

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Implementación de un prototipo de sistema automatizado de riego en un segmento de plantación de cacao en la Finca Florentino, recinto El Paraíso,
Milagro

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telecomunicaciones

Presentado por:

Fernando Gabriel Maldonado Narváez

Fernando Lautaro Sinchiguano Sabando

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

El presente proyecto quiero dedicarlo a Dios, por protegerme en cada momento y guiarme a lo largo de toda mi carrera para cumplir con cada uno de mis objetivos y cuidar a toda mi familia para poder estar junto a ellos.

A mis padres, Cesar Sinchiguano y Yolanda Sabando les dedico esta meta por siempre estar a mi lado y por todo el sacrificio que hicieron en todos estos años y las enseñanzas que me dieron para convertirme en todo un profesional.

A mis hermanos, Tania, Carlos, Milton y Claudio, por ser ejemplos de dedicación ante cualquier adversidad y por apoyarme en toda mi vida estudiantil.

A mi cuñado Sídney, a mi cuñada Melissa, a mi ñaña Kelly y a mis sobrinas Andrea y Carla, por su atención y tiempo en ayudarme en cualquier momento.

Fernando Lautaro Sinchiguano Sabando

Este proyecto quiero dedicarlo en primer lugar a Dios, quien provee en mi camino, para que mi vida profesional esté bajo su dirección.

A mi madre Inés que con paciencia ha sabido guiarme en cada paso de mi vida y por todo el sacrificio que ha significado sacar adelante a su familia.

A mi padre Julio, por brindarme siempre su amor incondicional y ser mi modelo de fidelidad a Dios, instruyéndome en la fe. Que brille para él la luz perpetua.

Fernando Gabriel Maldonado Narváez

AGRADECIMIENTOS

Agradezco a Dios por tener una buena familia y por cuidar a cada uno de ellos en cada momento y por siempre a ser que me supere como hijo, hermano, persona y ahora como un profesional.

A mi padre Cesar, por siempre darme consejos, enseñanzas e inculcarme principios y valores para forjar mi carácter, aprendí de ti a ser constante, perseverante y cumplir con todos mis objetivos, agradezco tener un padre maravilloso como tú que siempre se ha preocupado por toda su familia para que no les falte nada.

A mi madre bella Yolanda te agradezco por ser una madre maravillosa por siempre estar ahí dándome ánimos y apoyándome en todo, aprendí de ti a que siempre hay soluciones para cualquier problema y a nunca rendirme, eres el ejemplo de personalidad amable, dedicada y amorosa que siempre se ha preocupado por sus hijos, esposo y nietos, te agradezco mucho madre bella todo lo que haces por nosotros.

A mi ñaña Tania, por ser el pilar en la familia y por ser la motivación y el ejemplo a seguir y a nunca dejarnos vencer por adversidades .Te agradezco mucho ñaña por ser la mejor hermana que puedo tener porque sin ti esta meta que he cumplido no sería posible.

A mis hermanos Carlos, Milton, Claudio y a mi cuñado Sídney que es como un hermano más, les agradezco por estar siempre presentes y por cada consejo que me dieron para poder seguir adelante y cumplir con esta meta.

A mis sobrinas Andrea y Carla que me llenan de alegría cada día de mi vida. A mi tía Kelly que a pesar de la distancia ha presenciado toda mi trayectoria y a mi cuñada Melissa que es como una hermana más en mi familia, les agradezco a cada una de ellas por todo el cariño y motivación que me han dado.

Fernando Lautaro Sinchiguano Sabando

Agradezco a mi Dios por las bendiciones que he recibido durante mi tiempo en la carrera, por la familia que me ha regalado, por no dejarme flaquear en los momentos duros.

A mi madre Inés que siempre tuvo su esperanza firme, que me enseñó a siempre terminar todo proyecto que empezaba, que con su esfuerzo y dedicación me apoyó para poder convertirme en la persona que soy.

A mi hermana Mayra que con su ejemplo me inspiró a ser mejor cada día. Le agradezco por sus palabras de aliento en las situaciones difíciles, cada consejo que recibí y sobre todo por dedicarse enteramente al bienestar de nuestra familia.

Agradezco a mis amigos Alex, Nixon y Andrea por el ánimo que me han dado, por el apoyo incondicional que he recibido de cada uno.

De manera muy especial agradezco a CJ por ser mi soporte y mi empuje para continuar con mis estudios. Gracias por estar en esos momentos que tanto necesitaba tu compañía.

Fernando Gabriel Maldonado Narváez

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Fernando Gabriel Maldonado Narváez y Fernando Lautaro Sinchiguano Sabando damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Fernando Gabriel
Maldonado Narváez



Fernando Lautaro
Sinchiguano Sabando

EVALUADORES

PhD. Francisco Novillo Parales

PROFESOR DE LA MATERIA

Ing. Luis Vásquez Vera, MSc.

. PROFESOR TUTOR

RESUMEN

El cultivo de cacao requiere condiciones de suelo y de ambiente específicas para su desarrollo y producción. El problema que se presenta en esta plantación, es el control ineficiente en el riego del terreno, ya que se recurre únicamente a la experticia del agricultor para determinar en qué momento se debe realizar el riego de los sembríos. Esto provoca que las condiciones de humedad, tanto del suelo como del ambiente, no se mantengan constantes y uniformes, por lo que la productividad se ve disminuida. Además el riego excesivo permite la presencia de plagas y enfermedades en las plantas, tales como la Monilla y Mazorca Negra.

Para poder resolver este problema se propuso un prototipo de sistema que permite realizar el riego de la plantación de forma automática. En este prototipo se implementaron tres nodos periféricos con sensores para la medición de variables climáticas de la plantación y un nodo principal que controla el sistema. La comunicación que se estableció entre los nodos fue a través del protocolo de conexión inalámbrica Zigbee.

Los resultados derivados de la aplicación de este sistema mostraron una regulación eficiente en el sistema de monitoreo y control de riego, ya que las lecturas tomadas luego de la implementación del sistema indicaron una constancia y uniformidad de las variables medidas en el terreno.

La WSN implementada para el control de las variables que inciden en la producción de la plantación de cacao permite al agricultor ahorrar agua en el riego, evitar enfermedades y plagas en las plantas y mejorar la producción de la plantación.

Palabras Clave: Sistema de riego, Xbee, Proyecto Integrador, Zigbee.

ABSTRACT

Cocoa cultivation requires specific soil and environmental conditions for its development and production. The problem that arises in this plantation is the inefficient control in the irrigation of the land, since only the farmer's expertise is used to determine when the crop should be irrigated. This causes humidity conditions, both in the soil and in the environment, to not remain constant and uniform, so productivity is reduced. In addition, excessive irrigation allows the presence of pests and diseases in plants, such as Monilla and Mazorca Negra.

In order to solve this problem, a prototype system was proposed that allows the irrigation of the plantation to be carried out automatically. In this prototype, three peripheral nodes with sensors for the measurement of climatic variables of the plantation and a main node that controls the system were implemented. The communication that was established between the nodes was through the Zigbee wireless connection protocol.

The results derived from the application of this system showed an efficient regulation in the irrigation monitoring and control system, since the readings taken after the implementation of the system indicated a constancy and uniformity of the variables measured in the field.

The WSN implemented to control the variables that affect the production of the cocoa plantation allows the farmer to save water in irrigation, avoid diseases and pests in the plants and improve the production of the plantation.

Keywords: Irrigation system, Xbee, Integrative Project, Zigbee.

ÍNDICE GENERAL

EVALUADORES.....	8
RESUMEN.....	9
<i>ABSTRACT</i>	10
ÍNDICE GENERAL.....	11
ABREVIATURAS.....	14
ÍNDICE DE FIGURAS.....	15
ÍNDICE DE TABLAS.....	16
ÍNDICE DE ANEXOS.....	17
CAPÍTULO 1.....	18
1. Introducción.....	18
1.1 Antecedentes.....	19
1.2 Objetivos.....	20
1.2.1 Objetivo General.....	20
1.2.2 Objetivos Específicos.....	20
1.3 Justificación.....	20
CAPÍTULO 2.....	21
2. Marco teórico.....	21
2.1 Estándar IEEE 802.15.4.....	21
2.2 Protocolo Zigbee.....	22
2.2.1 Topología en malla.....	23
CAPÍTULO 3.....	24
3. DISEÑOS de la solución.....	24
3.1 Descripción del sistema.....	24
3.1.1 Obtención de datos.....	24
3.1.2 Módulo de transmisión.....	24

3.1.3	Módulo de recepción	25
3.1.4	Procesamiento	25
3.1.5	Diagrama de funcionamiento del sistema.....	25
3.2	Dispositivos y Protocolos seleccionados	26
3.2.1	Implementación en Modo API.....	26
3.2.2	Aplicación multiplataforma XCTU	26
3.2.3	Módulo Xbee S2C.....	27
3.2.4	Sensor de humedad de suelo HD-38.....	28
3.2.5	Sensor DHT22.....	29
3.2.6	Reloj RTC DS3231	29
3.2.7	Módulo TP4056.....	30
3.2.8	Módulo elevador de voltaje Mt3608.....	30
3.2.9	Panel Solar Innovatech	31
3.2.10	Batería Rakieta 18650 Li-ion recargable.....	31
3.2.11	Adaptador Xbee para Protoboard	32
3.3	Diseño de la WSN	32
3.3.1	Configuración de la WSN	32
3.3.2	Prueba de comunicación entre los nodos.....	33
3.3.3	Ubicación de los nodos en el terreno	34
3.4	Diseño de Nodo Principal y Periféricos.....	34
3.4.1	Diseño de Nodo Principal	34
3.4.2	Diseño de Nodo Periférico	35
3.5	Diseño del sistema de alimentación	36
3.6	Diseño del sistema de riego	37
CAPÍTULO 4.....		38
4.	Análisis de resultados.....	38

4.1	Análisis de datos de humedad del suelo de los Nodos periféricos	38
4.2	Análisis de datos de humedad del ambiente de los Nodos Periféricos	40
4.3	Análisis de datos de temperatura del ambiente de los nodos periféricos	42
CAPÍTULO 5.....		44
5.	Costos del proyecto.....	44
5.1	Análisis Económico	44
5.2	Gastos materiales.....	44
5.3	Análisis de Mantenimiento del Nodo Periférico	45
CAPÍTULO 6.....		46
6.	Conclusiones Y Recomendaciones.....	46
	Conclusiones.....	46
	Recomendaciones	46
BIBLIOGRAFÍA.....		48
7.	Bibliografía.....	48
PLANOS Y ANEXOS		49

ABREVIATURAS

API Application Programming Interface
ESPOL Escuela Superior Politécnica del Litoral
FFD Full Function Device
IEEE Institute of Electrical and Electronics Engineers
ISM Industrial, Scientific, and Medical
LR-WPAN Low-Rate Wireless Personal Area Networks
OSI Open Systems Interconnection Model
PAN Personal Area Network
RF Radio Frequency
RFD Reduced Function Device
WSN Wireless Sensor Networks
XCTU XBee Configuration and Test Utility

ÍNDICE DE FIGURAS

Figura 3.1 Diagrama de funcionamiento del Sistema	26
Figura 3.2 Dispositivos detectados y parámetros de configuración X-CTU.....	27
Figura 3.3 Red sensores tipo malla	33
Figura 3.4 Diagrama del Sistema del Nodo Principal.....	35
Figura 3.5 Diagrama del Sistema del Nodo Periférico.....	36
Figura 3.6 Diagrama del Sistema de Alimentación.....	37
Figura 3.7 Diagrama del Sistema de Riego.....	37
Figura 4.1 Gráfica de Humedad del Suelo del Nodo 1	39
Figura 4.2 Gráfica de Humedad del Suelo del Nodo 2	39
Figura 4.3 Gráfica de Humedad del Suelo del Nodo 3	40
Figura 4.4 Gráfica de Humedad del Ambiente del Nodo 1.....	41
Figura 4.5 Gráfica de Humedad del Ambiente del Nodo 2.....	41
Figura 4.6 Gráfica de Humedad del Ambiente del Nodo 3.....	41
Figura 4.7 Gráfica de Temperatura del Nodo 1	43
Figura 4.8 Gráfica de Temperatura del Nodo 2	43
Figura 4.9 Gráfica de Temperatura del Nodo	43

ÍNDICE DE TABLAS

Tabla 2.1 Banda de operación del Estándar IEEE 802.15.4	21
Tabla 3.1 Especificaciones técnicas del Xbee S2C [1]	28
Tabla 3.2 Especificaciones técnicas del Sensor de Humedad del Suelo HD-38 [4]	29
Tabla 3.3 Especificaciones técnicas del Sensor DHT22 [5]	29
Tabla 3.4 Especificaciones técnicas del Módulo DS3231 [6].....	30
Tabla 3.5 Especificaciones técnicas del Módulo TP4056 [2]	30
Tabla 3.6 Especificaciones técnicas de Módulo Elevador De Voltaje Mt3608	31
Tabla 3.7 Especificaciones técnicas de Panel Solar 5V [3]	31
Tabla 3.8 Especificaciones técnicas de Batería Rakieta 18650 Li-ion Recargable [4].....	32
Tabla 3.9 Especificaciones técnicas de Adaptador Xbee para Protoboard	32
Tabla 3.10 Parámetros de referencia para la activación del sistema de riego	35
Tabla 5.1 Cuadro general de costos de inversión.....	44
Tabla 5.2 Cuadro de costos de mantenimiento de Nodo Periférico	45
Tabla 5.3 Cuadro de costos de mantenimiento del Nodo Principal	45

ÍNDICE DE ANEXOS

Anexo No. 7-1 Xbee S2C [1]	49
Anexo No. 7-2 Sensor HD-38 [4]	49
Anexo No. 7-3 Sensor DHT22 [5]	50
Anexo No. 7-4 Módulo de Reloj DS3231 [6]	50
Anexo No. 7-5 Módulo TP4056 [2]	51
Anexo No. 7-6 Módulo Elevador De Voltaje Mt3608 [6]	51
Anexo No. 7-7 Panel Solar Innovatech [3]	52
Anexo No. 7-8 Batería Rakieta 18650 Li-ion Recargable [4].....	52
Anexo No. 7-9 Adaptador Xbee para Protoboard [5]	53
Anexo No. 7-10 Intensidad de señal red ZigBee con una distancia de 8 metros del Nodo 1 hacia al Nodo Coordinador.....	53
Anexo No. 7-11 Intensidad de señal red ZigBee con una distancia de 8 metros del Nodo 2 hacia al Nodo Coordinador.....	54
Anexo No. 7-12 Intensidad de señal red ZigBee con una distancia de 8 metros del Nodo 2 hacia al Nodo Coordinador.....	54
Anexo No. 7-13 Intensidad de señal red ZigBee con una distancia de 20 metros del Nodo 1 hacia al Nodo Coordinador.....	55
Anexo No. 7-14 Intensidad de señal red ZigBee con una distancia de 20 metros del Nodo 2 hacia al Nodo Coordinador.....	55
Anexo No. 7-15 Intensidad de señal red ZigBee con una distancia de 20 metros del Nodo 3 hacia al Nodo Coordinador.....	56
Anexo No. 7-16 Diseño del Nodo Periférico	56
Anexo No. 7-17 Diseño del Nodo Central	57
Anexo No. 7-18 Diseño de placa PCB del Nodo periférico.....	57
Anexo No. 7-19 Diseño de placa PCB del Nodo central	58
Anexo No. 7-20 Código General para los Nodos Periféricos	59
Anexo No. 7-21 Código General para el Nodo Central	59

CAPÍTULO 1

1. INTRODUCCIÓN

El cacao es uno de los cultivos puntales dentro del sector agrícola, tomando como referencia las cifras de exportación en el 2021, donde se exportó 360.714 toneladas de cacao y semielaborados, generando un ingreso de 940 millones de USD al país a pesar de que el valor del cacao por tonelada cayó en un 1.6% [1].

América Latina es considerada la cuna del cacao. Se han encontrado registros de huellas de ADN que sitúan el ancestro del cacao de hace 5000 años dentro América del sur [1]. La producción de la variedad de cacao nacional viene de una cepa cuyo ancestro ha sido determinado en Zamora Chinchipe, siendo inicialmente la variedad que identifica a la producción nacional, cuyo negocio fue rentable desde su inicio por parte de los españoles en el siglo XVI [1]. Sin embargo, debido a las plagas que azotaban el cultivo de cacao se diseñaron variedades injertadas que presentaban mayor resistencia. De estas variedades resulta el cacao CCN51 [1] que en sus inicios fue rechazado por la acidez, pero que ahora se destaca por la gran productividad y la calidad del grano, constituyendo más de la mitad de la producción nacional.

Ecuador es uno de los países de mayor exportación de cacao en grano siendo el primero en América y tercero a nivel mundial, después de Costa de Marfil y Ghana [1]. El sector cacaotero representa el 5% de la PEA a nivel nacional y el 15% de la PEA rural, principalmente en la zona costera del país, aportando un ingreso económico promedio de 800 millones de dólares al año para el país. Los pequeños productores de cacao constituyen el 90% de la producción, donde se aplica mayormente el conocimiento empírico para el cultivo, siendo el 10% restante, conformado por grandes productores [1].

Por tal razón en el presente documento se propone un prototipo de sistema automatizado

de riego en plantaciones de cacao CCN-51 con el objetivo de optimizar costos durante el cultivo y poder mejorar la producción. Dentro de la zona agrícola de la Finca Florentino, en la parroquia Roberto Astudillo, del cantón Milagro se implementará un sistema que permita realizar lecturas de valores de parámetros del suelo mediante el uso de una red de sensores desplegados por el terreno y poder procesar los datos para determinar los parámetros en los cuales es necesario el riego del suelo.

El presente documento se desarrolla de la siguiente forma: En el primer capítulo se detalla los fundamentos y conceptos teóricos sobre la producción de cacao CCN51 y la condiciones tomadas en cuenta en la implementación del proyecto; en el segundo capítulo se explica la metodología aplicada en el proyecto y las alternativas consideradas; en el tercer capítulo se esquematiza el diseño de la red de nodos sensores, los detalles técnicos de cada dispositivo vinculado y las etapas establecidas del proyecto; en el cuarto capítulo se analiza las mediciones de los nodos y la respuesta del sistema implementado; en el quinto capítulo se detalla el presupuesto aproximado de la implementación que permita determinar la rentabilidad del proyecto; en el sexto capítulo se exponen las conclusiones del proyecto y la recomendaciones para su aplicación.

1.1 Antecedentes

Dentro de la zona agrícola del recinto El Paraíso del cantón Milagro el cultivo de cacao es bastante extenso, siendo uno de los productos agrícolas principales. Uno de los problemas que se presenta en estos cultivos es un control ineficiente en el riego del terreno, recurriendo únicamente a la experticia del agricultor para determinar los momentos de riego de los sembríos. Esto ha provocado que las condiciones de humedad del suelo y de ambiente, no se mantengan constantes y uniformes, por lo que la productividad se ha visto disminuida.

1.2 Objetivos

1.2.1 Objetivo General

Implementar un prototipo de sistema automatizado de riego en la plantación de cacao en la Finca Florentino, en la parroquia Roberto Astudillo, del cantón Milagro.

1.2.2 Objetivos Específicos

- 1) Diseñar la red de nodos sensores para la lectura de variables.
- 2) Desplegar el sistema de comunicación y control de riego en el terreno.
- 3) Evaluar los datos recopilados para el análisis histórico.

1.3 Justificación

El cultivo de cacao es uno de los grandes sectores agrícolas que más ha aportado a la economía del país por lo que su cultivo se ha extendido, principalmente, a lo largo de la costa ecuatoriana [1]. Su alta demanda en el mercado internacional contribuye al desarrollo de pequeños y medianos productores. Los sembríos de cacao requieren cuidado para poder maximizar la producción y dentro de este cuidado se tiene el riego del terreno para mantener la humedad del suelo que necesita la planta de cacao [1]. Un suelo muy seco o con exceso de humedad puede afectar el crecimiento y producción de las mazorcas de cacao, así como la salud de la planta, lo cual se traduce en pérdidas económicas significativas para el agricultor debido a las formas tradicionales, sin sustento científico, que se emplea en los cultivos, lo cual no puede garantizar condiciones óptimas para el riego [1]. Por lo tanto, un sistema de monitoreo y automatización de riego es necesario para mejorar las condiciones del cultivo.

En este proyecto el beneficiario directo sería el agricultor propietario de la Finca Florentino, en el recinto “El Paraíso”, Milagro; sin embargo se podrá reproducir este sistema en otro sembrío con características similares.

CAPÍTULO 2

2. MARCO TEÓRICO

En este capítulo se presentan a detalle los conceptos y definiciones que fundamentan la propuesta de solución. Además, se muestra el esquema de cada uno de los procesos involucrados como la recolección de datos de los nodos, envío y recepción de estos datos, y el procesamiento de los mismos.

2.1 Estándar IEEE 802.15.4

Este estándar define una forma de acceso a redes inalámbricas de área personal LR-WPAN, es decir una red con una tasa de transmisión de datos baja. El objetivo de este estándar es establecer los niveles básicos de la red para poder brindar servicio a una red inalámbrica de área personal, estableciendo una comunicación entre los dispositivos de la red con un consumo bajo y una infraestructura mínima. El protocolo opera a nivel de capa 2, según el modelo OSI, en la parte de enlace de datos, donde se gestiona los bits de información que van a ser transmitidos por medio de impulsos electromagnéticos desde los dispositivos en la red [2].

Entre las características que definen al estándar 802.15.4 hay una cualidad fundamental que es la sencillez tecnológica, la cual minimiza los costos en la fabricación de dispositivos que funcionan bajo este estándar sin afectar la adaptabilidad de los mismos con las demás tecnologías. Además con el estándar 802.15.4, un dispositivo puede emplear bandas de frecuencia no licenciadas para la transmisión [2], tal como se detalla en la tabla 2.1.

Tabla 2.1 Banda de operación del Estándar IEEE 802.15.4 [2]

Frecuencia (MHz)	Bit rate (Kb/s)	Canales	Modulación	Región
868 – 868.6	20	3	BPSK	EUROPA
902 – 928	40	30	BPSK	EEUU
2400 – 2483.5	250	16	O-QPSK	MUNDIAL

Debido a que la banda de 2.4GHz es una banda libre [2], se tiene una extensa gama de tecnologías que la emplean, como WiFi y Bluetooth. Se pueden presentar interferencias no intencionadas producto de transmisiones en la banda. Sin embargo, existen mecanismos en el estándar 802.15.4 que permiten seleccionar el canal de comunicación de forma dinámica de tal manera que se escanean los canales por medio de la capa Mac para identificar y formar la red mediante la capa física, detectando la viabilidad de canales de comunicación dentro de la banda y la potencia de recepción. De esta manera se define el canal de operación y el cambio, en caso de saturación o desconexión [2].

Dentro del estándar IEEE 802.15.4 se definen 2 clases de nodos de una red [2]:

- **Nodo FFD (Full Function Device)**
 - Funciona como un coordinador dentro de la red PAN, ya sea dentro de un área de la red o para toda la red, y como un nodo limitado también.
 - Facilita un modelo de la red que permita la adición, disminución e intercambio de nodos en la red.
 - Funciona como un enrutador, para la transmisión de mensajes de los nodos RFD.
- **Nodo RFD (Reduced Function Device)**
 - Funcionan como recolectores de datos o ejecutores, con comunicación limitada únicamente a su nodo coordinador FFD.
 - No pueden ser coordinador, ni controlar otros RFDs.

2.2 Protocolo Zigbee

El conjunto de protocolos Zigbee se define bajo el estándar 802.15.4 que permite establecer conexiones inalámbricas entre dispositivos de diferentes fabricantes a distancias cortas con una velocidad relativamente baja [3]. Fue desarrollado en el año 2003 y finalmente fue publicado en el año 2005. Dentro de las características que presenta Zigbee se destaca el bajo consumo energético, útil para sistemas con fuentes de alimentación independiente ya que prolonga una vida útil de la batería, menor costo en los dispositivos

pudiendo manejar dentro de una sola red hasta 65536 dispositivos; además trabaja en frecuencias de 868 MHz a 915 MHz y 2.4 GHz con una cobertura de 10 a 75 metros [3].

Dentro de las aplicaciones de este protocolo se encuentra la domótica en diferentes áreas como la seguridad, control y automatización de procesos como la iluminación y climatización; además se puede utilizar en el control de acceso y monitoreo ambiental. [7]

El protocolo Zigbee, en relación con el estándar IEEE 802.15.4 se tomará en la banda de frecuencias ISM en torno a 2.4 GHz, dentro del proyecto en cuestión para la transmisión de datos en la red de nodos sensores.

2.2.1 Topología en malla

Zigbee emplea diversas topologías para poder realizar la transmisión de datos entre los nodos de la red utilizando en algunos casos nodos intermedios que hacen la función de ruteadores, es decir que retransmiten la información desde otro nodo por medio de la capa de red. Entre las topologías más comunes se encuentran la topología de árbol, topología estrella y topología de malla, además combinaciones entre sí [3].

La topología de malla comprende un esquema menos sencillo de una red Zigbee pero con una mayor eficiencia, la cual comprende un nodo central conectado directamente a uno o más nodos menores que a su vez pueden tomar la función de enrutadores para otros nodos. Esta topología presenta un costo mayor para alcanzar la comunicación en la red y se limita a las longitudes de los nodos; sin embargo cuenta con la ventaja de una fácil escalabilidad ya que se puede agregar múltiples nodos en su periferia sin afectar la configuración de los demás nodos, además presenta una gran facilidad para implementar su configuración [7].

CAPÍTULO 3

3. DISEÑOS DE LA SOLUCIÓN

3.1 Descripción del sistema

La propuesta de solución consiste en la implementación de un sistema que realice lecturas de las condiciones, tanto de suelo como ambiente, para mejorar el riego del terreno, manteniendo parámetros constantes para el cultivo. El sistema presenta una serie de procesos de recolección, envío y procesamiento de datos para su funcionamiento.

3.1.1 Obtención de datos

El módulo sensor de humedad relativa del suelo va a presentar valores que pueden variar dentro de un rango entre 0 y 1023 acorde a la humedad que presente el suelo. Si presenta un 100% de humedad tendrá el valor mínimo de 0 mientras que si presenta un 0% de humedad tendrá un valor máximo de 1023 [4].

En el caso del sensor de temperatura y humedad del ambiente se presenta la medición mediante una sola conexión directa al módulo central del nodo y se trabaja con librerías propias de arduino.

3.1.2 Módulo de transmisión

Cada uno de los nodos de la periferia son los que contienen los distintos módulos sensores y por lo tanto, son los que van a recoger la información obtenida por los sensores y los va a transmitir al módulo central. Para la transmisión de los datos cada nodo periférico tiene un dispositivo Xbee S2C el cual utiliza protocolo ZigBee para el envío de los datos; esto permite un bajo consumo de energía de los nodos y una fiabilidad en la comunicación [5].

3.1.3 Módulo de recepción

Empleando el mismo protocolo de comunicación el nodo central es el que actúa como módulo de recepción. En este módulo se recopila toda la información obtenida de la red de nodos periféricos, siendo este punto un centro de gestión y administración de la red [5].

3.1.4 Procesamiento

Con los datos obtenidos de cada punto de la red se realiza el proceso de validación, conversión, análisis y clasificación de cada uno de los valores enviados por los nodos sensores y se lo almacena en una tarjeta micro SD. También se determina las acciones que realiza de acuerdo al cumplimiento o incumplimiento de los parámetros establecidos.

3.1.5 Diagrama de funcionamiento del sistema

El sistema en detalle consta de tres etapas fundamentales que realizan acciones concretas dentro del mismo, como se describe en la figura 3.1. Dentro de este sistema se encuentran los nodos periféricos que tienen como función la recolección de los datos por medio de los sensores y el envío de los mismos al nodo principal a través de la comunicación ZigBee establecida empleando los módulos Xbee S2C.

El nodo principal se encarga de recibir la información enviada desde los nodos periféricos por medio de módulo Xbee y recopilar estos datos en una memoria extraíble, a su vez presenta la información en una pantalla y determina el accionamiento del sistema de riego acorde a los parámetros establecidos.

El sistema de riego consiste en la activación del módulo relay que permite el encendido de la bomba por intervalos de tiempo determinados por el sistema.

Los nodos cuentan con una etapa de alimentación que energiza el sistema, dándole una independencia que permite colocar los nodos en el sitio a conveniencia.

DIAGRAMA DEL SISTEMA

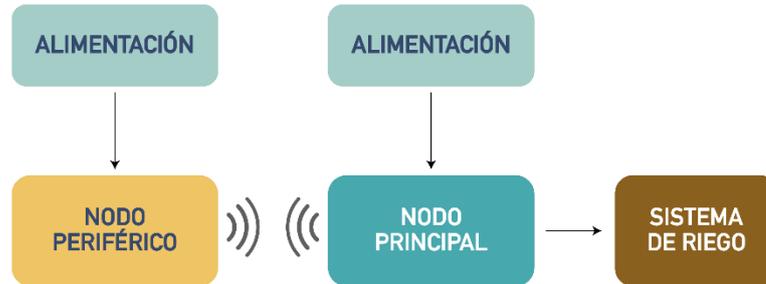


Figura 3.1 Diagrama de funcionamiento del Sistema

3.2 Dispositivos y Protocolos seleccionados

Conforme a las condiciones presentadas en el terreno, se han seleccionado los elementos adecuados para la construcción de la red de nodos de sensores. En concreto, se va a emplear módulos de Arduino, módulos Xbee S2C, sensores de humedad del suelo, relojes RTC, sensores de humedad y temperatura ambiente.

3.2.1 Implementación en Modo API

En la comunicación de módulos Xbee se toma en cuenta el modo API para enlazar los dispositivos. Este modo presenta algunas ventajas que permiten una eficiencia de la comunicación en la WSN y a su vez da paso a un escalamiento de la red en caso de requerirlo.

La comunicación en este modo se da por tramas en las que se incluyen los datos de cada uno de los sensores del nodo periférico para transmitir al nodo principal y la dirección del módulo destino. De esta manera también puede realizar la comunicación entre más nodos para extender la red y cubrir un área mayor.

3.2.2 Aplicación multiplataforma XCTU

El software XCTU es una interfaz GUI gratuita, desarrollada por la empresa Digi para

la programación de módulos RF, la cual permite realizar la configuración del firmware, de cada uno de los módulos Xbee empleados en el sistema y poder realizar el diseño de la red de nodos, tomando como coordinador al módulo que pertenece al nodo principal, de forma que todas las tramas se envíen a este nodo para el procesamiento de la información y los demás módulos como router, de manera que no solo transmitan la información sino que también sirvan de punto de enlace para nuevos nodos [6].

Para la configuración de los Xbee se conecta cada módulo a una computadora por medio de un adaptador USB y se verifica que en el software se detecte el dispositivo y el puerto de conexión.

Una vez seleccionado el dispositivo se muestra en la interfaz las configuraciones del mismo y se procede con el cambio de las mismas a conveniencia del sistema tal como se presenta en la Figura 3.2.

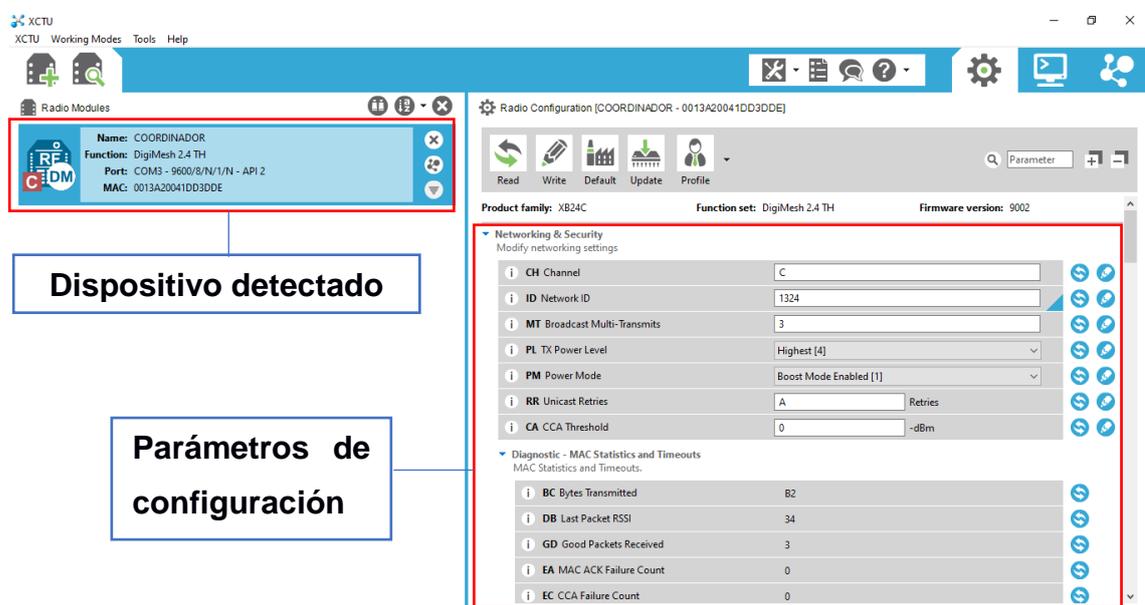


Figura 3.2 Dispositivos detectados y parámetros de configuración X-CTU

3.2.3 Módulo Xbee S2C

Para el proceso de transmisión y recepción de información dentro del sistema de comunicación con el protocolo ZigBee se emplea módulos Xbee S2C (Ver Anexo 1), cuyas

especificaciones se detallan en la Tabla 3.1. Este módulo se toma como una solución integrada que permite establecer la comunicación entre cada uno de los nodos de la red. Se puede crear conexiones punto a punto y punto a multipunto dependiendo de los requerimientos de la WSN, todo esto bajo el protocolo IEEE 802.15.4; estos módulos trabajan en frecuencias ISM. [1].

Tabla 3.1 Especificaciones técnicas del Xbee S2C [1]

Características técnicas del Xbee S2C	
Distancia	40 m
Rango Optimo	120 m
Corriente de transmisión /recepción	40/40 mA
Firmware	ZigBee
Entradas digitales	11
Entradas análogas	4
PWM	No
Topología punto a punto	Si
Topología Mesh y Cluster Tree	Si

3.2.4 Sensor de humedad de suelo HD-38

Este sensor tiene ciertas especificaciones detalladas en la Tabla 3.2, que permiten al dispositivo tomar lectura de humedad relativa del suelo donde se implementa, a través de dos varillas anticorrosivas. Estas varillas van incrustadas en el suelo donde se encuentra la plantación y miden la humedad relativa por medio del paso de corriente entre los dos terminales (Ver Anexo 2). Si la tierra se encuentra seca se presenta una resistencia bastante alta al paso de corriente; el valor de esta resistencia disminuye a medida que aumenta el porcentaje de humedad del suelo, permitiendo un control de esta variable climática en el cultivo de cacao. [4]

Tabla 3.2 Especificaciones técnicas del Sensor de Humedad del Suelo HD-38 [4]

Características del Sensor de Humedad del Suelo HD-38	
Voltaje de alimentación	3.3 V a 12 V
Corriente	30 mA
Chip controlador	LM393 VCC(+) GND(-)
Pines de salida	Salida analógica (A0) / Salida digital (D0)
Longitud de cable de sonda	1 m

3.2.5 Sensor DHT22

Para poder tomar mediciones de humedad y temperatura del ambiente dentro de la plantación de cacao se emplea el sensor DHT22 (Ver Anexo 3), el cual permite realizar la medición de estas dos variables al mismo tiempo. Entre las características que definen este dispositivo se encuentra un rango de medición de la humedad relativa del ambiente de 0% a 100 % con una precisión del 4%, además la medición de temperatura se puede precisar en un rango de 0 °C a 50 °C, tal como se detalla en la Tabla 3.3. El tiempo de muestreo de este dispositivo esta alrededor de los 12s. [5]

Tabla 3.3 Especificaciones técnicas del Sensor DHT22 [5]

Características del Sensor DHT22	
Voltaje de alimentación	3 V a 5 V
Temperatura	0 a 50 °C
Precisión de Temperatura	± 2°C
Humedad	0% a 100% RH
Precisión de Humedad	4% RH

3.2.6 Reloj RTC DS3231

Este dispositivo se emplea dentro del sistema para determinar intervalos de tiempo para las diferentes tareas que realiza cada nodo, como la toma de lectura en los nodos periféricos y el envío de los datos hacia el nodo principal; además permite controlar el intervalo de accionamiento de la bomba. Utiliza el protocolo I2C [7] para poder realizar una

sincronización dentro de los procesos que maneja el sistema. Puede funcionar de manera independiente con una fuente de alimentación de una pila de reloj CR2032, manteniendo una autonomía en la medición de los tiempos (Ver Anexo 4). En la Tabla 3.4 se puede verificar las características del módulo reloj [6].

Tabla 3.4 Especificaciones técnicas del Módulo DS3231 [6]

Características del Módulo de Reloj DS3231	
Voltaje de alimentación	3.3 a 5.5 V
Interfaz	I2C
Memoria	AT24C32
Capacidad de almacenamiento	32 KB
Precisión del reloj	0-40 °

3.2.7 Módulo TP4056

El módulo de carga TP4056 [8] posee características adecuadas, mostradas en la Tabla 3.5, para controlar la alimentación energética desde el panel solar, distribuir y regular la energía permitiendo la carga de la batería y el suministro de corriente al sistema, además de controlar la descarga de la batería para alimentar el circuito en ausencia de energía procedente del panel (Ver Anexo 5).

Tabla 3.5 Especificaciones técnicas del Módulo TP4056 [2]

Módulo TP4056	
Corriente	1 [A]
Voltaje de entrada	4.5 [V] a 5.5 [V]
Carga de precisión	1.5%
Tensión de carga completa	4.2 [V]
Temperatura de trabajo	-10 °C a 85 °C

3.2.8 Módulo elevador de voltaje Mt3608

El voltaje de entrada del sistema se debe mantener en torno a los 5V por lo que el módulo de elevación permite regular la tensión para que tenga una salida de 5V,

independientemente de la entrada que tenga, siempre que se encuentre dentro del rango de operación que se detalla en la Tabla 3.6. (Ver Anexo 6).

Tabla 3.6 Especificaciones técnicas de Módulo Elevador De Voltaje Mt3608 [8]

Módulo Elevador De Voltaje Mt3608	
Voltaje de entrada	2 a 24 [V]
Frecuencia de conmutación	1,2 [MHz]
Límite de corriente	4 [A]
MOSFET de potencia integrado	80 mΩ

3.2.9 Panel Solar Innovatech

La alimentación energética del sistema se la toma a partir de paneles solares fotovoltaicos (Ver Anexo 7) [3], cuyos factores de operación se muestran en la Tabla 3.7. Estos paneles generan una corriente con la presencia de luz solar. Esta energía se distribuye al sistema y también se almacena parte de ella en baterías.

Tabla 3.7 Especificaciones técnicas de Panel Solar 5V [3]

Panel Solar 5V	
Potencia	1.2 [W]
Voltaje	5 [V]
Dimensiones	98 x 68 mm

3.2.10 Batería Rakieta 18650 Li-ion recargable

La energía proveída por los paneles solares se almacena en baterías recargables [4] que permiten el funcionamiento continuo del sistema cuando la luz solar no tenga la suficiente intensidad para alimentar el circuito (Ver Anexo 8). En la Tabla 3.8 se detallan las condiciones energéticas bajo las cuales opera la batería.

Tabla 3.8 Especificaciones técnicas de Batería Rakieta 18650 Li-ion Recargable [4]

Batería Rakieta 18650 Li-ion Recargable	
Capacidad Nominal	12000 mAh
Voltaje	3.7 V
Tipo	li-ion
Medidas	6,6 x 1,8

3.2.11 Adaptador Xbee para Protoboard

Este dispositivo permite una conexión física más estable ya que adapta los terminales del módulo Xbee [10] para el armado del circuito de cada uno de los nodos (Ver Anexo 9).

Las especificaciones del adaptador Xbee se detallan en la Tabla 3.9.

Tabla 3.9 Especificaciones técnicas de Adaptador Xbee para Protoboard [10]

Módulo TP4056	
Voltaje	3.3 [V]
Comunicación	Paso en serie al módulo Xbee
Temperatura de funcionamiento	-40 a +85 °C
Dimensiones	29,4 mm x 25,6 mm x 14,8 mm

3.3 Diseño de la WSN

Para la implementación de la red de nodos del sistema se seleccionó la topología de estrella debido a que la comunicación es directa entre el nodo principal y cada uno de los nodos periféricos. Sin embargo se conserva la característica de escalabilidad para extender la red según la extensión del sitio donde se implementa. Se emplea un módulo Xbee S2C [6] para cada nodo periférico y el nodo principal para establecer la comunicación inalámbrica, además se utiliza una placa de arduino como unidad de central de cada nodo para el procesamiento de la data.

3.3.1 Configuración de la WSN

La configuración de los dispositivos Xbee de la WSN se realiza por medio de la herramienta de software libre XCTU. En esta red se definen algunos parámetros para la

conexión de los módulos como la ID de la red y el canal mediante el cual se establece la comunicación. Se configuran tres nodos periféricos como tipo router, de manera que puedan realizar la función de esclavos y a su vez funcionar como una ruta de transmisión de datos en caso de adicionar más nodos periféricos; y un nodo se configura como coordinador, el cual va a recibir los datos que se envían desde los nodos periféricos.

3.3.2 Prueba de comunicación entre los nodos

Con el uso del software XCTU se realiza la prueba de conexión entre los nodos de la red, de esta forma se puede verificar la comunicación establecida y la intensidad de señal de transmisión. La intensidad de señal que se recibe está en un rango entre -42 y -56 dBm, como se evidencia en la Figura 3.3.

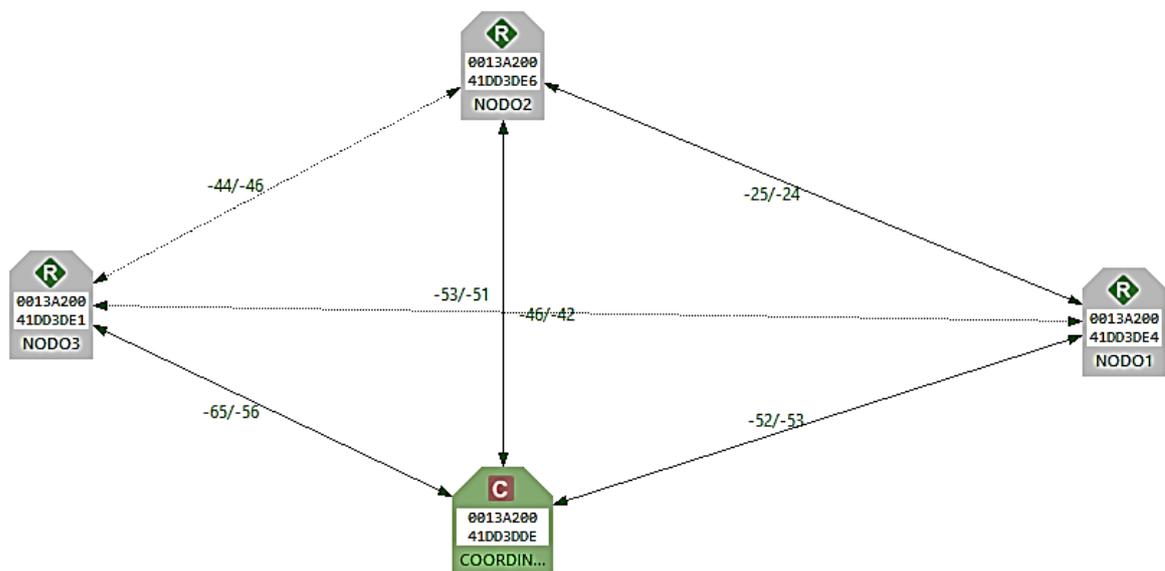


Figura 3.3 Red sensores tipo malla

Se realizaron pruebas de alcance con los nodos periféricos a 8 metros de distancia del nodo principal, (Ver Anexos 10, 11, 12); y 20 metros de distancia del nodo principal, (Ver Anexos 13, 14, 15); constatando una comunicación sin pérdida de información en las dos distancias de referencia.

3.3.3 Ubicación de los nodos en el terreno

Las características del lugar donde se implementa el sistema son favorables en cuanto a la uniformidad de las condiciones en toda la extensión del terreno de prueba, ya que al ser un terreno sin inclinación presenta una humedad relativa del suelo similar en cada punto del sitio, además las diferencias en las condiciones del ambiente, humedad relativa y temperatura, son prácticamente despreciables en todo el terreno. Por lo tanto, la distribución de los sensores obedece a una posición equidistante dentro del área considerando un radio de acción para cada nodo; sin embargo para aplicar en un terreno con pendiente se debe tomar en cuenta también esta condición debido a que la inclinación ocasiona diferentes valores del parámetro de humedad de suelo en cada punto de la plantación.

3.4 Diseño de Nodo Principal y Periféricos

3.4.1 Diseño de Nodo Principal

El nodo principal se compone de una tarjeta arduino como MCU, un módulo Xbee S2C encargado de la comunicación, un módulo de reloj que controla el tiempo de acción de la señal que activa la bomba, una pantalla LCD que muestra la información del sistema y un módulo para almacenamiento micro SD. Además, se conecta a un módulo de alimentación para una autonomía energética.

Este nodo funciona como coordinador del sistema, el cual se encarga de recibir la información de los nodos periféricos, procesarla y ejecutar las diferentes acciones en base a la información recibida. El funcionamiento consiste en la recepción de los valores de los parámetros de cada nodo, identificando el nodo del que proviene y el tipo de información que recibe, a su vez almacena la información en una memoria micro SD, para ser descargada por el usuario en caso de requerirlo. La información de la medición de los nodos

se presenta en una pantalla mediante la activación de un pulsador, que muestra uno a uno los nodos de la red de manera secuencial.

Con la información recibida, el nodo realiza una comparación con valores de los parámetros establecidos de tal manera que si un valor se encuentra fuera del rango de parámetros aceptable, se genera una señal hacia el sistema de activación de la bomba.

El diagrama del nodo principal se puede observar en la Figura 3.4.

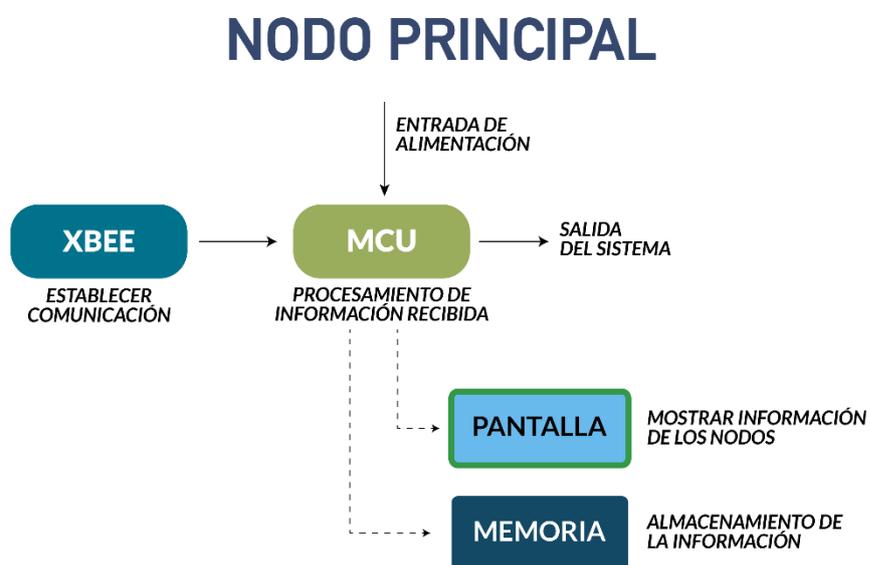


Figura 3.4 Diagrama del Sistema del Nodo Principal

Los parámetros determinados como aceptables para este proyecto se pueden evidenciar en la Tabla 3.10

Tabla 3.10 Parámetros de referencia para la activación del sistema de riego

Humedad del suelo	< 65 %	Se activa el sistema de riego
Humedad de ambiente	< 50 %	
Temperatura de ambiente	> 33 °C	

3.4.2 Diseño de Nodo Periférico

El nodo periférico se compone de una tarjeta arduino como MCU, un módulo Xbee

S2C encargado de establecer el enlace con el nodo principal, un módulo de reloj que regula los intervalos de tiempo en los cuales se toman las mediciones para enviar al nodo principal, un sensor de humedad relativa de suelo, un sensor de temperatura y humedad relativa de ambiente. Además, se conecta a un módulo de alimentación para una autonomía energética.

El funcionamiento de un nodo periférico consiste en administrar los sensores, recopilando la información proveniente de las lecturas de estos para enviar al nodo principal mediante el módulo Xbee. El diagrama del nodo periférico se puede observar en la Figura 3.5.

NODO PERIFÉRICO

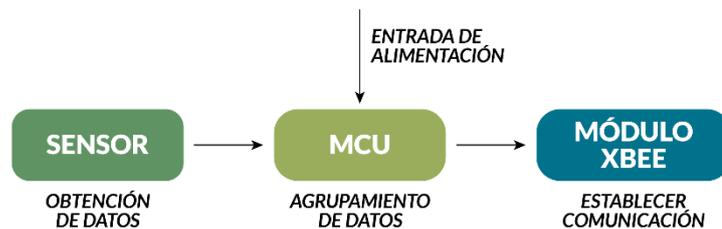


Figura 3.5 Diagrama del Sistema del Nodo Periférico

3.5 Diseño del sistema de alimentación

Para maximizar la eficiencia del sistema se debe lograr una independencia energética con la finalidad de poder extender los nodos a lugares donde no hay energía eléctrica pública, esto permitirá que el sistema sea autónomo. El sistema de alimentación está compuesto por paneles solares y baterías Li-ion de gran capacidad nominal. El diagrama del sistema de alimentación se puede observar en la Figura 3.6.

ALIMENTACIÓN



Figura 3.6 Diagrama del Sistema de Alimentación

3.6 Diseño del sistema de riego

El accionamiento de la bomba se da a través de una salida de voltaje desde el MCU del nodo principal para activar un relay que permita el paso de corriente hacia la bomba de agua, esta salida de voltaje se da en 2 casos que dependen de los valores de los parámetros recibidos desde los nodos periféricos. Según el parámetro que se encuentre fuera del rango aceptable se accionará bomba por un periodo de tiempo determinado entre 30 minutos y 120 minutos, acorde al requerimiento del cliente y basado en el tiempo de riego previo a la implementación del sistema. El tiempo de salida se regula por medio de un módulo de reloj. El diagrama del nodo periférico se puede observar en la Figura 3.7.

SISTEMA DE RIEGO



Figura 3.7 Diagrama del Sistema de Riego

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

Dentro de las mediciones realizadas están la humedad relativa del suelo cuyas gráficas de datos se observan en la Figuras 4.1, 4.2 y 4.3; también se muestran los datos obtenidos de humedad relativa de ambiente en las en las gráficas de las Figuras 4.4, 4.5 y 4.6; por último se evidencian las mediciones de temperatura de ambiente, cuyos datos está graficados en las figuras 4.7, 4.8 y 4.9. Las gráficas mostradas corresponden a los nodos periféricos 1, 2 y 3 respectivamente.

4.1 Análisis de datos de humedad del suelo de los Nodos periféricos

En la figura 4.1 se puede observar los datos de humedad de suelo tomados por el nodo periférico 1. En esta gráfica se observa el rango normal de porcentaje de humedad cuyo valor oscila entre 80 % y 90 %. Sin embargo, se muestran datos por encima del rango debido al riego activado en el terreno y también datos por debajo del rango promedio, hasta 68 % en días despejados.

En la figura 4.2 se presentan los datos de humedad de suelo correspondientes al nodo periférico 2. Se puede evidenciar un rango de mediciones entre 70 % y 77 %. Además se presentan valores que llegan por debajo del límite de parámetros aceptables, por lo cual se activa el sistema de riego para elevar la condición de humedad del suelo. Esto se puede comprobar verificando los valores de humedad luego de la activación de la bomba, que se elevan incluso hasta el 100% debido al riego del cultivo.

En la figura 4.3 se muestran los valores obtenidos del nodo periférico 3, cuyos valores reflejan un rango promedio de humedad de suelo entre 70 % y 80 %. Dentro de esta gráfica se verifican puntos por encima del rango promedio debido a la activación del sistema de

riego, así mismo se presentan puntos por debajo del promedio pero dentro del rango aceptable, a excepción de uno de los puntos por debajo del 65% que activa el riego.

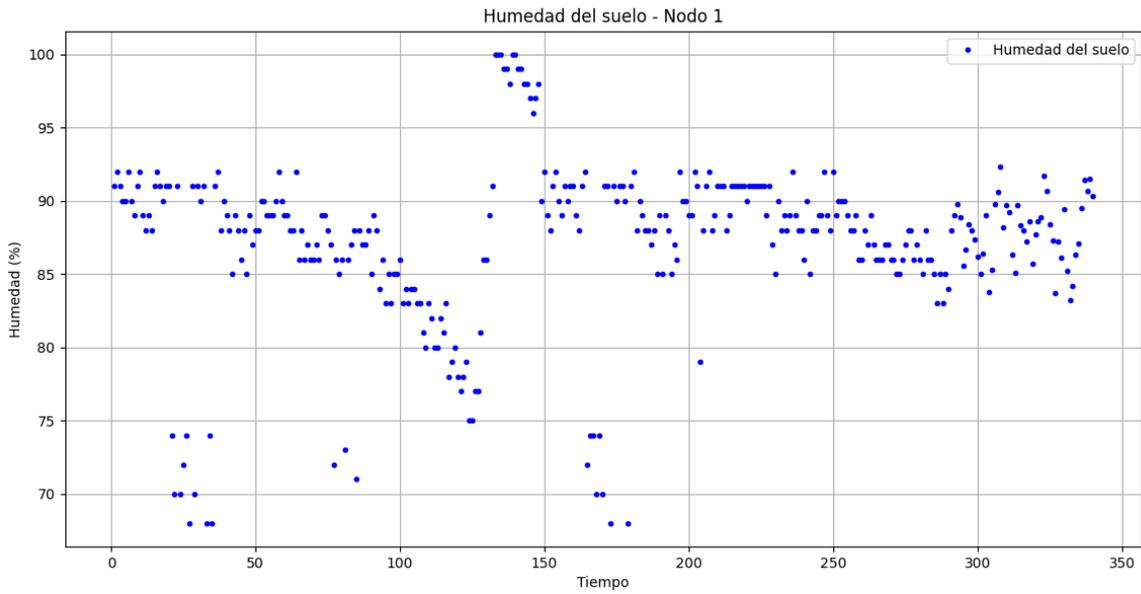


Figura 4.1 Gráfica de Humedad del Suelo del Nodo 1

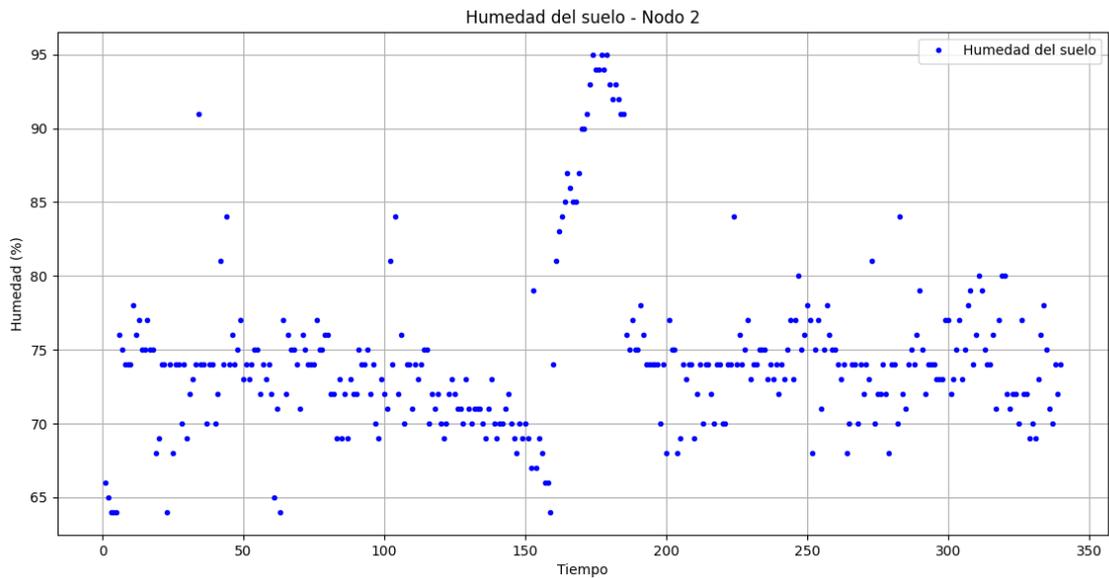


Figura 4.2 Gráfica de Humedad del Suelo del Nodo 2

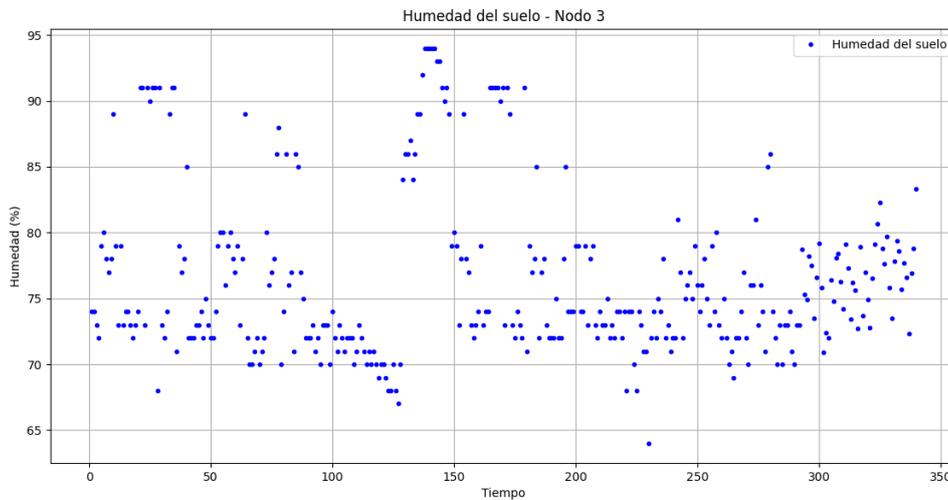


Figura 4.3 Gráfica de Humedad del Suelo del Nodo 3

4.2 Análisis de datos de humedad del ambiente de los Nodos Periféricos

En la figura 4.4 se evidencia la gráfica correspondiente a los datos de humedad de ambiente obtenidos en el nodo periférico 1. Los valores de humedad se encuentran en un rango porcentual entre 64 % y 70 %, manteniendo una constancia entre los valores de cada periodo de medición.

En la figura 4.5 se muestran los valores de humedad de ambiente tomados por el módulo sensor del nodo periférico 2. En la gráfica se evidencian valores de humedad en un rango entre 64 % y 70 %. Así mismo, se verifican muy pocos datos por debajo del rango promedio de humedad, mostrando una uniformidad de esta condición ambiental.

Los datos presentados en la gráfica de la figura 4.6 corresponden a las lecturas en la ubicación del nodo periférico 3. En esta gráfica también se evidencian valores de humedad entre 64 % y 70% con muy pocos datos fuera del rango promedio.

Los valores de humedad de ambiente registrados en cada uno de los nodos periféricos presentan una similitud entre sí, debido al microclima constante que se genera entre las plantas de cacao del cultivo. Este rango se mantiene por encima del límite establecido para la activación del sistema de riego.

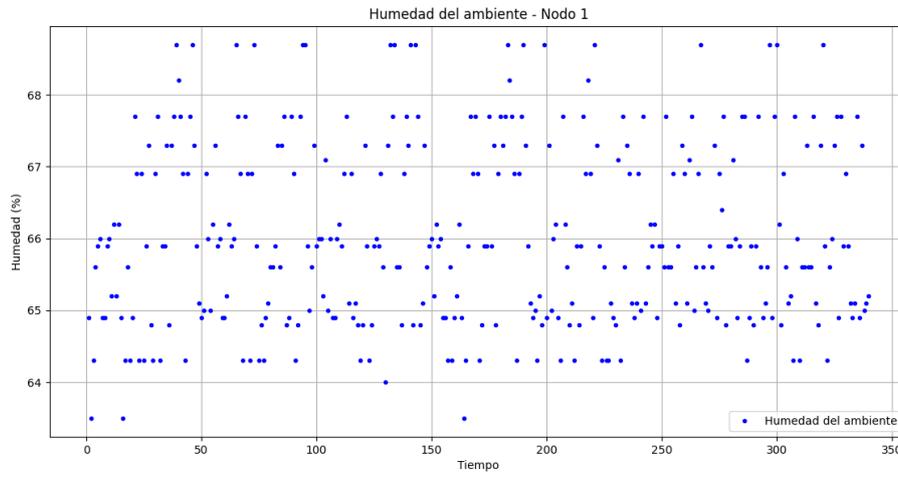


Figura 4.4 Gráfica de Humedad del Ambiente del Nodo 1

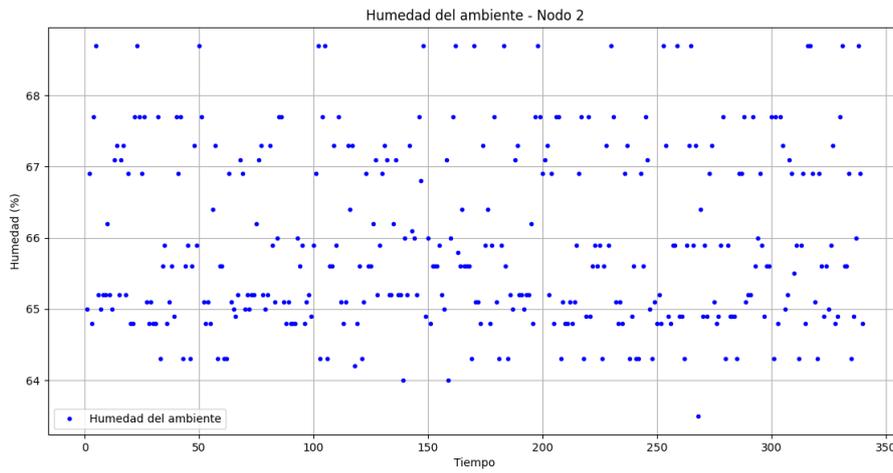


Figura 4.5 Gráfica de Humedad del Ambiente del Nodo 2

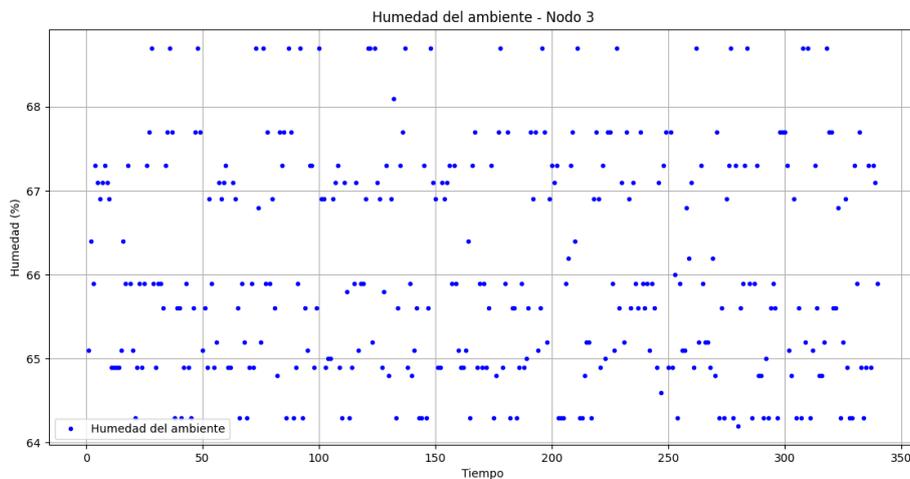


Figura 4.6 Gráfica de Humedad del Ambiente del Nodo 3

4.3 Análisis de datos de temperatura del ambiente de los nodos periféricos

Los valores de lectura tomados por el nodo periférico 1 se muestran en la gráfica de la figura 4.7, en la cual se aprecia una concentración de los valores en un rango de temperatura de ambiente entre 26 °C y 31°C. En las figuras 4.8 y 4.9 se representan los datos obtenidos de los nodos periféricos 2 y 3 respectivamente. En estas gráficas presentadas se puede observar una similitud con respecto a la gráfica del nodo 1 debido a que la temperatura en promedio dentro del área de implementación no varía de manera significativa. El rango de temperatura de ambiente se mantiene dentro de los parámetros ideales para el cultivo.

En las gráficas de temperatura, los tres nodos periféricos reflejan un patrón en los puntos. Este patrón describe el comportamiento de la temperatura a lo largo de cada día, empezando en la mañana con valores de temperatura entre 26 °C y 28 °C. Los valores de temperatura van aumentando hasta llegar a un promedio de 30 °C, aproximadamente al mediodía y decayendo ligeramente hacia el final del día con valores entre 29 °C y 30 °C.

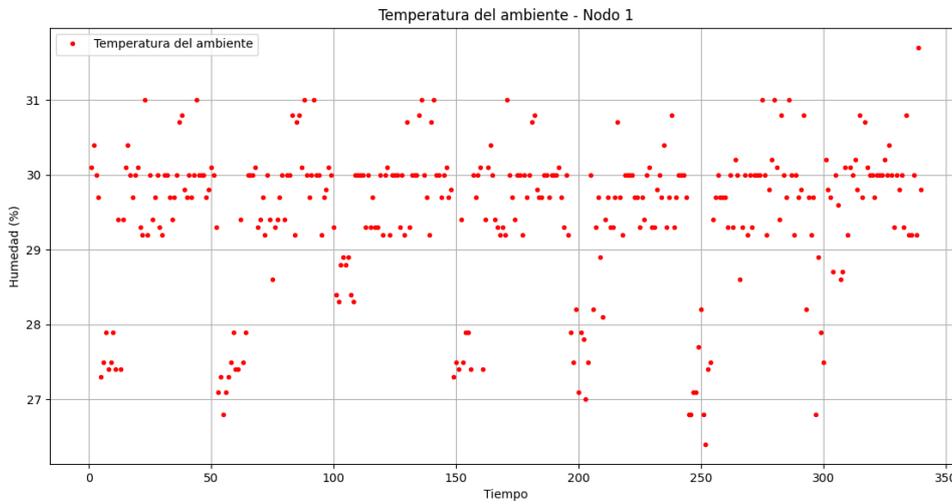


Figura 4.7 Gráfica de Temperatura del Nodo 1

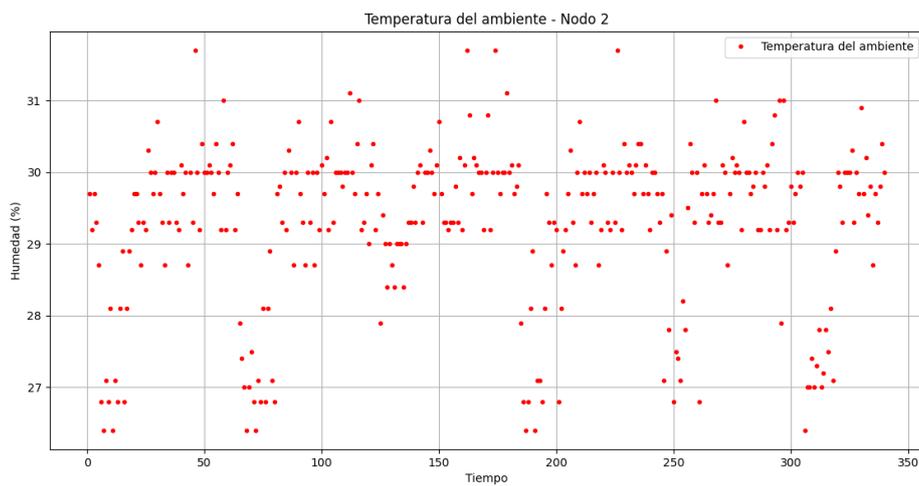


Figura 4.8 Gráfica de Temperatura del Nodo 2

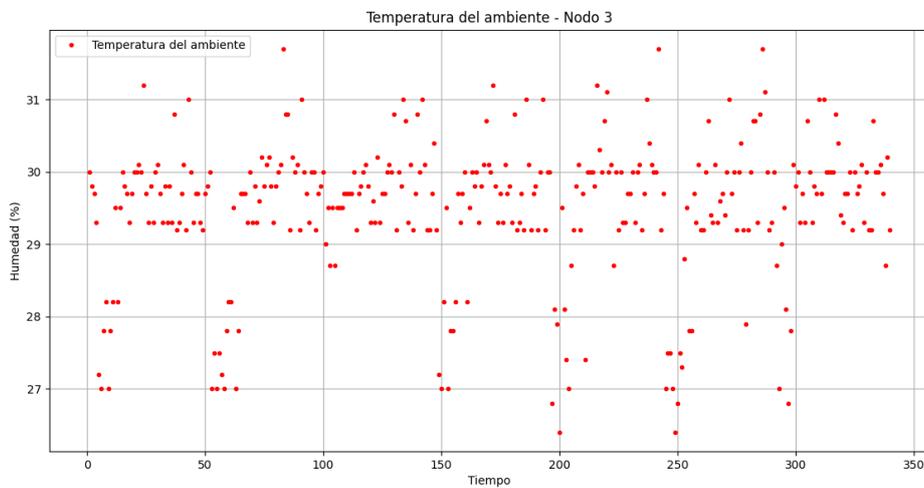


Figura 4.9 Gráfica de Temperatura del Nodo

CAPÍTULO 5

5. Costos del proyecto

5.1 Análisis Económico

La viabilidad del proyecto depende no solo de la parte técnica, sino también del costo de inversión respecto a los beneficios que representa para el usuario. Los costos de inversión y mantenimiento del sistema se aprecian en las tablas 5.1, 5.2 y 5.3, detallando los valores de los componentes utilizados; sin embargo, pueden variar dependiendo del proveedor.

5.2 Gastos materiales

Tabla 5.1 Cuadro general de costos de inversión

Equipos	Coste Unitario	Cantidad	Costo total
Placa Arduino Uno	\$25,00	3	\$75,00
Placa Arduino Mega	\$18,00	1	\$18,00
Módulo Zigbee Xbee S2C	\$40,00	4	\$160,00
Sensor de Temperatura y humedad de ambiente DHT22	\$8,00	3	\$24,00
Sensor de humedad de suelo HD-38	\$9,00	3	\$27,00
Módulo de Reloj DS3231	\$4,50	4	\$18,00
Módulo TP4056	\$3,00	4	\$12,00
Módulo USB Xplorer Para Xbee S2C	\$8,00	2	\$16,00
Adaptador Xbee para Protoboard	\$4,00	4	\$16,00
Batería Rakieta 18650 Li-ion Recargable	\$2,00	4	\$8,00
Panel Solar 5V	\$6,00	4	\$24,00
Módulo Relé 5V 2 Canales con acopladores	\$4,00	1	\$4,00
Módulo Elevador De Voltaje Mt3608	\$2,25	1	\$2,25
Pantalla LCD2004	\$12,00	1	\$12,00
Costo total			\$416,25

5.3 Análisis de Mantenimiento del Nodo Periférico

Tabla 5.2 Cuadro de costos de mantenimiento de Nodo Periférico

Mantenimiento de Nodo Periférico				
Equipos	Periodo	Coste unitario	Cantidad	Costo total
Cambio de módulo	12 Meses	\$40,00	1	\$40,00
Cambio de sensores DHT22	6 Meses	\$8,00	1	\$8,00
Cambio de sensores HD-38	6 Meses	\$9,00	1	\$9,00
Paneles solares	6 Meses	\$6,00	1	\$6,00
Pila del módulo de reloj	3 Meses	\$2,00	1	\$2,00
Cambio de batería	3 Meses	\$2,00	1	\$2,00
Mano de obra	-	-	-	\$15,00
Costo total anual				\$117,00

Tabla 5.3 Cuadro de costos de mantenimiento del Nodo Principal

Mantenimiento de Nodo Principal				
Equipos	Periodo	Coste unitario	Cantidad	Costo total
Cambio de módulo	12 Meses	\$40,00	1	\$40,00
Paneles solares	6 Meses	\$6,00	2	\$12,00
Pilas del módulo de reloj	3 Meses	\$2,00	1	\$2,00
Cambio de batería	3 Meses	\$2,00	2	\$4,00
Mano de obra	-	-	-	\$15,00
Costo total anual				\$103,00

CAPÍTULO 6

6. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Se pudo implementar la WSN para el control de las variables que inciden en la producción de la plantación de cacao de la Finca Florentino, en el recinto El Paraíso del cantón Milagro, logrando la automatización del riego en el área cubierta por el prototipo.
- La comunicación establecida en la WSN fue manejada a través del protocolo Zigbee empleando una topología de malla para una conexión internodal más efectiva y software de código abierto que evita costos adicionales.
- Se realizó la validación del sistema diseñado en el sitio de implementación, a través de la medición de los valores de humedad del suelo, humedad y temperatura de ambiente, contrastando los resultados con las condiciones ideales para el cultivo de cacao en la zona del recinto El Paraíso.
- La diferencia de los rangos obtenidos en la humedad del suelo de cada uno de los nodos indica que el terreno no tiene una humedad constante en todo el cultivo, a pesar de que el terreno no presenta una inclinación significativa. Esto puede deberse a la ubicación de los aspersores de riego distribuidos en el área, ya que el terreno circundante al aspersor tiende a presentar una mayor humedad. Sin embargo, los promedios de mediciones se presentan dentro de los parámetros ideales para el cultivo.

Recomendaciones

- La distancia de comunicación entre los nodos se ha establecido conforme al área de prueba determinado para este proyecto por lo que la antena del módulo Xbee S2C puede dar la cobertura necesaria. Sin embargo, para una implementación a una mayor distancia, se

puede adicionar una antena en cada módulo Xbee S2C, de manera que permita una mayor cobertura, de acuerdo a la ganancia de las antenas implementadas.

- La elección de los módulos sensores tiene relevancia al momento de realizar las mediciones. Esto debido a que la lectura realizada por los sensores puede verse afectada por la fiabilidad de los dispositivos, emitiendo datos inexactos o aberrantes que pueden activar el sistema de riego de forma innecesaria. Para prevenir estos sucesos se puede implementar un doble censado, de esta manera se activará el sistema solamente si se vuelve a recibir un dato similar.

- La implementación de los módulos Xbee S2C para la comunicación en red de nodos sensores es una buena opción en zonas rurales, ya que permite una conexión fiable en lugares donde no se tiene acceso a redes inalámbricas de gran alcance provistas por un ISP.

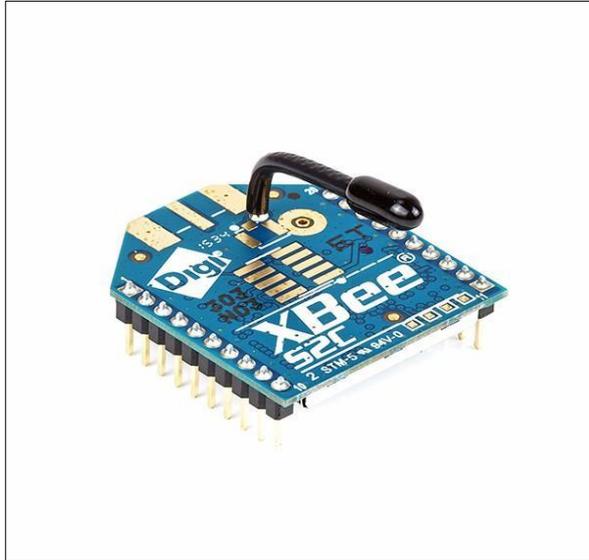
BIBLIOGRAFÍA

7. BIBLIOGRAFÍA

- [1] M. Banzi, D. Cuartielles, D. Mellis, T. Igoe y G. Martino, «arduino.cl,» 15 Octubre 2005. [En línea]. Available: <https://docs.arduino.cc/retired/shields/arduino-xbee-shield#references>.
- [2] Electrostore, «grupoelectrostore,» 08 Abril 2019. [En línea]. Available: <https://grupoelectrostore.com/shop/modulos-y-shields/cargadores-para-baterias/modulo-tp4056-5v-micro-usb-1a-cargador-para-baterias-litio-con-proteccion/>.
- [3] R. Konezny, «xbee.cl,» 15 Abril 2005. [En línea]. Available: <https://xbee.cl/que-es-xbee/>.
- [4] D. Cuartielles, T. Igoe, M. Banzi, D. Mellis y G. Martino, «arduino.cl,» 15 Octubre 2005. [En línea]. Available: <https://afel.cl/producto/shield-xbee-pro-para-arduino/>.
- [5] T. Liu, *Digital-output relative humidity & temperature sensor/module, ASAIR*, China,Guangzhou, Julio. 26, 2003.
- [6] V. Sas, «Vistronica,» 23 Abril 2014. [En línea]. Available: <https://www.vistronica.com/sensores/humedad/modulo-sensor-de-humedad-del-suelo-hd-38-detail.html>.
- [7] F. Carlile y L. Doughty, 24 Julio 1906. [En línea]. Available: <https://cdtecnologia.net/placas-electronicas/144-modulo-de-reloj-tiempo-real-ds3231-mas-bateria.html>.
- [8] J. P. Dignami, "*ANÁLISIS DEL PROTOCOLO ZIGBEE*", *Tesis de Ingeniería, Dept. Ing. Redes y Seguridad, Univ. Nacional de la Plata, La Plata, Argentina, 1897, 2011*.
- [9] ECO-RAEE, «LEDBOX,» 25 02 2019. [En línea]. Available: <https://www.ledbox.es/proyectores-led-para-extteriores/panel-solar-policristalino-5v-25w>.
- [10] I. A. Carrasco, «novatronicec.,» 21 06 2020. [En línea]. Available: <https://novatronicec.com/index.php/product/bateria-recargable-ultrafire-18650-3-7v-6800mah-li-ion/>.
- [11] P. C. Gracey, «PARALLAX,» 08 07 2002. [En línea]. Available: <https://www.mouser.com/datasheet/2/321/32403-XBee-Adapter-Board-v1.1-463992.pdf>.
- [12] L. Xi'an Aerosemi Technology Co., «MT3608,» CASC, China Shaanxi Aerosemi TechnologI. [En línea]. [Último acceso: 2011].

PLANOS Y ANEXOS

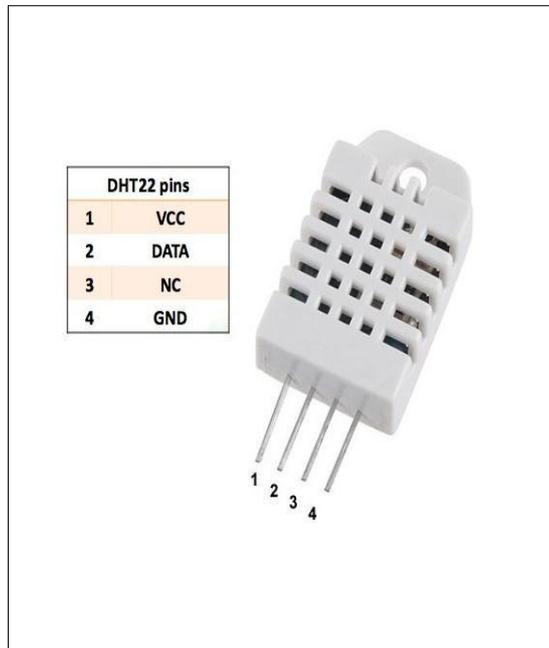
Anexo No. 7-1 Xbee S2C [1]



Anexo No. 7-2 Sensor HD-38 [4]



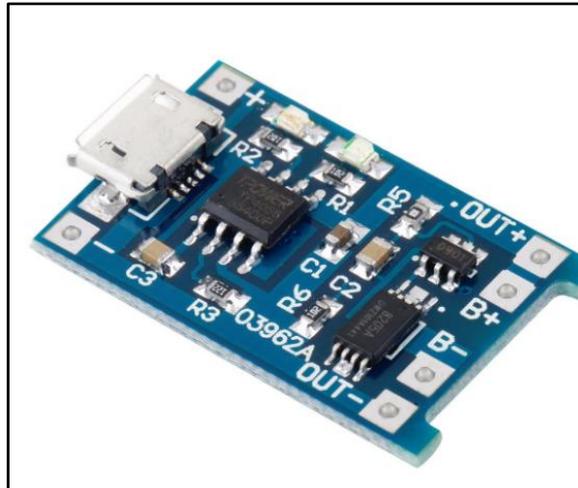
Anexo No. 7-3 Sensor DHT22 [5]



Anexo No. 7-4 Módulo de Reloj DS3231 [6]



Anexo No. 7-5 Módulo TP4056 [2]



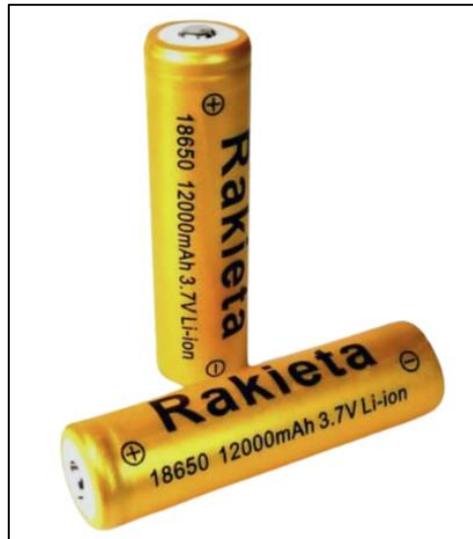
Anexo No. 7-6 Módulo Elevador De Voltaje Mt3608 [6]



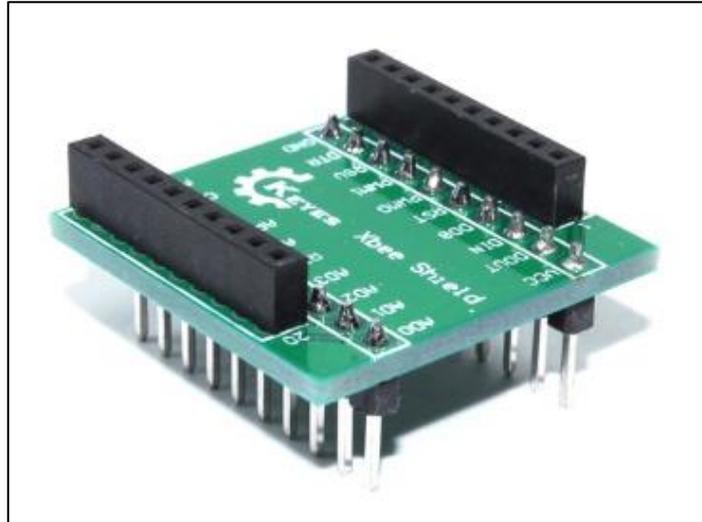
Anexo No. 7-7 Panel Solar Innovatech [3]



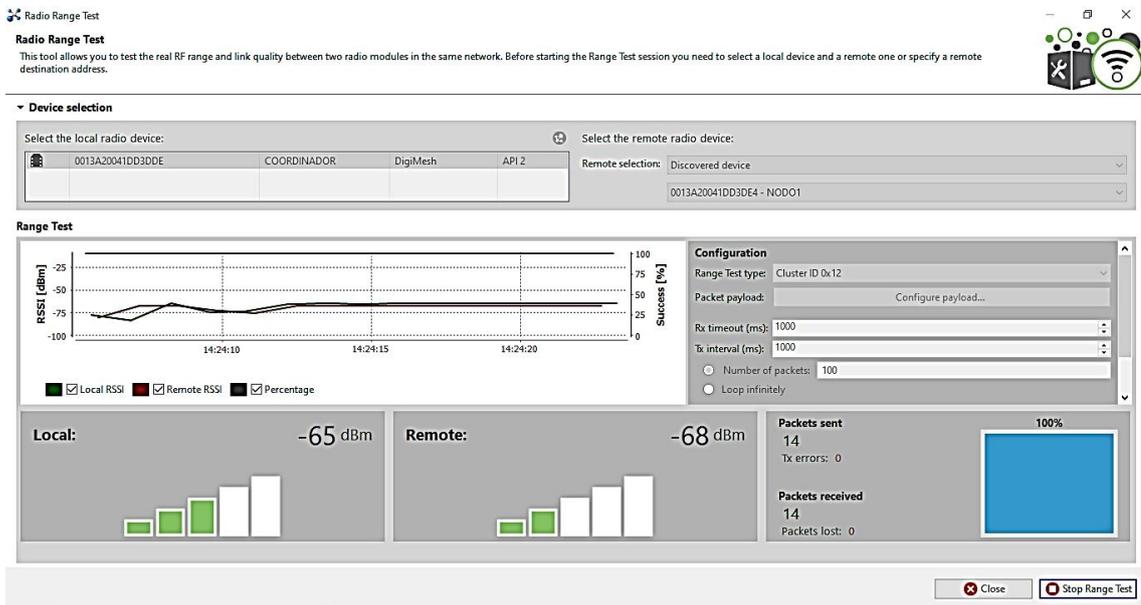
Anexo No. 7-8 Batería Rakieta 18650 Li-ion Recargable [4]



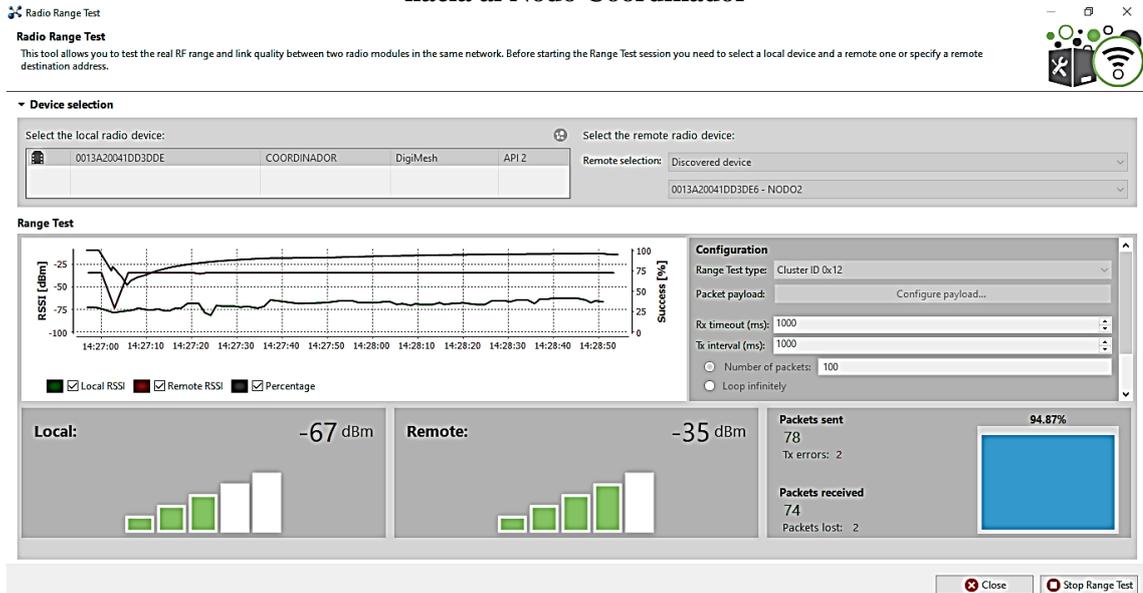
Anexo No. 7-9 Adaptador Xbee para Protoboard [5]



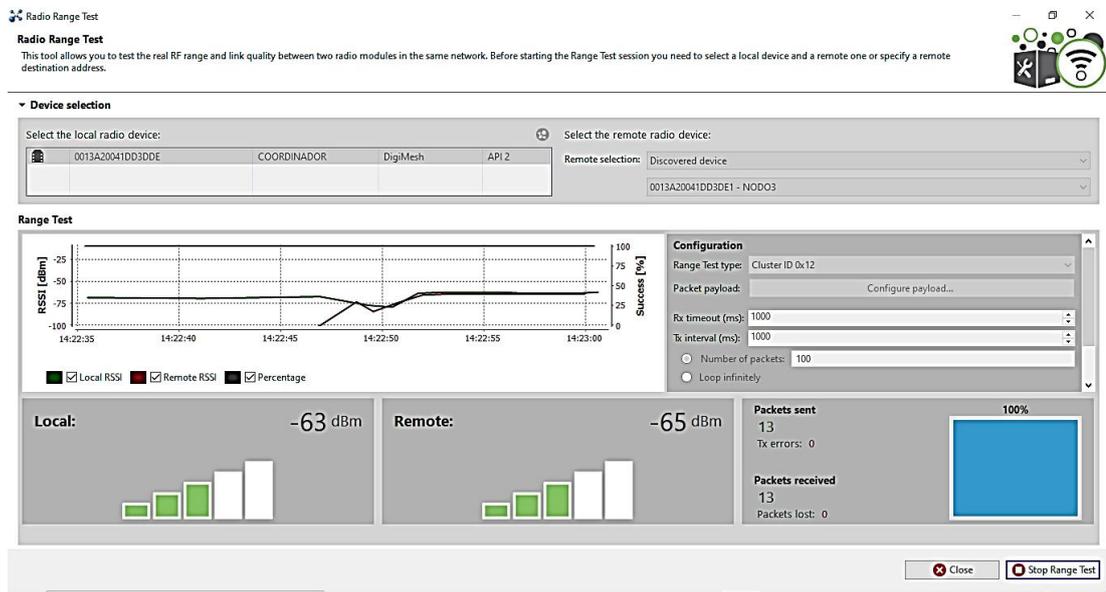
Anexo No. 7-10 Intensidad de señal red ZigBee con una distancia de 8 metros del Nodo 1 hacia al Nodo Coordinador



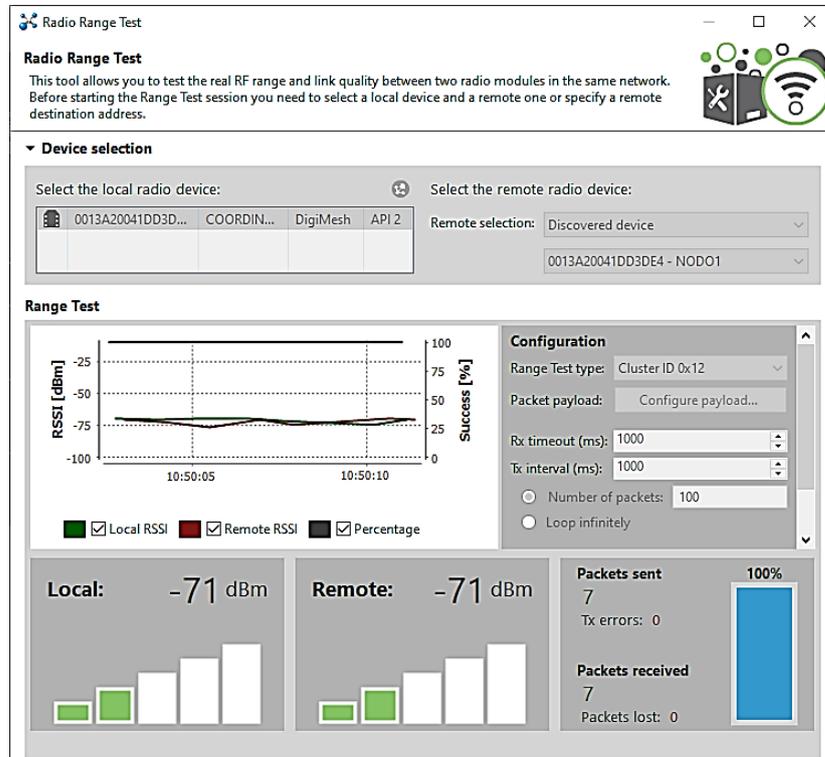
Anexo No. 7-11 Intensidad de señal red ZigBee con una distancia de 8 metros del Nodo 2 hacia al Nodo Coordinador



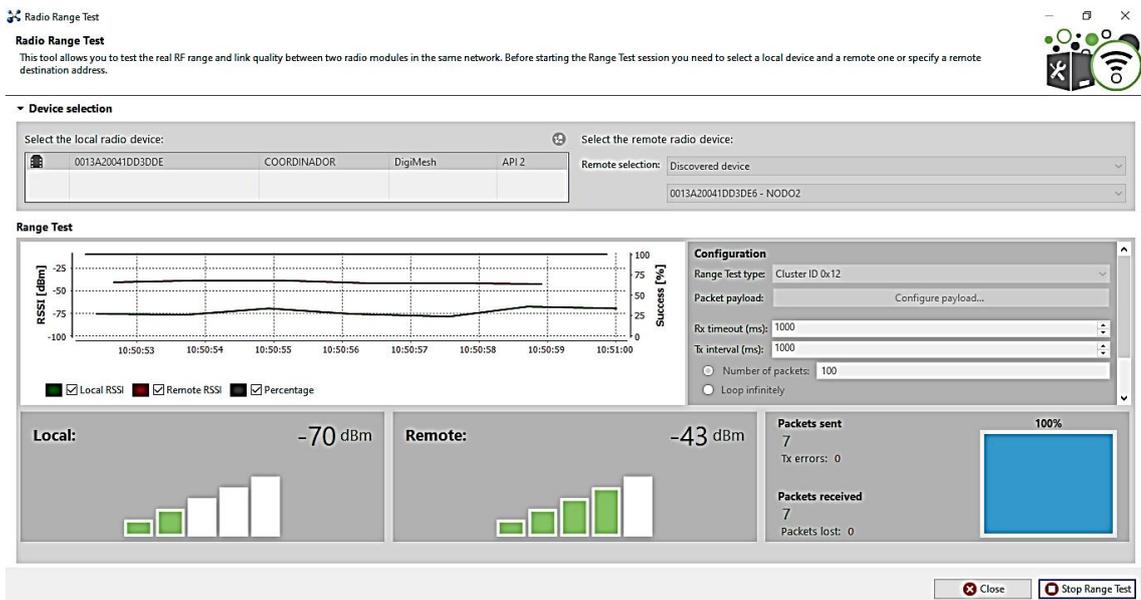
Anexo No. 7-12 Intensidad de señal red ZigBee con una distancia de 8 metros del Nodo 2 hacia al Nodo Coordinador



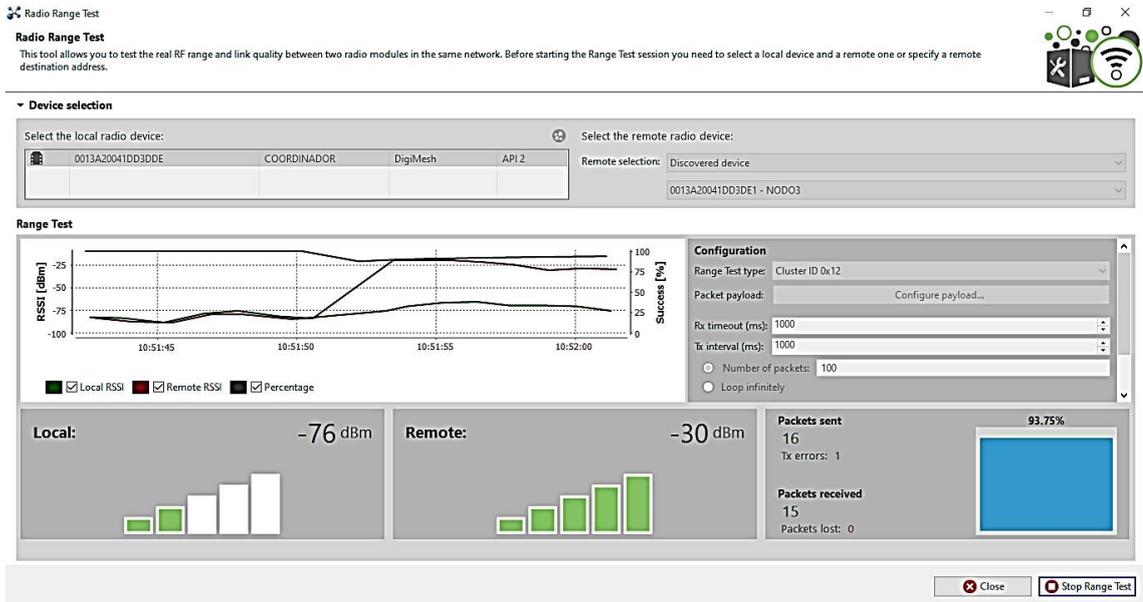
Anexo No. 7-13 Intensidad de señal red ZigBee con una distancia de 20 metros del Nodo 1 hacia al Nodo Coordinador



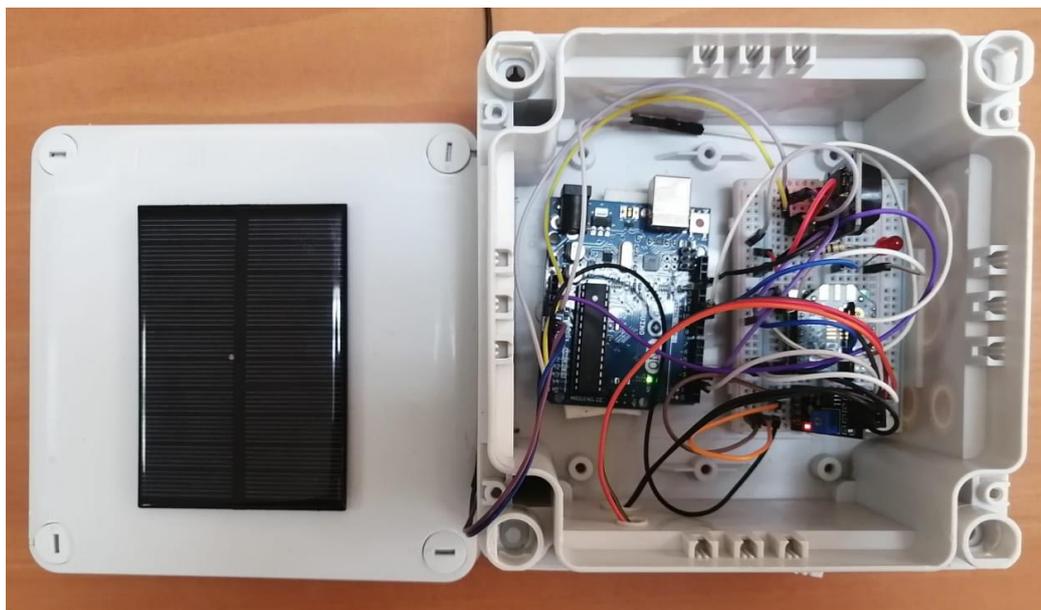
Anexo No. 7-14 Intensidad de señal red ZigBee con una distancia de 20 metros del Nodo 2 hacia al Nodo Coordinador



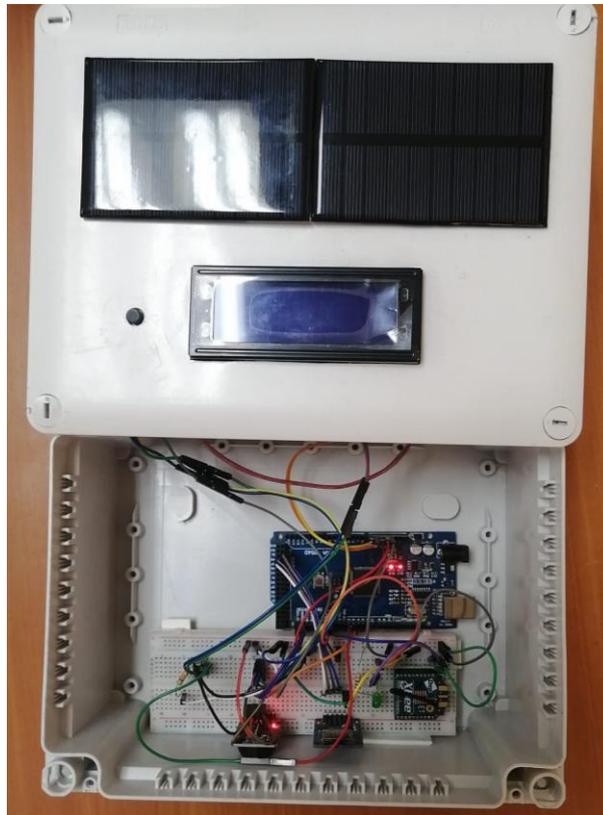
Anexo No. 7-15 Intensidad de señal red ZigBee con una distancia de 20 metros del Nodo 3 hacia al Nodo Coordinador



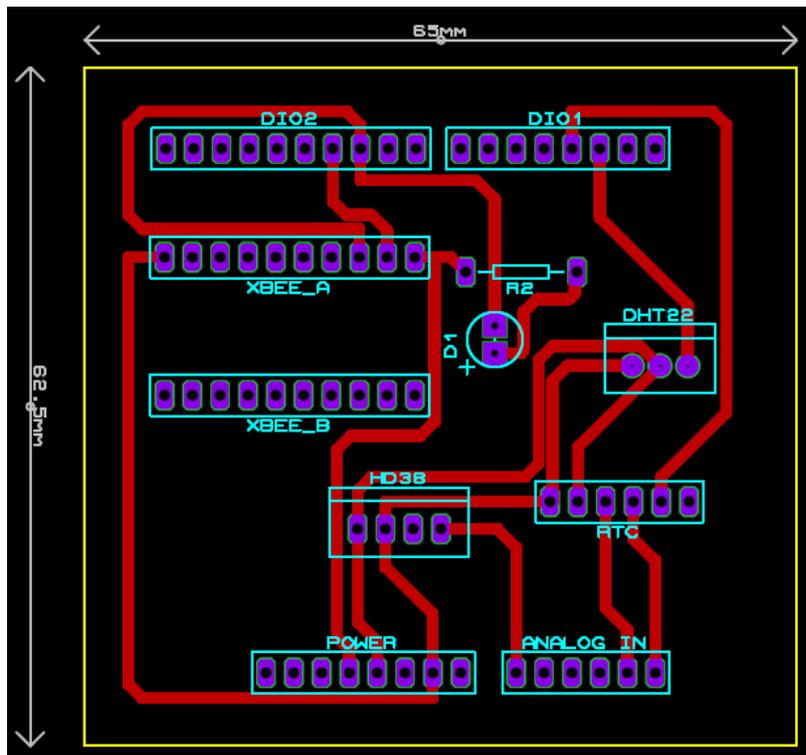
Anexo No. 7-16 Diseño del Nodo Periférico



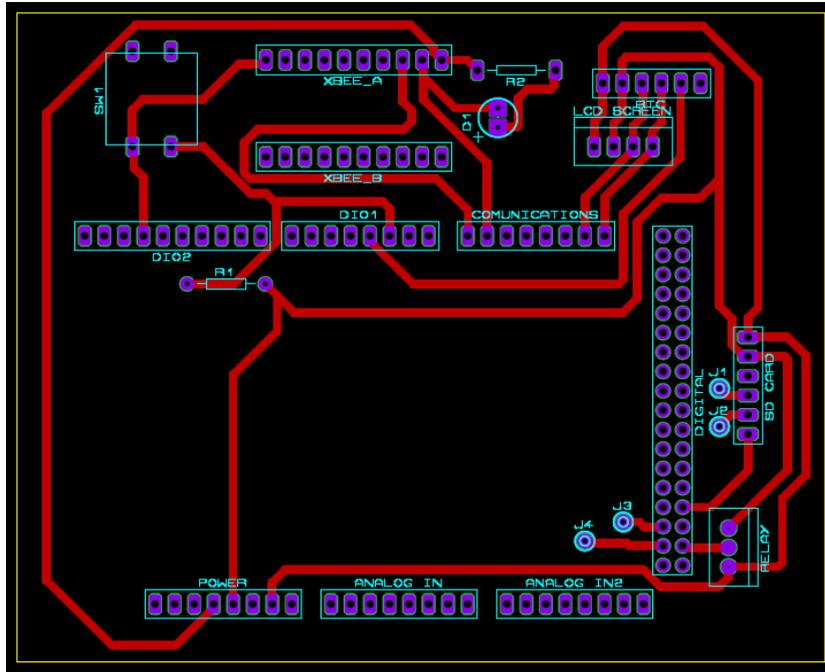
Anexo No. 7-17 Diseño del Nodo Central



Anexo No. 7-18 Diseño de placa PCB del Nodo periférico



Anexo No. 7-19 Diseño de placa PCB del Nodo central



Anexo No. 7-20 Código General para los Nodos Periféricos

```
#include "DHT.h"
#include <XBee.h>
#include <SoftwareSerial.h>
#include "RTCLib.h"
#include <avr/sleep.h>

// ----- Enviroment Variables ---- /
#define DHTPIN      2           // Digital pin connected to the DHT sensor
#define HD_38PIN    A0         //Analogic HD-38 PIN
#define XBEE_TX     10
#define XBEE_RX     11
#define RTC_INTERRUPT 3       //RTC alarm pin

#define DHTTYPE     DHT22      // DHT 22 (AM2302), AM2321
#define debug       Serial
#define WAKEUP_TIME 600       //Seconds to wakeup

const String NODE_ID = "NODE1";

// ---- Declare Instances --- //
DHT dht(DHTPIN, DHTTYPE);

XBee xbee= XBee();

ZBRxResponse rx = ZBRxResponse();
ZBTxStatusResponse s_tx = ZBTxStatusResponse();
ModemStatusResponse msr = ModemStatusResponse();
SoftwareSerial xbee_comm(XBEE_RX,XBEE_TX);

//rtc
RTC_DS3231 rtc;

uint32_t COORDINATOR_HIGH_ADDRESS = 0x0013A200; //SH address
uint32_t COORDINATOR_LOW_ADDRESS = 0x41DD3DDE; //SL address

XBeeAddress64 addr64 =
XBeeAddress64(COORDINATOR_HIGH_ADDRESS,COORDINATOR_LOW_ADDRESS);
//Create XBeeAddress64 objet

// --- Auxiliary Functions --- //
float HD_38_read(){
  int humidity = analogRead(HD_38PIN);           //analog value
  humidity = constrain(humidity, 400, 1023);     //define limits
  humidity = map(humidity, 400, 1023, 100, 0);   //convert analog value to 0-100%
  return humidity;
}
```

```

void to_array(uint8_t array[],String message){
  //array size
  int len = message.length();

  //fill array
  for (int i=0;i<len;i++){
    array[i]=message.charAt(i);
  }
}

void send_xbee(String message){

  //fills uint8_t array with all chars of message
  uint8_t a_message[message.length()];
  to_array(a_message,message);

  //Create a TX request
  ZBTxRequest request_tx = ZBTxRequest(addr64, a_message, sizeof(a_message));

  //Send TX request
  xbee.send(request_tx);
}

//Setup
void setup() {
  //Init instances
  xbee_comm.begin(9600);
  debug.begin(9600);           //Init Serial debugging
  debug.print("Power at ");
  debug.println(millis());

  dht.begin();                //DHT22
  debug.println("DHT22 Ready");

  xbee.setSerial(xbee_comm);   //XBee using Serial Communication

  rtc.begin();                //RTC
  rtc.disable32K();
  rtc.clearAlarm(1);
  rtc.clearAlarm(2);
  rtc.writeSqwPinMode(DS3231_OFF);

  //interrupts
  attachInterrupt(digitalPinToInterrupt(RTC_INTERRUPT), onAlarm, FALLING);
  //set first alarm
  if(!rtc.setAlarm1(
    rtc.now() + TimeSpan(1),
    DS3231_A1_Hour
  )

```

```

)) {
    debug.println("[TIMER]: Error, alarm wasn't set!");
} else {
    debug.println("[TIMER]: Activate at " +
rtc.now().timestamp(DateTime::TIMESTAMP_FULL));
}
}

//Loop Behavior
void loop() {

    xbee.readPacket(1500);

    //Got a packet
    if (xbee.getResponse().isAvailable()) {

        //Verify Frame Type
        if (xbee.getResponse().getApild() == ZB_RX_RESPONSE) {

            // zb rx packet
            debug.println("[XBEE] Frame Type: ZB_RX_RESPONSE");

            // now fill our zb rx class
            xbee.getResponse().getZBRxResponse(rx);

            //verify options

            //ACK
            if (rx.getOption() == ZB_PACKET_ACKNOWLEDGED) {
                //the sender got an ACK
                debug.println("[XBEE] Got ACK");
            }

            //BROADCAST
            else if (rx.getOption() == ZB_BROADCAST_PACKET) {
                debug.println("[XBEE] Broadcast");
            }

            //ANOTHER OPTION
            else {
                debug.print("[XBEE] option: ");
                debug.println(rx.getOption(),HEX);
            }

            //Get data

```

```

String data;
  for (int i= 0; i < rx.getDataLength(); i++) {
    data += (char)rx.getData(i);
  }
  debug.println("[XBEE] Arrived: "+ data);
}
//Modem Status
else if (xbee.getResponse().getApild() == MODEM_STATUS_RESPONSE) {
  //
  debug.println("[XBEE] Modem Status response");

  xbee.getResponse().getModemStatusResponse(msr);
  // the local XBee sends this response on certain events, like association/dissociation

  if (msr.getStatus() == ASSOCIATED) {
    //ASSOCIATED in network
    debug.println("[XBEE] Associated");
  }
  else if (msr.getStatus() == DISASSOCIATED) {
    //DISASSOCIATED in network
    debug .println("[XBEE] Disassociated");
  }
  else {
    //ANOTHER status
    debug.println("[XBEE] Another Status: ");
    debug.println(msr.getStatus(),HEX);
  }
}
//Delivery Status
else if (xbee.getResponse().getApild()==ZB_TX_STATUS_RESPONSE){
  //ZB_TX_STATUS_RESPONSE
  debug.println("[XBEE] ZB TX STATUS RESPONSE");
  xbee.getResponse().getZBTxStatusResponse(s_tx);
  //ROUTE NOT FOUND STATUS
  if (s_tx.getDeliveryStatus() == ROUTE_NOT_FOUND) {
    debug.println("[XBEE] Route not found");
  }
  //TX SUCCESS
  else if (s_tx.getDeliveryStatus() == SUCCESS) {
    debug.println("[XBEE] Success transmit");
  }
}

//ANOTHER TX STATUS
else{
  debug.print("[XBEE] Delivery Status: ");
  debug.println(s_tx.getDeliveryStatus(),HEX);
}

```

```

}
}
//Other API id
else {
    //Show another API ID
    debug.print("[XBEE] Another API ID");
    debug.println(xbee.getResponse().getApild());
}
}
//Error reading packet
else if (xbee.getResponse().isError()) {
    debug.print("[XBEE] Error reading packet. Error code: ");
    debug.println(xbee.getResponse().getErrorCode());
}
if (rtc.alarmFired(1)){
    rtc.clearAlarm(1);
    //readings from DHT22
    float humidity_dht22 = dht.readHumidity();
    float temperature_dht22 = dht.readTemperature();
    // Check if any reads failed and exit early (to try again).
    if (isnan(humidity_dht22) || isnan(temperature_dht22)) {
        debug.println(F("Failed to read from DHT sensor!"));
        return;
    }
    debug.print("[DHT 22] ");
    debug.print("Temperature: ");
    debug.print(temperature_dht22);
    debug.print("C");
    debug.print("; Humidity: ");
    debug.print(humidity_dht22);
    debug.println("%");
    float humidity_HD_38 = HD_38_read();
    debug.print("[HD-38] ");
    debug.print("Humidity: ");
    debug.print(humidity_HD_38);
    debug.println("%");
}
}

```

```

String message = NODE_ID + ";" +
    String(temperature_dht22) + ";" +
    String(humidity_dht22) + ";" +
    String(humidity_HD_38) + ";" ;
send_xbee(message);
if(!rtc.setAlarm1(
    rtc.now() + TimeSpan(WAKEUP_TIME),
    DS3231_A1_Hour // this mode triggers the alarm when the seconds match. See
Doxygen for other options
)) {
    debug.println("[TIMER]: Error, alarm wasn't set!");
} else {
    debug.println("[TIMER]:    Activate    for    30    minutes!    at    "    +
rtc.now().timestamp(DateTime::TIMESTAMP_FULL));
}
set_sleep_mode(SLEEP_MODE_PWR_DOWN);
sleep_cpu();
}
//uint32_t current_time = millis();
//while (millis() < current_time + WAKEUP_TIME*1000 ){
//delay(WAKEUP_TIME*1000);
}
//Interrupt functions
void onAlarm() {
    sleep_disable();
}

```

Anexo No. 7-21 Código General para el Nodo Central

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <XBee.h>
#include <SPI.h>
#include <SD.h>
#include "RTCLib.h"
// ---- Enviroment Variables ---- /
#define debug          Serial
#define REGULAR_MESSAGE_SPACES  4
#define XBEE_TX        11
#define XBEE_RX        10
#define LCD_ON_TIME    2000
#define SD_CHIPSELECT  49
#define WATERPUMP_PIN  53
#define MAX_ENV_TEMPERATURE  28
#define MIN_ENV_HUMIDITY    70
#define MIN_SOIL_HUMIDITY   50
#define INTERRUPT_BUTTON   0
#define RTC_INTERRUPT       3
const String SEPARATOR =  ";";
uint32_t last_lcd_turn_on;
bool displayed = false;
bool is_water_pump_on = false;
int count_messages = 0;
bool show = false;
bool turndown_waterpump = false;

// ---- Declare Instances --- //
LiquidCrystal_I2C lcd(0x27,16,2);
String message_list[3];

//xbee
XBee xbee= XBee();
ZBRxResponse rx = ZBRxResponse();
ZBTxStatusResponse s_tx = ZBTxStatusResponse();
ModemStatusResponse msr = ModemStatusResponse();
```

```

//rtc
RTC_DS3231 rtc;
//SoftwareSerial xbee_comm(XBEE_RX,XBEE_TX);
// --- Auxiliary Functions --- //
void parse_in_array(String message, String s_array[], String separator){
  int indexes = 0;
  while (message.indexOf(separator)!= -1 ){

    int foundit = message.indexOf(separator);
    s_array[indexes] = message.substring(0,foundit);
    indexes++;
    message = message.substring(foundit+1);
  }
}

void save_datalog(String node_id,String dht_temp, String dht_hum, String HD_38_hum){
  //timestamp
  DateTime time = rtc.now();
  if (!SD.begin(SD_CHIPSELECT)) {
    debug.println("[SD] Card failed, or not present");
  }
  else {
    String dataString = time.timestamp(DateTime::TIMESTAMP_FULL) + ";" +
      node_id + ";" +
      "TEMPERATURA_AMBIENTE: " + dht_temp + " C;" +
      "HUMEDAD_AMBIENTE: " + dht_hum + " %;" +
      "HUMEDAD_SUELO: " + HD_38_hum + " %;" ;
    if (dht_temp.toFloat() > MAX_ENV_TEMPERATURE){
      dataString += "TEMP AMBIENTE CRITICO: SI;";
    }
    else{
      dataString += "TEMP AMBIENTE CRITICO: NO;";
    }
    if (dht_hum.toFloat() < MIN_ENV_HUMIDITY){
      dataString += "HUMEDAD AMBIENTE CRITICA: SI;";
    }
  }
}

```

```

else{
    dataString += "HUMEDAD AMBIENTE CRITICA: NO;";
}
if (HD_38_hum.toFloat() < MIN_SOIL_HUMIDITY){
    dataString += "HUMEDAD SUELO CRITICA: SI;";
}
else{
    dataString += "HUMEDAD SUELO CRITICA: NO;";
}
if (is_water_pump_on) {
    dataString += "BOMBA ACTVADA: SI";
}else{
    dataString += "BOMBA ACTVADA: NO";
}
// open the file.
// so you have to close this one before opening another.
File dataFile = SD.open("datalog.csv", FILE_WRITE);
// if the file is available, write to it:
if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    // print to the serial port too:
    debug.println("[SD]: Writed ->" + dataString);
}
// if the file isn't open, pop up an error:
else {
    debug.println("error opening datalog.csv");
}
}
}

void manage_waterpump(String dht_temp, String dht_hum, String HD_38_hum){
    if(!is_water_pump_on){
        if (HD_38_hum.toFloat() < MIN_SOIL_HUMIDITY){
            if(!rtc.setAlarm1(
                rtc.now() + TimeSpan(3600*2),
                DS3231_A1_Hour
            )
        )
    }
}
}

```

```

)) {
    debug.println("[WATERPUMP]: Error, alarm wasn't set!");
} else {
    debug.println("[WATERPUMP]: Activate for 2 hours! at " +
rtc.now().timestamp(DateTime::TIMESTAMP_FULL));
}
is_water_pump_on = true;
digitalWrite(WATERPUMP_PIN,HIGH);
}
else if (dht_hum.toFloat() < MIN_ENV_HUMIDITY || dht_temp.toFloat() >
MAX_ENV_TEMPERATURE){
    if(!rtc.setAlarm1(
        rtc.now() + TimeSpan(1800),
        DS3231_A1_Minute
    )) {
        debug.println("[WATERPUMP]: Error, alarm wasn't set!");
    } else {
        debug.println("[WATERPUMP]: Activate for 30 min! at " +
rtc.now().timestamp(DateTime::TIMESTAMP_FULL));
    }
    is_water_pump_on = true;
    digitalWrite(WATERPUMP_PIN,HIGH);
}
}
}

void notify_message(int node_num,String dht_temp, String dht_hum, String HD_38_hum){
    String node_id = "";
    switch(node_num){
        case 0:
            node_id = "NODO1";
            break;
        case 1:
            node_id = "NODO2";
            break;
        case 2:
            node_id = "NODO3";

```

```

break;
  default:
    break;
}
//Save time when LCD is ON
last_lcd_turn_on = millis();
//Turn On BackLight.
lcd.backlight();
lcd.print(node_id);
if (!dht_temp.equals("") && !dht_hum.equals("") && !HD_38_hum.equals("")){
  lcd.setCursor(0, 1);
  // Write Enviroment Sensor
  lcd.print("A:T ");
  lcd.print(dht_temp);
  lcd.print("C,H ");
  lcd.print(dht_hum);
  lcd.print("%");

  // Write Soil Sensor
  lcd.setCursor(0, 2);
  lcd.print("S:H ");
  lcd.print(HD_38_hum);
  lcd.print("%");
}else {
  lcd.setCursor(0, 1);
  lcd.print("Aun sin");
  lcd.setCursor(0, 2);
  lcd.print("informacion");
}
}
//Setup
void setup() {

  pinMode(WATERPUMP_PIN, OUTPUT);
  pinMode(RTC_INTERRUPT,INPUT_PULLUP);

```

```

debug.begin(9600);
//xbee_comm.begin(9600);
Serial3.begin(9600);
// Initialize LCD
lcd.init();
//xbee.setSerial(xbee_comm); //XBee using Serial Communication
xbee.setSerial(Serial3);
//initialize RTC
rtc.begin();
rtc.disable32K();
rtc.clearAlarm(1);
rtc.clearAlarm(2);
rtc.writeSqwPinMode(DS3231_OFF);
//interrupts
attachInterrupt(INTERRUPT_BUTTON, display_messages, FALLING);
attachInterrupt(digitalPinToInterrupt(RTC_INTERRUPT), onAlarm, FALLING);
}
//Loop Behavior
void loop() {
  if (rtc.alarmFired(1)){
    rtc.clearAlarm(1);
    debug.println("[WATERPUMP]: turn down waterpump at: " +
rtc.now().timestamp(DateTime::TIMESTAMP_FULL));
    turndown_waterpump = false;
    is_water_pump_on = false;
    digitalWrite(WATERPUMP_PIN,LOW);
  }
  xbee.readPacket(1500);
//Got a packet
if (xbee.getResponse().isAvailable()) {
  //Verify Frame Type
  if (xbee.getResponse().getApild() == ZB_RX_RESPONSE) {
    // zb rx packet
    debug.println("[XBEE] Frame Type: ZB_RX_RESPONSE");
    // now fill our zb rx class
    xbee.getResponse().getZBRxResponse(rx);
  }
}
}

```

```

//verify options
//ACK
if (rx.getOption() == ZB_PACKET_ACKNOWLEDGED) {
    //the sender got an ACK
    debug.println("[XBEE] Got ACK");
}
//BROADCAST
else if (rx.getOption() == ZB_BROADCAST_PACKET) {
    debug.println("[XBEE] Broadcast");
}
//ANOTHER OPTION
else {
    debug.print("[XBEE] option: ");
    debug.println(rx.getOption(),HEX);
}
//Get data
String data;
for (int i= 0; i < rx.getDataLength(); i++) {
    data += (char)rx.getData(i);
}
String s_array [REGULAR_MESSAGE_SPACES];
parse_in_array(data,s_array,SEPARATOR);
String device_id = s_array[0];
String env_temperature = s_array[1];
String env_humidity = s_array[2];
String soil_humidity = s_array[3];
int device_number = String(device_id[4]).toInt();
manage_waterpump(env_temperature, env_humidity, soil_humidity);
save_datalog(device_id,env_temperature,env_humidity,soil_humidity);
debug.println("[XBEE] Arrived: "+ data);
switch (device_number) {
    case 1:
        message_list[0] = data;
        break;
    case 2:
        message_list[1] = data;

```

```

break;
    case 3:
        message_list[2] = data;
        break;
    default:
        break;
}
}
//Modem Status
else if (xbee.getResponse().getApild() == MODEM_STATUS_RESPONSE) {
    //
    debug.println("[XBEE] Modem Status response");
    xbee.getResponse().getModemStatusResponse(msr);
    // the local XBee sends this response on certain events, like association/dissociation
    if (msr.getStatus() == ASSOCIATED) {
        //ASSOCIATED in network
        debug.println("[XBEE] Associated");
    }
    else if (msr.getStatus() == DISASSOCIATED) {
        //DISASSOCIATED in network
        debug .println("[XBEE] Disassociated");
    }
    else {
        //ANOTHER status
        debug.println("[XBEE] Another Status: ");
        debug.println(msr.getStatus(),HEX);
    }
}
}
//Delivery Status
else if (xbee.getResponse().getApild()==ZB_TX_STATUS_RESPONSE){
    //ZB_TX_STATUS_RESPONSE
    debug.println("[XBEE] ZB TX STATUS RESPONSE");
    xbee.getResponse().getZBTxStatusResponse(s_tx);
//ROUTE NOT FOUND STATUS
    if (s_tx.getDeliveryStatus() == ROUTE_NOT_FOUND) {
        debug.println("[XBEE] Route not found");
    }
}
}

```

```

}
//TX SUCCESS
else if (s_tx.getDeliveryStatus() == SUCCESS) {
    debug.println("[XBEE] Success transmit");
}
//ANOTHER TX STATUS
else{
    debug.print("[XBEE] Delivery Status: ");
    debug.println(s_tx.getDeliveryStatus(),HEX);
}
}
//Other API id
else {
    //Show another API ID
    debug.print("[XBEE] Another API ID");
    debug.println(xbee.getResponse().getApild());
}
}
//Error reading packet
else if (xbee.getResponse().isError()) {
    debug.print("[XBEE] Error reading packet. Error code: ");
    debug.println(xbee.getResponse().getErrorCode());
}
else{
    if (show) {
        if (!displayed){
            String data = message_list[count_messages];

            String s_array [REGULAR_MESSAGE_SPACES];
            parse_in_array(data,s_array,SEPARATOR);
String device_id = s_array[0];
            String env_temperature = s_array[1];
            String env_humidity = s_array[2];
            String soil_humidity = s_array[3];
notify_message(count_messages,env_temperature,env_humidity,soil_humidity);
            displayed = true;

```

```
count_messages++;
    if (count_messages == 3){
        show = false;
        count_messages = 0;
        attachInterrupt(INTERRUPT_BUTTON, display_messages, FALLING);
    }
}

if (last_lcd_turn_on + LCD_ON_TIME < millis() ){
    lcd.clear();
    if (!show){
        lcd.noBacklight();
    }
    displayed = false;
}
}
}

//interrupt functions
void display_messages(){
    detachInterrupt(INTERRUPT_BUTTON);
    show = true;
}

void onAlarm() {
    turndown_waterpump = true;
}
```