

## ANEXOS

### **I. Configuración de los módulos utilizando la aplicación X-CTU:**

1. Descargar la aplicación de la dirección indicada e instalarlo.

<http://www.digi.com/support/productdetail?pid=3352>

2. Conectar el módulo de programación a un puerto USB del computador con el Xbee ZB montado sobre los sockets.
3. Ingresar al programa, haciendo clic sobre el ícono del X-CTU.

En la pestaña PC Settings, elegir el puerto COM correcto y configurar los parámetros para la comunicación serial: 9600bps, sin control de flujo, 8bits de datos, sin paridad y 1bit de parada. Asimismo, seleccionar el modo API (Application Programming Interface), debido a que el modo API del Xbee ZB nos permite interactuar a nivel de aplicación con el dispositivo. La figura I muestra la configuración especificada para la comunicación serial.

Finalmente, se presiona el botón de Test para comprobar la correcta conectividad del módulo a la PC.

Luego, para la configuración de los parámetros, ingresar a la pestaña de Configuración de modem. Este puede ser usado para leer, modificar y escribir cambios de la configuración de los módulos Xbee y actualizar el firmware de estos. Esta aplicación al detectar el módulo conectado, lee el tipo de modem encontrado, el tipo de firmware y la versión que posee.

1. Darle click en el botón de Lectura. Si la conectividad está bien, se debería visualizar la ventana de configuraciones por defecto del modem.
2. Hacer click en el botón de Restaurar. Una vez que se complete lo deseado en las áreas de texto, hacer click en el botón de Lectura nuevamente y se mostrará la configuración que realizaste.

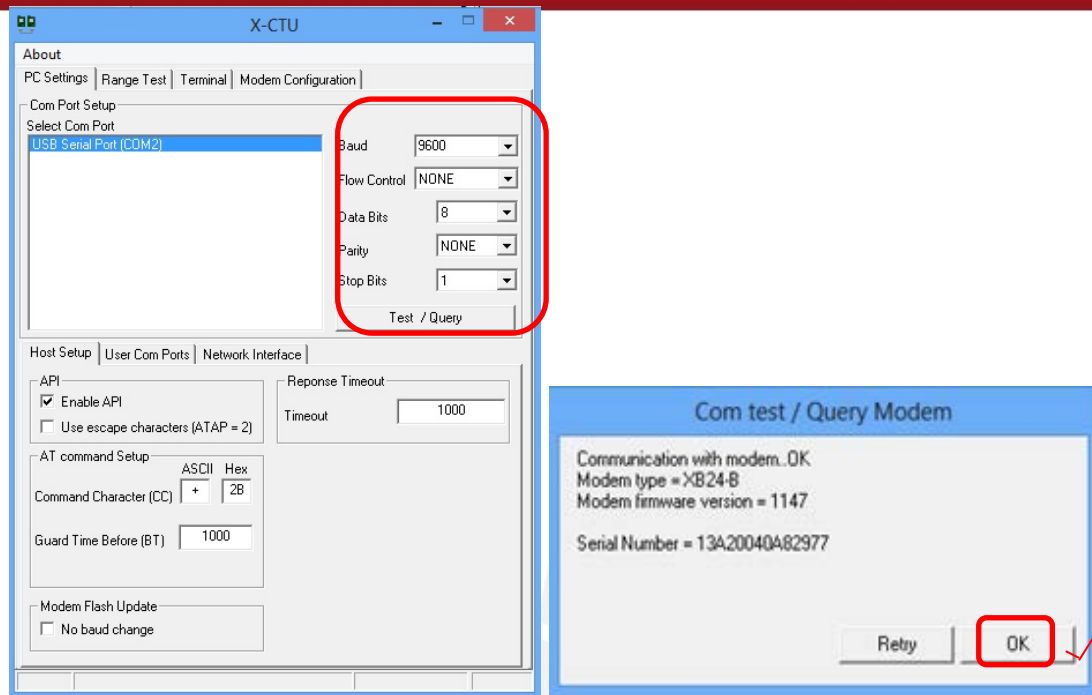


Figura I: Configuración del Xbee ZB para la comunicación serial.

## II. Programación de los elementos del sistema

- Programa Microcontrolador – Control

```
#include <avr/io.h>
#include <util/delay.h>

#include "lcd.h"
#include "keypad.h"
#include "user_interface.h"
#include "password_manager.h"

#define F_CPU 1000000UL // Clock Speed
#define BAUD 9600
#define MYUBRR (((F_CPU / (BAUD * 8UL))) - 1)

//LCD Backlight i/o configuration
#define LCDBL_PORT PORTD //PORB
#define LCDBL_DDR DDRD
#define LCDBL_PIN PD1

//Output relay i/o configuration
#define RELAY_PORT PORTC
#define RELAY_DDR DDRC
#define RELAY_PIN PC0
#define RELAY_PIN1 PC2

int master_password=5555;

//Funcion de retardo
void Wait(uint8_t n);
```

```

//Funciones del Control de relé de salida
void RelayOn();
void RelayOn1();

void RelayOff();
void RelayOff1();

//Funciones del Sistema
void SystemInit();

// Funciones USART
void USART_Init();
void USART_putstring();

void main()
{
    uint16_t password;

    SystemInit();
    USART_Init(MYUBRR);

int i=1;
    while(1)
    {
        DOS:
        password=InputNumber("INGRESE CLAVE ");

//Compara Password

if(password==ReadPassFromEEPROM1() || password==ReadPassFromEEPROM2() || password=
=ReadPassFromEEPROM3() || password==ReadPassFromEEPROM4())
    {
        LCDClear();
        LCDWriteString("Acceso concedido");
        //Si es correcta la contraseña se habilita el relay
        RelayOn();
        Wait(15);

        if(password==ReadPassFromEEPROM1())
        {
            USART_putstring("TM0100");
            LCDClear();
            LCDWriteString(" Bienvenido ");
            Wait(5);
            LCDClear();
            LCDWriteString(" USUARIO 1 ");
            i=0;
        }

        if(password==ReadPassFromEEPROM2())
        {
            USART_putstring("TM0200");
            LCDClear();
            LCDWriteString(" Bienvenido ");
            Wait(5);
            LCDClear();
        }
    }
}

```

```

LCDWriteString("  USUARIO 2  ");
i=0;
}

if(password==ReadPassFromEEPROM3())
{
USART_putstring("TM0300");
LCDClear();
LCDWriteString("  Bienvenido  ");
Wait(5);
LCDClear();
LCDWriteString("  USUARIO 3  ");
i=0;
}

if(password==ReadPassFromEEPROM4())
{
USART_putstring("TM0400");
LCDClear();
LCDWriteString("  Bienvenido  ");
Wait(5);
LCDClear();
LCDWriteString("  USUARIO 4  ");
i=0;
}

//Desactivo relay
_delay_loop_2(10);
RelayOff();

Wait(2);
}
else if(password==0)
{
/*
Si el usuario ingresa 0000 como password se indica que
quiere cambiar su password.
*/
LCDClear();
password=InputNumber2("Indique # clave");

if(password==0001)
{
        _delay_loop_2(10);
        LCDClear();
        password=InputNumber("Clave anterior

1");

        uint16_t NewPassword;
        NewPassword=InputNumber("Nueva clave

1");

        WritePassToEEPROM1(NewPassword);

        LCDClear();
        LCDWriteString("Listo !");
        USART_putstring("TM0101");

```

```
        Wait(15);
        goto DOS;
    }

    if(password==0002)
    {
        LCDClear();
        password=InputNumber("Clave anterior
2");

        uint16_t NewPassword;
        NewPassword=InputNumber("Nueva clave
2");

        WritePassToEEPROM2(NewPassword);

        LCDClear();
        LCDWriteString("Listo !");
        USART_putstring("TM0202");

        Wait(15);
        goto DOS;
    }

    if(password==0003)
    {
        LCDClear();
        password=InputNumber("Clave anterior 3");

        uint16_t NewPassword;
        NewPassword=InputNumber("Nueva clave 3");
        WritePassToEEPROM3(NewPassword);

        LCDClear();
        LCDWriteString("Listo !");
        USART_putstring("TM0303");

        Wait(15);
        goto DOS;
    }

    if(password==0004)
    {
        LCDClear();
        password=InputNumber("Clave anterior 4");

        uint16_t NewPassword;
        NewPassword=InputNumber("Nueva clave 4");
        WritePassToEEPROM4(NewPassword);

        LCDClear();
        LCDWriteString("Listo !");
        USART_putstring("TM0404");

        Wait(15);
        goto DOS;
    }
}
```

```
else
{
    //No se permite el cambio de clave
    LCDClear();
    LCDWriteString("Clave errada !");

    Wait(15);

}

}

else
{
    if(i<=3)
    {
        LCDClear();
        LCDWriteString("Acceso denegado");
        RelayOff();
        USART_putstring("TM0X00");
        Wait(15);

        i+=1;
    }

    else {
        UNO:
        LCDClear();
        LCDWriteString("BLOQUEADO");
        USART_putstring("TMXX00");

        LCDClear();
        LCDWriteString("ALARMA ACTIVADA");
        RelayOn1();

        Wait(5);

        password=InputNumber(" CLAVE MAESTRA ");

        if(password==master_password)
        {
            LCDClear();
            LCDWriteString(" DESBLOQUEADO ! ");
            RelayOff1();
            USART_putstring("TMY00");
            i=0;//1
            Wait(5);

        }
        else
        goto UNO;

    }

}

}

}
```

```
void SystemInit()
{
    LCDBL_DDR|=(1<<LCDBL_PIN);

    RELAY_DDR|=(1<<RELAY_PIN);

    _delay_loop_2(0);

    LCDInit(LS_NONE);

    LCDClear();
    LCDWriteString("BIENVENIDO A");

    Wait(10);
    LCDClear();
    LCDWriteString("KYARI IMPORT");

    LCDBacklightOn();

    if(ReadPassFromEEPROM1()==25755)
    {
        //Password is blank so store a default password
        WritePassToEEPROM1(1111);
    }
    if(ReadPassFromEEPROM2()==25755)
    {
        WritePassToEEPROM2(2222);
    }
    if(ReadPassFromEEPROM3()==25755)
    {
        WritePassToEEPROM3(3333);
    }
    if(ReadPassFromEEPROM4()==25755)
    {
        WritePassToEEPROM4(4444);
    }
}

void Wait(uint8_t n)
{
    uint8_t i;
    for(i=0;i<n;i++)
        _delay_loop_2(0);
}

void RelayOn()
{
    RELAY_PORT|=(1<<RELAY_PIN);
}

void RelayOn1()
{
    RELAY_PORT|=(1<<RELAY_PIN1);
}

void RelayOff()
```

```

{
    RELAY_PORT&=~(1<<RELAY_PIN));
}

void RelayOff1()
{
    RELAY_PORT&=~(1<<RELAY_PIN1));
}

void USART_Init(unsigned int ubrr)
{
    /* Configuración serial */
    UBRRH = (unsigned char)(ubrr>>8);
    UBRRL = (unsigned char)ubrr;

    UCSRB = (1 << RXEN) | (1 << TXEN);
    UCSRA = (1<<U2X);

    UCSRC = (1 << URSEL) | (0 << USBS) | (1 << UCSZ0) | (1 << UCSZ1);
}

void USARTWriteChar(char data)
{
    while(!(UCSRA & (1<<UDRE)))
    {
        //Do nothing
    }
    UDR=data;
}

void USART_putstring(char* StringPtr)
{
    while(*StringPtr != 0x00)
    {
        USARTWriteChar(*StringPtr);
        StringPtr++;
    }
}

```

- Código del programa módulo remoto Control de acceso (php)

```

<?php
$usuario = $argv[1];
$estado = $argv[2];

if ($estado==1)
$evento="Ingreso";
if ($estado==2)
$evento="Cambio de clave";
if ($estado=='X')
$evento="Clave errada";
if ($estado=='B')
$evento="Sistema Bloqueado";

```



```

if ($estado=='D')
$evento="Sistema Desbloqueado";

$idconnect=mysqli_connect("Localhost", "root", "", "tesis");
if ($idconnect==0)
echo "Lo sentimos no se ha podido conectar con la base de datos.";
else
echo "Enhorabuena, se ha conectado con la base de datos.";

date_default_timezone_set("America/Bogota");
$d=date("Y-m-d H:i:s"); // gives 201101

mysqli_query($idconnect,"INSERT INTO tecladomatricial (Usuarios, Evento,
FechaHoraIngreso) VALUES('$usuario','$evento','$d')");

mysqli_close($idconnect);

?>

```

- Código del programa módulo remoto Sensores (php)

```

<?php
$sensor = $argv[1];
$evento = $argv[2];

if ($sensor=='m')
$nombre="ContactoMagnetico";
if ($sensor=='h')
$nombre="SensorHumo";
if ($sensor=='i')
$nombre="SensorInundacion";

if($evento==1)
{$estado="Activo";
$to = "urpi56@gmail.com";
$subject = "Hi!";
$body = "Hi,\n\nHow are you?";
if (mail($to, $subject, $body))
{echo("<p>Email successfully sent!</p>");}
else {echo("<p>Email delivery failed...</p>");}
};
else if($evento=2)

```

```
{$estado="Apagado";}
```

```
$idconnect=mysqli_connect("Localhost", "root", "", "tesis");
```

```
if ($idconnect==0)
```

```
echo "Lo sentimos no se ha podido conectar con la base de datos.\n";
```

```
else
```

```
echo "Enhorabuena, se ha conectado con la base de datos.";
```

```
date_default_timezone_set("America/Bogota");
```

```
$d=date("Y-m-d H:i:s"); // gives 201101
```

```
mysqli_query($idconnect, "UPDATE sensores SET Estado='$estado' WHERE  
Sensor='$nombre'");
```

```
mysqli_close($idconnect);
```

```
?>
```

- Código del programa gestor de eventos (C++)

```
#define __WINDOWS_COM__ // #define __LINUX_COM__
```

```
//#define ENABLE_SERIAL_PORT_EVENT
```

```
#include "serial.h"
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int numero=0;
```

```
/*void SERIAL_PORT_EVENT(HANDLE * hPort)
```

```
{
```

```
    char c;
```

```
    Getc_Port(*hPort,& c);
```

```
    printf("[%d]=%c\n",numero,c);
```

```
    numero++;
```

```
}*/
```

```
int main()
```

```
{
```

```
    HANDLE fd;
```

```
    //DCB OldConf;
```

```
    char cad[23]="";
```

```
    int n;
```

```
    int sec;
```

```
fd=Open_Port("COM1"); // Abre el puerto serie.
// fd=Open_Port("/dev/ttyS0");
//OldConf=Get_Configure_Port(fd); // Guardo la configuración del
puerto.
```

```
Configure_Port(fd,B9600,"8N1"); // Configuro el puerto serie.
//IO_Blocking(fd,TRUE); // Bloqueante por defecto, pero también
// se puede usar:
// IO_Blocking(fd,TRUE);
//n=Write_Port(fd,cad,16); // Escribo en el puerto serie.
for (;;)
{
n=Read_Port(fd,cad,23);

    if((cad[16]==0x54)&(cad[17]==0x4D))
    //(cad[16]==0x54)&(cad[17]==0x4D))
    {
    if ((cad[19]==0x31)&(cad[21]==0x30))
    {printf("Usuario 1 ha ingresado \n");
    system("php C:/Users/URPI/actualizarSQL.php 1 1");
    //goto principal;
    }
    else if (cad[21]==0x31)
    {printf("Usuario 1 cambio de clave\n");
    system("php C:/Users/URPI/actualizarSQL.php 1 2");}

    if((cad[19]==0x32)&(cad[21]==0x30))
    {printf("\nUsuario 2 ha ingresado \n");
    system("php C:/Users/URPI/actualizarSQL.php 2 1");}
    else if (cad[21]==0x32)
    {printf("\nUsuario 2 cambio de clave\n");
    system("php C:/Users/URPI/actualizarSQL.php 2 2");}

    if ((cad[19]==0x33)&(cad[21]==0x30))
    {printf("\nUsuario 3 ha ingresado \n");
    system("php C:/Users/URPI/actualizarSQL.php 3 1");}
    else if (cad[21]==0x33)
    {printf("\nUsuario 3 cambio de clave\n");
    system("php C:/Users/URPI/actualizarSQL.php 3 2");}

    if ((cad[19]==0x34)&(cad[21]==0x30))
```

```

{printf("\nUsuario 4 ha ingresado \n");
system("php C:/Users/URPI/actualizarSQL.php 4 1");}
if (cad[21]==0x34)
{printf("\nUsuario 4 cambio de clave\n");
system("php C:/Users/URPI/actualizarSQL.php 4 2");}

if((cad[18]=='0')&(cad[19]=='X'))
{printf("\nIngreso de clave errada \n");
system("php C:/Users/URPI/actualizarSQL.php 0 X");}

if((cad[18]=='X')&(cad[19]=='X'))
{printf("\nSistema bloqueado");
system("php C:/Users/URPI/actualizarSQL.php 0 B");}

if((cad[18]=='Y')&(cad[19]=='Y'))
{printf("\nSistema DESbloqueado");
system("php C:/Users/URPI/actualizarSQL.php 0 D");}
}

if((cad[11]==0x36)&(cad[12]==0xFFFFF8E)&(cad[21]==0x00))
{printf("\nContacto magnetico activado \n");
system("php C:/Users/URPI/sensores.php m 1");}
else if((cad[21]==0x01)&(cad[22]==0x00))
{printf("\nContacto magnetico desactivado \n");
system("php C:/Users/URPI/sensores.php m 2");}

if((cad[11]==0x37)&(cad[12]==0x44)&(cad[21]==0x00))
{printf("\nSensor de Humo activado \n");
system("php C:/Users/URPI/sensores.php h 1");}
//printf("%x",cad[22]);
else if((cad[21]==0x01)&(cad[22]==0xFFFFFEB))
{printf("\nSensor de humo desactivado \n");
system("php C:/Users/URPI/sensores.php h 2");}

if((cad[11]==0x36)&(cad[12]==0xFFFFFFE)&(cad[21]==0x00)&(cad[22]==0x7F
))

{printf("\nSensor de inundacion activado \n");
system("php C:/Users/URPI/sensores.php i 1");}
//printf("%x",cad[22]);
else if((cad[21]==0x01)&(cad[22]==0x7D))

```

```

        {printf("\nSensor de inundacion desactivado \n");
        system("php C:/Users/URPI/sensores.php i 2");
        /*break;*/}

    }
    Close_Port(fd);
}

```

- Código de la interfaz web (php y html)

```

<form action="seguridad.php" method="post">
<h2>Bienvenido</h2>
<p>Indique su nombre de usuario y clave</p>

Usuario: <input type="text" name="usuario"><br>
Clave:<input type="password" name="password"><br><br>

<input type="submit" value="Enviar">
</form>

<?php
$us = $_POST['usuario'];
$cl = $_POST['password'];
$mius = "admin";
$miclave = "1234";
if(($us != $mius) or ($cl != $miclave))
{echo "Tu nombre de usuario y contraseña son incorrectos";
exit;}
else
{echo "Has accedido a la zona restringida";}
?>

<?php header("Refresh: 5; URL='actualizacion.php'"); ?>
<H1>Sistema de Seguridad Kyari Import</H1>

<?php
include("conex.php");
$link=Conectarse();
$result1=mysql_query("select * from sensores",$link);
?>

<TABLE BORDER=2 CELLSPACING=1 CELLPADDING=1 CELLPADDING=1>

```

```
<TR><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD></TR>
```

```
<?php
echo "Monitoreo: ESTADO DE SENSORES";
while($row = mysql_fetch_array($result1)) {

printf("<tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td></tr>",
$row["ID"],$row["Sensor"],$row["Estado"]);
}
mysql_free_result($result1);
mysql_close($link);
?>
```

```
<?php
$link=Conectarse();
$result2=mysql_query("select * from tecladomatricial",$link);
?>
```

```
<TABLE BORDER=2 CELLSPACING=1 CELLPADDING=1 CELLPADDING=1>
```

```
<TR><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>&nbsp;</TD></TR>
```

```
<?php
echo "Monitoreo: CONTROL DE ACCESO";
while($row = mysql_fetch_array($result2)) {

printf("<tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td></tr>",
$row["ID"],$row["Usuarios"],$row["Evento"],$row["FechaHoraIngreso"]);
}
mysql_free_result($result2);
mysql_close($link);

?>
```

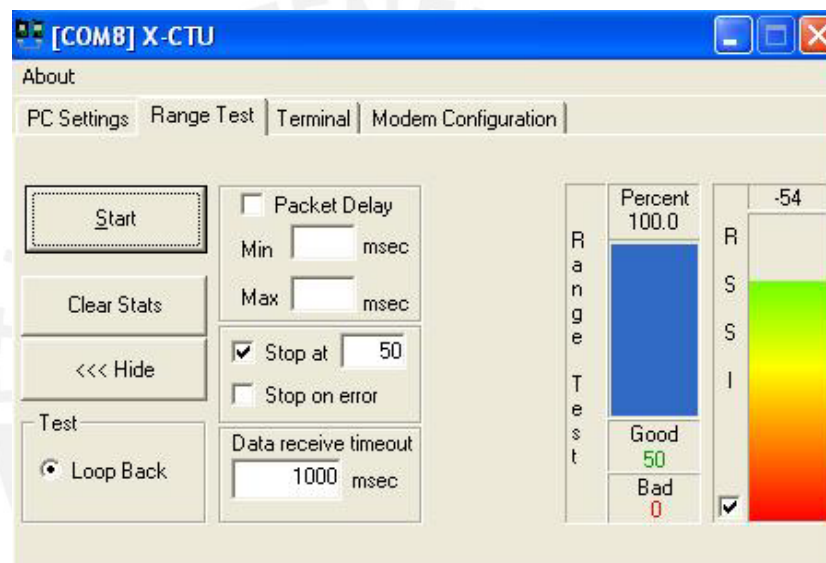
```
</table>
</body>
</html>
```

### III. Pruebas de RSSI:

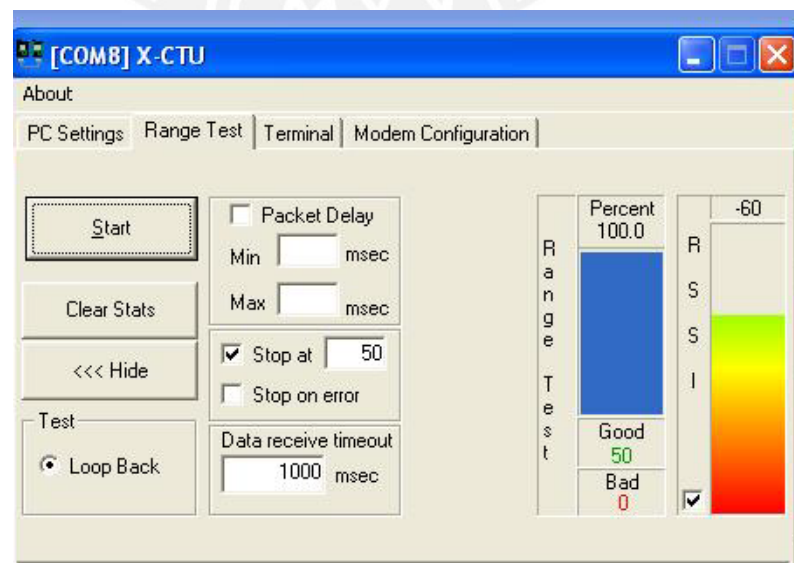
Se realizaron distintas pruebas de RSSI para medir la fuerza de la señal recibida por los módulos zigbee en diferentes escenarios. Básicamente se realizaron mediciones de RSSI entre los nodos remotos y el nodo base(almacén/sótano y segundo piso donde están las oficinas). Adicionalmente considerando las mismas distancias se realizaron pruebas en campo libre donde se verifica la diferencia de potencia debido a la atenuación por la infraestructura.

Por otro lado realizaron pruebas entre el almacén/sótano y el tercer piso donde se verifica en los resultados algunos paquetes no enviados.

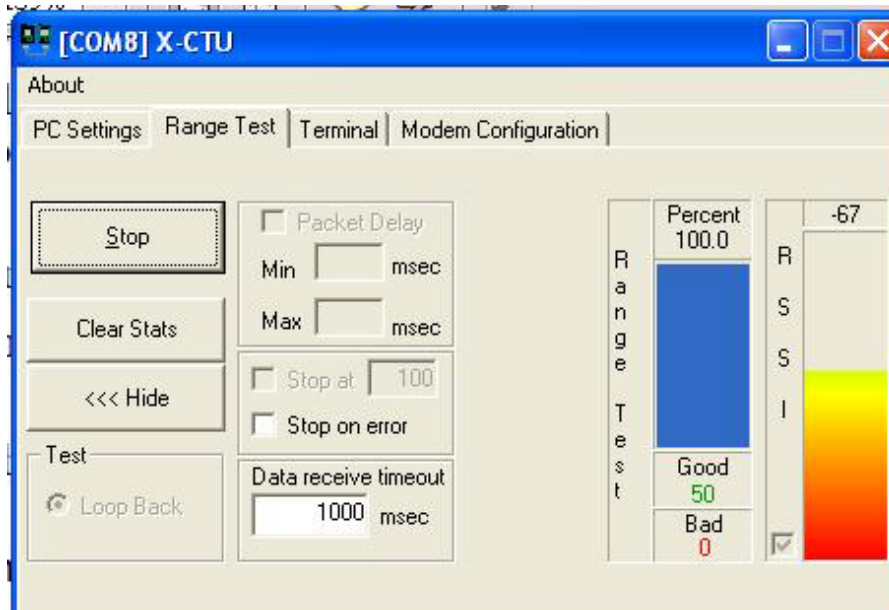
- Prueba Campo libre distancia de 4.5 metros



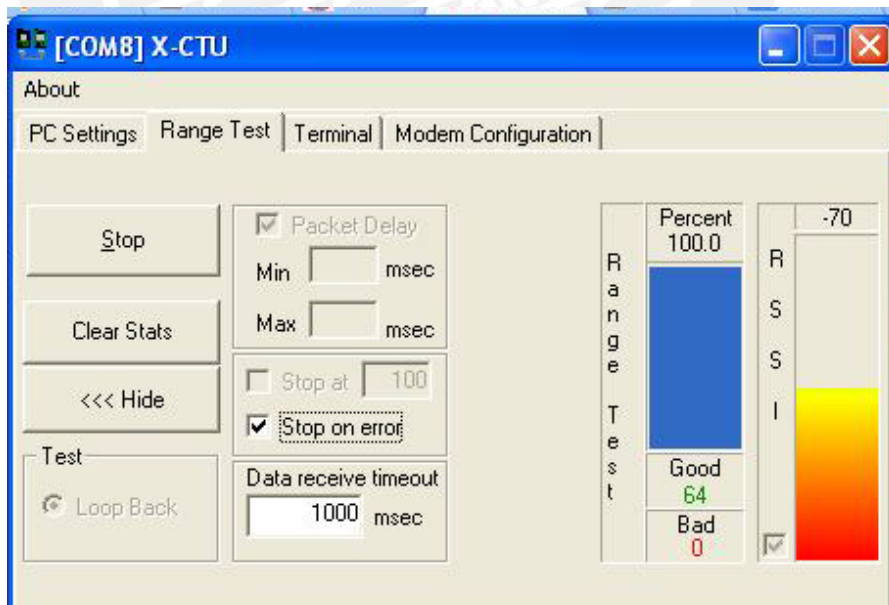
- Prueba Campo libre distancia de 6 metros



- Prueba en el almacén con distancia de 4.5 metros. Fuente: En el Almacén/sótano. Receptor: 2 piso oficina

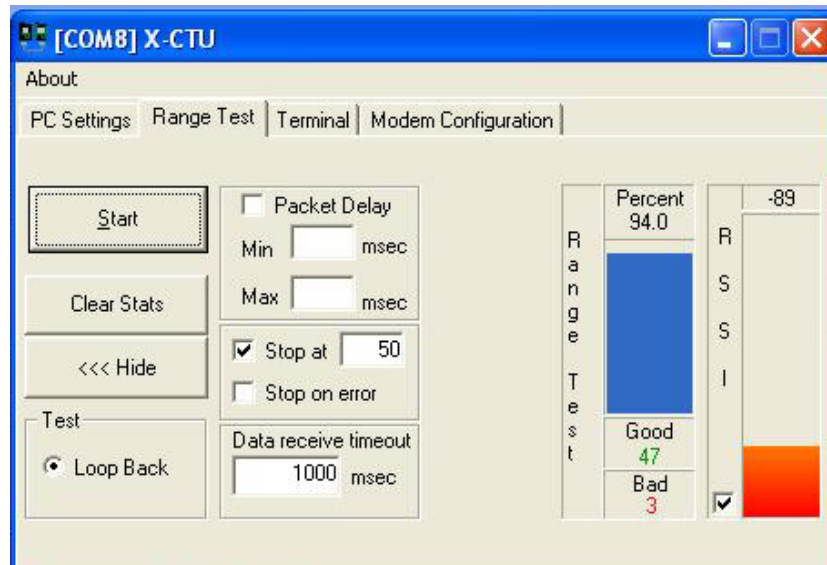


- Prueba en el almacén con distancia de 6 metros. Fuente: En el Almacén/sótano. Receptor: 2 piso oficina





- Prueba en el almacén en el 3er Piso aproximadamente 7 metros. Fuente: En el Almacén/sótano. Receptor: 3piso



- Prueba en el almacén en el 3er Piso aproximadamente 7 metros. Fuente: En el Almacén/sótano. Receptor: 3piso

