

Apéndice A

1) Conexión de un dispositivo a un computador.

El módulo XBee se conecta mediante la placa EXPLORER USB y un cable mini USB al computador, la información se transmite mediante un puerto serial, si el computador lo requiere se debe actualizar el *driver* del puerto serial. Si el equipo reconoce el módulo, se despliega la pantalla que se ve en la figura 1, donde se debe buscar y seleccionar el dispositivo.

En la primera pantalla se debe buscar el dispositivo conectado, esto se realiza mediante el botón de *Discover devices*, posteriormente se selecciona el puerto COM al cual está conectado el módulo, después se configuran algunos parámetros iniciales, generalmente se dejan por defecto. Este proceso garantiza que el dispositivo sea controlado por XCTU.

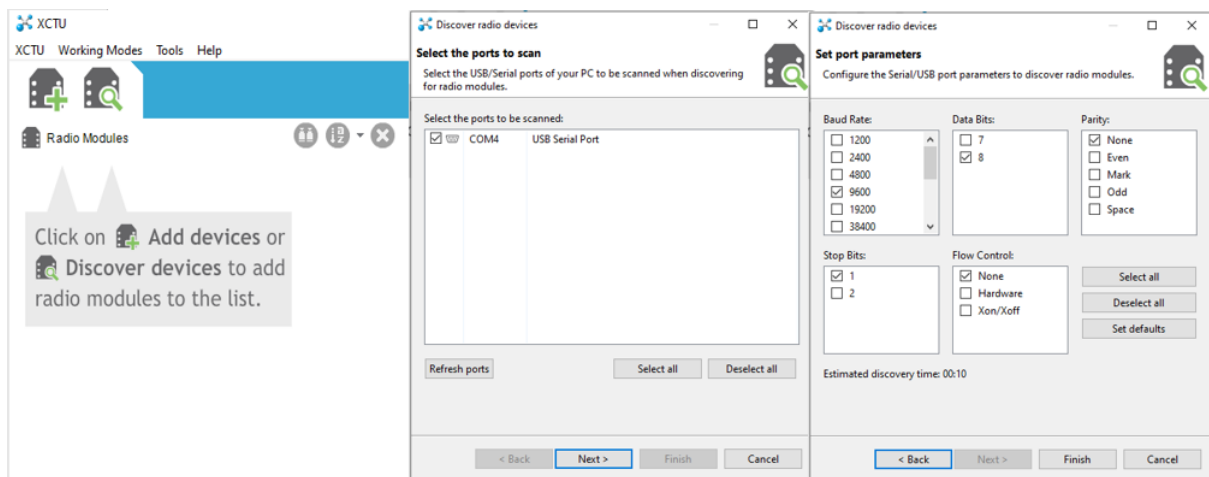


Figura 1: Configuración para agregar módulo en XCTU.

Después de este proceso, XCTU agrega el dispositivo y despliega la pantalla de todas las configuraciones y parámetros que se pueden modificar, en la figura 2 se muestra la pantalla inicial de XCTU.

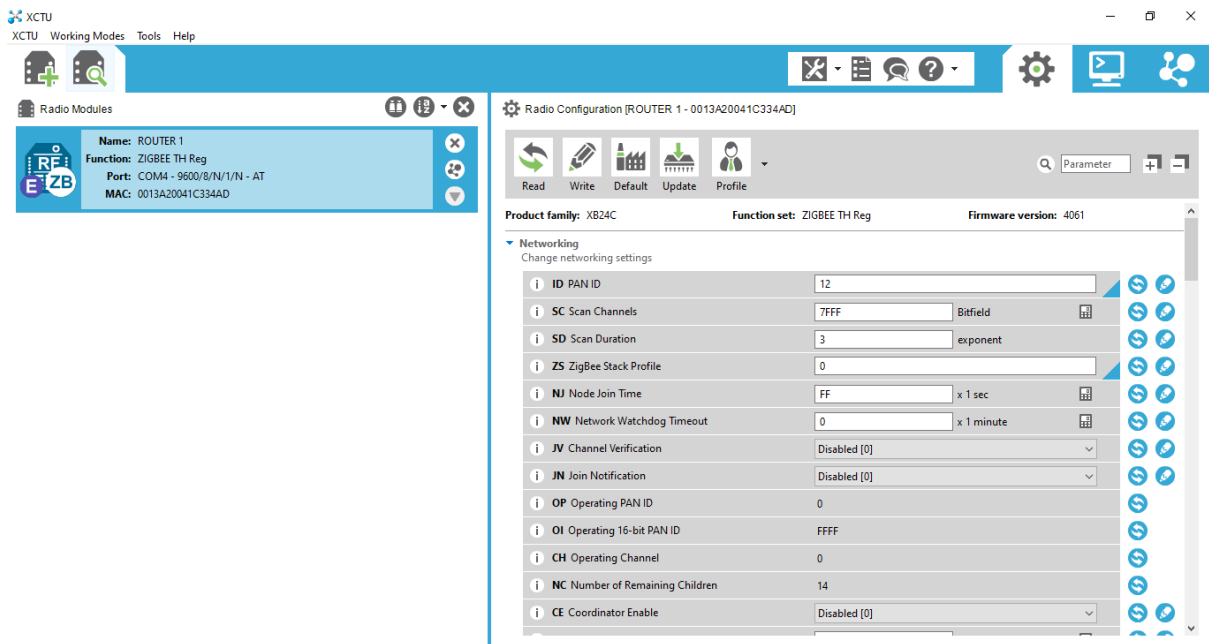


Figura 2: Pantalla inicial en XCTU.

2) Configuración de los parámetros

La figura 2 denota la interfaz inicial para modificar los parámetros de los módulos XBee, en la parte izquierda se puede ver el dispositivo conectado, con su respectivo nombre, el puerto al cual está conectado y su dirección MAC que es su identificador. En la parte superior derecha aparecen algunas opciones como *Read* que es la encargada de leer la configuración actual, *Write* escribe o guarda la información modificada, *Default* restaura las opciones de fábrica, *Update* actualiza el *firmware* y *Profiles* gestiona los diferentes perfiles creados. En la parte inferior derecha se muestran los parámetros modificables que tienen los módulos XBee, generalmente los parámetros que se modifican son pocos y varían de acuerdo con el tipo de perfil que se vaya a emplear. A continuación, se especifican los parámetros más importantes.

- ID (Pan ID): Es el identificador para la red, todos los dispositivos dentro de la red deben tener el mismo ID.
- JV (*Channel Verification*): Este parámetro sirve para activar el modo router en los dispositivos, 1 es activado, 0 desactivado.
- CE (*Coordinator Enable*): Es la configuración para activar el modo coordinador, 1 es activado, 0 desactivado.
- SM (*Sleep Mode*): Es el parámetro para establecer el modo *sleep* o *end device*, 1 es activado, 0 desactivado.
- DH (*Destination Address High*) y DL (*Destination Address Low*): Es la asignación de direcciones MAC para la comunicación entre dispositivos, existen 3 formas de comunicación. Punto a punto, aquí se debe poner la dirección MAC del otro dispositivo al cual se desea conectar. Conexión a coordinador, de esta forma el dispositivo solo se podrá comunicar con el coordinador y para esto se debe colocar 0 en ambos parámetros. Comunicación *broadcast*, el dispositivo emite en todos sentidos, la información

puede ser leída por los router y el coordinador, los caracteres designados para realizar esta acción es FFFFF en ambos parámetros.

- AP (*API Enable*): Este parámetro sirve para configurar el tipo de comunicación, el modo transparente permite una conexión serial mediante comandos por la consola de XCTU, el modo API brinda la capacidad de comunicarse mediante tramas.
- PINES I/O (DIGITALES/ANALÓGICOS): estos pines pueden ser configurados dependiendo la necesidad.

3) Conformación y configuración de trama ZigBee

De esta forma se configuran los parámetros del nodo destino, sin embargo, se debe conformar la trama que va a transmitir el ND a los nodos referencia, esta tarea se puede realizar desde XCTU, donde se conforman las tramas y sus parámetros. Para esto se selecciona la opción de herramientas y se escoge el generador de tramas, que despliega una ventana la cual permite seleccionar el tipo de trama que se desea conformar, en la figura 3 se muestran estas ventanas.

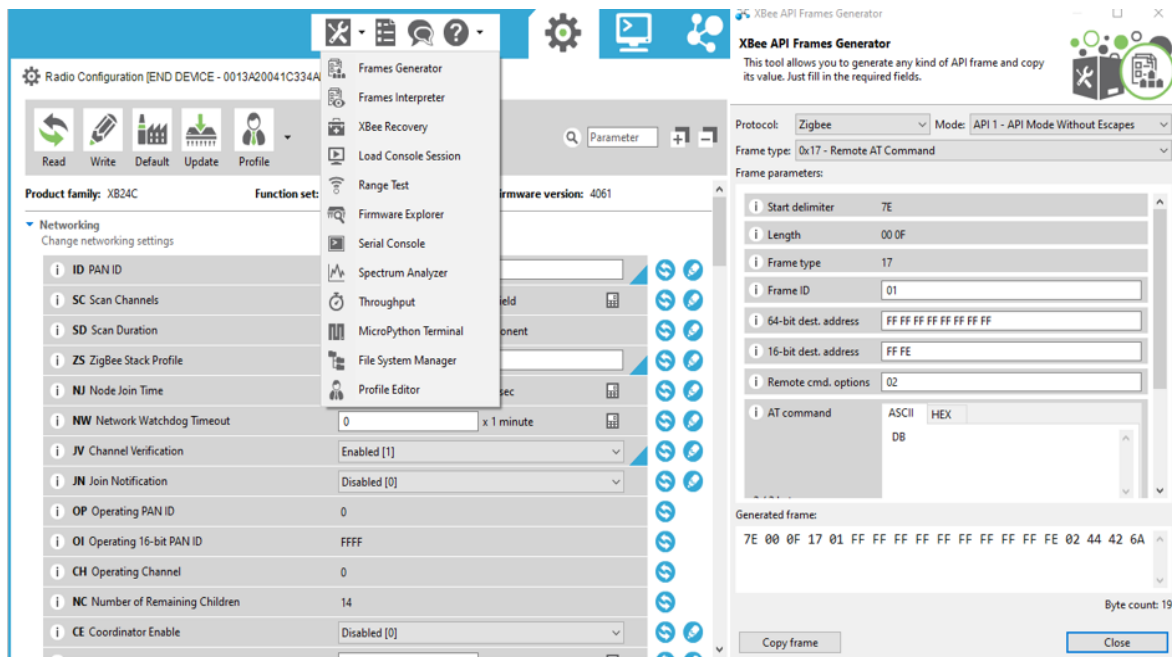


Figura 3 Ventana para generación de tramas.

El generador de tramas API permite seleccionar el protocolo y el tipo de trama, dependiendo el tipo, varía la longitud de bytes transmitidos. Para este caso, se debe seleccionar el protocolo ZigBee y el tipo de trama *Remote AT Command*, ya que esta permite incluir el valor RSSI. Una vez conformada la trama, los campos se especifican en la tabla 1.

Bytes	Descripción
7E	Delimitador: inicio del paquete
00 0F	Longitud: número de bytes

17	Tipo de mensaje: 0x17 Remote AT Command
01	Trama ID: 0x01 se solicita respuesta
FF FF FF FF FF FF FF FF	Dirección de destino 64-bits: MAC de dispositivo destino
FF FE	Dirección de destino 16 bits.
02	Opción del comando remoto: 0x02 para guardar cambios
44 42	Nombre del comando AT: DB en ASCII
6A	Checksum: suma de comprobación

Tabla 1: Descripción de los bytes en trama API.

El delimitador de inicio, es el byte que anuncia el inicio de la trama, generalmente su valor no cambia y es 7E. El tamaño de la trama varía dependiendo la carga útil que se transmita. El tipo de trama se especifica anteriormente, pero se debe establecer, cambiando su nombre por un valor numérico. La dirección de destino de 64 bytes especifica la dirección MAC del destinatario, se pueden emplear las 3 formas ya descritas. La dirección de destino de 16 bytes es el resumen de la dirección destino de 64 bytes. Las opciones de comando remotas usualmente no cambian y se dejan por defecto. El comando AT es el encargado de llevar el comando o sigla que permite realizar diferentes acciones, en este caso se emplea las siglas DB para transmitir el valor RSSI de la comunicación. La carga útil es el lugar donde se almacena la información de la transmisión. Finalmente, la suma de comprobación es la encargada de corroborar que no haya errores dentro de la trama. Establecidos los parámetros, las configuraciones y los dispositivos que componen el nodo destino, se conforma el ND como se ve en la figura 4.

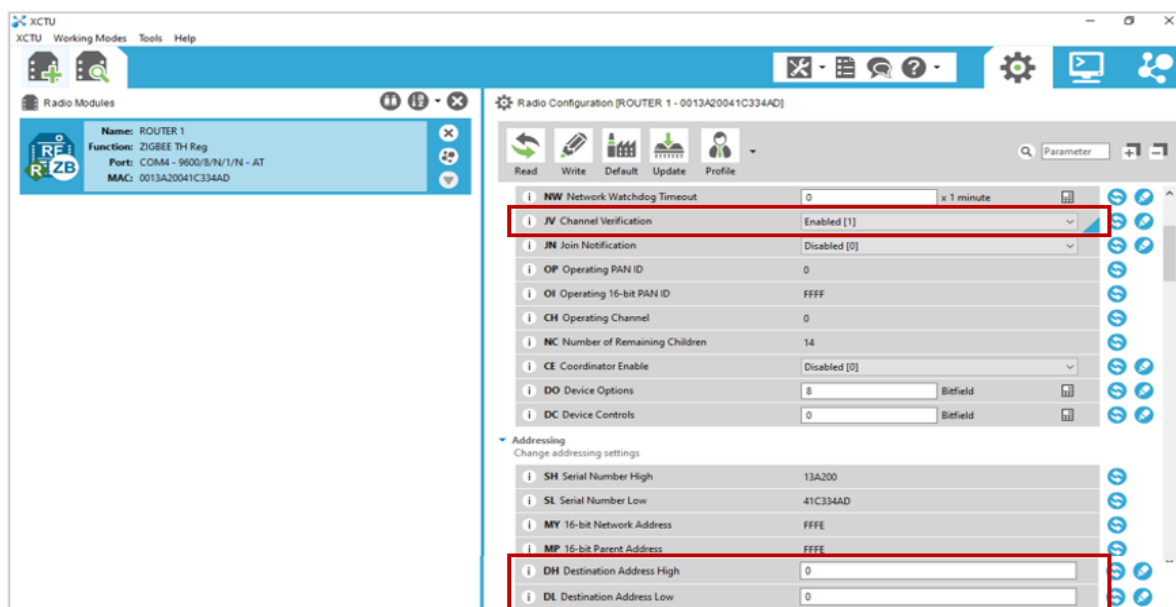


Figura 4: Configuración inicial

- Configuraciones para nodo referencia
- ID (*PAN ID*): Es el identificador de la red, se coloca el mismo para todos los dispositivos, para este caso 12.
- JV (*Channel Verification*): Este parámetro sirve para activar el modo router en los dispositivos, se establece en 1.
- DH y DL: los router dentro del sistema se deben conectar directamente con el nodo procesamiento o coordinador de la red, por esto se emplea el prefijo 0 en ambos campos para especificar la comunicación directa.
- AP: modo API.

La figura 5 muestra las configuraciones realizadas para el nodo referencia.

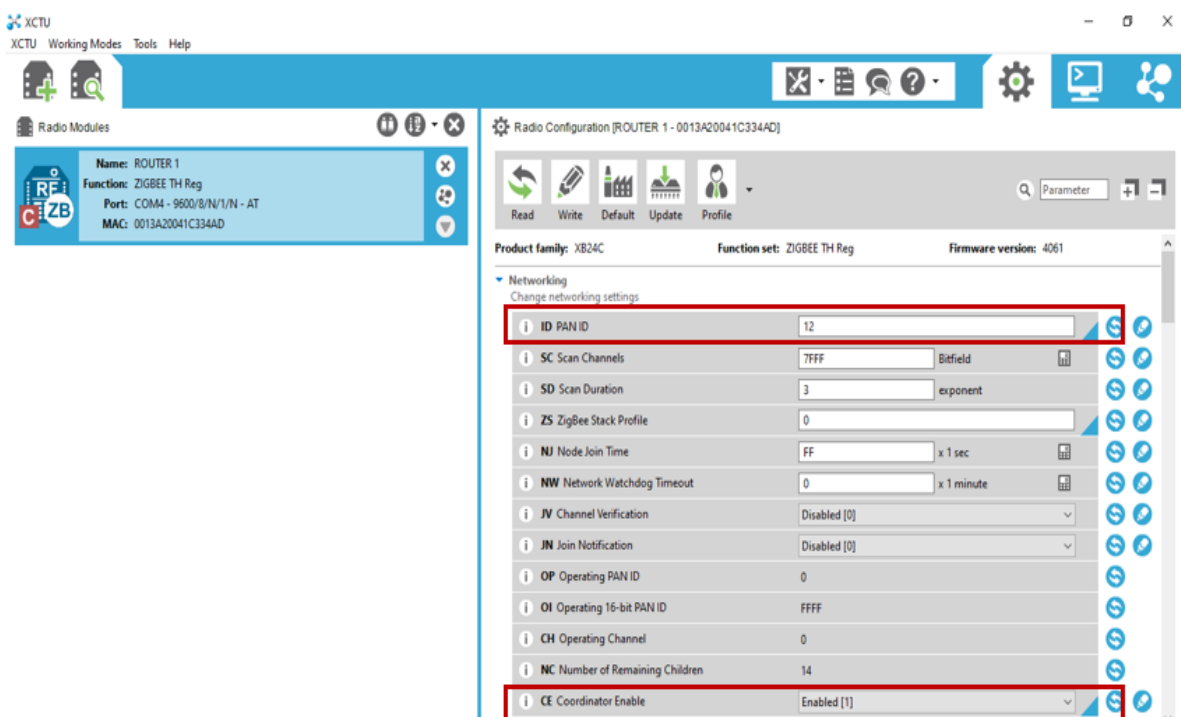


Figura 5: Configuraciones para nodo referencia

4) Conexión entre Arduino y ZigBee

Arduino puede controlar cualquier tipo de dispositivo ZigBee gracias a su compatibilidad con las librerías XBee, esto permite realizar múltiples tareas basadas en radiofrecuencia. Para la conexión entre XBee y Arduino se emplea un puerto serial y se declara la librería *Software Serial*, este proceso se muestra en la figura 6.

```
//Conexión
//Declaración de librerías
#include <SoftwareSerial.h>
SoftwareSerial Xbee(11,10);
```

Figura 6: Conexión entre Arduino y XBee.

Sobre el puerto serial se incluye la librería XBee y se inicializan los pines que se van a utilizar en la tarjeta Arduino, para este caso se emplean los pines 10 y 11. Después de esto se hace la configuración de los puertos serial con su respectivo retardo, estas modificaciones se observan en la figura 7.

```

//Configuración de puertos serial
//Se inicializan los puertos
void setup() {

    Serial.begin(9600);
    Xbee.begin(9600);
    delay(1000);
}

```

Figura 7: Configuración de los puertos serial.

5) Código de trama parametrizada en Arduino

XCTU no exporta datos a otros entornos de desarrollo, por esto el valor RSSI de las tramas provenientes de los NR no pueden ser separadas. Para obtener el valor RSSI de una comunicación entre un ND y los NR, es necesario configurar desde Arduino una petición a los NR, esto se realiza parametrizando una trama API con las direcciones MAC de cada uno de los NR. Como el NR tiene almacenada la trama API recibida desde el ND, el nodo procesamiento solo debe generar la misma trama cambiando las direcciones MAC de los router o nodos referencia. Para esto se emplea el comando Xbee.write (“parámetro”) como se observa en la figura 8.

```

#include <SoftwareSerial.h>
SoftwareSerial Xbee(11,10);
char Data;
void NR1();
void NR2();
void Dato_Xbee();
}
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
    Serial.begin(9600);
    Xbee.begin(9600);
    delay(1000);
}
void loop() {
    // put your main code here, to run repeatedly:

    if(Serial.available()>0){
        Data=Serial.read();
        if(Data=='A'){
            //for (int j=0;j<20;j++){
                NR1();
                delay(100);
                Dato_Xbee();
            }
        }
    }

    if (Data=='B'){
        //digitalWrite(LED_BUILTIN, HIGH);
        NR2();
        delay(100);
        Dato_Xbee();
    }
}
//*****CIERRE DE CICLO*****
//+++++FUNCIONES+++++//

void Dato_Xbee(){
    if (Xbee.available()){
        Data=Xbee.read();
        Serial.print(Data,HEX);
        Serial.print(",");
        if(Data==0x7E){
            for (int i=0;i<19;i++){
                Data=Xbee.read();
                Serial.print(Data,HEX);
                Serial.print(",");
            }
            Serial.println();
        }
    }
}

```



```

void NR1() {
  Serial.print("A,");
  Xbee.write(0x7E);
  Xbee.write((byte)0x0);
  Xbee.write(0x0F);
  Xbee.write(0x17);
  Xbee.write(0x01);
  Xbee.write((byte)0x0);
  Xbee.write(0x13);
  Xbee.write(0xA2);
  Xbee.write((byte)0x0);
  Xbee.write(0x41);
  Xbee.write(0xC3);
  Xbee.write(0x33);
  Xbee.write(0x6F);
  Xbee.write(0xFF);
  Xbee.write(0xFE);
  Xbee.write(0x02);
  Xbee.write(0x44);
  Xbee.write(0x42);
  Xbee.write(0x07);
}

void NR2() {
  Serial.print("B,");
  Xbee.write(0x7E);
  Xbee.write((byte)0x0);
  Xbee.write(0x0F);
  Xbee.write(0x17);
  Xbee.write(0x01);
  Xbee.write((byte)0x0);
  Xbee.write(0x13);
  Xbee.write(0xA2);
  Xbee.write((byte)0x0);
  Xbee.write(0x41);
  Xbee.write(0x97);
  Xbee.write(0x7E);
  Xbee.write(0x30);
  Xbee.write(0xFF);
  Xbee.write(0xFE);
  Xbee.write(0x02);
  Xbee.write(0x44);
  Xbee.write(0x42);
  Xbee.write(0x27);
}

```

Figura 8: Código para extraer trama API.

6) Código para extracción de valor RSSI entre nodos.

```

clear all;
clc;
K=28;% Valor RSSI tomado a una distancia de 1m
n=2.9;% Valor de la atenuación del entorno
x1=0; y1=0;% Posición de Nodo Referencia 1
x2=0; y2=3.15;% Posición de Nodo Referencia 1
X=0; Y=0;
XX=0; YY=0;
Xx=0; Yy=0;
s = serialport("COM11",9600,"Timeout",1); % CONEXIÓN A ARDUINO
configureTerminator(s,"CR/LF");
flush(s);
disp('Inicio');
writeline(s,"A");
pause(2);
for i=1:2
  G=plot(X,Y,'ow');
  H=plot(Xx,Yy,'ow');
  I=plot(XX,YY,'ow');
  disp('NODO REFERENCIA 1');
  writeline(s,"A");%Solicitar Valor Xbee

```

```

pause(0.5)
data = readline(s);%Recibir Dato Xbee
Trama = strsplit(data, ',');%Separar la Trama
DB_A=Trama(20);%Seleccionar valor DB
A=hex2dec(DB_A);%Convertir a Decimal
d1=10^((A-K)/(10*n));%Obtener la Distancia desde Xbee A hasta PC
pause(0.2);

```

7) código de sistema de posicionamiento para interiores.

```

clear all;
clc;
K=28;% Valor RSSI tomado a una distancia de 1m
n=2.9;% Valor de la atenuación del entorno
x1=0; y1=0;% Posición de Nodo Referencia 1
x2=0; y2=3.15;% Posición de Nodo Referencia 1
X=0; Y=0;
XX=0; YY=0;
Xx=0; Yy=0;
s = serialport("COM11",9600,"Timeout",1); % CONEXIÓN A ARDUINO
configureTerminator(s,"CR/LF");
flush(s);
disp('Inicio');
writeline(s,"A");
pause(2);
for i=1:2
    G=plot(X,Y,'ow');
    H=plot(Xx,Yy,'ow');
    I=plot(XX,YY,'ow');
    disp('NODO REFERENCIA 1');
    writeline(s,"A");%Solicitar Valor Xbee
    pause(0.5)
    data = readline(s);%Recibir Dato Xbee
    Trama = strsplit(data, ',');%Separar la Trama
    DB_A=Trama(20);%Seleccionar valor DB
    A=hex2dec(DB_A);%Convertir a Decimal
    d1=10^((A-K)/(10*n));%Obtener la Distancia desde Xbee A hasta PC
    pause(0.2);
    %disp(d1);
    %*****
    %disp('Router B');
    writeline(s,"B");%Solicitar Valor Xbee
    pause(0.5)
    data = readline(s);%Recibir Dato Xbee
    Trama = strsplit(data, ',');%Separar la Trama
    DB_B=Trama(20);%Seleccionar valor DB
    B=hex2dec(DB_B);%Convertir a Decimal
    d2=10^((B-K)/(10*n));%Obtener la Distancia desde Xbee A hasta PC
    pause(0.2);
end
%*****Metodo Angulación*****
theta=acosd((d2^2-d1^2-y2^2)/(-2*d1*y2));
X=abs(d1*sin(theta));
Y=abs(d1*cos(theta));
%*****Valor promedio*****
disp('Posicion del nodo movil en el plano x,y');
axis([-0.5 5 -0.5 4]);

```



```
hold on
title('Localizacion de Nodos');
A=plot(x1,y1,'dr');
B=plot(x2,y2,'dg');
E=plot(X,Y,'og');
legend([A,B,E], 'Nodo R1', 'Nodo R2', 'Nodo Destino');
drawnow;
grid on;
pause(0.5);
flush(s);
clear s;
```