arduino Nano debido al bajo consumo que éste tiene. Además, cumple con los requisitos básicos necesarios para la comunicación con el Shield GPRS y por ende cumple los requisitos especificados en la sección 4.1

4.4. Firmware del módulo

4.4.1. Funcionamiento

El funcionamiento del Módulo GPRS es el siguiente:
Se realiza una consulta al Servidor y éste le devuelve las acciones que deben realizar ciertos sensores o algún parámetro a modificar de la Red. El Módulo se lo envía directamente al Módulo LoRa y queda esperando la respuesta de este último.

Capítulo 4. Módulo GPRS

Características [11]	Uno	MEGA	DUE	Nano
μ C.	ATMega328	ATMega2560	AT91SAM3X8E	ATMega328
Tension Op. (V)	5	5	3.3	5
Digital I/O	14	54	54	14
PWM Pins	6	15	12	6
Memoria Flash (KB)	32	256	512	32
SRAM (KB)	2	8	96	2
EEPROM (KB)	1	4	-	1
Consumo (mA)	46	93	75	15

Tabla 4.3: Comparativa Arduinos

Una vez recibida la respuesta, la envía al Servidor. Por último el Servidor le indica cuándo volver a preguntar y el Módulo se va a un modo de ahorro de energía hasta que sea la hora de la próxima consulta. EL servidor no despierta al módulo para realizar acciones, es siempre el módulo quien realiza la consulta.

En la figura 4.3 se detalla una secuencia exitosa que realiza el Módulo GPRS.

En el Capítulo 8 se describe la secuencia completa de intercambio de mensajes del Módulo GPRS con el Servidor y con el Módulo LoRa.

4.4.2. Máquinas de Estado

Para lograr el funcionamiento antes descrito, el Arduino se maneja mediante máquinas de estado que manejan el estado global del Módulo. Entre las que se utilizan se destacan las 3 principales las cuales funcionan en “paralelo” y están interconectadas:

- *Arduino*
- *Modem*
- *Consultas*

Las anidaciones entre las máquinas se realiza con banderas que se levantan. En la figura 4.3 se pueden observar las Máquinas de estados junto a sus estados.

Máquina Arduino

La máquina de estados *Arduino* se encarga de controlar el modo de funcionamiento del módulo. Es necesario que el Arduino Nano, que controla la comunicación con el Servidor, se encuentre en un modo de bajo consumo mientras no sea necesaria la comunicación. Por lo tanto se tienen dos estados (Activo y Bajo Consumo). En el estado bajo consumo es el Módulo está la mayor parte del tiempo.

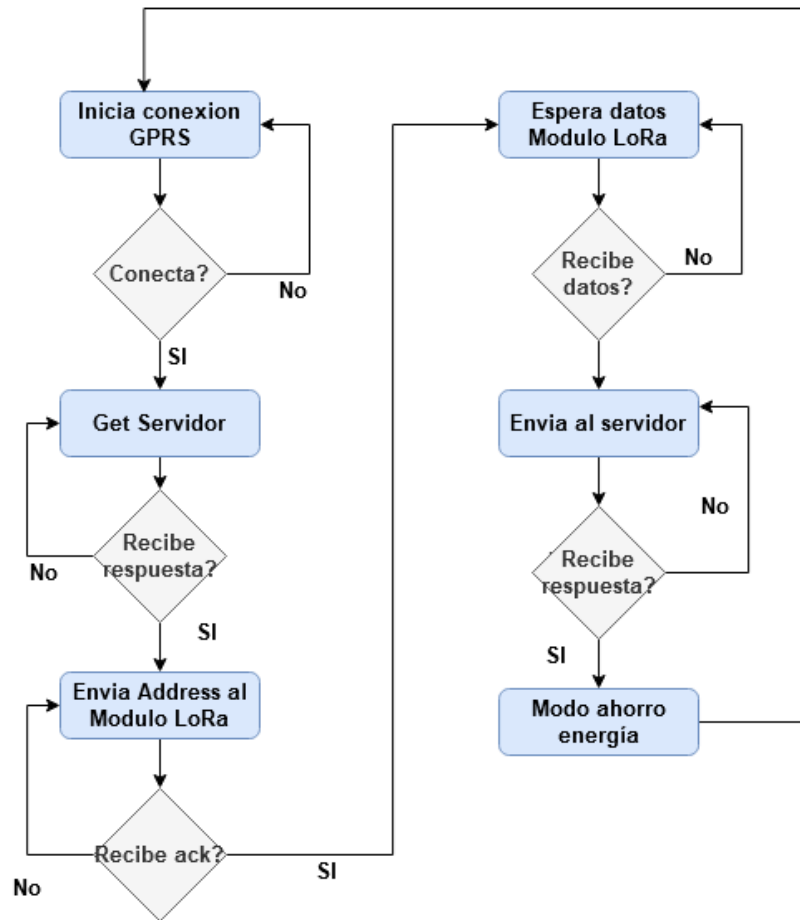


Figura 4.3: Secuencia de funcionamiento Módulo GPRS

Máquina Modem

La máquina de estados *Modem* es la que se encarga de realizar la conexión de datos del Modem (SIM900) a la red y establecer una comunicación HTTP. Para minimizar el consumo, mientras no es necesario, el SHIELD está apagado.

El Modem funciona como se indica a continuación. Luego de encendido, se registra en la red del ISP Antel y solicita una conexión de datos con lo cual se le asigna una IP. A continuación, debido al protocolo establecido, inicia una comunicación HTTP, para luego de realizar los GET correspondientes, finalizar la comunicación HTTP.

Por lo tanto la máquina cuenta con los estados de Apagado, Encendido, Registro en la Red, Conexion de datos, Inicio de comunicación HTTP y finalización de comunicación HTTP.

Estos estados funcionan uno a continuación del otro. Con la salvedad de que

Capítulo 4. Módulo GPRS

entre que los estados Inicio HTTP, y Terminar HTTP se pasa a la máquina de estados *Consulta* para realizar los GET.

Máquina Consultas

Esta máquina es la que se encarga de realizar los dos Get con el servidor. Entre ambos Get se envía a Nodo LoRA Principal y se espera por las medidas que este devolverá.

Todos los estados de esta máquina cuentan con una cierta cantidad de reintentos antes de indicar error. En caso de suceder un error, se reinicia la máquina para poder establecer una conexión exitosa con el servidor. En la figura 4.4 se puede observar un diagrama con los estados de cada máquina.

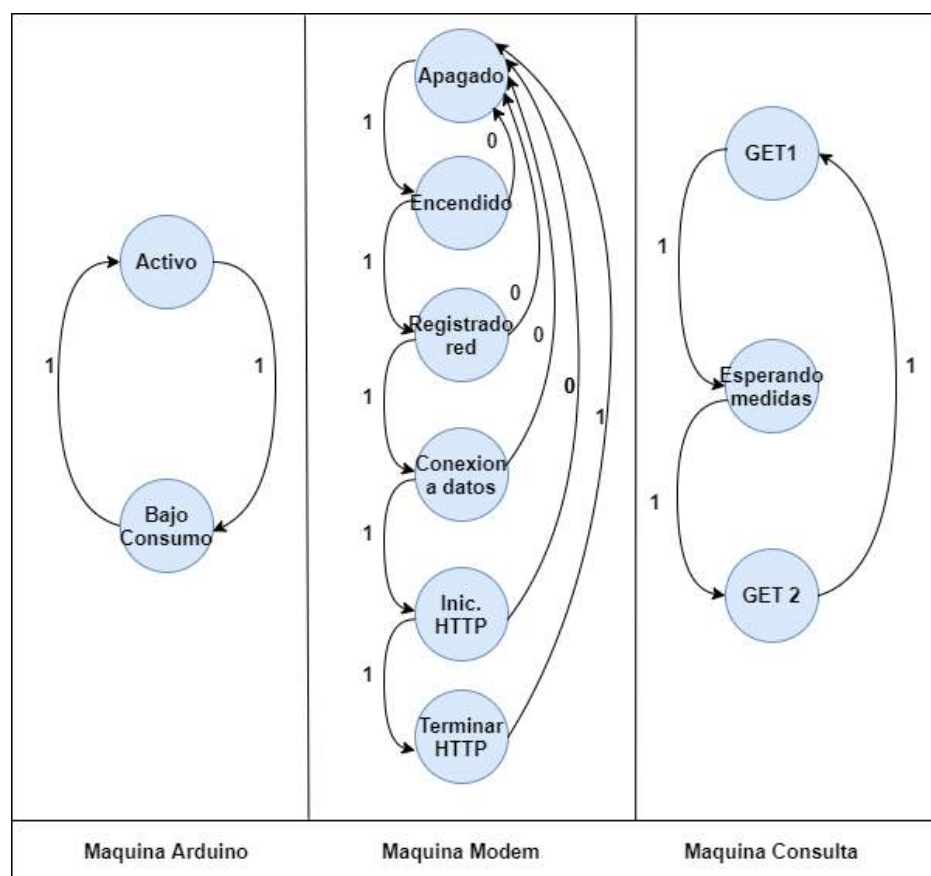


Figura 4.4: Maquinas de estado del Módulo GPRS

4.4.3. Librerías auxiliares

Para la comunicación y la configuración del Modem se utilizan librerías ya existentes. Debido a la popularidad del GSM para Arduino, existen varias librerías. Se elige la librería de *Seeed Technology Co, LTD* [8], escrita por Lawliet Zou y

4.4. Firmware del módulo

licenciada por las licencias MIT.

Esta librería cuenta con todas las funciones necesarias para una completa comunicación con el Shield GSM.

La librería esta dividida en 2 partes. Están las funciones que permiten la comunicación del Arduino con el shield y luego las funciones específicas para la configuración del Modem SIM900 mediante el envío de los comandos AT.

Se le realizan algunos cambios a las funciones de *httpSendGetRequest* y *httpReadResponseData* para mejorar la optimización de la misma y adaptarlas a las necesidades del proyecto.

- *httpSendGetRequest* - Es la función encargada de realizar el Get al servidor, en nuestro caso enviar algún mensaje al servidor.
- *httpReadResponseData* - Es la función encargada de leer la respuesta del servidor al Get previamente hecho.

Para el modo Bajo Consumo del Arduino se utiliza la librería *LowPower*. Ésta librería es la que obtiene el menor consumo cuando se está en modo ahorro de energía. Permite diferentes modos de bajos consumos, configurables a través de los siguientes parámetros:

- *ADC_Off*: Apaga los convertidores Analógico/Digital.
- *BOD_On*: Enciende el circuito de Brown Out Detection, que es un circuito para detectar niveles peligrosamente bajos de tensión. Esto evita que se dañe la memoria el controlador.
- *Time_Sleep*: Máximo tiempo del Watchdog con el cual el sistema se despertará.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Módulo LoRa

5.1. Características que debe cumplir el módulo

Si bien existe una diferencia en cuanto a las tareas que deben desarrollar el Módulo LoRa del Nodo Principal y el de los Nodos N, constructivamente serán iguales, por lo que todos deben cumplir las siguientes características:

- Comunicación inalámbrica con otros módulos del mismo tipo:
 - Ser de una tecnología de bajo consumo
 - Tener un alcance de al menos 1 Km
- Tener comunicación I2C con los módulos sensores
- Tener comunicación UART con el Módulo GPRS
- Tener un consumo mínimo de energía
- Poder de procesamiento ya que administrará la Red LoRa y la comunicación I2C

5.2. Tecnología de comunicación inalámbrica

Posibles tecnologías IoT para establecer la comunicación en un principio:

- Zigbee
- LoRa

5.2.1. Zigbee

Está diseñado sobre el estándar IEEE 802.15.4. Es un protocolo de recuperación automática, seguro, robusto y con capacidad de implementar una red mallada. Se puede escalar a miles de nodos en grandes áreas.

Capítulo 5. Módulo LoRa

Dentro de una arquitectura de sistema ZigBee, hay 3 tipos de dispositivos. Coordinador, enrutador y el dispositivo final. Solo hay un coordinador en la red que selecciona la topología de la red, establece la red y administra la información de configuración. Actúa como la puerta de enlace dentro y fuera de la red, por lo que debe estar encendido y funcionando en todo momento. Los enrutadores transmiten información y mueven datos a través de la red y como representan la columna vertebral de la red, siempre deben estar activados. El dispositivo final está en el borde de la red y es la fuente o el usuario de los datos de la red. Por lo general, funciona con batería y se puede colocar en modo de reposo y baja potencia durante largos períodos de tiempo.

Es una solución de radio con un rango medio de señal no mayor a 100 metros pudiendo utilizar una arquitectura en estrella para lograr rangos mayores.

5.2.2. LoRa

LoRa es modulación de largo alcance y bajo consumo desarrollado por SEMTECH. Se basa en la técnica de modulación por espectro expandido SS.

La modulación LoRa ha crecido enormemente como una de las principales tecnologías de transmisión de datos para IoT debido a las características que se mencionan a continuación.

Esta tecnología opera en las frecuencias de sub GHz (menores a 1GHz). Las grandes ventajas de esta modulación son el gran alcance que se logra (15Km) y el bajo consumo que tienen. Como desventaja se tiene una tasa de transmisión bastante baja que va desde 0.3 a 50 kbps. Variando ciertos factores (CR y SF) se puede modificar la tasa de transferencia en función del alcance o viceversa. Esto afecta a su vez el consumo, por ejemplo, para tener un mayor alcance se baja la tasa de transmisión con lo que se requiere más tiempo en transmisión para transmitir la misma cantidad de datos. En el Apéndice A se describe de forma más detallada la comunicación LoRa junto con los parámetros antes mencionados.

Por lo tanto, esta tecnología de IoT es buena para aplicaciones que envían pocos bytes de datos, aunque lo hagan muchas veces al día.

Dado que LoRa cumple con todos los requerimientos para la comunicación inalámbrica del proyecto, especialmente un largo alcance y bajo consumo, se optó por utilizar esta tecnología para implementar la red de nodos que se encargará de la adquisición de medidas.

5.3. Hardware principal del módulo

El módulo de comunicación esta compuesto por dos componentes fundamentales:

- Módulo de Comunicación LoRa - RN2903
- Microcontrolador dsPIC33

5.3.1. Módulo de Comunicación LoRa - RN2903

Para la modulación y transmisión de datos en la comunicación LoRa se utiliza el módulo RN2903 de Microchip. Este módulo cuenta con un transeiver y un microcontrolador. Ver figura 5.1a.

En la tabla 5.1 se presentan las principales características del módulo elegido.

Característica	Valor
Fabricante	Microchip
Frecuencia central	915 MHz
Potencia Tx	18.5 dBm
Consumo Tx	124 mA (max)
Sensibilidad Rx	-146 dBm

Tabla 5.1: Características del chip RN2903 [12]

Así como en GSM se utilizan los comandos AT, el RN2903 utiliza los siguientes comandos:

- ***“radio set «comando» «valor»”*** para configurar parámetros de la red.
- ***“radio get «comando»”*** para leer la configuración
- ***“radio «comando» «valor»”*** para el manejo de la comunicación por radio frecuencia.
- ***“sys «comando»”*** para manejar el chip.
- ***“mac «comando»”*** en caso de querer utilizar el chip para conectarse a una red LoRaWAN. Este protocolo no se utiliza en el proyecto, pero se estudio su implementación

Se observan en la tabla 5.2 los comandos necesarios para configurar el módulo acorde a las especificaciones del proyecto.

Tipo	Comando	Valor	Descripción
radio set/get	frec	902 a 928 MHz	Configura/Lee frecuencia central de la Red.
	pwr	2 a 20	Configura/Lee potencia de transmisión .
	sf	sf7 a sf12	Configura/Lee spread factor.
	cr	4/5 a 8	Configura/Lee code rate.
	bw	125, 250, 500 kHz	Configura/Lee ancho de banda.
	wdt	0 a 4294967295 ms	Configura/Lee ancho de ventana de tiempo de transmisión y recepción. '0' para desactivar.
radio	rx	0 a 65535 simbolos	Configura cantidad de caracteres a recibir. '0' activa modo escucha continua.
	tx	<<mensaje>>	Transmite mensaje por radio frecuencia. <<mensaje>>en representación hexadecimal.
sys	reset		Resetea y reinicia el módulo RN2903.
mac	pause		Pausa funcionalidades de LoRaWAN.

Tabla 5.2: Principales comandos de configuración LoRa [13]

Se monta según la recomendación del fabricante, con capacitores de desacople en todos los pines de alimentación y un conector SMA para la antena. También se monta con un conector de cinco pines con el fin de poder actualizar el firmware del chip y una resistencia en forma de pull up para el pin de reset ya que este se activa por nivel bajo.

5.3.2. Microcontrolador - dsPIC33

El microcontrolador que tienen los Módulos LoRa es el encargado de realizar toda la administración de la Red. Se encarga tanto de gestionar las acciones solicitadas desde el servidor como de solicitar medidas de los distintos sensores de los todos los nodos.

Se optó por un microcontrolador del fabricante Microchip (dsPIC33ep128GM604), ya que se tenía conocimiento de este hardware y además se tenía disponibilidad

5.3. Hardware principal del módulo

del componente en plaza. Esto permitía disminuir considerablemente la curva de aprendizaje y el tiempo de importación en el caso de elegir otro. Ver figura 5.1b.

En la tabla 5.3 se presentan las principales características del microcontrolador elegido.

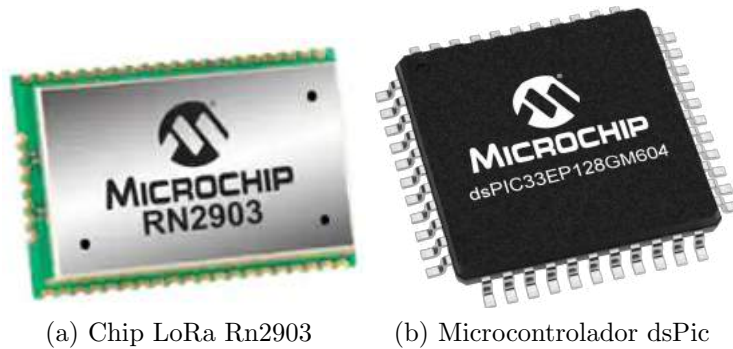
Característica	Descripción
Fabricante	Microchip
Modelo	dsPIC33EP128GM604
Arquitectura	16-bit
Frecuencia reloj (max.)	70 MHz
Tension operación	3.3 V
Memoria ROM	128 KB
SRAM	16 KB
Pines	44
CAN	2
SPI	3
UART	4
I2C	2
Timers	9
PWM	12
ADC	2
ADC Channel	18
I/O pins	35

Tabla 5.3: Principales características del microcontrolador elegido [14]

Se utiliza el microcontrolador con un oscilador externo de 20 MHz y capacitores cerámicos de desacople en todos los pines de alimentación como recomienda el fabricante. También se agrega un capacitor de tantalio (por su baja resistencia) con el fin de mantener estable el regulador interno que trae integrado el microcontrolador. El reset del microcontrolador, al igual que el chip LoRa cuentan con una señal de reset por nivel bajo por lo que se coloca una resistencia como pull up a la tensión de alimentación en dicho pin.

5.3.3. Diseño del PCB

Para diseñar el PCB se utiliza el programa “Eagle” como herramienta CAD. Se diseñaron varias versiones para evaluar el hardware y luego para corregir errores que se cometieron en las primeras versiones.



(a) Chip LoRa Rn2903 (b) Microcontrolador dsPic

Figura 5.1: Hardware principal del Módulo LoRa

5.3.3.1. Placa ReSeDEM v1.0

En una primera instancia se realiza un prototipo con un CNC con el fin de evaluar el hardware. La idea principal era poder enviar comandos desde un puerto serial a través del microcontrolador al chip LoRa. Es con este fin que se agrega un MAX232. Una vez comprendido el funcionamiento del chip y determinados los comandos necesarios para enviar y recibir datos, se implementa en el microcontrolador una primer versión de firmware que inicializa el chip y es capaz de enviar y recibir datos. Con dos prototipos armados, se prueba la comunicación punto a punto entre ellos. Esta primera versión fue exitosa ya que se comprueba el largo alcance de la red. Ver sección 10.1.1. Con este prototipo validado se procede a realizar una nueva versión del PCB.

En la figura 5.2 se muestra el primer prototipo realizado del módulo de comunicación.



Figura 5.2: Módulos LoRa v1.0

5.3. Hardware principal del módulo

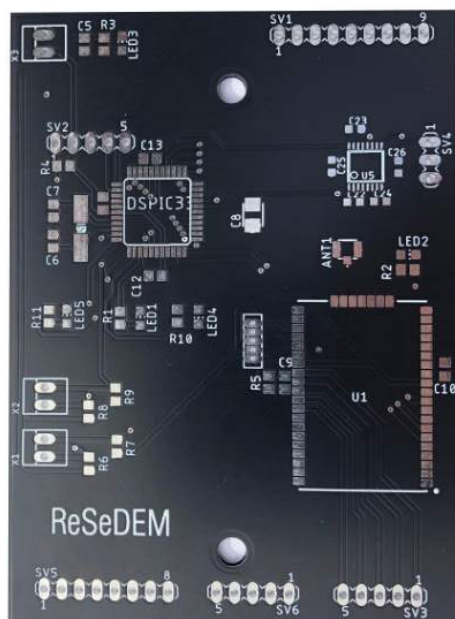


Figura 5.3: Módulos LoRa v2.0

5.3.3.2. Placa ReSeDEM v2.0

Con la base del primer prototipo se diseña un módulo incorporando el hardware para el I2C y se agrega Leds para debugging. Este prototipo se diseña y luego se manda fabricar al exterior.

Se realiza una importación de componentes y una vez que se recibieron los PCB y los componentes, se montaron las placas.

Debido a que se encuentran fallas en el diseño, como no tener en cuenta las diferencias de tensiones que hay en el bus I2C entre los dsPic del Módulo LoRa (3.3V) y los Arduinos de los Módulos Sensores (5V) y en el UART del dsPic y el Arduino del Módulo Principal. Se procede entonces a realizar una versión final.

De todas formas se agregaron shift level converters externos para poder avanzar en la implementación de la Red mientras se diseña y espera por el nuevo PCB con las correcciones realizadas.

En la figura 5.3 se muestra la segunda versión del PCB diseñado.

5.3.3.3. Placa ReSeDEM v3.0 - Versión final

Además de corregir los errores encontrados en la versión 2.0 del PCB, incorporando a la misma los shift level converters, se decide por razones de optimización de espacio incorporar el Módulo GPRS al PCB del Módulo LoRa. Para ello se

Capítulo 5. Módulo LoRa

agregan sockets y se realizan las pistas de conexión para que se pueda conectar tanto el Shield GPRS y el Arduino Nano del Módulo GPRS. De esta forma los Módulos se vayan a utilizar como Nodo Principal, con comunicación GPRS, basta con agregarle estos 2 componentes.

En la figura 5.4 se muestra versión final del PCB módulo de comunicación LoRa diseñado.

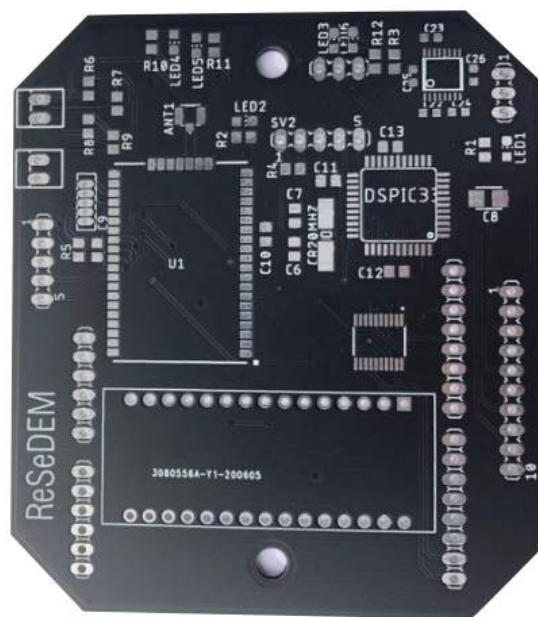


Figura 5.4: Módulos LoRa v3.0

En el Apéndice B se muestra el CAD del PCB final, junto con la lista de componentes y el esquemático.

5.4. Firmware de los Módulos

Los Módulos LoRa son constructivamente iguales, y también lo es el firmware. Dependiendo en que Nodo se conecte (Principal o Secundario) es la configuración que le vamos a grabar. A través del puerto serie montado en la placa es posible configurar los Módulos LoRa de cada Nodo. Es el Nodo Principal quien realiza el manejo de toda la Red LoRa, mientras que los Módulos LoRa de todos los Nodos Secundarios solamente responden a los pedidos de acciones del Nodo Principal.

5.4.1. Funcionamiento Nodo Principal

El Módulo LoRa del Nodo Principal es quien administra la Red de Sensores. Recibe del Módulo GPRS direcciones e instrucciones que debe realizar sobre Red.

5.4. Firmware de los Módulos

Estas direcciones identifican a los sensores que se encuentran conectados en los Nodos. Ningún Nodo tiene conocimiento de cuáles direcciones se encuentran en su bus I2C, por lo tanto lo primero que realiza es una consulta I2C con las direcciones solicitadas y las acciones que deben realizar. Las direcciones que respondan localmente en el bus I2C, las envía por la red LoRa y espera por la repuesta de los Nodos Secundarios. Luego de tener todas las respuestas de los sensores solicitados, las empaqueta en un mensaje y se las envía al Módulo GPRS para que éste las envíe al Servidor.

5.4.2. Funcionamiento Nodo N

El funcionamiento del Módulo LoRa en estos Nodos es más simple. Este nodo recibe las direcciones provenientes de la Red LoRa. Al igual que el Nodo principal, no tiene conocimiento de cuáles sensores tiene en su bus I2C, por lo tanto, cuando llegan por la Red LoRa ciertas direcciones, realiza las consultas en su bus y luego de tener respuesta las envía por radiofrecuencia al Nodo Principal. De no recibir respuesta en cierto tiempo, implica que no pertenece a su Nodo, por lo que no responde y se queda esperando la próxima consulta que debe realizar.

5.4.3. Maquinas de estado

Al igual que en los otros módulos se utilizaron máquinas de estado. Las siguientes máquinas de estado se diseñaron para controlar de manera eficiente y ordenada la comunicación entre todas las partes que involucran al sistema:

- *RN2903*
 - *Communication*
- *RS232*
- *I2C*
- *Server*

RN2903

Esta máquina es quien se encarga de configurar los parámetros de la comunicación LoRa. Para esto envía los comandos almacenados en el dsPic al chip LoRa. Esta inicialización ocurre al inicio, cada vez que el Módulo sale de ahorro de energía y en caso que falle la comunicación con el chip LoRa. Una vez inicializado el chip LoRa, ejecuta de forma anidada la máquina de estados **Communication**, que se encarga del manejo de la Red.

Communication

Capítulo 5. Módulo LoRa

Una vez que el chip LoRa está inicializado, la máquina de estados en cuestión se ejecuta.

En el caso de un Nodo Principal, en el estado inicial se encuentra esperando que el módulo GPRS envíe la solicitud de acciones que le llegan del servidor. Una vez recibida la solicitud, le ordena a **I2C** que escanee el bus en busca de la primera dirección I2C solicitada desde el servidor. Si obtiene una respuesta continua con la siguiente. En caso de no obtener respuesta de la dirección I2C en cuestión, envía la solicitud a través de la Red LoRa y se queda esperando la recepción de un paquete por un tiempo definido. Si se recibe correctamente un paquete de datos desde la Red LoRa, se continúa con la siguiente dirección, si no se asigna un mensaje de error y también se continúa. Una vez finalizado el proceso de solicitud de acciones, empaqueta los datos de todas las direcciones y le da la orden a **Server** de enviar los datos pendientes hacia el servidor y se queda esperando por el tiempo que debe ir a ahorro de energía.

En un Nodo Secundario, en el estado inicial se espera por una solicitud de acciones a través de la Red LoRa. Una vez recibida, se procesa a través de **Server** simulando que la solicitud proviene del servidor. Con eso se desencadena el mismo proceso que en un Nodo Principal, la diferencia es que en caso de no obtener localmente respuesta de la dirección I2C solicitada por el principal, desecha la solicitud. Si la Red LoRa se encuentra inactiva por un tiempo dado, se inicializa el proceso de ir a ahorro de energía por un tiempo definido.

RS232

El Módulo LoRa cuenta con un puerto Serial que se utiliza para configurar los distintos parámetros de chip LoRa entre otras configuraciones del Módulo. La máquina de estados en cuestión permite en cualquier momento cambiar los parámetros de la comunicación de radio como lo son el SF, CR y BW. Una vez recibido un comando a través del puerto serie, pausan las demás tareas y se abre un modo transparente donde todas las respuestas que da el chip LoRa se retransmiten hacia el puerto Serial permitiéndole al usuario ver en qué estado se encuentra el chip. Para más información de estos parámetros ver Anexo A.

I2C

La máquina I2C es la que se encarga de realizar la comunicación entre el ds-Pic y los módulos sensores que estén conectados en el bus I2C. Aquí es donde se le solicita a los sensores que realicen las acciones que son solicitadas por el Servidor.

Server

En el Nodo Principal esta máquina, es la que se encarga de realizar la comunicación con el Módulo GPRS. Se comunica con el GPRS recibiendo las acciones que

5.4. Firmware de los Módulos

debe realizar la Red de sensores y luego le devuelve el resultado de dichas acciones.

En un Nodo Secundario, espera por la solicitud de acciones a través de la Red LoRa para luego desencadenar el escaneo del bus I2C y por último de enviar los datos recibidos de los sensores a travesa de LoRa, cuando corresponde.

En la figuras 5.5 y 5.6 se representa una secuencia de funcionamiento de un Módulo LoRa perteneciente al Nodo Principal y a un Nodo N. Estas secuencias se encuentran simplificadas para su entendimiento. El sistema incorpora además, reintentos en las transmisiones cuando no se tiene respuesta y reconocimientos (ack) de llegada de paquetes. Luego de cada ciclo de trasmisión de datos y cada cierto tiempo, tanto el Módulo LoRa del Nodo Principal como el del Nodo N se reinician para de esta forma evitar que el sistema entre en bucles y se caiga la Red.

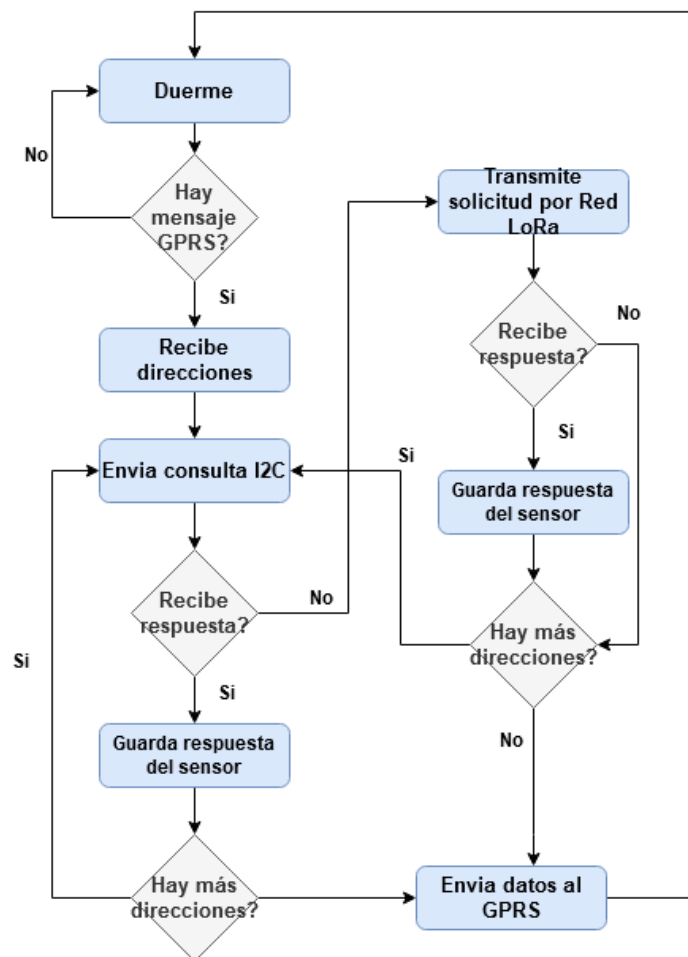


Figura 5.5: Secuencia funcionamiento del Módulo LoRa del Nodo Principal

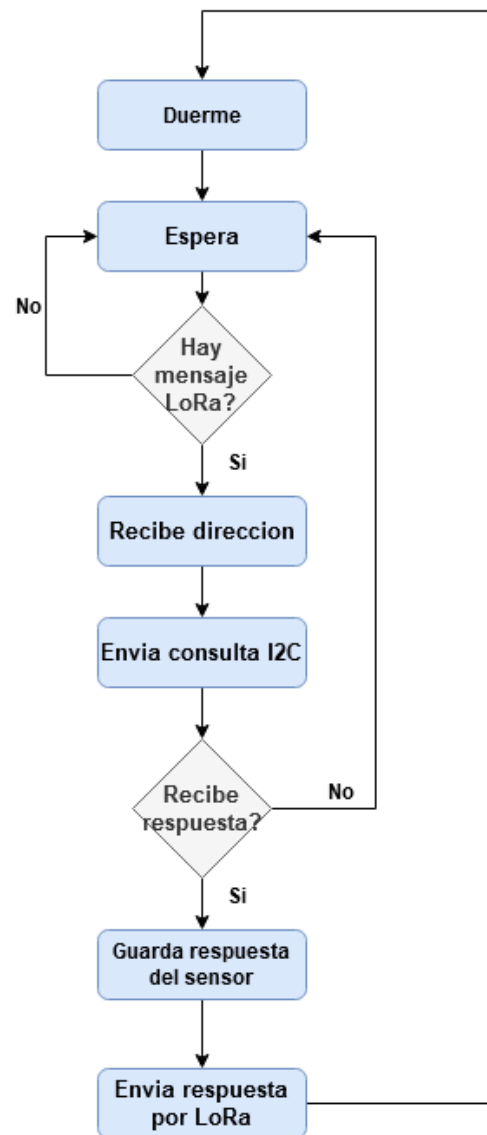


Figura 5.6: Secuencia funcionamiento del Módulo LoRa del Nodo N

Capítulo 6

Módulo Sensor

6.1. Características que debe cumplir el módulo

Las principales características que debe cumplir el módulo son:

- Tener capacidad para conectar sensores
- Tener comunicación I2C con el Módulo LoRa
- Tener un consumo mínimo de energía

6.2. Hardware del Módulo

El Módulo Sensor es el encargado de realizar las acciones que el Servidor solicite. Está compuesto por un microcontrolador y un sensor y la comunicación con el resto del sistema es mediante I2C.

6.2.1. Microcontrolador

Al igual que en el Módulo GPRS, se eligió como microcontrolador, un Arduino Nano “v3.0 bordo ATmega328”. Ver 4.3.2. Este microcontrolador es el encargado de procesar la orden que le llega desde la Red LoRa, realizarla y devolver una respuesta nuevamente cuando se le ordene.

6.2.2. Sensores

La Red de sensores está pensada para cualquier tipo de sensores. A modo de ejemplo, en el proyecto se utilizaron sensores de medición de humedad del suelo, temperatura ambiente y humedad ambiente.

Capítulo 6. Módulo Sensor

6.2.2.1. Sensor de Temperatura y Humedad Ambiente - DT11

El DT11 [15] es un componente capaz de medir 2 magnitudes físicas. Posee un termistor para medir la temperatura ambiente ($^{\circ}C$) y un sensor resistivo para sensar humedad ambiente (%). Cuenta con un microcontrolador de 8 bit y un ADC para medir las condiciones climáticas y transmitirlos digitalmente mediante un solo hilo. Pose 3 conexiones, VCC, GND y DATA (pin de transmisión de datos en half duplex). En la figura 6.1 se puede ver el sensor DT11.

Los datos entre el DT11 y Arduino, en este caso, se transmiten mediante un string de 40 bits en serie de los cuales se desglosa de la siguiente manera:

- 8b - parte entera de la humedad
- 8b - parte decimal de la humedad
- 8b - parte entera temperatura ambiente
- 8b - parte decimal temperatura ambiente
- 8b - checksum

A nivel de programación la lectura de los datos y comunicación con el DT11 ya está resuelta. Se utiliza la librería *dht.h* quien devuelve las lecturas por separado, tanto de humedad como de temperatura.

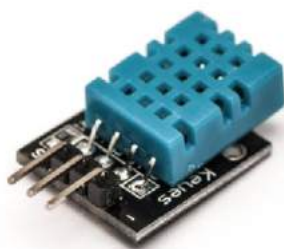


Figura 6.1: Sensor DT11

A continuación se describen las principales características del sensor [15]:

- Tensión de alimentación: $[3,3, 5]VDC$
- Consumo: $1mA$
- Rango de Humedad $[20, 80] \% \pm 5 \%$
- Rango de temperatura $[0, 50]^{\circ}C \pm 2^{\circ}C$

6.2.2.2. Sensor Humedad Suelo - FC-28

El sensor de suelo elegido es un FC-28. Este es un sensor que mide la humedad del suelo por conductividad. El valor de la medida dependerá del tipo de suelo y la presencia de elementos químicos, como fertilizantes.

6.3. Firmware del módulo

Este sensor viene con un circuito de acondicionamiento con un comparador que permite obtener tanto una medida analógica como una medida digital. La salida digital se puede calibrar desde el sensor para que cuando supere cierto umbral se active. Mientras que la salida analógica presenta una tensión de salida que varía entre los 0 y 5VDC. Esta tensión corresponde con la humedad del suelo siendo los 5VDC una humedad del 0% y 0V una humedad del 100%. Posee 4 pines de conexión (VCC, GND, DO (salida digital), AO (salida analógica)) [16].

Para el proyecto se utilizó la salida analógica ya que lo que interesaba era saber el valor de humedad y no obtener un umbral disparador.

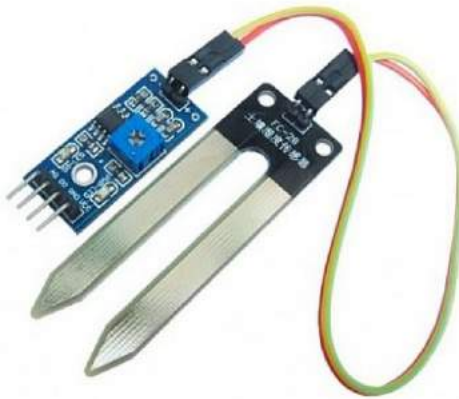


Figura 6.2: Sensor de humedad de suelo

6.3. Firmware del módulo

6.3.1. Funcionamiento

El módulo es identificado con una Dirección I2C única en la Red. Desde el Servidor se le puede, en principio, solicitar 3 acciones al Sensor: configurarle una dirección específica y agregarlo a la Red (AGREGAR), borrar la dirección del mismo y poner una genérica (ELIMINAR) o solicitarle Medida (MEDIR).

El Módulo se encuentra esperando que le llegue una solicitud desde el Bus I2C. Cuando identifica que es una solicitud para él, realiza la acción solicitada y responde al Módulo LoRa del Nodo al cual pertenece para que éste lo reenvíe por la red.

Tanto en el sensor DT11 como en el de humedad de suelo se toman varias muestras consecutivas y se realiza un promedio para eliminar el ruido blanco que

Capítulo 6. Módulo Sensor

pueda existir a la hora de realizar el muestreo.

6.3.2. Máquinas de estado

El firmware de los Módulos Sensor consta de una máquina de estados que controla la comunicación I2C y las de acciones que debe realizar el módulo. Se tienen estados de espera por solicitud, identificación de acción, realización de acción y para enviar la medida/acción realizada. En la figura 6.3 se describe tanto la secuencia de funcionamiento del módulo como los estados de la máquina.

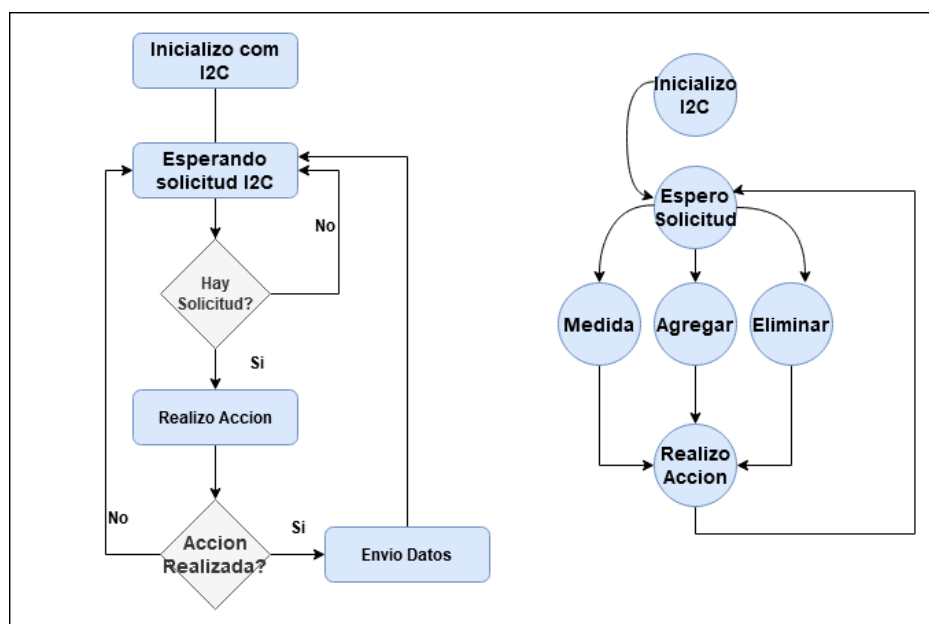


Figura 6.3: Secuencia funcionamiento Módulo Sensor y estados

Capítulo 7

Arquitectura de la Red LoRa

7.1. Topologías de red posibles para implementar

Existen diversas topologías de red, de las cuales el estudio se centró en las 3 siguientes por considerarlas de las mejores opciones:

- Estrella
- Multisalto
- Anillo

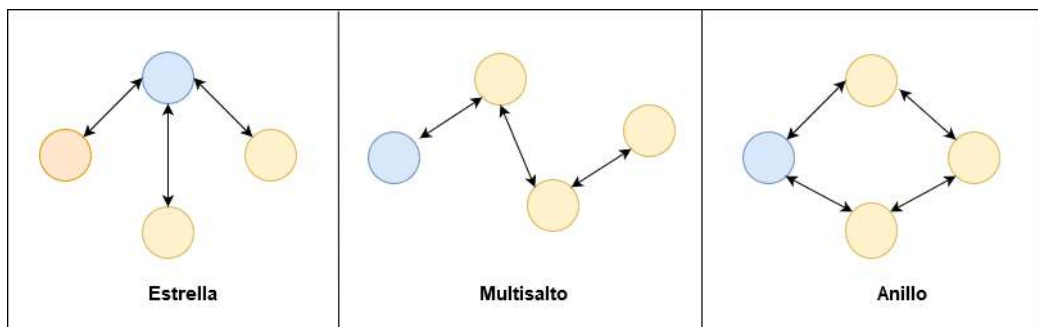


Figura 7.1: Topologías de Red

Estrella

Es una de las topologías más conocidas. Se tiene un nodo central, por el cual pasa toda la información.

Características:

- Comunicación punto a punto entre nodo central y cada nodo.
- Facilidad para agregar nuevos nodos

Capítulo 7. Arquitectura de la Red LoRa

- El alcance máximo está dado por cobertura de la tecnología de enlace.
- Ante una falla de un nodo secundario, la red continúa funcionando sin problemas, salvo por ese enlace.

Multisalto

Este tipo de red es muy utilizado cuando se quiere cubrir grandes superficies. .
Características:

- No requiere una comunicación punto a punto con el nodo principal.
- Facilidad para agregar nuevos nodos.
- El alcance máximo está dado por cobertura de la tecnología de enlace y la cantidad de nodos disponibles (N° Nodos \times Alcance tecnología de enlace)
- Ante una falla de un nodo, se pierde comunicación hacia adelante por lo tanto la red puede quedar casi inhabilitada por completo en caso de fallar, por ejemplo, el segundo nodo.

Anillo

La topología de red en anillo presenta características similares a la red multisalto pero la diferencia es que a cada nodo de la red se le puede llegar por 2 caminos diferentes.

Características:

- Robusta ante la caída de un nodo, sigue disponible el otro camino (redundante).
- Costosa de implementar, requiere agregar más nodos, para cubrir la misma distancia que la red multisalto. Sin embargo tiene una cobertura mucho mayor a la red en estrella.
- Agregar más nodos tiene cierta complejidad, ya que hay que implementar tablas de ruteo.

7.2. Elección de la topología de red para el sistema

Luego de analizar las 3 tecnologías de red se eligió la Red en estrella.

Uno de los requisitos del sistema es que los nodos estén a 1 Km de distancia uno de otro, por lo que implementando este tipo de red se lograría al menos 1 Km de radio desde del nodo principal. Como se analizará más adelante, con las pruebas de medición de alcance, se lograron distancias mucho mayores, por lo que se consideró que el área de cobertura es más que suficiente.

7.3. Arquitectura de la Red LoRa

Otra de las ventajas de esta red, considerando que los nodos estarán en el campo, lejos de la ciudad, es la robustez. De fallar un nodo, se pueden seguir utilizando todos los demás sin quedar inhabilitado el sistema hasta ir a solucionar el problema.

La facilidad de agregar un nuevo nodo es otro factor que se consideró importante, ya que para agregar un nuevo nodo, no hay que reprogramar ningún nodo anterior que esté instalado en campo.

Por último, de tener una tecnología de red multisalto, implicaría que ciertos nodos, además del nodo principal, tengan consumos diferentes que pueden llegar a ser muy altos. Esto se debe que los nodos más cercanos al nodo Principal, tendrían un tráfico mayor de datos, respecto a los nodos más lejanos. Esto requeriría pensar en una capacidad de batería para cada nodo, o sobredimensionar todo el sistema de almacenamiento para cumplir con el requisito de 1 semana de autonomía como se especificó.

7.3. Arquitectura de la Red LoRa

La Red está pensada para ser transparente a los efectos del sistema de nodos. Solo reciben información y envían cuando les corresponde. Todo lo que se envía a través de esta red es en forma de Broadcast. Es decir todos los receptores que se encuentren en zona de cobertura recibirán el mensaje.

Cuando el Nodo Principal comprueba que una dirección solicitada por el Servidor no pertenece a su nodo, la transmite en Broadcast hacia todos los Nodos, a la espera de que el Nodo N que la tenga realice la acción y responda.

Los Nodos N, tampoco saben que sensores tienen conectados por el bus I2C. Por lo tanto realizan una consulta por el bus. Si no reciben respuesta desechan el pedido que les llega por la Red. En caso de recibir respuesta esperan por la medida y lo envían por Broadcast nuevamente. Todos los Nodos a su alcance recibirán la respuesta pero solo el Nodo Principal será el que reconozca dicho mensaje.

En caso de tener varias direcciones que no pertenezcan al Nodo Principal, éste va realizando las consultas de a una por la Red y va almacenando las respuestas para enviar al Servidor un paquete único.

En la figura 7.2 se muestra como un mensaje es enviado por el Nodo Principal y solo recibe respuesta de uno. Esa respuesta contiene la acción realizada y el dato.

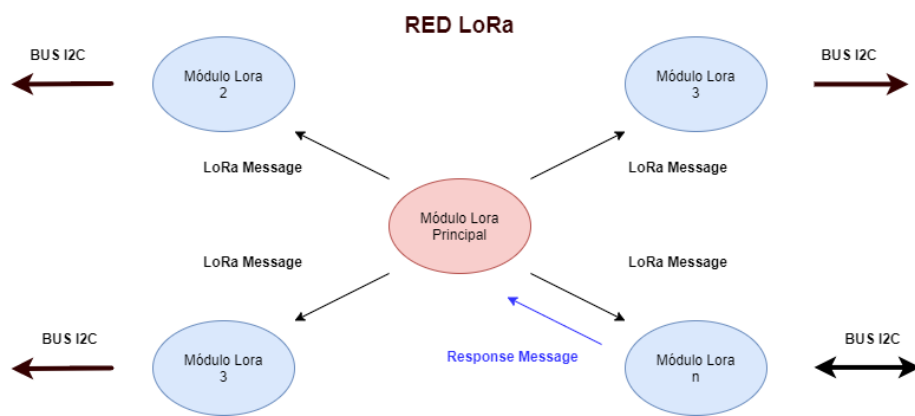


Figura 7.2: Red LoRa

Capítulo 8

Interfaces y protocolos de comunicación

En este capítulo se describen las interfaces utilizadas y los protocolos de mensajes implementados para la comunicación entre los distintos puntos del sistema.

8.1. Interfaces utilizadas

8.1.1. UART

Para la comunicación entre Arduino-GPRS, Arduino-Pic y Pic-Chip LoRA se utiliza el protocolo UART.

La UART se encuentra integrada en los microcontroladores y se utiliza para la comunicación en serie entre dispositivos. La comunicación es asíncrona y por lo tanto tanto el receptor como el transmisor deben establecer la velocidad de transmisión (boudrate) de antemano. Los boudrates más utilizados son 9600 bps y 19200 bps.

En la figura 8.1 se muestra la trama de una comunicación UART. LA misma incluye 1 bit de arranque (S), 5-8 bits de datos, un bit paridad (P) y un bit de parada (Sp).

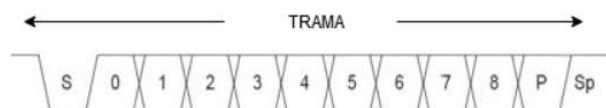


Figura 8.1: Trama mensaje UART

Su uso es muy extendido a nivel mundial por su fácil implementación pues no requiere un reloj entre emisor y receptor y se puede implementar con 2 cables como se muestra en la figura 8.2 . Como desventaja este sistema solo permite la comunicación entre 2 dispositivos por una misma UART, es decir, no permite la implementación de un sistema multipunto.

Capítulo 8. Interfaces y protocolos de comunicación

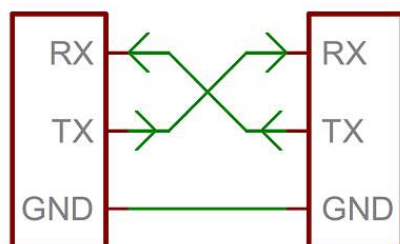


Figura 8.2: Conexión para una comunicación UART

8.1.2. RS232

Esta interfaz es utilizada para poder configurar el Chip LoRa desde una computadora.

La interfaz RS-232 es una norma recomendada que establece los criterios de comunicación serial entre una computadora y un dispositivo final, como por ejemplo un Modem. Esta interfaz se implementa sobre la UART del dispositivo final. Está pensada para transmitir datos a no más de 15 metros, y a una baja velocidad de transmisión (20kbps). Si bien este estándar tiene diversos tipos de conectores, en la Tabla 8.1 se indica el pinout de un conector RS232 en su versión más popular “DB-9”.

Pinout	Señal
1	CD - Carrier Detect
2	RD - Received Data
3	TD - Transmitted Data
4	DTR - Data Terminal Ready
5	CG - Common Ground
6	DSR - Data Set Ready
7	RTS - Request To Send
8	CTS - Clear To Send
9	RI - Ring Indicator

Tabla 8.1: Pinout - Conector RS232

Debido a las diferencias de tensión entre la computadora y el dispositivo final, generalmente se utiliza el “MAX232”, un circuito integrado de Maxim para convertir las señales del RS-232 en señales de tensiones a las que trabaja el microcontrolador, señales TTL.

8.1. Interfaces utilizadas

8.1.3. I2C

La comunicación entre los Módulos sensores y el Módulo LoRa se realiza mediante el protocolo I2C. Este protocolo fue diseñado por Philips en el año 1982 para la comunicación entre periféricos y el microcontrolador.

Este protocolo permite la comunicación maestro-esclavo entre varios dispositivos o configuraciones de varios maestros y esclavos en la misma línea. AL igual que UART, sea realiza con 2 cables (SDA y SCL). La señal SCL es el clock que es generada por el maestro, ya que es un protocolo síncrono, y SDA, la señal de datos. El clock es quien genera la velocidad de transferencia de datos siendo la usual de 100 kHz aunque también se puede llegar a un modo de transición rápida donde el clock funcione a 5 MHz.

Se le llama bus I2C al par de hilos SDA y SCL, en conjunto con GND en el cual el Maestro y los esclavos se conectan al mismo. Cada dispositivo conectado al bus debe tener una dirección I2C a la cual responder cuando el maestro la solicita. En la figura 8.3 se muestra el bus de comunicación con diversos dispositivos conectados. El protocolo prevé el uso de resistencias de pull-up para las líneas.

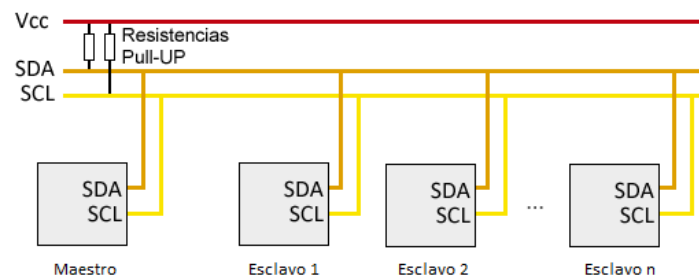


Figura 8.3: Bus I2C [18]

Por lo general, la dirección I2C está especificada por hardware para el caso de sensores que se utilicen este protocolo. La dirección está compuesta por 8 bits, 7 corresponden a la dirección propiamente dicha y el octavo bit al modo que se quiere comunicar con el dispositivo, lectura o escritura. Por lo tanto se pueden tener hasta 128 dispositivos en un mismo bus I2C. El protocolo también presenta adaptaciones para direcciones de 10 bits con lo que se pueden obtener hasta 1008 dispositivos en un mismo bus ya que se tienen ciertas direcciones reservadas.

En la figura 8.4 se presenta un ejemplo donde el maestro solicita escritura a una dirección específica, enviándola por el bus, por lo que le llega a todos los dispositivos. A continuación, envía el dato para que el sensor cuya dirección coincide con la enviada previamente la guarde.

En ReSeDEM las direcciones I2C de los módulos sensores que se conectan al bus de datos corresponden con direcciones de 8 bits. Estas direcciones se deben

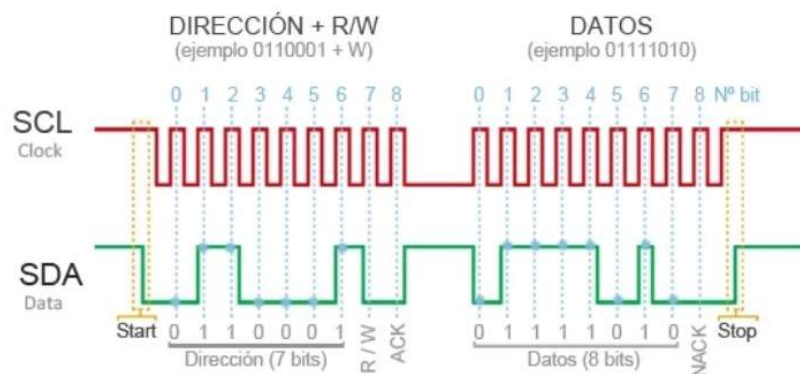


Figura 8.4: Ejemplo de una comunicación I2C y las señales generadas en las líneas SDA y SCL [19]

establecer desde el Servidor ya que al conectar un Módulo Sensor al bus, por primera vez, este inicia con una dirección hexadecimal genérica (0x30) y cambia su dirección a la dirección que se indica desde el servidor.

El proyecto incluyo el desarrollo de una librería I2C para el dsPic ya que no se contaba con una, mientras que para los Arduino, se utilizó una librería muy extendida “Wire.h” [17].

8.2. Intercambio de mensajes

Por sobre las interfaces antes mencionadas, se han establecidos ciertos intercambios de mensajes propios para que todas las partes puedan interpretar las acciones que deben realizar y así lograr la comunicación de extremo a extremo.

8.2.1. Servidor- Nodo principal

A continuación se describe la secuencia que se realiza entre el Servidor y el Nodo Principal a través del Módulo GPRS .

1. El GPRS inicia la comunicación con al servidor realizando un primer GET1 a la URL “/consulta.php” y este le devuelve el mensaje *ServerDOWN1* como se muestra en la tabla 8.2. Este mensaje indica la acción (Acción), cuantos sensores tienen que realizar dicha acción (Nº direcciones) y las direcciones correspondientes (Direcciones). Un mensaje del servidor puede contener varias acciones, pero para cada sensor, solo debe solicitar una acción. No necesariamente tienen que estar las 3 acciones ya que en determinado momento puede que solo solicite medidas a sensores, o solo solicite agregar determinado sensor, por ejemplo.

8.2. Intercambio de mensajes

Acción	Nº direcciones	Direcciones	...	Acción	Nº direcciones	Direcciones
--------	----------------	-------------	-----	--------	----------------	-------------

Tabla 8.2: Mensaje del servidor solicitando acciones - ServerDOWN1

Al momento se pueden realizar 3 acciones sobre la red:

- **01 - Solicitar Medida** - Solicita media a un determinado sensor.
- **02 - Agregar Sensor** - Agrega un sensor a la red. Establece una dirección específica a dicho sensor. Esta dirección es ingresada por el usuario desde el Servidor. Una vez establecida la dirección el Servidor puede empezar a solicitarle datos.
- **03 - Eliminar Sensor** - Elimina un sensor de la red. Le establece la dirección reservada (0x30) y borra todos los datos del sensor en el servidor. Esta dirección reservada permite al sensor ser inicializado en otro momento desde el Servidor.

Sin embargo se puede ampliar la cantidad de acciones ya que están identificadas con 2 caracteres ASCII. Por ejemplo se podría configurar algún sensor, cambiar parámetro de la red Lora como puede ser el Code Rate, Spread Factor, etc y muchas otras funcionalidades.

2. Luego el GPRS realiza un segundo GET2 a la URL *“/proc2.php”*, enviando el mensaje *ServerUP* con información de la Red LoRa al servidor y finalmente, este le devuelve el mensaje *ServerDOWN2*.

El mensaje ServerUP se muestra en la tabla 8.3. Los mensajes incluyen la dirección del sensor (Direccion), el largo de la medida (Largo) y el valor de la medida (Valor). En caso de que ocurra un error con determinado sensor, o no se haya podido establecer una comunicación el parámetro “Largo”- contendrá los caracteres “SS” significando un mensaje de estado. El código de estado es de tres caracteres ASCII. Por último se concatena un caracter de fin de mensaje con la letra *f*.

Direccion	Largo	Valor	...	Direccion	Largo	Valor	f
-----------	-------	-------	-----	-----------	-------	-------	---

Tabla 8.3: Mensaje hacia el servidor - ServerUP

3. Como respuesta al *Get2* que le envía las medidas/errores al Servidor este responde con los minutos restantes para la próxima medida que se va a solicitar. En la tabla 8.4 se indica el mensaje.

En la figura 8.5 se representa la secuencia de intercambio de mensajes entre el Servidor y el Módulo GPRS.

Tabla 8.4: Respuesta del servidor informando minutos de la próxima solicitud - ServerDOWN2

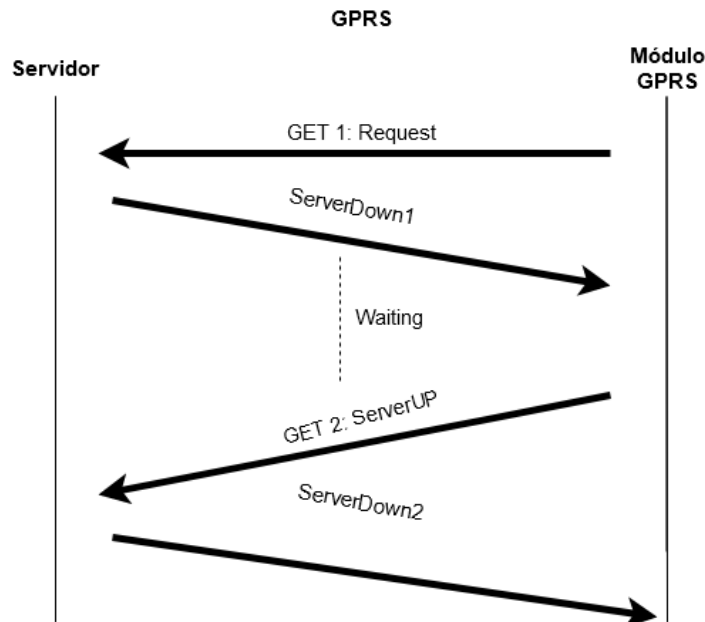


Figura 8.5: Mensajes: Servidor - GPRS

8.2.2. Módulo GPRS - Módulo LoRa

Luego de recibido el mensaje del servidor, el Módulo GPRS envía el mensaje al módulo LoRa para que éste lo distribuya por la red. Para esta comunicación se utiliza la interfaz UART. Por lo tanto, por sobre el mensaje se le agregan caracteres de inicio (S) y fin de mensaje (E). Además se agrega un char de checksum (CS) para comprobar que los datos lleguen correctamente.

Tanto a la ida como a la vuelta, el receptor del mensaje revisa que la suma corresponda con la del checksum y devuelve un ack informando que le llegaron correctamente.

En la figura 8.6 se representa la secuencia de intercambio de mensajes entre el Módulo GPRS y el Módulo LoRa.

8.2.3. Módulo Lora - Bus I2C

Cuando un Modulo Lora recibe un mensaje ya sea por aire (Nodos N) o desde el Módulo GPRS para el caso del Nodo Principal, indicando acciones y direcciones, lo transmite por su bus I2C y espera respuesta del Sensor, de recibir respuesta

8.2. Intercambio de mensajes

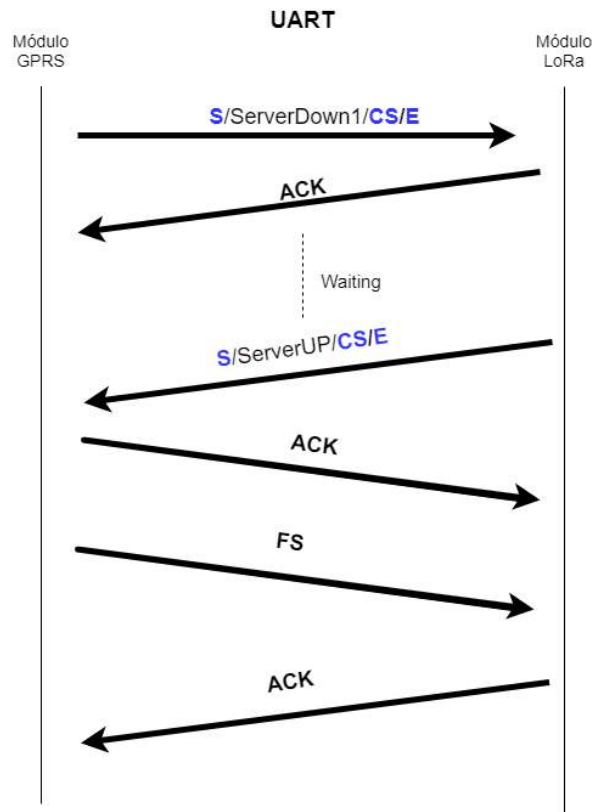


Figura 8.6: Mensajes: GPRS-LoRa

procede a realizar la acción con dicho Sensor.

Cuando un sensor responde por el bus I2C, se realiza el siguiente intercambio de mensajes:

1. Se indica la acción que se va a realizar y espera el ACK del sensor.
2. Se solicita el largo de la medida y espera respuesta.
3. Se le solicita el dato.

En la figura 8.7 se muestra el intercambio de mensajes que se realiza por sobre el protocolo I2C.

8.2.4. Estado y Manejo de errores del sistema

Desde el Servidor se tiene un monitor donde se informa el estado de situación del sistema. Este monitor sirve para visualizar en tiempo real el tiempo que falta para la próxima solicitud de medida, si hay retraso en la comunicación con el Servidor o si hay algún problema.

Capítulo 8. Interfaces y protocolos de comunicación

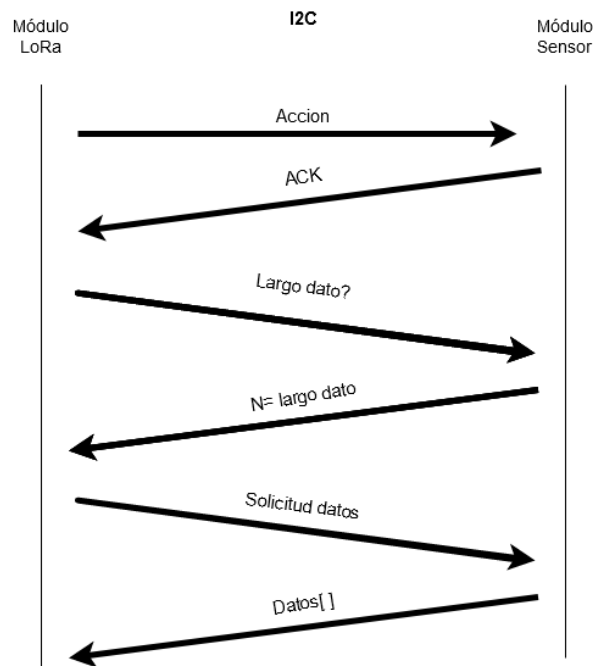


Figura 8.7: Mensajes: Nodo - I2C - Sensor

Estado	Informa el evento	Descripción del evento/error
100	Servidor	Sensor inicializado
200	Servidor	Dato correcto
220	Servidor	Esperando Conexión GPRS - Hay acciones a solicitar
250	Servidor	GET:Consulta
300	Servidor	Se perdió conexión con Módulo GPRS
400	GPRS	No se recibió ack de solicitud de acciones desde LoRa principal
423	GPRS	Timeout en la recepción de datos desde LoRa principal
424	LoRa Ppal	Error en transmisión de acciones hacia la red LoRa
425	LoRa Ppal	Error en recepción de datos desde la red LoRa
426	LoRa Ppal	Timeout en recepción de datos desde la red LoRa
600	Servidor	Sensor eliminado
604	Servidor	Sensor para eliminar

Tabla 8.5: Estado general del sistema

Además se registran todos los eventos/errores que se han detectado ya sea datos cargados correctamente como errores en sensores, en la Red LoRa o pérdida de conexión con el Servidor.

En tabla 8.5 se describen los eventos/errores detectados.

8.2.5. Ejemplo de un intercambio de mensajes extremo a extremo

En la figura 8.8 se ilustra el intercambio extremo a extremo del sistema siguiendo la secuencia ilustrada en la figura 2.3

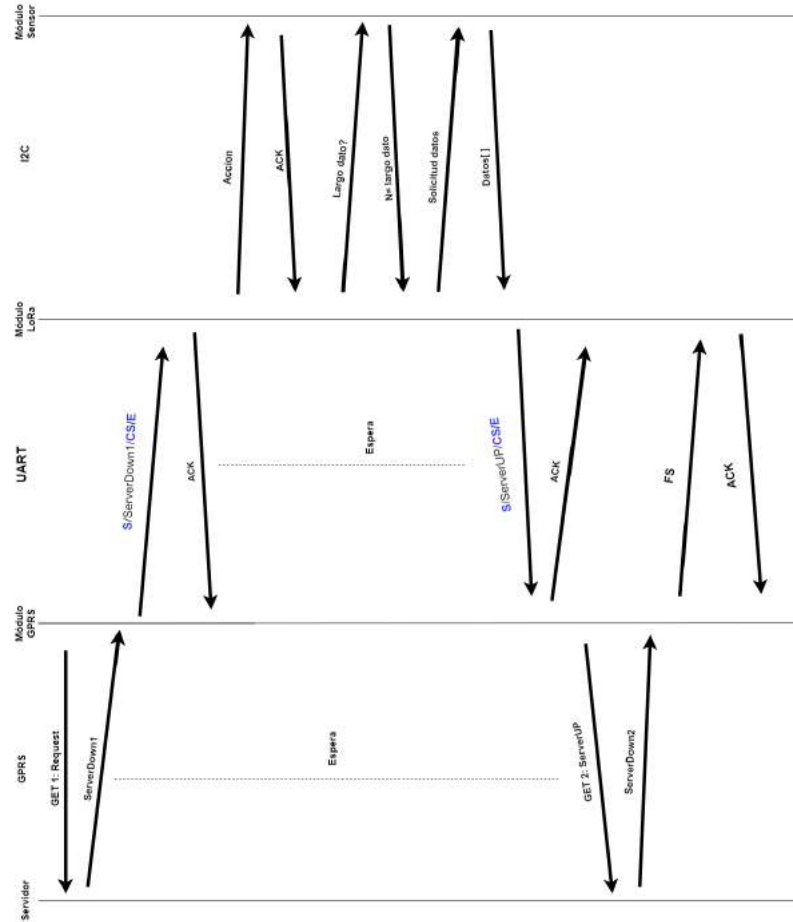


Figura 8.8: Mensajes extremo a extremo - Servidor - GPRS - LORA - I2C

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 9

Diseño final

9.1. Componentes para el ensamblado

Cajas

Debido a que los módulos deben estar a la intemperie, se utilizan cajas estancas IP 65. Con esto se logra que la lluvia y humedad no ingrese al interior y deteriore los componentes electrónicos.

Conectores

Cada módulo tiene 2 conectores GX16-7 para el bus de datos y alimentación. De esta forma se puede agregar a la línea varios Módulos Sensores, simplemente interconectándolos con conectores estos modelos.

Además se utilizaron GX16-5p para la conexión de los sensores de medición a los módulos sensores. Se eligieron estos, ya que como se ha mencionado antes, permiten conectar cualquier dispositivo de medida. En caso de necesitar más se puede cambiar estos conectores por unos de más pines.

En la imagen 9.1 se pueden apreciar los conectores utilizados para los módulos.



Figura 9.1: Conector Gx16-7p (izq), Gx16-5p (der)

9.2. Ensamblado final

9.2.1. Módulo Lora, Sensor, GPRS

En las siguientes figuras se pueden observar los diferentes componentes del sistema de forma individual, así como también un ejemplo de un Nodo conformado por un Módulo LoRa y un Módulo Sensor.

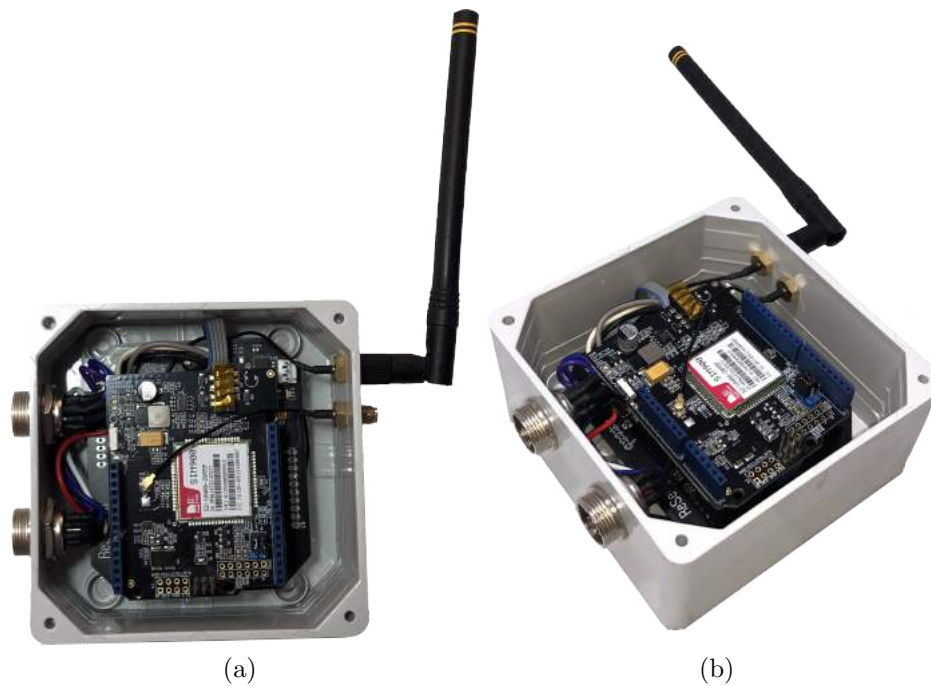


Figura 9.2: Módulo LoRa-GPRS, Nodo Principal

9.2. Ensamblado final

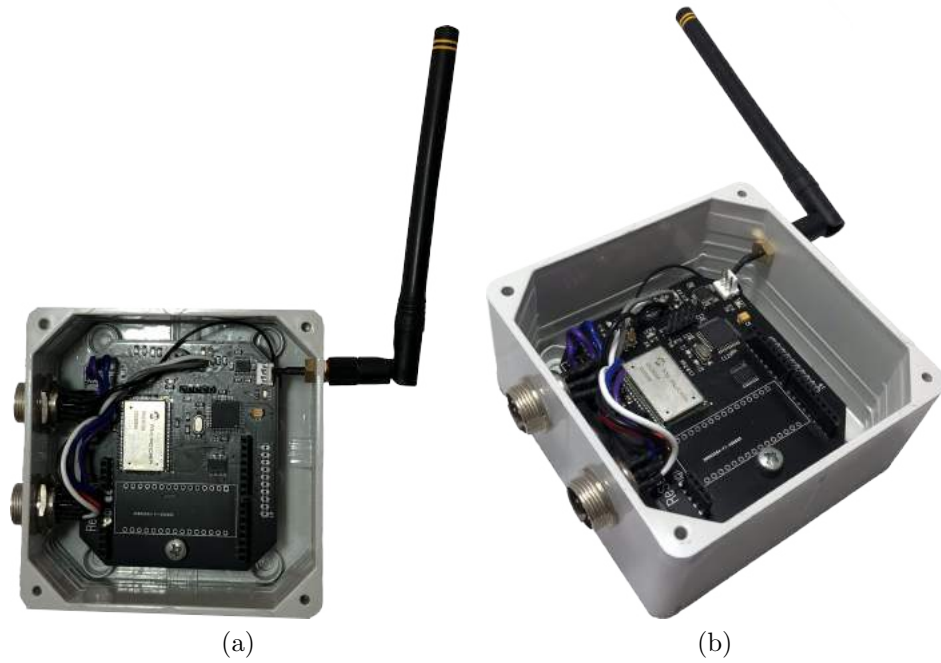


Figura 9.3: Módulo LoRa, Nodos Secundarios

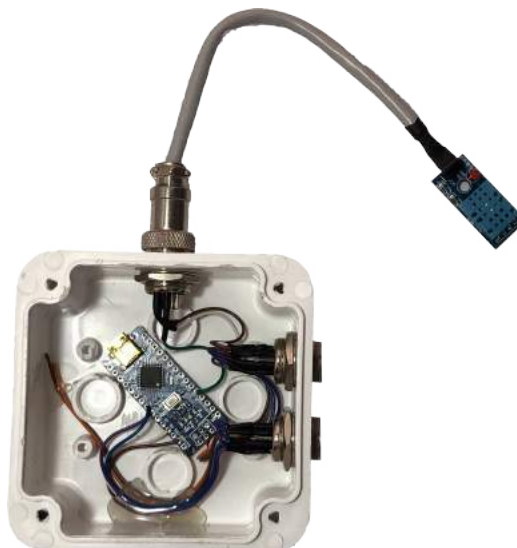


Figura 9.4: Módulo Sensor

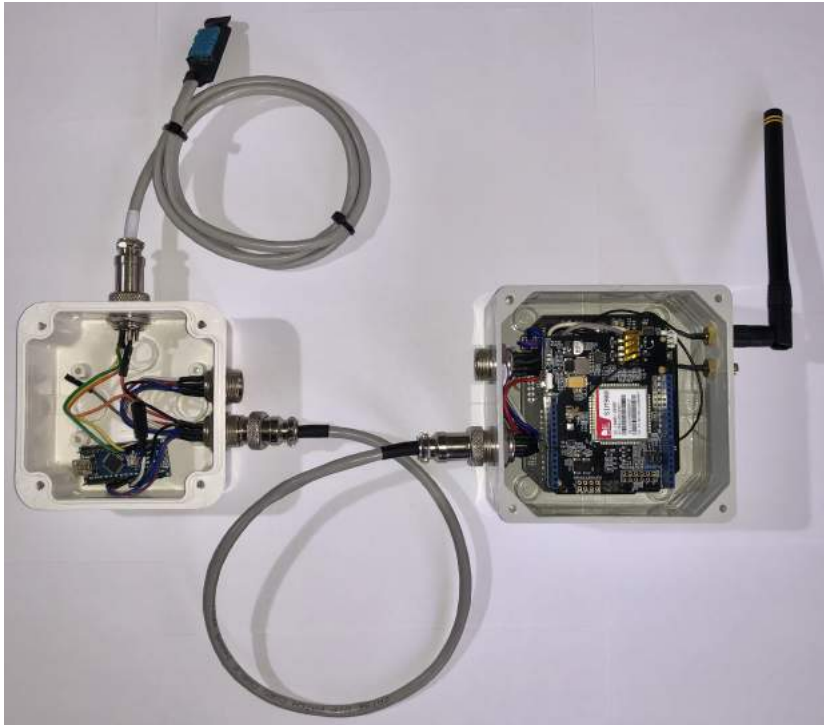


Figura 9.5: Sistema ensamblado: Nodo Ppal y Módulo Sensor

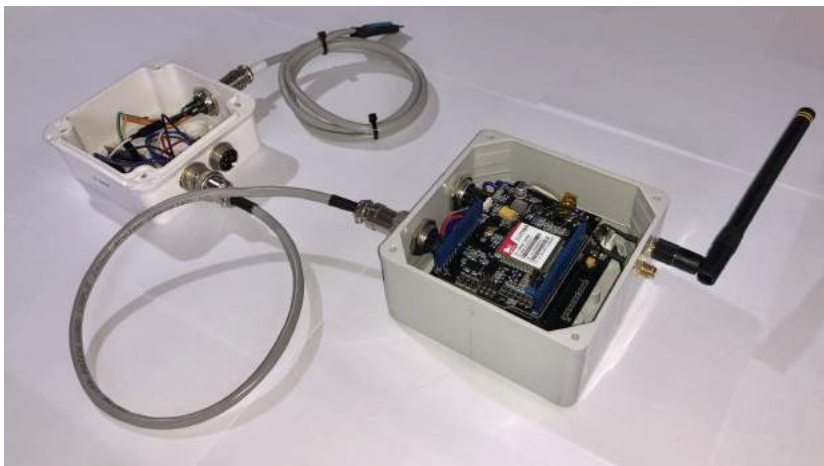


Figura 9.6: Sistema ensamblado: Nodo Ppal y Módulo Sensor

Capítulo 10

Pruebas del Sistema

En este capítulo se presentan las pruebas que se le realizaron al sistema. Las mismas incluyen pruebas de alcance y de consumo, siendo estas los requisitos fundamentales que tenía el proyecto. También se realizaron pruebas de desempeño que permiten caracterizar el sistema bajo ciertas condiciones de prueba y son útiles para las aplicaciones que se quieran implementar con ReSeDEM.

10.1. Pruebas de Alcance

Uno de los requisitos del sistema es que la Red LoRa tenga un alcance mínimo de 1 Km entre nodos. A continuación se presentan los resultados de las pruebas de alcance realizadas.

10.1.1. Primera prueba

Luego de realizar el primer prototipo de Módulo LoRa v1.0, con el fin de tener una aproximación de distancia, se programó una comunicación punto a punto entre 2 de estos módulos y se probó el alcance.

Configuración del módulos

Los módulos fueron programados con la siguiente configuración:

- SF: 7
- CR: 5/7
- BW: 500 kHz
- potencia TX: 18 dBm

Lugar de la prueba:

Rambla de Montevideo.

Capítulo 10. Pruebas del Sistema

10.1.1.1. Resultados

Se alcanzó una distancia de 3.63 Km, con línea vista, es decir, sin ningún obstáculo. Esta distancia no fue la máxima, ya que se podría haber seguido, pero como el prototipo demostró que cumplía con el objetivo mínimo se lo dio por válido. La máxima cobertura se comprobó en campo, como se indica en la sección 10.1.2

En la figura 10.1 se muestra la distancia lograda en la primer prueba.

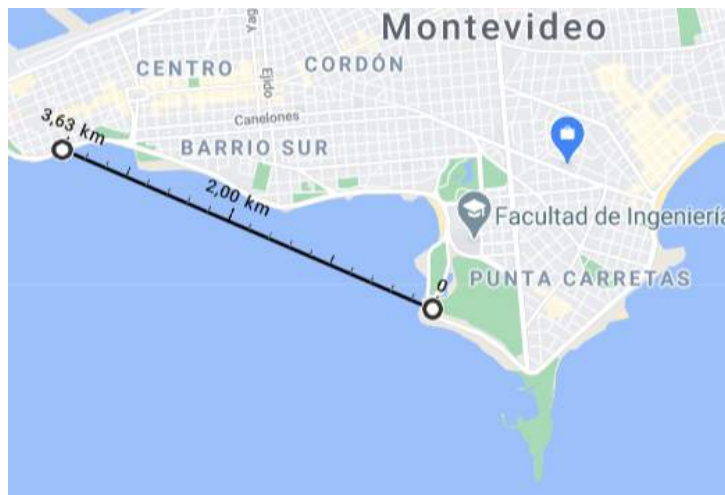


Figura 10.1: Primer prueba alcance

10.1.2. Prueba con el sistema final

Para comprobar el alcance de la red con el sistema final se realizaron 2 pruebas.

Primero se probó el sistema con 2 Nodos Secundarios ubicados en estrella con el Nodo Principal. Uno de los Nodos Secundario se ubicó a 3.97 Km del Nodo Principal, con línea vista, mientras el segundo nodo se encontraba a 758 m con línea vista parcial (NearLOS). Ver figura 10.2

La segunda prueba consistió en alejar lo más posible el Nodo Secundario del Principal, siempre teniendo línea vista, para comprobar el alcance máximo posible.

Se cambiaron los parámetros de la red para obtener una mejor sensibilidad y por lo tanto mayor alcance.

Configuración del módulos

Los módulos fueron programados con las siguiente configuración:

- SF: 12

10.1. Pruebas de Alcance

- CR: 4/7
- BW: 500 kHz
- potencia TX: 18 dBm

Lugar de la prueba:

Rambla de Montevideo, Faro Punta carretas, Escollera del Puerto del Buceo - Playa Malvin.

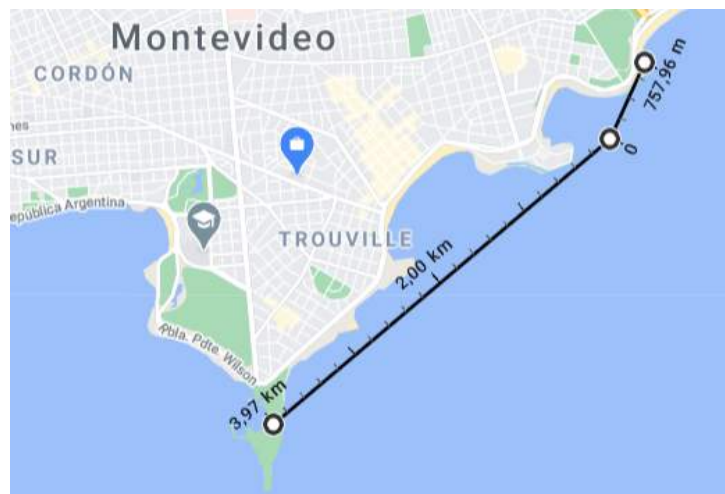


Figura 10.2: Prueba alcance Sistema final



Figura 10.3: Prueba alcance máxima Sistema final

Capítulo 10. Pruebas del Sistema

10.1.2.1. Resultados

La prueba de campo de la red fue exitosa. El sistema funcionó correctamente, permitió comprobar el largo alcance de la red y que no es necesario tener línea vista para funcionar.

La segunda prueba permitió comprobar que en línea vista la distancia máxima que se tiene es de 6.4 km. Ver figura 10.3. Por lo tanto el sistema excede el requisito de alcance original que se tenía.

10.2. Pruebas de Consumo

Uno de los requisitos del proyecto es que el sistema tenga una autonomía de una semana. Para ello se debe establecer un consumo nominal.

Se divide la prueba en 3. Se mide el consumo en un ciclo predefinido, del Módulo GRPS y Módulo LoRa Principal, de un Módulo LoRa Secundario y el consumo de un Módulo Sensor.

Las pruebas se realizaron utilizando una fuente DC que muestra el consumo de corriente que está teniendo el dispositivo alimentado. La fuente es “Zhaoxin PS-305D (Linear DC Power Supply PS-305D (0-30V, 0-5A, 1mA Display))”. Se registra un data log de los consumos para luego realizar un consumo medio en un período.

Se alimenta con la fuente en 5V y se utiliza un convertidor de 5/3.3V para tener las 2 tensiones de trabajo deseadas. Se midió el consumo del convertidor en vacío y el mismo fue de 0.08 mA.

A partir de los resultados de estas pruebas, en el Apéndice D, a modo de ejemplo, se recomiendan baterías en función de ciertos ciclos de funcionamientos preestablecidos.

10.2.1. Módulo LoRa Principal + Módulo GPRS

Para evaluar el consumo de estos módulos se utilizó la siguiente configuración de la red por considerarse la más exigente, donde solo se solicita datos a un sensor externo al Nodo. De esta forma se evalúa también el consumo en LoRa, ya que el consumo de los sensores se establece en la prueba de la Sección 10.2.3.

1. Sistema conformado por 2 Nodos:

- a) Nodo Principal sin Módulo Sensor, solicita datos a Nodo 1
- b) Nodo N1 con 1 Módulo Sensor - [Dir:08] - FS= 2 min (establecida desde servidor).

10.2. Pruebas de Consumo

2. Configuración de la comunicación LoRa:

- a) SF: 12
- b) CR: 4/7
- c) BW: 500 kHz
- d) potencia TX: 18 dBm

3. Solicitud de medida: Un ciclo (2 minutos + tiempo de conexión con el servidor)

10.2.1.1. Resultados

En la figura 10.4 se presentan los resultados obtenidos en la prueba.

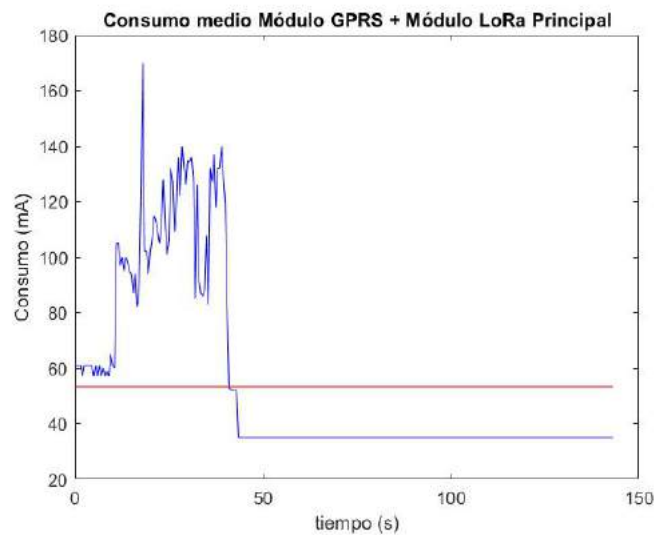


Figura 10.4: Consumo Módulo LoRa Principal + Módulo GPRS (azul), Consumo Medio (rojo)

Los resultados demuestran 2 fases de consumos correspondientes a las etapas de transmisión y bajo consumo. En la tabla 10.1 se muestran los consumos medios de cada etapa del ciclo y el consumo medio total. Se destaca que al disminuir la frecuencia en la solicitud de datos, bajan los consumos considerablemente ya que aumenta el porcentaje de tiempo en el que el dispositivo duerme.

Etapas	Consumo (mA)
Tx GPRS y LoRa	111.3
Bajo consumo	35
Consumo Medio Total	53.2

Tabla 10.1: Consumos medios por ciclo LoRa Principal + GPRS

Capítulo 10. Pruebas del Sistema

10.2.2. Módulo LoRa Secundario

Para evaluar el consumo de estos módulos se utilizó la misma configuración que en el Módulo de GPRS y LoRa Principal.

10.2.2.1. Resultados

En la figura 10.5 se presentan los resultados obtenidos en la prueba.

En estos módulos se pueden observar 3 fases de consumo correspondientes a las etapas de “esperando solicitud”, transmisión y bajo consumo. En la tabla 10.2 se muestran los consumos medios de cada etapa del ciclo y el consumo medio total. En este módulo el sistema duerme de forma fija 1 minuto en cada ciclo. No es variable como el principal, por lo tanto la mayor parte del tiempo se queda en el estado “esperando solicitud”. Esto evidencia un consumo medio que tiende al consumo de la etapa “esperando solicitud”, a medida que disminuye la frecuencia de solicitud de datos.

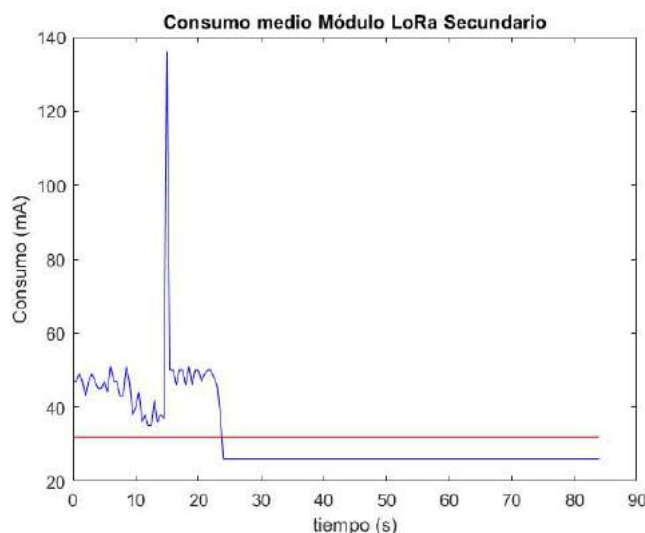


Figura 10.5: Consumo Módulo LoRa Secundario (azul), Consumo Medio (rojo)

Etapas	Consumo (mA)
Tx LoRa	140
Espera	43
Bajo consumo	26
Consumo Medio Total	32

Tabla 10.2: Consumo medio en un ciclo LoRa secundario

10.3. Pruebas de Desempeño del Sistema

10.2.3. Módulo Sensor

Para establecer el consumo medio del Módulo Sensor se conectó el módulo directamente a la fuente y se midió el consumo. En este caso el consumo fue el mismo todo el tiempo ya que no se implementó un modo de bajo consumo. En la figura 10.6 y en tabla 10.3 se muestra el consumo del mismo.

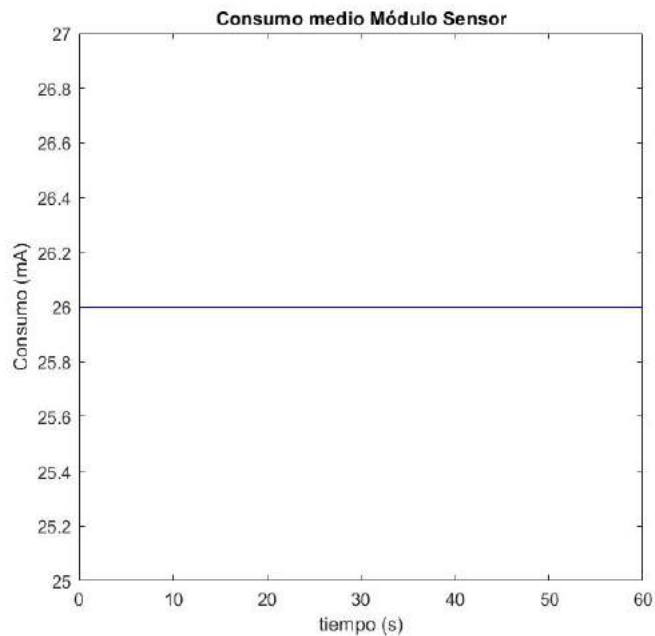


Figura 10.6: Consumo Módulo Sensor

Etapa	Consumo (mA)
Sensor	26
Consumo Medio Total	26

Tabla 10.3: Consumo medio Módulo Sensor

10.3. Pruebas de Desempeño del Sistema

Las pruebas de desempeño buscan conocer datos más allá del consumo. Estas pruebas permitieron conocer la tasa de fallas del sistema.

Capítulo 10. Pruebas del Sistema

10.3.1. Tasa de fallas

La tasa de fallas del sistema es la cantidad de intentos fallidos que tiene la red a la hora de guardar datos en el servidor. Se considera falla cuando el Servidor le solicita un reintento de un dato a la red, y en el reintento tampoco le llega un dato válido para ese sensor. Cuando al Servidor le llega un mensaje de error, solicita un reintento de dato al minuto nuevamente para ese sensor. Este reintento se puede visualizar, ya que luego de realizar consulta, no cambia el estado del sensor y se da alguno de los errores marcados en la Sección 8.5.

Para la prueba de Tasa de fallas se deja el sistema funcionando durante 24 horas. Se registran en el “LOG” del Servidor todos los eventos y luego se calcula el porcentaje de fallas.

Hipótesis de trabajo

1. Lugar de prueba: Laboratorio - Se considera una potencia de recepción máxima del Módulo GPRS y entre los Módulos LoRa.
2. Sistema conformado por 3 Nodos:
 - a) Nodo Principal con 1 Módulo Sensor - [Dir:09] - FS= 2 min.
 - b) Nodo N1 con 1 Módulo Sensor - [Dir:07] - FS= 2 min.
 - c) Nodo N2 con 1 Módulo Sensor - [Dir:08] - FS= 2 min.
3. Tiempo de funcionamiento: 24 horas.

10.3.1.1. Resultados

Luego de realizada la prueba se registran un total de 2477 eventos. En la tabla 10.4 se detallan los distintos eventos ocurridos y la cantidad de veces que se dio cada uno. En la Tabla 10.5 se distingue la cantidad de reintentos solicitados por el servidor y cuantas veces ese reintento no fue exitoso, además se distinguen las fallas por sensor.

Evento	Descripción	Cantidad
250	GET1:Consulta (primer intento)	824
250	GET1:Consulta (reintentos)	36
200	Dato Correcto (todos los sensores)	1581
425	Error en la recepción de datos desde la Red LoRa	21
426	TimeOut en recepción de datos desde la Red LoRa	15
Total		2477

Tabla 10.4: Eventos registrados en la prueba de Tasa de fallas

10.3. Pruebas de Desempeño del Sistema

Dirección	Reintentos solicitados	Fallas
09	0	0
08	14	0
07	22	2
Total	36	2

Tabla 10.5: Fallas por sensor

Las solicitudes de acciones y direcciones (estado 250) por parte del Módulo GPRS fue 824. En 36 de esas solicitudes existió algún inconveniente por lo que se solicita reintento de esos datos. Los eventos que produjeron una solicitud de reintento por parte del Servidor fueron los estados estados 425 y 426. De esos 36 reintentos, solo 2 no fueron exitosos.

Por lo tanto la tasa de fallas respecto a las solicitudes originales (824) totales informadas por el Servidor fue de:

$$TF = \frac{100 \times \text{Fallas}}{\text{Solicitudes totales}} = \frac{2}{824} = 0,24\%$$

Analizando las Tablas 10.4 y 10.5 se constata que las fallas provienen de la Red ya que el Sensor del Nodo Principal [Dir:09] no falló en ninguno de sus intentos. En esta prueba se comprueba la robustez del I2C, del Nodo Principal y del GPRS.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 11

Trabajos Futuros

En este capítulo se presentan los posibles desarrollos a futuro que puede tener el sistema y las líneas de investigación que se pueden seguir.

11.1. Nodo Principal

Si bien en un principio se pensó el Módulo GPRS independiente al Módulo LoRa principal y por lo tanto se incluyó un Arduino Nano para controlar el Modem, luego se decidió incluirlo en un mismo PCB.

El PCB que se desarrolló tiene el controlador principal de la red (DsPic), Chip Lora, controlador GPRS (Arduino nano) y Modem (SIM900). Esto es ineficiente, ya que el DSPic podría controlar todo.

Por lo tanto un desarrollo inmediato sería cambiar toda la comunicación con el SIM900 al DsPic. No es un cambio drástico ya que ambos lenguajes de programación son C/C++. Además del cambio en la programación se debe realizar un nuevo PCB, quitando las pistas que iban al Arduino y conectándolas directamente al DsPic.

11.2. Nuevas funcionalidades del Servidor

Una vez almacenado los datos en el servidor se puede hacer cualquier procesamiento con ellos, a continuación se muestran algunos ejemplos:

- Alarmas: a partir de distintos tipos de medidas de un nodo, se puede configurar una alarma que relacione las mediciones a un evento. (ej: condiciones dadas para un incendio)
- Envío de reportes por email a administrador del sistema.
- Aviso por SMS de alarma (ej: temperatura alta)

Capítulo 11. Trabajos Futuros

- Creación de usuarios con contraseña para ingresar al Servidor con distintos permisos, por ejemplo administradores que puedan crear/borrar nodos y sensores y usuarios que solo pueda visualizar datos. Esto ofrece seguridad de la información.
- Análisis de datos combinados entre varios sensores de un nodo .

11.3. Seguridad en el sistema

11.3.1. Transferencia de datos Servidor-Nodo Principal

La información desde el Nodo principal al Servidor se envía mediante HTTP, los mensajes de esta forma van en texto plano, alguien en algún punto de la cadena de transmisión IP podría ver que datos se están enviando. Una manera de mitigar este problema es utilizar HTTPS que encripta la información de punta a punta.

11.3.2. Comunicación entre Nodos LoRa

Si bien es necesario tener los 3 parámetros correctos (BW, SF y CR) para poder demodular los mensajes de una comunicación LoRa, el sistema actualmente no tiene una barrera que impida acceder a las transmisiones, más allá de tener estos parámetros. Una manera de brindar mayor seguridad a la Red sería agregar cifrado a los datos a transmitir, por ejemplo, realizando una encriptación simétrica.

11.4. Narrow band

En lugar de implementar la comunicación Módulo GPRS-Servidor mediante GPRS, se puede implementar mediante Narrowband-IOT.

Con esta tecnología se puede lograr bajar los consumos del Nodo principal y la cobertura es tanto o más extensa que la tecnología utilizada en nuestra investigación.

Actualmente solo esta disponible para desarrollo por lo que es una línea de investigación que no se ha profundizado. Debido a que aún no es una tecnología comercial en Uruguay, es necesario contactar con el ISP ANTEL, para obtener un chip de comunicación NB-IOT.

11.5. Modo Bajo Consumo

En el Nodo Principal tanto el Módulo LoRa como el Módulo GPRS cuentan con un modo de ahorro energético ya que este nodo tiene el mayor consumo. Sin embargo se podría aplicar modos de bajo consumo para los Nodos secundarios

11.5. Modo Bajo Consumo

y para los Módulos Sensores. Esto significaría un gran ahorro de energía lo que permitiría utilizar baterías de menor capacidad o bien extender en el tiempo la autonomía del sistema. Una de las propuestas es desplegar por toda la red, llegando hasta los Módulos Sensores inclusive, la hora de próxima despertada. Esto haría que todo el sistema este en modo bajo consumo hasta la próxima conexión con el Servidor.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 12

Conclusiones

12.1. Evaluación de resultados

Se desarrolló un sistema modular que cumple con todos los requisitos especificados de consumo, alcance y modularización.

La modularización lograda permite fácilmente intercambiar e incorporar los sensores que se precisen respetando el protocolo de comunicación propuesto por el Departamento de Sistemas y Control del IIE.

Se logró una Red de sensores inalámbrica que es capaz de ser gestionada desde un servidor, lo que permite la automatización en la adquisición de medidas. La Red es de una tecnología de bajo consumo, lo que permite al sistema funcionar largos periodos de tiempo.

Se pudo desarrollar un módulo de comunicación que cumplió con creces el requisito de gestión y alcance de la Red que se buscaba ya que este módulo tiene un gran poder de computo y largo alcance.

Por último, el sistema adquirió una robustez y confiabilidad excelentes lo que asegura una autonomía de funcionamiento ideal para aplicaciones donde su instalación sea en lugares remotos o de difícil acceso.

12.2. Aprendizajes

El proyecto requirió tener que estudiar e investigar por cuenta propia las distintas soluciones a las dificultades que se presentaban. A lo largo del mismo se tuvieron que poner en practica muchos de los conceptos que se adquieren a lo largo de la carrera. También permitió generar conocimientos en áreas donde en la carrera no se realiza un gran enfoque, por ejemplo, el desarrollo de Hardware.

Capítulo 12. Conclusiones

La realización y planificación de este proyecto deja experiencias que serán de gran utilidad a futuro en el desempeño profesional, ya que en el campo laboral es muy común que sea orientado a proyectos donde el éxito dependen fuertemente de la planificación y gestión.

Otro punto a destacar es el trabajo en equipo, en varias oportunidades en las diferentes materias se realizan tareas colectivas, pero para este proyecto la dedicación y carga horaria es mucho mayor, lo que requirió saber dividir y delegar tareas, planificar encuentros para alinear el trabajo y para seguir con lo planificado.

Referencias

- [1] Alejandro Draper, Nicolas Obrusnik y Pablo Zinemanas, “Pestibee”, Facultad de Ingeniería - UDELAR, 2015
- [2] Francisco de Izaguirre, Maite Gil, Marco Rolon, “Sistema Autonomo y Remoto de Adquisicion”, Facultad de Ingeniería - UDELAR, 2020
- [3] Ignacio Abadie, Mauro Martínez, Aldo Vignone, “AGROVISIÓN : Red de sensores inalámbricos para la previsión y tratamiento de plagas en cultivos frutales”, Facultad de Ingeniería - UDELAR, 2016
- [4] Cecilia Cardozo, Ignacio Camps, Martín Driedger, “Red de Sensores Inalámbricos para monitoreo de condiciones micro climáticas en cultivos de cítricos”, Facultad de Ingeniería - UDELAR, 2015
- [5] SIMCOM, SIM900_Hardware Design_V2.05 https://simcom.ee/documents/SIM900/SIM900_Hardware%20Design_V2.05.pdf. Acceso: Abril 2019
- [6] GEETECH, Shield GPRS https://www.geeetech.com/wiki/index.php/Arduino_GPRS_Shield. Acceso: Abril 2019
- [7] SIMCOM, Commands Manual_V1.11, Datasheet https://simcom.ee/documents/SIM900/SIM900_AT%20Command%20Manual_V1.11.pdf. Acceso: Abril 2019
- [8] Seeed-Studio/GPRS_SIM900 https://github.com/Seeed-Studio/GPRS_SIM900 Acceso: Abril 2019
- [9] Lowpower.h <https://github.com/rocketscream/Low-Power> Acceso: Agosto 2019
- [10] Cobertura Antel, <http://www.antel.com.uy/personas-y-hogares/movil/cobertura> Acceso Junio 2019
- [11] Arduino, <https://www.arduino.cc/>
- [12] Microchip, Low-Power Long Range LoRa Technology Transceiver Module, Datasheet
- [13] Microchip, RN2903 LoRa Technology Module Command Reference User's Guide, Datasheet

Referencias

- [14] Microchip, dsPIC33EPXXXGM3XX/6XX/7XX, Datasheet
- [15] DT11, https://components101.com/sites/default/files/component_datasheet/DHT11-Temperature-Sensor.pdf Acceso: Junio 2020
- [16] FC-28, <https://artofcircuits.com/product/fc-28-soil-moisture-sensor-analog-and-digital-outputs> Acceso: Mayo 2020
- [17] Arduino, Wire.h, <https://www.arduino.cc/en/Reference/Wire>
- [18] Bus I2C, <https://www.luisllamas.es/arduino-i2c/>
- [19] Trasmisión I2C, <https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/>
- [20] Semtech, N1200.22 LoRa Modulation Basics
- [21] Semtech, “LoRaTMModulation Basics, Application Note”, 2015
- [22] Daniel Antonio Jimenez Ballart, “Estudio de la cobertura de la modulación LoRa en la banda de 915 Mhz en un ambiente urbano”, Universidad de Chile, 2018
- [23] Leonardo Horovitz, Raúl Mayobre, “Telemetría de contenedores de residuos”, Universidad ORT, 2018
- [24] DecodingLora, <https://revspace.nl/DecodingLora> Acceso: Julio 2019
- [25] Spreading Factor, Bandwidth, Coding Rate and Bit Rate in LoRa, <https://josefmt.com/2018/08/14/spreading-factor-bandwidth-coding-rate-and-bit-rate-in-lora-english/> Acceso: Julio 2020
- [26] How Spreading Factor affects LoRaWAN device battery life, [https://www.thethingsnetwork.org/article/how-spreading-factor-affects-lorawan-device-battery-life#:~:text=Spreading%20Factor%20\(SF\)%20decides%20on,communication%20device%20and%20the%20gateway](https://www.thethingsnetwork.org/article/how-spreading-factor-affects-lorawan-device-battery-life#:~:text=Spreading%20Factor%20(SF)%20decides%20on,communication%20device%20and%20the%20gateway) Acceso: Julio 2020

Apéndice A

Comunicación LoRa

A continuación desarrolla el protocolo de comunicación LoRa y los parámetros más importantes que se deben tener en cuenta para lograr mayores distancias o mejores bitrates.

A.1. Modulación de espectro ensanchado

LoRa es un protocolo de capa física que utiliza la técnica de espectro expandido mediante chirps Css para transmitir la información. La modulación Css deriva de la modulación por espectro expandido SS.

Contrariamente a las técnicas de modulación clásicas donde se busca tener un ancho de banda mínimo, que cumpla la teoría de Nyquist-Shannon, esta técnica aumenta el ancho de banda de la señal. Esparce la potencia de la señal a lo largo de la banda. La señal original a transmitir es codificada con una secuencia de códigos de esparcimiento (Spread code) que lo que hacen es esparcir esa señal a lo largo de un ancho de banda determinado (spread spectrum). El receptor utiliza la misma secuencia para demodular la señal original. En la figura A.1 se muestra una modulación de espectro expandido.

La modulación por espectro expandido presenta las siguientes ventajas:

- Inmunidad frente al ruido: Al no concentrar la potencia en una sola frecuencia, evita que ruidos/interferencias en ciertas frecuencias hagan que no se pueda reconstruir la señal.
- Seguridad: Se utiliza para encriptar la señal. Solo el receptor que tiene el correcto spread code puede demodular la señal.
- Múltiple acceso por división de código: Muchos usuarios pueden compartir el medio, ya que este es multiplexado donde a cada usuario se le asigna un código diferente. Esta técnica deriva del CDMA del cuál proviene el SS.

Apéndice A. Comunicación LoRa

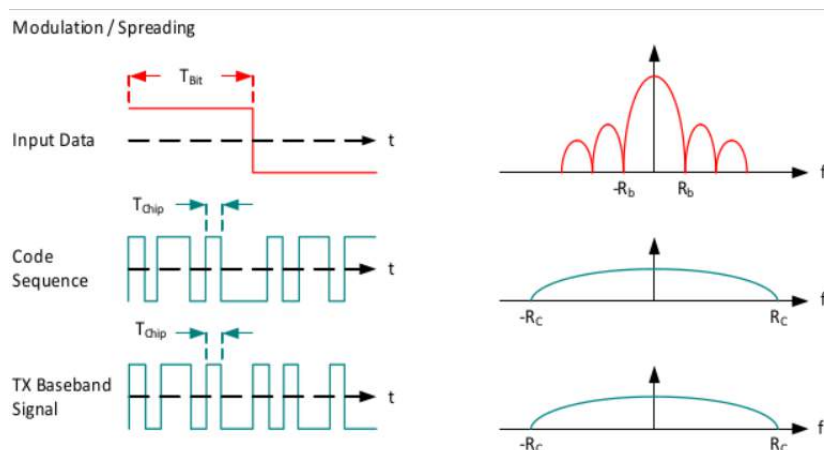


Figura A.1: Modulación de frecuencia expandida [21]

A.2. Parámetros de la comunicación LoRa

A.2.1. Spread Factor, Bandwidth, Code Rate

En la modulación LoRa se tiene 3 parámetros que son los que se utilizan para definir la comunicación que son:

- Spread Factor SF - Factor de esparcimiento de la señal
- Bandwidth - BW - Ancho de banda de la señal transmitida
- Code Rate - CR - Tasa de codificación de errores

Estos parámetros definen la velocidad de transmisión de datos lo que afecta el consumo y la sensibilidad que se requiere para demodular las señal lo que se refleja en el alcance máximo que se tiene.

El Spread Code, aquí, es conocido como chirp sequence. Estos pulsos son de una frecuencia variable linealmente en el tiempo, mucho mayor que la frecuencia de la señal de entrada. Este código se multiplica con la señal de entrada para luego transmitirla. En la figura A.2 se observa a modo de ejemplo una transmisión LoRa en la frecuencia de 869 MHz y su variación en frecuencia debido a la variación de los pulsos chirp.

El chirp rate es la cantidad de chirps por segundo que tiene el código: $Chirp\ Rate = [chirps/seg]$. Un símbolo LoRa está compuesto por 2^{SF} chirps. Por lo tanto un símbolo LoRa puede codificar SF bits de información. La relación entre el chip rate y la velocidad de símbolo es lo que se define como Spread factor [20]:

$$SF = \frac{Chirp\ rate}{Symbol\ rate}$$

A.2. Parámetros de la comunicación LoRa

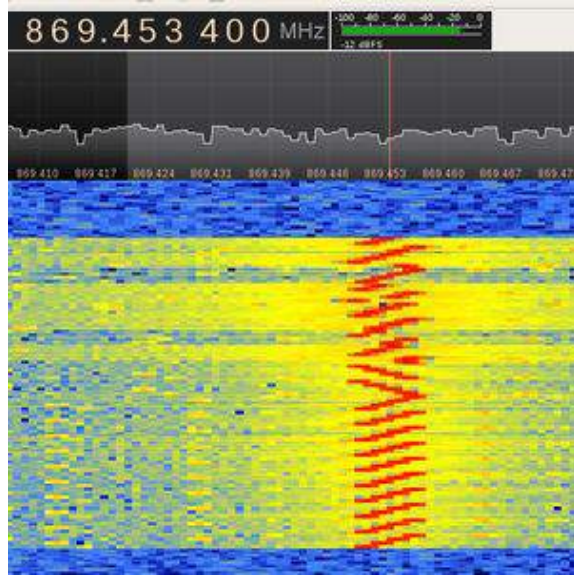


Figura A.2: Espectro de una transmisión LoRa [22]

SF	Chirps/símbolo
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096

Tabla A.1: Valores típicos de SF y chirps por símbolo

Un mayor número de SF implica un mayor número de chirps por símbolo lo que implica una mayor ganancia de sensibilidad en recepción. En LoRa los SF típicos son los expresados en la tabla A.1.

Los anchos de banda (BW) típicos de LoRa son 125 KHz, 250KHz y 500KHz y estos determinan el ancho de banda efectivo que ocupa la transmisión.

A partir del SF y del BW es que se define el tiempo de símbolo T_s y la velocidad de símbolo R_s de una transmisión LoRa:

$$T_s = \frac{1}{R_s} = \frac{2^{SF}}{BW}$$

Debido a que LoRa agrega una corrección de errores a cada transmisión. Esta corrección de errores CR también se denomina FEC y se basa en agregar cierta cantidad de bits redundantes por 4 cada bits de información. Los valores típicos

Apéndice A. Comunicación LoRa

de CR son: 4/5, 4/6, 4/7, 4/8. Donde el primer número corresponde a los bits de información y el segundo a los bits totales enviados.

Por lo tanto el bitrate final R_b en una transmisión depende de los 3 parámetros antes explicados SF, CR y BW:

$$R_s = \frac{SF \times 2^{SF} \times CR}{BW} bps$$

A.2.2. Relación entre los parámetros, el bitrate y la sensibilidad

A continuación se describen los efectos más importantes de variar estos parámetros en la transmisión.

- Aumentar el BW, disminuye el tiempo de bit por lo que aumenta el bitrate de la comunicación ya que aumenta la capacidad del canal. También disminuye la sensibilidad del receptor
- Aumentar el SF, baja el bitrate ya que aumenta el tiempo de bit, a pesar que un símbolo transporta más bits. Una de las contras de aumentar el SF es que al bajar el bitrate, implica que se tenga que transmitir una mayor cantidad de tiempo para enviar la misma información, por lo tanto aumenta el consumo (20 veces más un SF12 respecto a SF7 en un ciclo de transmisión [26]). Sin embargo una de las ventajas es el aumento en la sensibilidad del receptor.

En la tabla A.2 se muestra la relación entre el BW, y el SF en el bitrate de la comunicación y la sensibilidad del receptor, para un CR de 4/7.

BW (kHz)	SF	Bitrate (bps)	Sensibilidad (dB)
125	7	4557	-123
125	12	244	-136
500	7	18229	-117
500	12	976	-137

Tabla A.2: Bitrate y Sensibilidad en función del BW y SF para un CR de 4/7 [23] y [22]

Apéndice B

PCB, lista de componentes y esquemático del Módulo LoRa

A continuación se muestran el cad del PCB final del módulo de comunicación LoRa, figura B.1, la lista de componentes, Tabla B.1 y el esquemático del mismo, figura B.2.

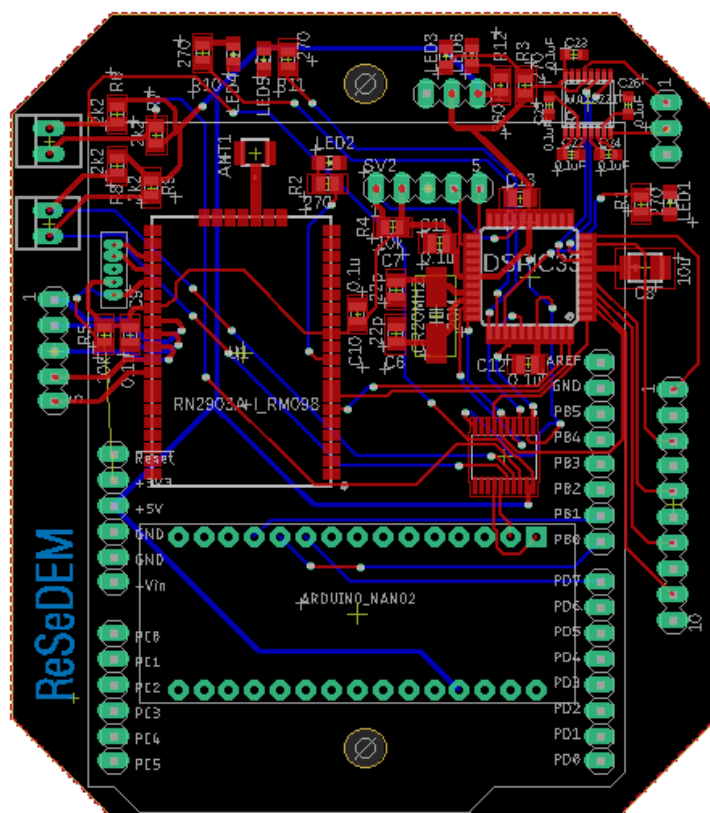


Figura B.1: Cad del PCB del módulo de comunicación

Apéndice B. PCB, lista de componentes y esquemático del Módulo LoRa

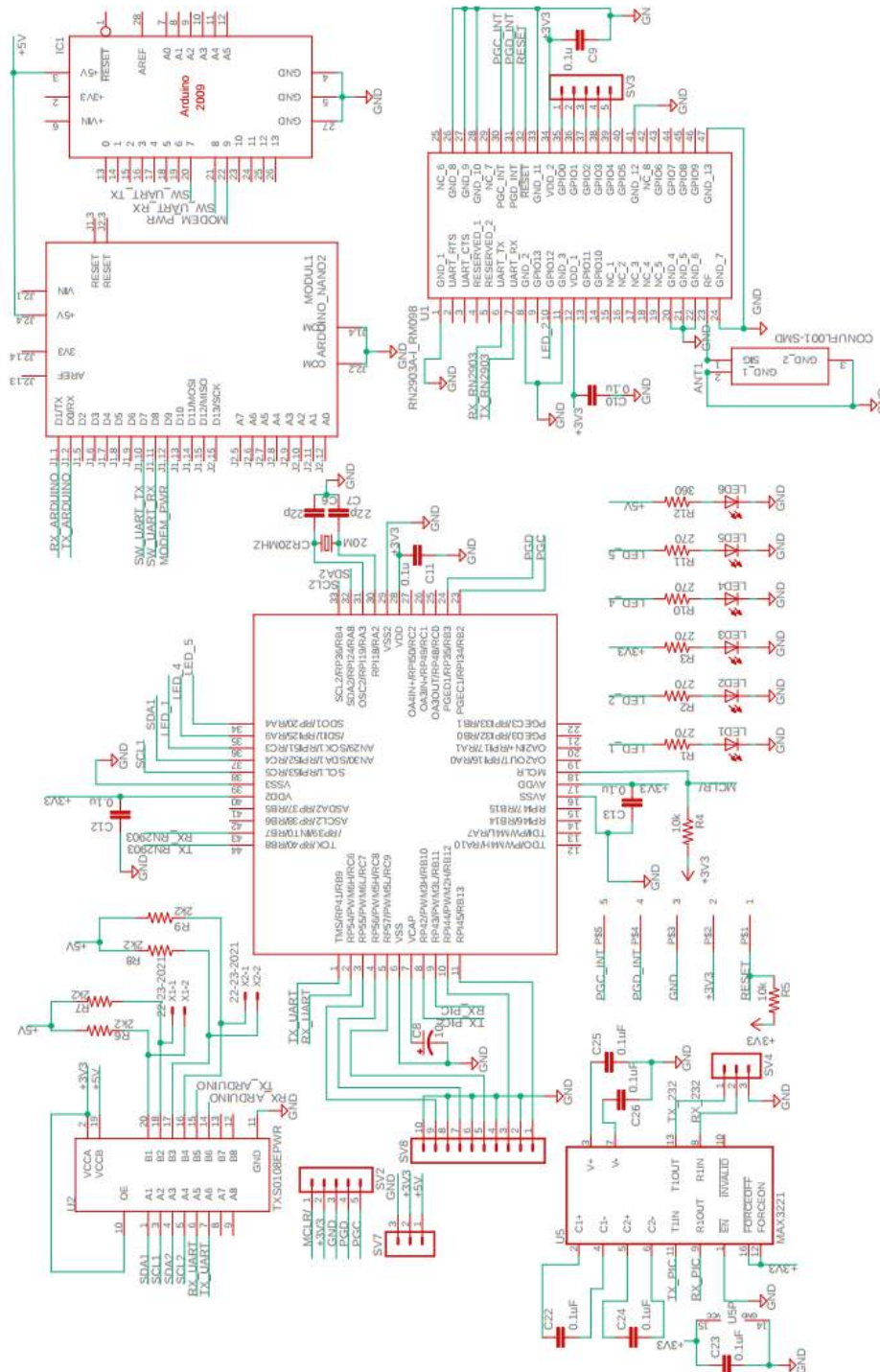


Figura B.2: Esquemático de la placa ReSeDEM

Componente	Valor	Código	Descripción
ANT1	CONUFL001-SMD	CONUFL001-SMD	Antena con cable coaxial
C6	22p	C-EUC0805	Capacitor
C7	22p	C-EUC0805	Capacitor
C8	10u	CPOL-USCT3528	Capacitor
C9	0.1uF	C-EUC0805	Capacitor
C10	0.1uF	C-EUC0805	Capacitor
C11	0.1uF	C-EUC0805	Capacitor
C12	0.1uF	C-EUC0805	Capacitor
C13	0.1uF	C-EUC0805	Capacitor
C22	0.1uF	C-EUC0603K	Capacitor
C23	0.1uF	C-EUC0603K	Capacitor
C24	0.1uF	C-EUC0603K	Capacitor
C25	0.1uF	C-EUC0603K	Capacitor
C26	0.1uF	C-EUC0603K	Capacitor
CR20MHZ	20M	CRYSTAL	Oscilador 20MHz
DSPIC33	DSPIC33EP128GM604	DSPIC33EP128GM604	Microcontrolador
IC1	Arduino Shield	ARDUINO2009_11	Arduino Shield
LED1	Led	LEDCHIP-LED0805	Led
LED2	Led	LEDCHIP-LED0805	Led
LED3	Led	LEDCHIP-LED0805	Led
LED4	Led	LEDCHIP-LED0805	Led
LED5	Led	LEDCHIP-LED0805	Led
LED6	Led	LEDCHIP-LED0805	Led
MODUL1	Arduino Nano	ARDUINO_NANO2	Arduino Nano
R1	270 Ohm	R-US_R0805	Resistencia
R2	270 Ohm	R-US_R0805	Resistencia
R3	270 Ohm	R-US_R0805	Resistencia
R4	10k Ohm	R-US_R0805	Resistencia
R5	10k Ohm	R-US_R0805	Resistencia
R6	2k2 Ohm	R-US_R0805	Resistencia
R7	2k2 Ohm	R-US_R0805	Resistencia
R8	2k2 Ohm	R-US_R0805	Resistencia
R9	2k2 Ohm	R-US_R0805	Resistencia
R10	270 Ohm	R-US_R0805	Resistencia
R11	270 Ohm	R-US_R0805	Resistencia
R12	360 Ohm	R-US_R0805	Resistencia
SV2	-	MA05-1	Cabezal Pin
SV3	-	MA05-1	Cabezal Pin -
SV4	-	MA05-1	Cabezal Pin
SV7	-	MA05-1	Cabezal Pin
SV8	-	MA05-1	Cabezal Pin
U\$1	CONN1.27MM5P	CONN1.27MM5P	-
U1	RN2903A-IRM098	RN2903-I/RM095	Chip LoRa Rx/TX
U2	TXS0108EPW	SOP65P640X120-20N	Conversor de niveles de tensión
U5	MAX3221	MAX3221EUE	Transceiver RS232
X1	22-23-2021	-	Cabezal Pin
X2	22-23-2021	-	Cabezal Pin

Tabla B.1: Lista componentes de la placa ReSeDEM del Módulo LoRa

Esta página ha sido intencionalmente dejada en blanco.

Apéndice C

Comandos GPRS

En la Tabla C.1 se describen comandos, a modo informativo, que pueden ser de utilidad para agregarle prestaciones a la comunicación GPRS. Estos comandos son de utilidad para establecer la QOS mínima sabiendo la intensidad en la señal recibida, establecer alarmas a partir de una hora determinada o enviar alertas por SMS en determinadas circunstancias. También se describen algunos Comandos AT para realizar una conexión TCP o UDP y comandos FTP para transmisión de archivos. La conexión FTP es útil en caso de querer mandar paquetes grandes de datos. Si bien en el proyecto se implemento HTTP, se estudió la posibilidad de realizarlo por alguna de estas formas también.

Comando AT	Descripción del comando
AT+CREG	Estado de conexión en la Red
AT+CGATT	Conectarse a la Red
AT+CSQ	Nivel de señal del dispositivo
AT+CMGF	Establece el Modo SMS
AT+CMGS	Enviar un SMS
AT+CIPSTART	Iniciar comunicación TCP o UDP
AT+CIPSEND	Enviar datos a través de la conexión TCP o UDP
AT+CIPCLOSE	Cierra conexión TCP o UDP
AT+QFTPUSER	Usuario FTP
AT+QFTPPASS	Contraseña FTP
AT+QFTPPATH	Establece ruta destino
AT+QFTPPUT	Envia archivo mediante FRP

Tabla C.1: Comandos AT adicionales

Esta página ha sido intencionalmente dejada en blanco.

Apéndice D

Módulo Alimentación - Baterías

En este Apéndice se brindan recomendaciones de baterías para una autonomía de una semana del sistema en función de las frecuencias con la que se solicitan datos. La frecuencia depende de la finalidad con la que se instala la Red ReSeDEM.

Estas recomendaciones se calculan en base a los consumos estimados en la Sección 10.2 y tomando todas las frecuencias de los sensores instalados iguales. De tener frecuencias diferentes los sensores se puede tener un consumo mixto.

Frecuencia de solicitud de datos: 2 minutos

Nodo	Módulo	Consumo (mA)
Principal	LoRa + GPRS	53
Principal	Sensor	26

Batería Recomendada (mAh) $8900 + 4000 \times \text{sensor}$

Secundario	Secundario	32
Secundario	Sensor	26

Batería Recomendada (mAh) $5400 + 4000 \times \text{sensor}$

Tabla D.1: Baterías para autonomía 1 semana. $f_s=2$ minutos

Frecuencia de solicitud de datos: 10 minutos

Apéndice D. Módulo Alimentación - Baterías

Nodo	Modulo	Consumo (mA)
Principal	LoRa + GPRS	40
Principal	Sensor	26
Batería Recomendada (mAh)		6700 + 4000 × sensor

Secundario	Secundario	45
Secundario	Sensor	26
Batería Recomendada (mAh)		7500 + 4000 × sensor

Tabla D.2: Baterías para autonomía 1 semana. fs=10 minutos

Frecuencia de solicitud de datos: Mayor/igual a 15 minutos

Nodo	Modulo	Consumo (mA)
Principal	LoRa + GPRS	36
Principal	Sensor	26
Batería Recomendada (mAh)		6000 + 4000 × sensor

Secundario	Secundario	45
Secundario	Sensor	26
Batería Recomendada (mAh)		7500 + 4000 × sensor

Tabla D.3: Baterías para autonomía 1 semana. fs=15 minutos

chapterManual de Usuario

Este manual tiene la intención de brindar una descripción de los componentes del sistema, el conexionado de los componentes, la interfaz de usuario y la inicialización del sistema. Además se describe como detectar posibles fallas que puedan estar sucediendo en el sistema.



ReSeDEM

MANUAL DE USUARIO

Contenido

Descripción.....	1
Componentes de una Red.....	1
Módulo Principal	1
Modulo Secundario	2
Juego de Cables	3
Módulo Sensor	3
Módulo Alimentación.....	4
Interfaz de Usuario/ Página WEB.....	4
Página de Inicio	4
Configuración de la Red	5
Crear Nodo	5
Eliminar Nodo.....	5
Configuración de los Sensores	5
Monitoreo de la Red	6
Despliegue de datos	6
Monitor	6
LOG	7
Puesta en funcionamiento	8
Elementos necesarios.....	8
Inicialización de sensores	8
Eliminado de sensores.....	9

Descripción

Con este manual el usuario será capaz de poner en funcionamiento una red de sensores ReSeDEM. Una vez inicializada será capaz de realizar un seguimiento activo de los datos relevados por los sensores instalados desde la página web.

Componentes de una Red

Módulo Principal

Módulo encargado de establecer la conexión con el servidor y luego escanear la Red LoRa solicitando las acciones pendientes.

Consta de una caja estanca con dos conectores de 7 pines para interconectarse con cualquier otro elemento de la red. A su vez cuenta con dos conectores para las antenas LoRa y GPRS.



Modulo Secundario

Este módulo recibe solicitud de acciones desde el Módulo Principal para luego escanear su red local I2C y devolver las medidas obtenidas.

Consta de una caja estanca con dos conectores de 7 pines para interconectarse con cualquier otro elemento de la red. A su vez cuenta con un conector para la antena LoRa.



Juego de Cables

Necesarios para configurar y alimentar el Módulo Principal y Secundario e integrar un sensor a la red.

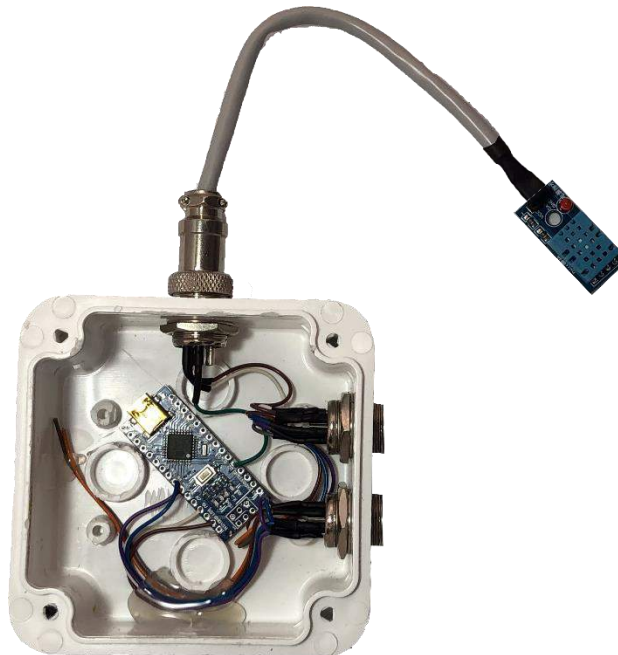
Consta de dos cables con conectores de 7 pines. Uno para la alimentación del Módulo y otro para conectar el sensor a integrar a la red. Por otro lado, un conector DB9 para la configuración de los Módulos y de los parámetros de la red LoRa.



Módulo Sensor

Módulo encargado de relevar la medida de interés.

Consta de una caja estanca con dos conectores de 7 pines para interconectarse con cualquier otro elemento de la red y un conector de 5 pines para la conexión del sensor.



Módulo Alimentación

Módulo encargado de alimentar los elementos de la red.

Cada Nodo necesita de un Módulo de Alimentación. La elección de este dependerá de la autonomía requerida por el usuario.



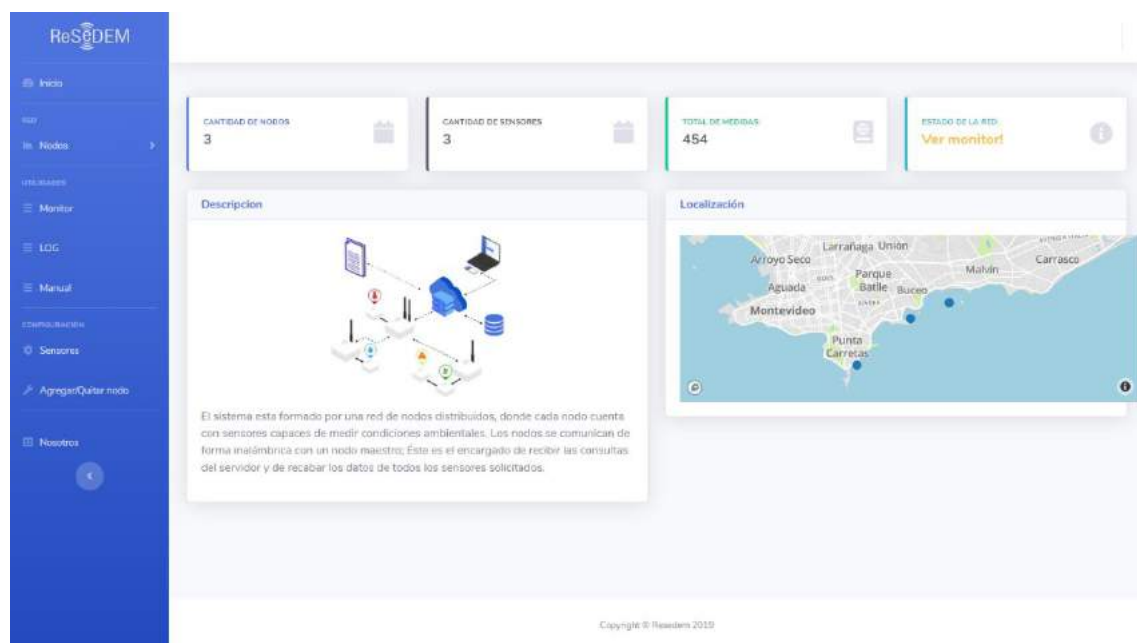
Interfaz de Usuario/ Página WEB

Para ingresar al portal web del sistema dirigirse a la siguiente URL:

<http://resedem.bitnamiapp.com/resedemNew>

Página de Inicio

La página de inicio del Servidor muestra información general de la Red como la cantidad de nodos configurados, cantidad de sensores, cantidad de medidas realizadas y ubicación geográfica de los nodos.



Configuración de la Red

Crear Nodo

Para agregar un Nodo dirigirse a la pestaña ``Agregar/Quitar nodo''. Será necesario ingresar las coordenadas geográficas de la posición del nodo (Formato UTM). Luego dar clic en “Agregar”. Automáticamente se le asignará el número de identificación al Nodo.

Eliminar Nodo

En la pestaña ``Agregar/quitar Nodo'' seleccionar el Nodo a eliminar y dar clic en “Eliminar”.

La imagen muestra dos secciones de la interfaz de usuario. La primera sección, titulada "Agregar nuevo nodo", indica "Se agregará el nodo N° 4" y contiene campos de entrada para "Latitud:" y "Longitud:". Debajo de estos campos hay un botón azul que dice "Agregar". La segunda sección, titulada "Eliminar nodo", tiene un menú desplegable etiquetado "Nodo:" con el valor "1" seleccionado. Debajo del menú hay un botón rojo que dice "Eliminar".

Configuración de los Sensores

Para configurar los sensores dirigirse a la pestaña “Sensores”. En esta sección podremos ver que sensores hay en la Red, agregar y quitar sensores y modificar los tiempos de muestreo de estos o cambiarlos de Nodo.

La imagen muestra la interfaz de usuario para la configuración de sensores. En la parte superior, hay un encabezado "Sensores". Debajo, hay un control "Show 10 entries" y un campo de búsqueda "Search:". El contenido principal es una tabla con las siguientes columnas: DIR, Nodo, Tipo, FS y Estado. La tabla contiene tres filas de datos. Debajo de la tabla, hay un indicador "Showing 1 to 3 of 3 entries" y botones de navegación "Previous", "1" (seleccionado) y "Next". En la parte inferior, hay una barra de herramientas con los enlaces "Agregar sensor", "Eliminar sensor", "Modificar muestreo" y "Cambiar de Nodo".

DIR	Nodo	Tipo	FS	Estado
07	1	TemAmb1	15	300
08	2	HumAmb	15	300
09	3	TempAmb2	15	300

Para agregar un sensor se debe especificar la dirección que tendrá, el Nodo al cual pertenece y la frecuencia de medias que tendrá. La dirección “1E” está reservada para el sistema. Una vez agregado, el sensor queda en estado “100 inicializando” a la espera de la confirmación de la Red LoRa. Al recibir confirmación, el sensor pasa a estado “200” y empieza a solicitar sus medidas. Sólo se puede inicializar un sensor a la vez.

Para eliminar un sensor basta con ingresar la dirección del sensor. De esta forma el sensor pasa a estado “604” esperando la confirmación de la Red LoRa. Una vez recibida la confirmación el sensor es eliminado completamente del sistema.

De forma similar para modificar los tiempos de muestro o para cambiarlo de Nodo se selecciona la opción correspondiente y se indica la dirección del sensor.

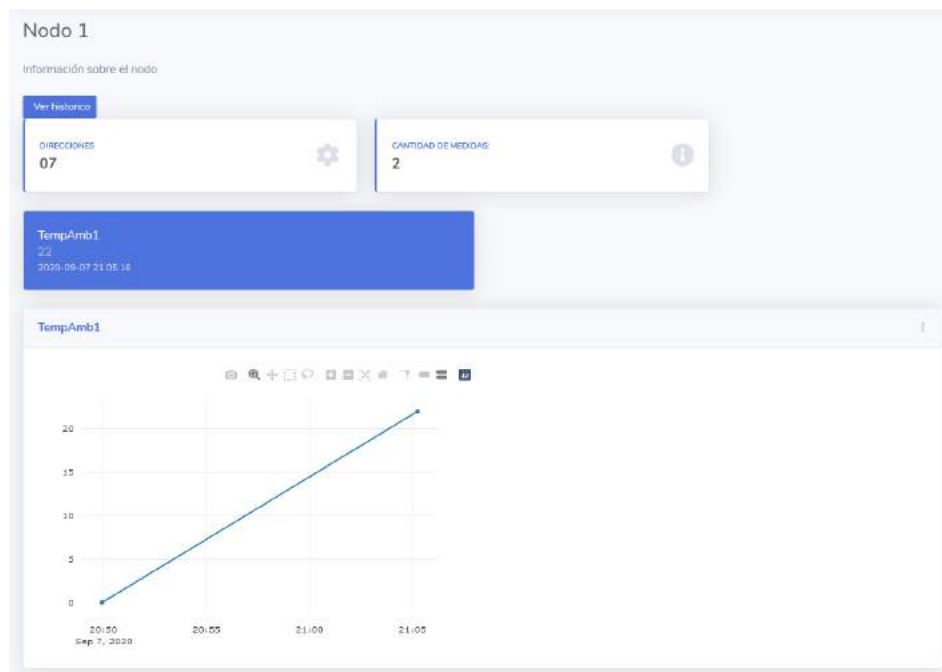
Monitoreo de la Red

En el servidor se lleva registro de todos los datos enviados por la Red. Se puede ver las medidas, el estado de los sensores y un histórico de los mismos.

Despliegue de datos

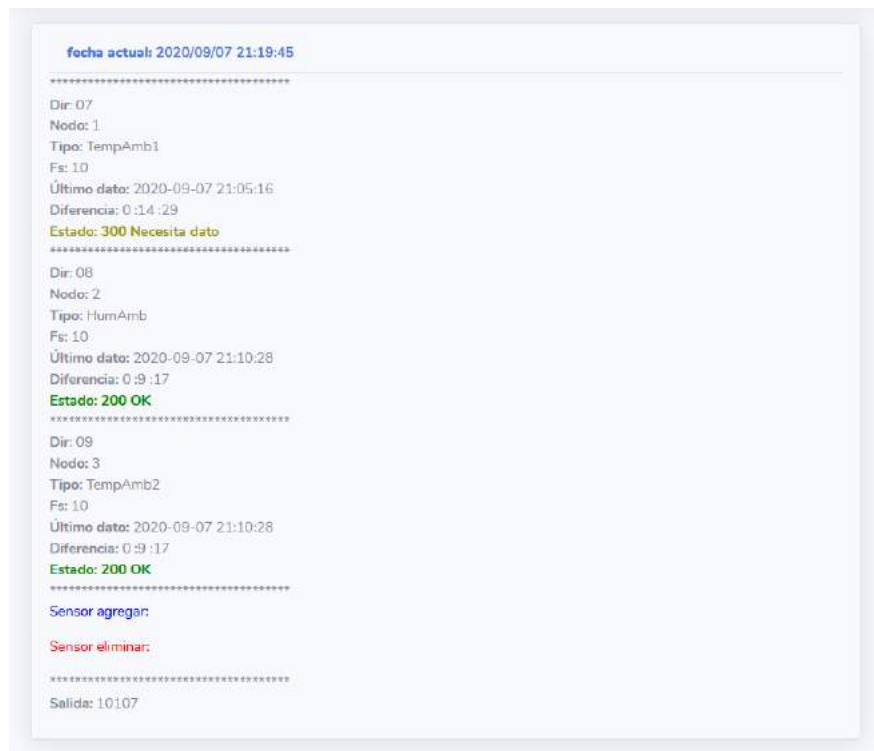
Para ver todas las medidas (válidas) que han sido almacenadas en el servidor hay que dirigirse a la pestaña “Nodos” y seleccionar el Nodo al cual pertenece el sensor que se quiere ver las medidas.

En esa pestaña se informa cuantas medidas se han realizado. Para ver el histórico en forma gráfica se debe dar clic en “Datos” y seleccionar el sensor que se quiera visualizar. Para ver los datos en una de tabla se debe dar clic en “Ver histórico” y seleccionar el sensor que se quiera ver. En las tablas se indicará número de medida, fecha de almacenada y el valor.



Monitor

La pestaña “Monitor”, nos permite ver el estado actual de todos los sensores, aquí podremos ver si están enviando datos correctamente o si algún sensor está reportando algún error.



LOG

Esta pestaña guarda el historial de las comunicaciones del Nodo Principal con el servidor, se guardan todos los mensajes reportados por la Red. Se registra el evento, la fecha y una descripción de este. Es una herramienta útil para detectar errores y evaluar el desempeño de la Red

LOG			
Fecha	Sensor	Estado	Descripción
2020-09-07 21:18:45	00	0	next_dato: 1 minutos
2020-09-07 21:18:45	07	200	Dato correcto
2020-09-07 21:18:36	0	250	GET:CONSULTA
2020-09-07 21:17:31	00	0	next_dato: 1 minutos
2020-09-07 21:17:31	07	200	Dato correcto
2020-09-07 21:17:22	0	250	GET:CONSULTA
2020-09-07 21:16:17	00	0	next_dato: 1 minutos
2020-09-07 21:16:17	07	200	Dato correcto
2020-09-07 21:16:08	0	250	GET:CONSULTA
2020-09-07 21:10:28	08	200	Dato correcto
2020-09-07 21:10:28	09	200	Dato correcto
2020-09-07 21:10:28	00	0	next_dato: 5 minutos
2020-09-07 21:10:11	0	250	GET:CONSULTA
2020-09-07 21:06:07	0	250	GET:CONSULTA
2020-09-07 21:05:16	00	0	next_dato: 2 minutos

Puesta en funcionamiento

Elementos necesarios

Para inicializar la red se necesita al menos:

1. Dispositivo con acceso a internet.
2. Módulo LoRa Principal.
3. Módulo Alimentación.
4. Módulo Sensor.
5. Juego de cables.



En caso de necesitar desplegar la red de sensores, se debe anexar el siguiente paquete de ítems. Tantos como el usuario desee anexar a la red.

1. Módulo LoRa Secundario.
2. Módulo Alimentación.
3. Módulo Sensor.
4. Juego de Cables.



Observaciones:

- Cada Nodo soporta hasta dos sensores. En caso de necesitar más sensores es necesario agregar más Módulos Secundarios.
- El Módulo Principal debe contener una tarjeta SIM de algún proveedor de telecomunicaciones con soporte de la red GPRS.

Inicialización de sensores

Los sensores deben ser agregados **uno a la vez**. Por lo tanto, el procedimiento a seguir es el siguiente:

1. Ingresar a http://resedem.bitnamiapp.com/resedemNew/add_sensor.php y completar los datos del sensor.
 - a. Dir.: Dirección I2C a grabar en el sensor conectado.
 - b. Nodo: En que Nodo de la red va a quedar conectado.
 - c. Tipo: Nombre indicando el tipo de medida que va a tomar el sensor.
 - d. Muestreo: Cada cuanto el sensor debe reportar una nueva medida.
2. Conectar con el Juego de Cables el sensor a agregar en el Módulo Principal o en el Módulo Secundario.
3. Conectar con el Juego de Cables el Módulo Principal y el o los Módulos Secundarios a los Módulo de Alimentación.
4. Ingresar a http://resedem.bitnamiapp.com/resedemNew/consulta_d.php y verificar el estado del sensor a agregar. Una vez inicializado el estado será: **200 OK**.

Observación:

El proceso de inicialización puede tomar hasta un minuto.

5. En caso de necesitar dar de alta un nuevo sensor, volver al paso número 1 y saltar el paso número 3.

Observación:

Al agregar un segundo sensor, puede que el Módulo Principal se encuentre en ahorro de energía. En este caso tenemos dos opciones:

- i. Reiniciar el Módulo Principal.
- ii. Esperar a que salga de ahorro de energía. Este tiempo va a depender del Muestreo de los sensores agregados anteriormente.

Eliminado de sensores

Para eliminar sensores el procedimiento es el siguiente:

1. Ingresar a http://resedem.bitnamiapp.com/resedemNew/delete_sensor.php
2. Ingresar la dirección del sensor a eliminar.
3. Clic en el botón Eliminar.
4. Verificar en el monitor el estado del sensor eliminado.
5. Una vez que en el monitor no se visualice más el sensor, desconectarlo de la red.

Observación:

El sistema permite eliminar muchos sensores a la vez. Es fundamental que, si se eliminan más de un sensor, los eliminados sean desconectados de la red.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice E

Manual Técnico

Este manual tiene la intención de brindar una descripción técnica de los componentes. Incluye el Manual de Usuario.



ReSeDEM

MANUAL TÉCNICO

Contenido

Descripción.....	1
Ensamblado.....	1
Datos y alimentación.....	1
Conexión del sensor	1
Módulo LoRa	2
PCB	2
Esquemático.....	3
Lista de Componentes.....	4
Puesta en funcionamiento	5
Configuración de Red LoRa	5
Modificación de los parámetros.....	5
Configuración del tipo de Módulo.	5
Inicialización de sensores	6
Posibles problemas	6
Eliminado de sensores.....	7
Posibles problemas	7
Funcionamiento	8
Verificación de alimentación.....	8
Verificación de funcionamiento	8
LoRa Principal	8
Verificación de funcionamiento	8
LoRa Secundario.....	8
Verificación de funcionamiento	8
Modem	8
Verificación de funcionamiento	8
Comandos Configuración Red LoRa	9

Descripción

Con este manual el técnico será capaz de verificar el correcto funcionamiento de una Red LoRa y en caso de falla, determinar en qué punto de la red se encuentra. También será capaz de reparar los distintos módulos en caso de daños.

Ensamblado

Datos y alimentación

Se debe utilizar un cable de 7 hilos para el armado de los conectores. El cable debe ser armado con un conector GX16-7p en cada extremo.



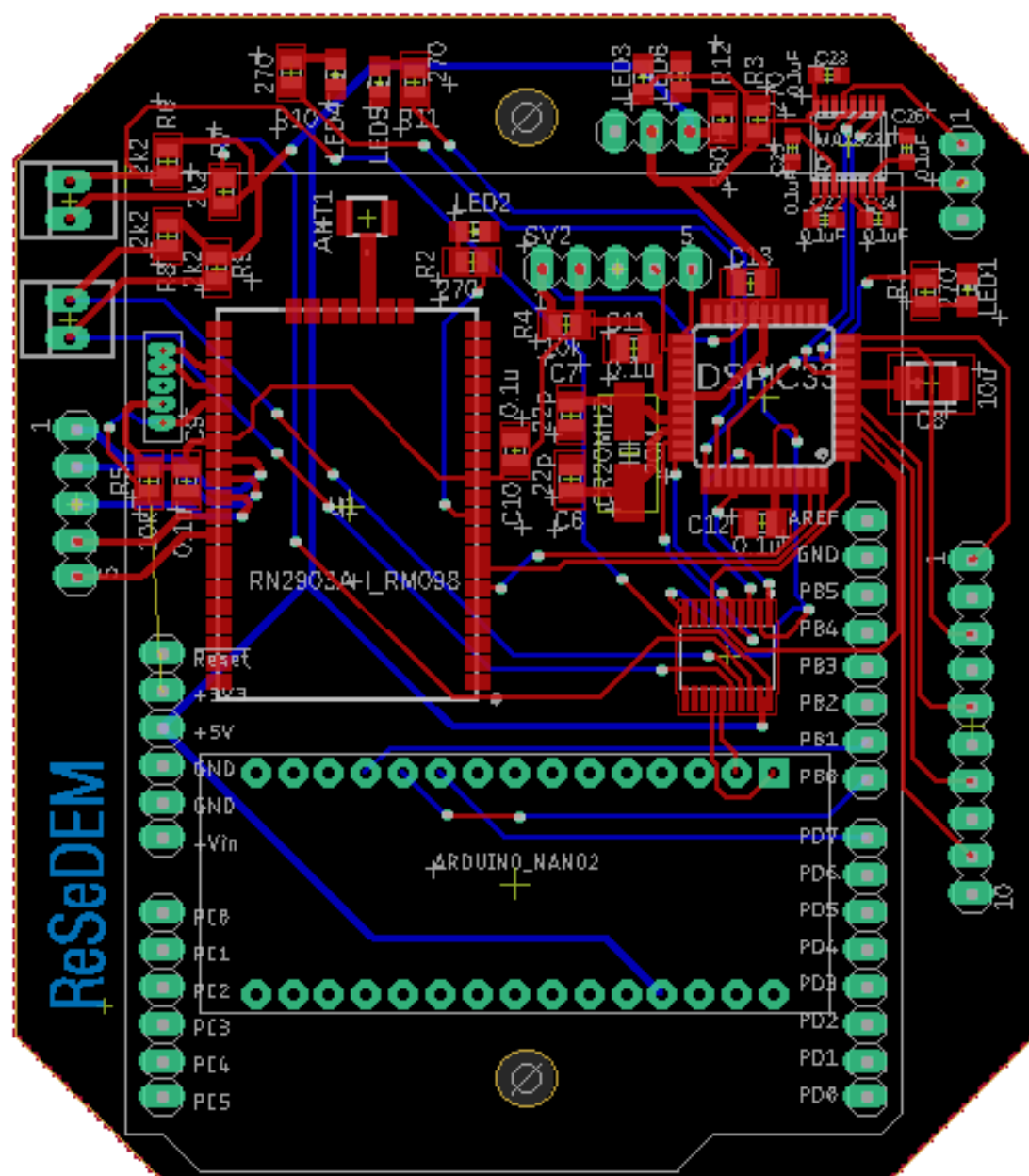
Conexión del sensor

Se debe utilizar un cable de 5 hilos para el armado de los conectores. El cable debe ser armado con un conector GX16-5p en un extremo y en el otro el sensor a utilizar.

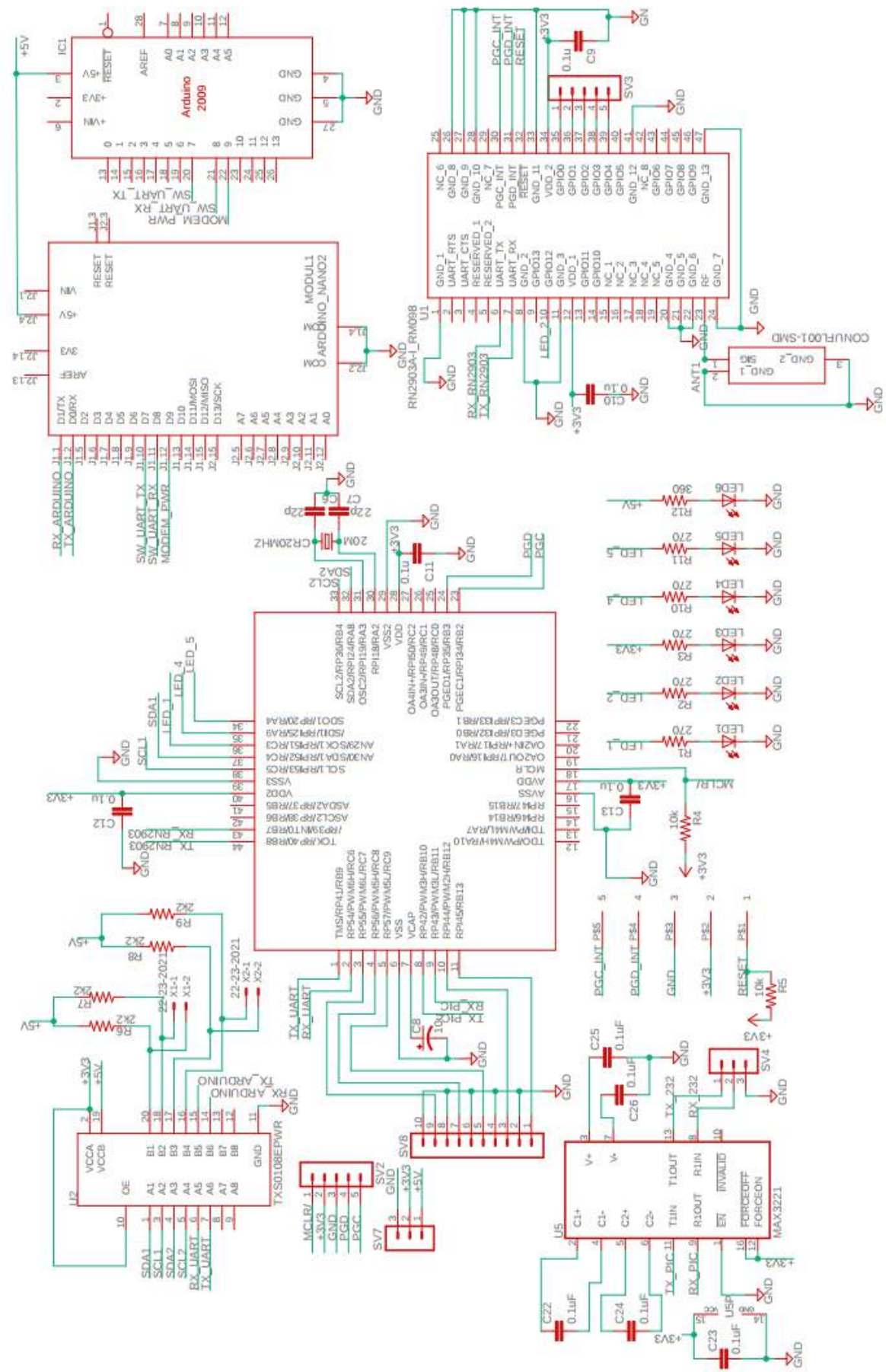
El conexionado de los pines depende del tipo de sensor y de los pines utilizados dentro del Módulo Sensor para relevar la medida en cuestión.



Módulo LoRa
PCB



Esquemático



Lista de Componentes

Componente	Valor	Código	Descripción
ANT1	CONUFL001-SMD	CONUFL001-SMD	Antena con cable coaxial
C6	22p	C-EUC0805	Capacitor
C7	22p	C-EUC0805	Capacitor
C8	10u	CPOL-USCT3528	Capacitor
C9	0.1uF	C-EUC0805	Capacitor
C10	0.1uF	C-EUC0805	Capacitor
C11	0.1uF	C-EUC0805	Capacitor
C12	0.1uF	C-EUC0805	Capacitor
C13	0.1uF	C-EUC0805	Capacitor
C22	0.1uF	C-EUC0603K	Capacitor
C23	0.1uF	C-EUC0603K	Capacitor
C24	0.1uF	C-EUC0603K	Capacitor
C25	0.1uF	C-EUC0603K	Capacitor
C26	0.1uF	C-EUC0603K	Capacitor
CR20MHZ	20M	CRYSTAL	Oscilador 20MHz
DSPIC33	DSPIC33EP128GM604	DSPIC33EP128GM604	Microcontrolador
IC1	Arduino Shield	ARDUINO2009_11	Arduino Shield
LED1	Led	LEDCHIP-LED0805	Led
LED2	Led	LEDCHIP-LED0805	Led
LED3	Led	LEDCHIP-LED0805	Led
LED4	Led	LEDCHIP-LED0805	Led
LED5	Led	LEDCHIP-LED0805	Led
LED6	Led	LEDCHIP-LED0805	Led
MODUL1	Arduino Nano	ARDUINO_NANO2	Arduino Nano
R1	270 Ohm	R-US_R0805	Resistencia
R2	270 Ohm	R-US_R0805	Resistencia
R3	270 Ohm	R-US_R0805	Resistencia
R4	10k Ohm	R-US_R0805	Resistencia
R5	10k Ohm	R-US_R0805	Resistencia
R6	2k2 Ohm	R-US_R0805	Resistencia
R7	2k2 Ohm	R-US_R0805	Resistencia
R8	2k2 Ohm	R-US_R0805	Resistencia
R9	2k2 Ohm	R-US_R0805	Resistencia
R10	270 Ohm	R-US_R0805	Resistencia
R11	270 Ohm	R-US_R0805	Resistencia
R12	360 Ohm	R-US_R0805	Resistencia
SV2	-	MA05-1	Cabezal Pin
SV3	-	MA05-1	Cabezal Pin -
SV4	-	MA05-1	Cabezal Pin
SV7	-	MA05-1	Cabezal Pin
SV8	-	MA05-1	Cabezal Pin
U\$1	CONN1.27MM5P	CONN1.27MM5P	-
U1	RN2903A-LRM098	RN2903-I/RM095	Chip LoRa Rx/TX
U2	TXS0108EPW	SOP65P640X120-20N	Convertidor de niveles de tensión
U5	MAX3221	MAX3221EUE	Transceiver RS232
X1	22-23-2021	-	Cabezal Pin
X2	22-23-2021	-	Cabezal Pin

Puesta en funcionamiento

Configuración de Red LoRa

Los nodos inician con una configuración LoRa default la que se detalla a continuación que maximiza el alcance:

- Frequency 928 MHz
- Spreading Factor 12
- Coding Rate 4/7
- Bandwidth 500kHz
- Power 20

Esta configuración puede ser modificada por el usuario en caso de ser necesario. Por ejemplo, si coexisten dos o más redes de sensores instaladas en la misma zona, los parámetros elegidos no deben coincidir. De lo contrario se producirá conflictos entre las redes desplegadas.

Modificación de los parámetros

La configuración default de la Red LoRa puede ser modificada y leída a través de un puerto RS232. La configuración del puerto debe ser la siguiente:

- Baud rate 9600.
- Data: 8bits.
- Parity: none.
- Stop Bits: 1.
- Flow Control: none.

Dentro del Juego de Cables se incluye un cable con conector DB9 que se utiliza para conectarse al Módulo.

Dirigirse a la sección **Comandos Configuración Red LoRa** para detalles y ejemplos de los comandos de configuración.

Observación:

Todos los nodos de la red deben tener **exactamente la misma configuración** de los parámetros de la Red LoRa.

Configuración del tipo de Módulo.

Inicialmente los Módulos no están configurados para ser Principal o Secundario.

Análogamente a la Modificación de parámetros, es posible configurar el tipo de Módulo. Debe ser configurado como Módulo Principal aquel que tenga integrado el módulo de comunicación GPRS.

Dirigirse a la sección **Comandos Configuración Red LoRa** para acceder a los comandos de configuración.

Observación:

Dado que el hardware base de los Módulos es el mismo, el tipo no está especificado en la configuración default. Es necesario configurarlo al menos una vez para poder inicializar la red.

Inicialización de sensores

Dirigirse al manual de usuario para inicializar la Red ReSeDEM.

Posibles problemas

Al momento de inicializar un sensor el usuario puede encontrarse con algunos problemas. Los detallamos a continuación y como se podrían solucionar.

Problema 1:

El sensor no enciende led de alimentación.

Posible causa:

Esto se puede deber a un problema en el cable o en el módulo de alimentación.

Posible solución:

Si el Nodo al que está conectado el sensor funciona correctamente probar utilizar otro cable.
Si la falla también afecta al Módulo al que está conectado el sensor, verificar el Módulo de Alimentación.

Problema 2:

El sensor no se inicializa correctamente.

Posible causa:

Si el sensor está siendo reutilizado, es posible que no haya sido eliminado ordenadamente.
Esto hace que la dirección I2C del mismo no sea la necesaria para poder inicializar el sensor desde la web.

Verificación:

Observando la forma en que parpadea el led L del sensor podemos saber en qué estado está.

- Si enciende cada 2 segundos, el sensor se encuentra inicializado.
- Si enciende cada 1 segundo, el sensor esta sin inicializar.
- Si el parpadeo es de más de una vez por segundo, el sensor se encuentra tomando medidas.

Posible solución:

Conectar mediante un cable USB el sensor a un PC. Abrir un terminal serie con la siguiente configuración:

- Baud rate 9600.
- Data: 8bits.
- Parity: none.
- Stop Bits: 1.
- Flow Control: none.

Con el sensor conectado, al presionar el botón “reset”, se imprimirá en pantalla la dirección I2C que tiene grabada el sensor. En la web, el sensor debe ser dado de alta con esa dirección.

Eliminado de sensores

Dirigirse al manual de usuario para eliminar uno o más sensores de la Red ReSeDEM.

Observación:

El sistema permite eliminar muchos sensores a la vez. Es fundamental que, si se eliminan más de un sensor, los eliminados sean desconectados de la red ya que quedarían varios sensores con la dirección default conectados en el mismo bus I2C.

Posibles problemas

Al momento de eliminar un sensor el usuario puede encontrarse con algunos problemas. Los detallamos a continuación y como se podrían solucionar.

Problema 1:

En el LOG se visualiza el siguiente mensaje:

- *Estado 428, Sensor borrado con falla*

Posible causa:

Falla la comunicación LoRa al confirmar la eliminación del sensor.

Posible solución:

Si se tiene acceso al sensor dirigirse a la **Posible solución** del **Problema 2** en la Inicialización del sensor y verificar que la dirección del sensor es el default (30). Si la dirección no es el valor default, conectarlo nuevamente en la red y darlo de alta con la dirección que tiene grabada. Luego inicializar el proceso de eliminación nuevamente.

Si no se tiene acceso, se puede verificar que el sensor se eliminó correctamente al dar de alta un nuevo sensor y verificar que este se inicialice correctamente. Luego inicializar el proceso de eliminación nuevamente.

Problema 2:

En el LOG se visualiza el siguiente mensaje:

- *Estado 424, LoRa ppal: error en transmisión de acciones hacia red LoRa.*

Posible causa:

Falla en la transmisión del mensaje hacia la Red LoRa.

Posible solución:

Esperar a que la Red LoRa solicite acciones. Se va a reintentar ejecutar la acción de eliminar el sensor en cuestión.

Funcionamiento

Dependiendo la de la configuración del Módulo LoRa como Principal o Secundario, el funcionamiento va a ser diferente.

Verificación de alimentación

Para el correcto funcionamiento del Módulo se necesitan dos voltajes, 3,3V y 5V.

- LED3 representa el estado de la alimentación de 3,3V.
- LED6 representa el estado de la alimentación de 5V.

Verificación de funcionamiento

- LED1 debe cambiar de estado cada 0,5 segundos.
Si está siempre encendido o apagado, el programa no se está ejecutando.
Es necesario reprogramar el microcontrolador.
- LED2 Debe estar encendido fijo.
Este led encendido representa que el módulo de radio RN2903 está configurado.

LoRa Principal

Verificación de funcionamiento

- LED5 debe cambiar de estado cuando se recibe un comando desde el Módulo GPRS.
Si no enciende nunca verificar que el modem funcione correctamente.
- LED4 debe cambiar de estado cuando se recibe un comando desde la Red LoRa.
Si no enciende nunca verificar que los LoRa secundarios estén funcionando correctamente.

LoRa Secundario

Verificación de funcionamiento

- LED4 debe cambiar de estado cuando se recibe un comando desde el Módulo LoRa Principal.
Si no enciende nunca verificar que el Módulo LoRa Principal este recibiendo correctamente la solicitud de acciones desde el Módulo GPRS y al menos uno de los sensores con acciones pendientes conectado en el LoRa Secundario.
- LED5 debe cambiar de estado cuando se procesa correctamente el comando recibido desde la Red LoRa.
Si no enciende nunca verificar que los LoRa secundarios estén funcionando correctamente.

Modem

Verificación de funcionamiento

Dirigirse al manual de usuario del Shield GPRS de seeedstudio 3.0.

https://wiki.seeedstudio.com/GPRS_Shield_V3.0/

Comandos Configuración Red LoRa

Command format ASCII														
STX	Command ID	Command Group	Command Type	Command	Length	Value	Checksum	ETX						
						<table><tr><th>STX</th><th>ETX</th></tr><tr><td>02</td><td>03</td></tr></table>			STX	ETX	02	03		
						STX	ETX							
						02	03							
						sys0x01		-	0x00					
								set	0x01					
								get	0x02					
						mac0x02		-	0x00					
								set	0x01					
								set channel						
								get	0x02					
						radio0x03		get channel	0x02					
								-	0x00					
								set	0x01					
								get	0x02					
Example	"radio set bw 500"													
STX	Command ID	Command Group	Command Type	Command	Length	Value	Checksum	ETX						
02	00	03	01	48	01	03	52	03						
		"sys reset"												
STX	Command ID	Command Group	Command Type	Command	Length	Checksum	ETX							
02	00	01	00	02	00	05	03							

	Command Group	Command Type	Command	Length	Value	
//System Commands	0x01	0x00	-	-	-	-
'sys sleep '<length>			0x01	0x04	<0x00000000..0xFFFFFFFF>	<0..4294967295>
'sys reset'			0x02	0x00		
'sys eraseFW'			0x03	0x00		
'sys factoryRESET'			0x04	0x00		
//System Set Commands	0x01	0x01	-	-	-	-
'sys set nvm '<address> <data>			0x05	0x04	<0x300..0x3FF><0x00..0xFF>	
'sys set pindig '<pinname> <pinstate>			0x06	0x03	<0x0000..0xFFFF> <0x00..0x01>	<GPIO0-GPIO13, UART_CTS, UART_RTS, TEST0 and TEST1><0..1>
'sys set pinmode '<pinname> <pinmode>			0x07	0x03		
//System Get Commands	0x01	0x02	-	-	-	-
'sys get ver'			0x08	0x00		
'sys get nvm '<address>			0x09	0x02	<0x300..0x3FF>	
'sys get vdd'			0x0A	0x00		
'sys get hweui'			0x0B	0x00		
'sys get pindig '<pinname>			0x0C	0x01	<0x00..0x12>	
'sys get pinana '<pinname>			0x0D	0x01	<0x00..0x12>	
//MAC Commands	0x02	0x00	-	-	-	-
'mac reset'			0x0E	0x00		
'mac tx '<type> <portno> <data>			0x0F	0x02	<0x01..0x02><0x00..0xDF><data>	<cnf-uncnf><0..223><data>
'mac join '<mode>			0x10	0x01	<0x01..0x02>	<otta..abp>
'mac save'			0x11	0x00		
'mac forceENABLE'			0x12	0x00		
'mac pause'			0x13	0x00		
mac resume'			0x14	0x00		

//MAC Set Commands	0x02	0x01	-	-	-	-
'mac set appkey '<appKey>			0x15	0x10	<16bytes HEX>	
'mac set appskey '<appSessKey>			0x16	0x10	<16bytes HEX>	
'mac set ar '<state>			0x17	0x01	<0x00..0x01>	<off..on>
'mac set bat '<level>			0x18	0x01	<0x00..0xFF>	<0..255>
'mac set class '<class>			0x19	0x01	<0x01..0x02>	<A..C>
'mac set devaddr '<address>			0x1A	0x04	<4bytes HEX>	
'mac set deveui '<devEUI>			0x1B	0x08	<8bytes HEX>	
'mac set dnctr '<FCntDown>			0x1C	0x04	<0x00000000..0xFFFFFFFF>	
'mac set dr '<dataRate>			0x1D	0x01	<0x00..0x04>	
'mac set linkchk '<linkCheck>			0x1E	0x02	<0x0000..0xFFFF>	<0..65535>
'mac set mcast '<state>			0x1F	0x01	<0x00..0x01>	<off..on>
mac set mcastappskey '<mcastApplicationSessionkey>			0x20	0x10	<16bytes HEX>	
'mac set mcastdevaddr '<mcastAddress>			0x21	0x04	<4bytes HEX>	
mac set mcastdnctr '<fMcastCntDown>			0x22	0x04	<0x00000000..0xFFFFFFFF>	<0..4294967295>
'mac set mcastnwkskey '<mcastNetworkSessionkey>			0x23	0x10	<16bytes HEX>	
'mac set nwkskey '<nwkSessKey>			0x24	0x10	<16bytes HEX>	
'mac set pwrldx '<pwrIndex>			0x25	0x01	<0x05..0x0A>	
'mac set retx '<reTxNb>			0x26	0x01	<0x00..0xFF>	
'mac set rx2 '<dataRate> <frequency>			0x27	0x06	<0x08..0x0D><0x370870A0..0x374886E0>	<923300000..927500000>
'mac set rxdelay1 '<rxDelay>			0x28	0x02	<0x0000..0xFFFF>	<0..65535>
'mac set sync '<synchWord>			0x29	0x01	<0x00..0xFF>	
'mac set upctr '<fCntUp>			0x2A	0x04	<0x00000000..0xFFFFFFFF>	<0..4294967295>

//MAC Get Commands	0x02	0x02	-	-	-	-
'mac get adr'			0x2B	0x00		
mac get appeui'			0x2C	0x00		
'mac get ar'			0x2D	0x00		
'mac get class'			0x2E	0x00		
mac get dcycleps'			0x2F	0x00		
mac get devaddr'			0x30	0x00		
mac get deveui'			0x31	0x00		
mac get dnctr'			0x32	0x00		
'mac get dr'			0x33	0x00		
'mac get gwnb'			0x34	0x00		
'mac get mcast'			0x35	0x00		
mac get mcastdevaddr'			0x36	0x00		
mac get mcastdnctr'			0x37	0x00		
'mac get mrgn'			0x38	0x00		
mac get pwridx'			0x39	0x00		
'mac get retx'			0x3A	0x00		
'mac get rx2'			0x3B	0x00		
mac get rxdelay1'			0x3C	0x00		
'mac get rxdelay2'			0x3D	0x00		
'mac get status'			0x3E	0x00		
'mac get sync'			0x3F	0x00		
'mac get upctr'			0x40	0x00		
//Radio Commands	0x03	0x00	-	-	-	-
'radio rx '<rxWindowSize>			0x41	0x02	<0x00..0xFF><0x00..0xFF>	0..65535
'radio tx '<data>			0x42	0x00	<data>	
radio cw '<state>			0x43	0x01	<0x00..0x01>	<off..on>
'rxstop'			0x44	0x00		

//Radio Set Commands	0x03	0x01	-	-	-	-
'radio set afcbw '<autoFreqBand>			0x45	0x04	float	
'radio set bitrate '<fskBitrate>			0x46	0x03	<0x01..0x493E0>	<1..300000>
'radio set bt '<gfbT>			0x47	0x01	<0x01..0x03>	<1.0, 0.5, 0.3>
'radio set bw '<bandWidth>			0x48	0x01	<0x01..0x03>	125,250,500
'radio set cr '<codingRate>			0x49	0x01	<0x01..0x04>	4/5, 4/6, 4/7, 4/8
'radio set crc '< crcHeader >			0x4A	0x01	<0x00..0x01>	<off..on>
'radio set fdev '<freqDev>			0x4B	0x03	<0x01..0x30D40>	<1..200000>
radio set freq '<frequency>			0x4C	0x04	<0x35C36D80..0x37502800>	902000000..928000000 en Hz
'radio set iq i'<iqInvert>			0x4D	0x01	<0x00..0x01>	<off..on>
'radio set mod '<mode>			0x4E	0x01	<0x00..0x01>	<LoRa..fsk>
'radio set prlen '<preamble>			0x4F	0x02	<0x00..0xFF><0x00..0xFF>	0..65535
'radio set pwr '<pwrOut>			0x50	0x01	<0x02..0x14>	2..20
'radio set rxbw '<rxBandwidth>			0x51	0x04	float	
'radio set sf '<spreadingFactor>			0x52	0x01	<0x07..0x0C>	7..12
'radio set sync '<syncWord>			0x53	0x08	<8bytes HEX>	
'radio set wdt '<watchDog>			0x54	0x04	<0x00000000..0xFFFFFFFF>	0..4294967295
//Radio Get Commands	0x03	0x02	-	-	-	-
'radio get afcbw'			0x55	0x00		
'radio get bitrate'			0x56	0x00		
'radio get bt'			0x57	0x00		
'radio get bw'			0x58	0x00		
'radio get cr'			0x59	0x00		
'radio get crc'			0x5A	0x00		
'radio get fdev'			0x5B	0x00		
'radio get freq'			0x5C	0x00		
'radio get iq i'			0x5D	0x00		
'radio get mod'			0x5E	0x00		
'radio get prlen'			0x5F	0x00		

'radio get pwr'			0x60	0x00	
'radio get rssi'			0x61	0x00	
'radio get rxbw'			0x62	0x00	
'radio get sf'			0x63	0x00	
'radio get snr'			0x64	0x00	
'radio get sync'			0x65	0x00	
radio get wdt'			0x66	0x00	
//MAC Set Channel Commands					
	0x02	0x01	-	-	-
mac set ch drrange' <channelID> <minRange> <maxRange>			0x67	0x03	<0..71><0..3><0..3>
mac set ch status'<channelID> <status>			0x68	0x02	<0..71><off..on>
//MAC Get Channel Commands					
	0x02	0x02	-	-	-
mac get ch freq' <channelID>			0x69	0x01	
mac get ch drrange' <channelID>			0x6A	0x01	
mac get ch status' <channelID>			0x6B	0x01	
//CONFIGURE NODE					
	0x08				
node type		0x00	0x6C	0x01	<0x01..0x02> 0x01 Master 0x02 Slave
//SAVE CONFIGURATION					
save	0x04	0x00	0x00	0x00	Save actual configuration
//DELETE CONFIGURATION					
delete	0x05	0x00	0x00	0x01	If command = 0xff, delete all saved config else delete only command
//LOAD DEFAULT CONFIGURATION					
default	0x06	0x00	0x00	0x00	Commands in red (delete alctual config and save default parameters)
//REBOOT					
reboot	0x07	0x00	0x00	0x00	Reboot device without saving

LoRa Node Commands	
NODE TYPE	
principal	020008006C01017803
secondary	020008006C01027903
save	
	0200040000000603
delete "cm"	
	020005000001cmcs03
delete all	
	020005000001FF0703
default	
	0200060000000803
reboot	
	0200070000000903
RN2903 Commands	
sleep	0200010001041194D7FF8303
sys reset	0200010002000503
radio set bw 500	020003014801035203
radio set cr 4/5	020003014901015103
radio set pwr 20	020003015001146B03
radio set sf 7	020003015201076003
radio set wdt 0	02000301540400000005E03
radio set freq 915000000	020003014C043689CAC09F03
mac pause	0200020013001703
radio rx 0	02000300410200004803
sys set pingdig GPIO12 1	020001010603000C011A03
radio get freq	020003025C006303
radio get cr	0200030259006003
radio get sf	0200030263006A03
radio get pwr	0200030260006703

Índice de tablas

1.1. Correspondencia Pin - Función del Bus de datos/alimentación . . .	3
4.1. Tecnologías para acceso a internet	18
4.2. Comandos AT para conexión a servidor WEB [7]	21
4.3. Comparativa Arduinos	22
5.1. Características del chip RN2903 [12]	29
5.2. Principales comandos de configuración LoRa [13]	30
5.3. Principales características del microcontrolador elegido [14]	31
8.1. Pinout - Conector RS232	48
8.2. Mensaje del servidor solicitando acciones - ServerDOWN1	51
8.3. Mensaje hacia el servidor - ServerUP	51
8.4. Respuesta del servidor informando minutos de la próxima solicitud - ServerDOWN2	52
8.5. Estado general del sistema	54
10.1. Consumos medios por ciclo LoRa Principal + GPRS	65
10.2. Consumo medio en un ciclo LoRa secundario	66
10.3. Consumo medio Módulo Sensor	67
10.4. Eventos registrados en la prueba de Tasa de fallas	68
10.5. Fallas por sensor	69
A.1. Valores típicos de SF y chirps por símbolo	81
A.2. Bitrate y Sensibilidad en función del BW y SF para un CR de 4/7 [23] y [22]	82
B.1. Lista componentes de la placa ReSeDEM del Módulo LoRa	85
C.1. Comandos AT adicionales	87
D.1. Baterías para autonomía 1 semana. fs=2 minutos	89
D.2. Baterías para autonomía 1 semana. fs=10 minutos	90
D.3. Baterías para autonomía 1 semana. fs=15 minutos	90

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

2.1. Descripción de la red	5
2.2. Esquema de la arquitectura del sistema	7
2.3. Ejemplo de funcionamiento del sistema	8
3.1. Estructura de la base de datos	13
3.2. Script consulta.php	14
3.3. Script proc.php	15
4.1. Cobertura ISP ANTEL - Inicio proyecto (Izq), Actualidad (Der) [10]	19
4.2. Hardware Módulo GPRS [11]	20
4.3. Secuencia de funcionamiento Módulo GPRS	23
4.4. Maquinas de estado del Módulo GPRS	24
5.1. Hardware principal del Módulo LoRa	32
5.2. Módulos LoRa v1.0	32
5.3. Módulos LoRa v2.0	33
5.4. Módulos LoRa v3.0	34
5.5. Secuencia funcionamiento del Módulo LoRa del Nodo Principal . .	37
5.6. Secuencia funcionamiento del Módulo LoRa del Nodo N	38
6.1. Sensor DT11	40
6.2. Sensor de humedad de suelo	41
6.3. Secuencia funcionamiento Módulo Sensor y estados	42
7.1. Topologías de Red	43
7.2. Red LoRa	46
8.1. Trama mensaje UART	47
8.2. Conexion para una comunicación UART	48
8.3. Bus I2C [18]	49
8.4. Ejemplo de una comunicación I2C y las señales generadas en las líneas SDA y SCL [19]	50
8.5. Mensajes: Servidor - GPRS	52
8.6. Mensajes: GPRS-LoRa	53
8.7. Mensajes: Nodo - I2C - Sensor	54
8.8. Mensajes extremo a extremo - Servidor - GPRS - LORA- I2C . . .	55

Índice de figuras

9.1. Conector Gx16-7p (izq), Gx16-5p (der)	57
9.2. Módulo LoRa-GPRS, Nodo Principal	58
9.3. Módulo LoRa, Nodos Secundarios	59
9.4. Módulo Sensor	59
9.5. Sistema ensamblado: Nodo Ppal y Módulo Sensor	60
9.6. Sistema ensamblado: Nodo Ppal y Módulo Sensor	60
10.1. Primer prueba alcance	62
10.2. Prueba alcance Sistema final	63
10.3. Prueba alcance máxima Sistema final	63
10.4. Consumo Módulo LoRa Principal + Módulo GPRS (azul), Consumo Medio (rojo)	65
10.5. Consumo Módulo LoRa Secundario (azul), Consumo Medio (rojo)	66
10.6. Consumo Módulo Sensor	67
A.1. Modulación de frecuencia expandida [21]	80
A.2. Espectro de una transmisión LoRa [22]	81
B.1. Cad del PCB del módulo de comunicación	83
B.2. Esquemático de la placa ReSeDEM	84