RED DE SENSORES DE XBEE

INFORME FINAL

Sara Álvarez Bello

[salvarezb@eafit.edu.co](mailto:salvarezb@eafit.edu.co)

Juan Diego Agudelo Balvin

[jdagudelob@eafit.edu.co](mailto:jdagudelob@eafit.edu.co)

Jacobo Chica Quintero

[jchicaq@eafit.edu.co](mailto:jchicaq@eafit.edu.co)

Profesor: Hugo Alberto Murillo Hoyos

LENGUAJES TÉCNICOS DE PROGRAMACIÓN

UNIVERSIDAD EAFIT

ESCUELA DE CIENCIAS APLICADAS E INGENIERÍA

Noviembre de 2023

# INTRODUCCIÓN

En el ámbito de las comunicaciones inalámbricas, la tecnología XBee ha demostrado ser una opción robusta y eficiente para el intercambio de datos en entornos diversos. Este informe se centra en la implementación de un servidor compuesto por tres placas XBee, utilizando tramas API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés) como protocolo de comunicación.

Las placas XBee, desarrolladas por Digi International, son dispositivos compactos y versátiles que utilizan el estándar Zigbee para facilitar la comunicación inalámbrica. La elección de implementar un servidor con tres placas XBee abre un abanico de posibilidades para aplicaciones que requieren una transmisión confiable de datos entre múltiples nodos.

El uso de tramas API agrega una capa de control avanzada a la comunicación entre los nodos, permitiendo una mayor flexibilidad y precisión en el manejo de los datos transmitidos. A lo largo de este informe, se explorarán los aspectos clave de la implementación, desde la configuración inicial hasta la manipulación de tramas API para lograr una comunicación eficiente y segura entre el servidor y los dispositivos remotos.

En resumen, la implementación de un servidor XBee con tres placas utilizando tramas API representa un paso significativo hacia la creación de una red inalámbrica confiable y eficiente. Este informe guiará al lector a través de los detalles técnicos esenciales para comprender y aplicar este enfoque en sus propios proyectos de comunicación inalámbrica.

# OBJETIVOS

## Objetivo general:

Implementar un sistema de sensores remotos que se comuniquen a través del módulo inalámbrico “Xbee’s” para adquirir y enviar señales a diferentes actuadores para que respondan de una forma determinada.

## Objetivos específicos:

* Crear las maquetas de los sensores y actuadores que se distribuirán en diferentes zonas, instalando una red de sensores que establezca la conexión por medio de la unidad Xbee’s
* Comprobar el funcionamiento de la red inalámbrica Xbee’s mediante la realización de una serie de pruebas que verifiquen cada uno de los componentes de la planta.

# JUSTIFICACIÓN

Las redes de sensores están formadas por un grupo de sensores que adquieren información a partir de una comunicación inalámbrica. Para llegar a esto, se tienen diferentes protocolos como Bluetooth, WIFI – IEEE 802.11x, IEEE 802.15.4, IEEE 802.15.6, ZigBee y WirelessHart, el uso de cada una de estas tecnologías varia en las diferentes aplicaciones, programación, módulos, configuración, entre otras.

Por lo que, para esta aplicación se escoge el protocolo ZigBee, pues los nodos de sensores son pequeños, tienen la posibilidad de cambiar baterías, pueden tener varías funcionalidades (pueden ser Coordinador o Router), puede tener una gran densidad de nodos, manejan una tasa de transmisión de máximo 250 kbps, tienen un rango amplio y el acceso a sensores es fácilmente reemplazable.

Todas las características anteriores son precisas para la aplicación del presente proyecto, además de que la siguiente figura, presenta un diagrama de cómo sería el esquema general de red, relacionado con los objetivos de la red de sensores por el módulo Xbee.

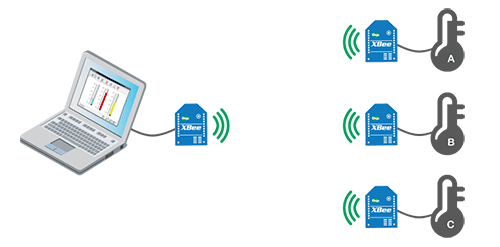


Fig. 1. Protocolo de red inalámbrica.

# MARCO TEORICO

La implementación de un servidor compuesto por tres placas XBee mediante tramas API se basa en una serie de conceptos y tecnologías clave que son fundamentales para comprender el funcionamiento de este sistema de comunicación inalámbrica. A continuación, se presenta el marco teórico que sustenta esta implementación:

## XBee y Tecnología ZigBee:

* **XBee:** Desarrollado por Digi International, XBee es una familia de módulos de comunicación inalámbrica que utiliza el estándar Zigbee. Estos módulos proporcionan una solución eficiente y confiable para la transmisión de datos en redes de área personal y local.

Los módulos XBee son dispositivos que integran un transmisor – receptor de ZigBee y un procesador en un mismo modulo, lo que permite a los usuarios desarrollar aplicaciones de manera rápida y sencilla. Los módulos XBee son versátiles a la hora de establecer diversas topologías de red, dependiendo de la serie XBee que se adquiera pueden crearse redes.

* **ZigBee:** Es un protocolo de comunicación inalámbrica de bajo consumo energético y alta eficiencia diseñado para la interconexión de dispositivos en redes de área personal.

Con el fin de agregar más información, se presenta con más detalle cada componente. En el caso del módulo Xbee, este se alimenta con 3.3 V en el pin 1, en el pin 2 se tiene la comunicación Tx, en el pin 3 la comunicación Rx y en el pin 10 la tierra GND. El resto de pines están distribuidos entre operaciones preestablecidas y puertos A/D.



Fig. 2. Módulo Xbee.

A parte de la visualización del módulo, los puertos del módulo son de gran interés para el presente proyecto, por lo que en la siguiente figura se presentan.

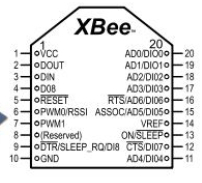


Fig. 3. Salidas y entradas del Módulo Xbee.

De las cuales, las más importantes para la comunicación son las entradas AD0 – 3, en donde se pueden conectar las salidas de los sensores de interés para la red de sensores inalámbrica o el valor analógico a transmitir.

## Modo API (Interfaz de Programación de Aplicaciones):

La utilización del modo API en los módulos XBee permite una mayor flexibilidad en el manejo de datos. En este modo, la información se envía en forma de tramas estructuradas, que contienen campos específicos para la identificación del tipo de mensaje, dirección del destinatario, datos y sumas de verificación. De este modo, se facilita el control preciso sobre el proceso de comunicación, permitiendo a los desarrolladores manipular directamente los paquetes de datos que se envían y reciben.

En el modo API, los datos se encapsulan en paquetes con un encabezado y una estructura específica, lo que permite una comunicación más estructurada y versátil entre los dispositivos. Esto es diferente al modo UART (Universal Asynchronous Receiver/Transmitter), que es el modo por defecto en los módulos XBee y que envía los datos de manera directa, sin estructura adicional.

El modo API ofrece varias ventajas:

* Control mejorado: Permite enviar y recibir no solo datos, sino también comandos y configuraciones específicas a través de paquetes con estructuras definidas.
* Gestión de errores: Los paquetes API contienen información adicional que puede utilizarse para verificar la integridad de los datos, lo que ayuda en la detección y corrección de errores de transmisión.
* Notificación de estado: Proporciona información de estado sobre la entrega de los datos, lo que permite a los dispositivos emisores recibir confirmación de que los datos se han recibido correctamente.
* Capacidad de manejar múltiples tipos de datos: Permite la transferencia de datos complejos, como comandos de configuración, datos de sensores, mensajes de control, etc.

El modo API tiene dos submodos principales: el modo API transparente y el modo API explícito. En el modo API transparente, los datos se envían sin estructura adicional, similar al modo UART. Por otro lado, en el modo API explícito, se emplean paquetes con estructura específica que contienen información adicional además de los datos a enviar.

## Comunicación entre Nodos XBee:

La implementación de un servidor con tres placas XBee implica la configuración de un nodo central (maestro) y dos nodos remotos (esclavos), esto es posible, gracias a herramientas como XCTU (Xbee Configuration and Test Utility) proporcionada por Digi International, el fabricante de los módulos XBee.

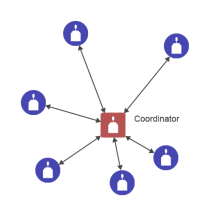


Fig. 4. Comunicación entre nodos.

Igualmente, en la siguiente figura se proporciona una ejemplificación de la forma de comunicación entre los Xbee en modo API.

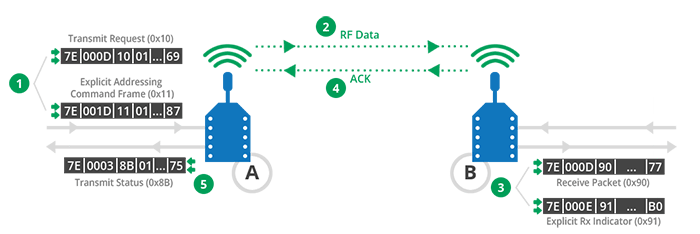


Fig. 5. Transmisión y adquisición de datos.

## Manejo de Tramas API:

Cada trama API consta de varios campos, como el delimitador, la longitud, el tipo de trama, datos, y la suma de verificación. El conocimiento detallado de la estructura de estas tramas es esencial para interpretar y generar mensajes de manera adecuada.

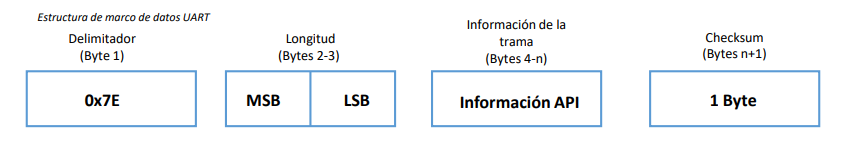


Fig. 5. Estructura general de las tramas API.

En la figura anterior, se presenta la estructura general, en donde el delimitador, determina el inicio de la trama; la longitud, se refiere a qué tan larga es la trama; la información de la trama es el paquete de datos a transmitir; y el Checksum, es un byte de verificación de trama enviada. Además, también hay paquetes que indican el tipo de trama y la dirección de destino dadas en la longitud de la trama. A continuación, se presenta un ejemplo de cómo se organiza una trama API completa.

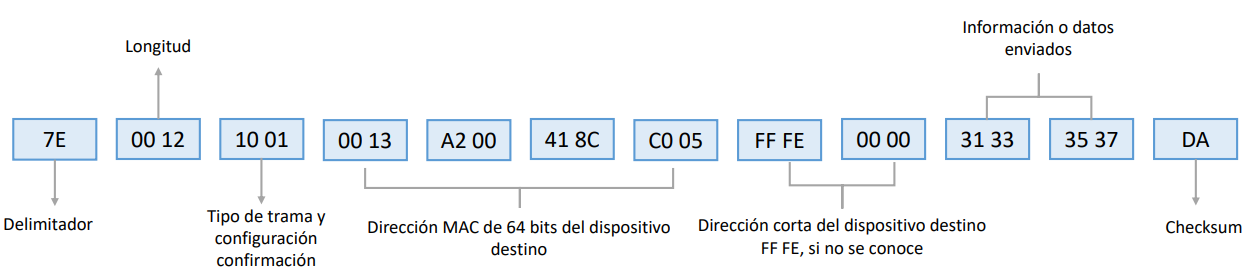


Fig. . Ejemplo de trama API.

Una vez se tiene presenta dicha información para la comunicación de las tramas, se propone la utilización del programa de computo Python para la programación de dicha serie inalámbrica, con la librería digi.xbee.

Este marco teórico sienta las bases para comprender los aspectos esenciales de la implementación de un servidor XBee con tramas API. En las secciones siguientes de este informe, se proporcionará una guía práctica para la configuración y operación de este sistema de comunicación inalámbrica.

# METODOLOGÍA

Para el presente proyecto, se tienen una guía de pasos a seguir para realizar la configuración inicial de los Xbee, teniendo en cuenta los parámetros de red necesarios para establecer comunicación. A continuación, se expondrá una secuencia de pasos para dicha configuración.

* **Requisitos de configuración:** 
  + XCTU instalado en tu computadora.
  + Adaptadores o módulos XBee conectados a la computadora a través de USB.
* **Pasos:**
  + Conexión física: Conecta cada módulo XBee a tu computadora utilizando adaptadores shields XBee conectados por USB.
  + Abre XCTU: Abre el software XCTU en tu computadora.
  + Identificación de los módulos: En la pestaña "Discover Devices" de XCTU, presiona el botón de búsqueda para detectar los módulos XBee conectados a la computadora.
  + Configuración de los módulos: Selecciona uno de los módulos detectados y haz clic en el botón "Add selected devices to My XBee List". Repite este proceso para cada módulo que desees configurar.
  + Configuración del módulo maestro (coordinador): Selecciona el módulo XBee que actuará como maestro (coordinador). En la pestaña "Configuration", selecciona "API Enable" en el menú desplegable "API Enable". Establece el parámetro "CE" (Coordinator Enable) en el valor 1 (esto lo encontrarás en la sección "Function Set"). Configura otros parámetros según las necesidades de tu red (por ejemplo, PAN ID, canal, etc.).
  + Configuración de los módulos esclavos: Selecciona cada uno de los módulos XBee que actuarán como esclavos. En la pestaña "Configuration", también activa "API Enable" en el menú desplegable "API Enable". Establece el parámetro "CE" (Coordinator Enable) en el valor 0 para configurarlos como dispositivos finales (esto lo encontrarás en la sección "Function Set"). Configura los parámetros de red (PAN ID, canal, etc.) para que coincidan con el módulo coordinador.
  + Aplicar cambios y escribir configuración: Una vez que hayas realizado todas las configuraciones necesarias en cada módulo, haz clic en el botón "Write" en XCTU para escribir la configuración en los módulos.
  + Verificación y pruebas: Desconecta los módulos XBee de la computadora. Conecta cada módulo a su fuente de alimentación correspondiente y verifica que los módulos esclavos se conecten correctamente al módulo maestro. Puedes usar la función de consola de XCTU para enviar comandos API y verificar la comunicación entre los módulos.
  + 9. Ajustes adicionales: Si experimentas problemas de comunicación, verifica las configuraciones y los parámetros de red para asegurarte de que estén correctamente sincronizados en todos los módulos. Siguiendo estos pasos, deberías poder configurar tres módulos XBee en modo esclavo y uno como maestro en modo API mediante XCTU. Recuerda que estos pasos son generales y pueden variar ligeramente dependiendo de la versión de XCTU o de los modelos específicos de los módulos XBee que estés utilizando.

Una vez se establece la comunicación se procede con la inclusión al código en Python, en donde se establecen los parámetros configurados con anterioridad y se procede a destinar funciones a cada Xbee.

# RESULTADOS

En esta sección, se dispondrán las configuraciones utilizadas en la sección anterior, para presentar un reporte del proceso seguido en el proyecto, de esta forma, se tendrá un seguimiento de cada paso y de los resultados obtenidos.

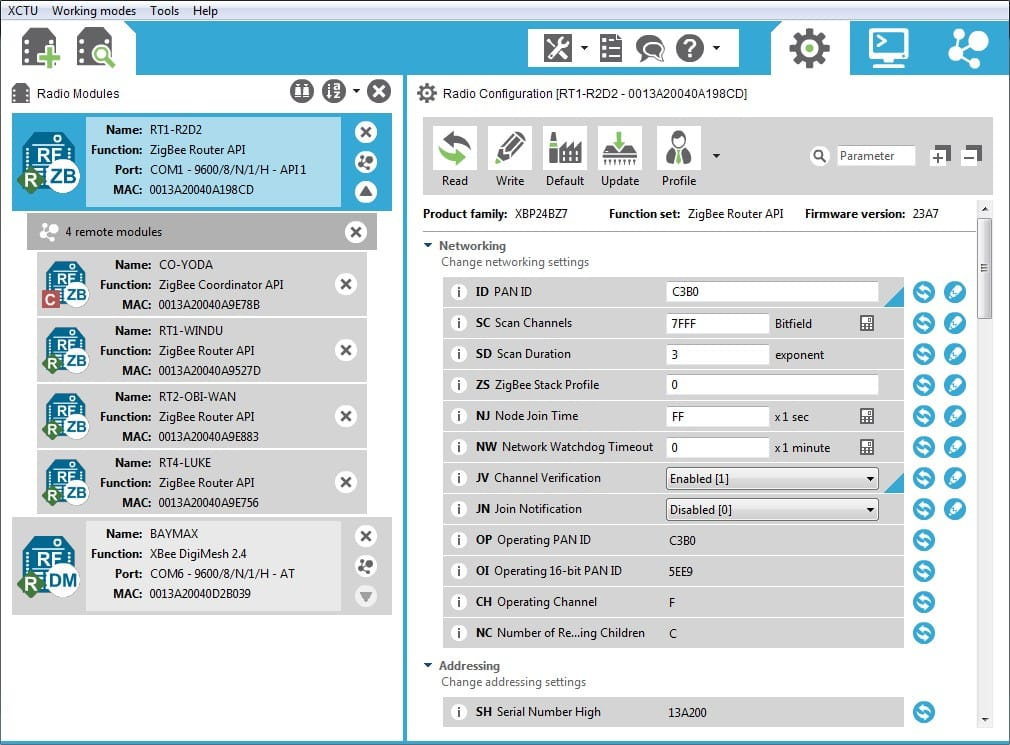


Fig. . Interfaz gráfica de XCTU.

En la figura anterior, se presenta la interfaz gráfica de XCTU, la aplicación para configurar los módulos XBee. En la cuál podemos observar, las secciones para buscar y agregar módulos, y la sección para configurarlos. Una vez se configuran, se procede a verificar la comunicación de la red. En la siguiente figura se obtiene dicho esquema.

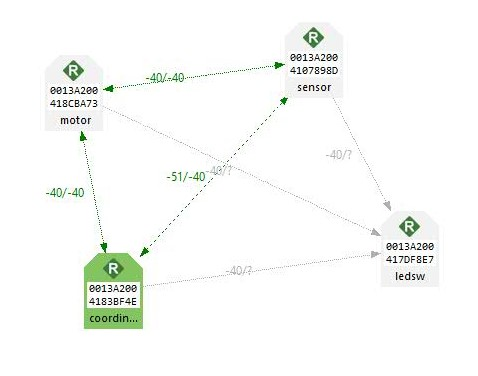


Fig. . Red Inalámbrica.

En esta, se puede verificar, la trama API que identifica cada Xbee, el nombre de cada uno y se resalta en verde el Xbee coordinador o maestro. De esta forma, se considera que hay conexión entre los Xbee y a partir de aquí se pueden programar en Python.

Una vez se tiene la comunicación entre las Xbee, se procede a conectarlas en los reguladores que soportan los módulos. A continuación, se presentan los modulos de los nodos de la red.

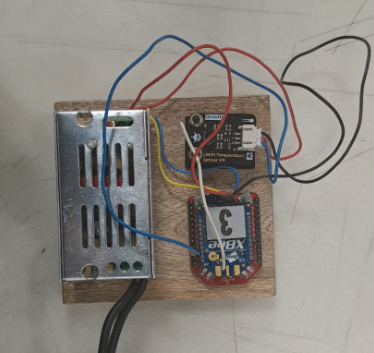


Fig. . Nodo Xbee “sensor”.

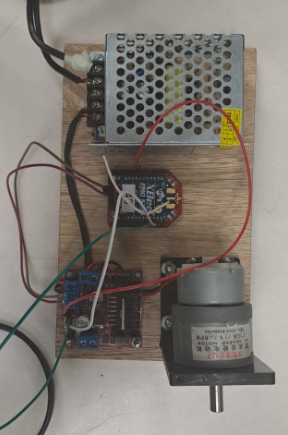


Fig. . Nodo Xbee “motor”.

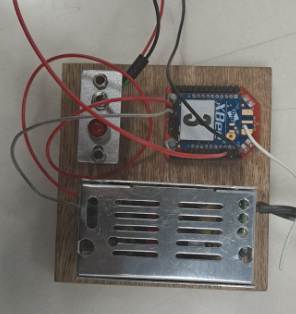


Fig. . Nodo Xbee “ledsw”

A partir de este momento, se procede con la implementación del código, el cual fue basado en una máquina de estados finitos, y esta es:

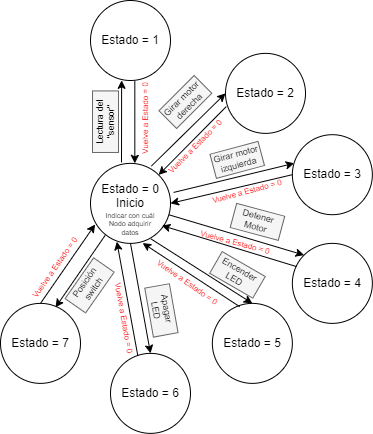


Fig. . Máquina de estados finitos.

Con el anterior diagrama, se evidencia la teoría del código en cuestión, de tal forma que luego de escoger el nodo a utilizar, se procede a realizar la acción respectiva de cada uno.

# IMPLEMENTACIÓN EN PYTHON

En esta sección se expone la implementación en Python de la red inalámbrica por medio de una explicación detallada de cada una de las funciones y variables del código fuente.

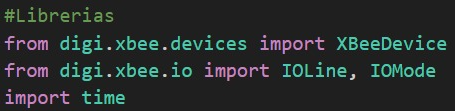


Fig. . Librerías utilizadas.

La librería “digi.xbee.devices” es necesaria para controlar los dispositivos Xbee dentro de la red API, la librería “digi.xbee.io” se utiliza para manejar las terminales análogas, digitales y de PWM, y la librería “time” para general tiempos muertos dentro del código.

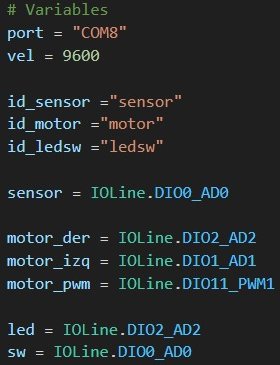


Fig. . Variables.

Las variables “port” y “vel”, son respectivamente, el puerto de la Xbee coordinadora (local) y la velocidad de datos en la red (bps “baud rate”).

Las identificaciones id son los nombres que se le dan a cada una de las Xbee remotas. Por último, el resto de variables son las entradas y salidas de cada uno de los actuadores.

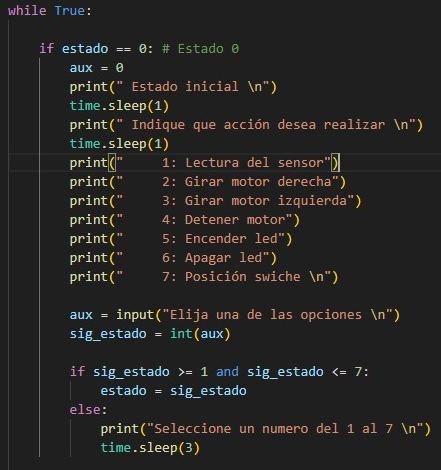


Fig. . Máquina de estado finito.

La máquina de estado finito, es un bucle infinito en el que cada estado es llamado por las funciones “IF”. En el Estado = 0, se le pregunta al usuario, cuál de los actuadores quiere utilizar y lleva la MEF al estado correspondiente.

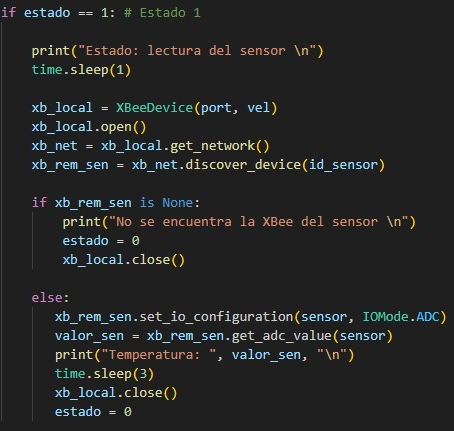


Fig. . Estado = “sensor”.

Creamos el objeto “xb\_local” con el puerto y velocidad seleccionados, después, abrimos la comunicación y la network API. En segundo lugar, se crea el objeto “xb\_rem\_sen” que hace referencia a la Xbee remota conectada al actuador que debe de estar en la red API con el nombre previamente seleccionado. Si no se encuentra dicha Xbee, procede a imprimir un mensaje de alerta y volver al Estado = 0, pero si encuentra la Xbee, selecciona el pin del sensor como ADC (analog digital conversor) e imprime este valor, luego vuelve al Estado = 0.

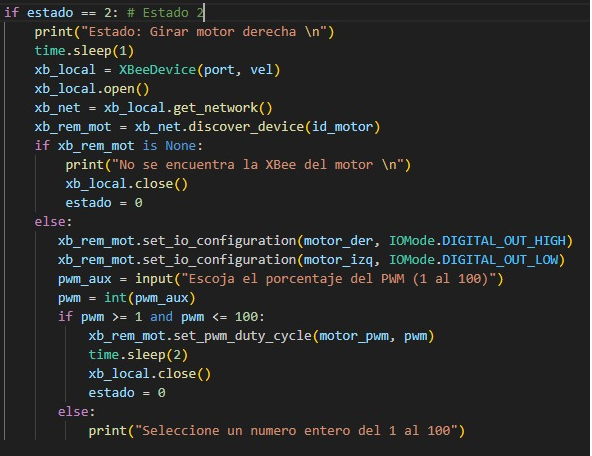


Fig. . Estado = “motor derecha”.

Todos los estados tienen el mismo protocolo de encontrar la Xbee remota que el Estado = 1 “sensor”. En este estado, se llaman las dos variables que controlan la dirección del motor como salidas digitales, después, se le pide al usuario que introduzca el valor porcentual del PWM (1 a 100), luego se le asigna este valor a la salida PWM de la Xbee “motor”.

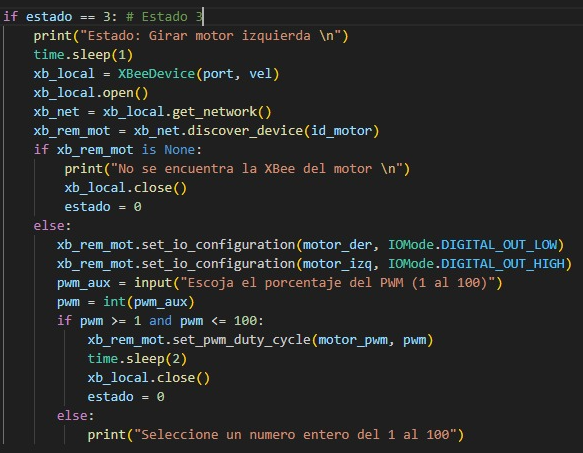
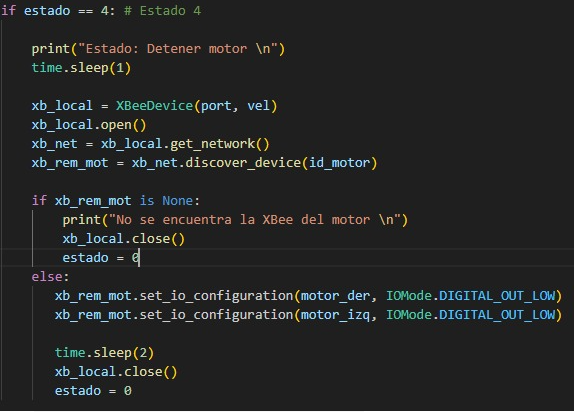


Fig. . Estado = “motor izquierda” y Estado = “Detener motor”.

El Estado “3” y “4” son muy similares al estado “2”, ya que se diferencian en que uno solo cambia la dirección de motor y el otro lo apaga, respectivamente.

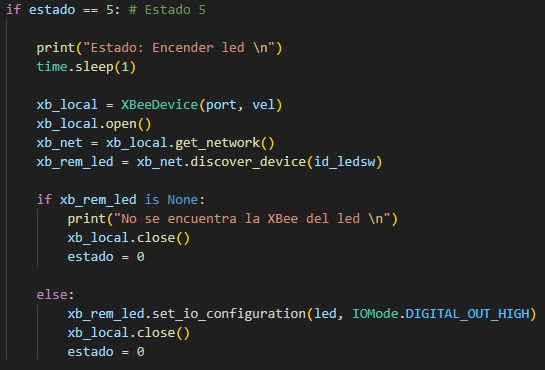


Fig. . Estado = “Encender LED”.

En este estado se abre la comunicación la Xbee remota “ledsw” y se posiciona la salida del led en Alto (High). El Estado = “Apagar LED” es lo mismo, solo cambia el High por Low (de encendido a apagado).

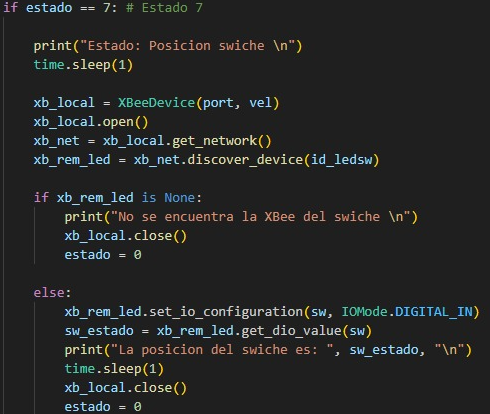


Fig. . Estado = “Posición sw”.

En este estado, se selecciona el pin “sw” como una entrada digital, en la cuál se lee el valor del actuador y se imprime en la consola, y vuelve al Estado = 0 al finalizar, volviendo así, a continuar con el bucle.

# CONCLUSIONES

* Los módulos Xbee son una gran herramienta para controlar actuadores a distancia, lo que permite comunicar varios dispositivos con un solo Xbee maestro, por lo que las redes inalámbricas son un avance en la forma de utilizar los dispositivos.
* El modo API para controlar la red inalámbrica permite crear con mayor flexibilidad la red, de forma que permite la configuración de forma intuitiva y rápida de los nodos Xbee.
* En el proyecto, se obtuvieron resultados variados, pues la comunicación es rápida para obtener resultados instantáneos, y en cambio la programación en Python, permitió la configuración de los actuadores de los nodos, sin embargo, la información que se tiene de las librerías no es intuitiva, por lo que su aplicación se complica.