UNIVERSIDAD AUTONOMA METROPOLITANA AZCAPOTZALCO

DIVISION DE CIENCIAS BASICAS E INGENIERIA
LICENCIATURA EN INGENIERIA EN COMPUTACIÓN

PROYECTO TECNOLOGICO

“RED DE MONITOREO REMOTO INALAMBRICO DE SENSORES CON DISPOSITIVOS ZIGBEE”

Anahid Hernández Hernández 208202551

Asesor:

José Ignacio Vega Luna

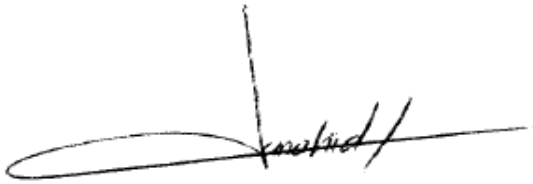
TRIMESTE 14-P

29-Agosto-2014

Yo, José Ignacio Vega Luna, *declaro que aprobé el contenido del presente Reporte de Proyecto de Integración* y *doy mi autorización para su* publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

José Ignacio Vega Luna
Asesor

Yo, Anahid Hernández Hernández, *doy mi autorización* a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Anahid Hernández Hernández
Alumna

Resumen

La temperatura y humedad son elementos naturales los cuales tienen unidades de medición y existen diversos aparatos de distintos materiales, formas, etc., que permiten que esto se realice de manera eficiente, es por ello que en este documento se desarrolla un proyecto que permite que este proceso se realice de manera rápida, menos costosa, de forma inalámbrica, y en tiempo real lo que implica que la recopilación de información sea de manera más rápida, eficiente, y confiable.

En este documento de describe como se realizó la implementación de un sistema el cuan tiene como principal objetivo crear red de monitoreo remoto inalámbrico de sensores con dispositivos zigbee, mediante el uso conocimientos adquiridos en cuanto a tecnologías inalámbricas, conceptos relacionados con la electrónica, configuración de dispositivos, así como la generación de sistemas desarrollado métodos y estrategias para la obtención de información. De forma general se realizó un diseño de un sistema de monitoreo utilizando dispositivos que permitieran la recopilación de la información, así como la configuración de dispositivos que permitieran el almacenamiento de dicha de información para posteriormente interpretada mediante una interfaz, lo que permite reducción de recursos, tiempo.

Este proyecto estuvo compuesto por distintos bloques que se describirán más adelante, y cabe mencionar que también se realizó mediante la combinación de la carrera de Ingeniería en Computación e Ingeniería en Electrónica.

Tabla de contenido

Lista de figuras

Figura Pág.

Lista de tablas

Tabla Pág.

1 .Introducción

Actualmente la mayoría de sistemas y redes de monitoreo remoto de variables y procesos se realiza de forma alambrada y punto a punto. No existen redes de sensores, presentándose los siguientes inconvenientes en estas redes: mayor costo y dificultad de instalación, operación, mantenimiento y crecimiento; vulnerabilidad al medio ambiente; límite en la cantidad de sensores y alcance en la transmisión de la información. Esto ha traído como consecuencia que los fabricantes de estos sistemas estén proponiendo usar en el futuro tecnologías y protocolos de comunicación inalámbrica en aplicaciones de medición distribuida y sistemas de supervisión, para implantar lo que algunos denominan cloud device (dispositivos en la nube) y otros la planta inteligente o PlantWeb. Con esto, se pretende tener mejor rendimiento de los procesos; alta disponibilidad de equipos; incremento de la calidad; reducción de tiempos muertos y riegos; reducción de costos y visibilidad y confiabilidad de información para la toma de decisiones. Las principales tecnologías inalámbricas propuestas son X10, WiFi, Bluetooth y ZigBee [10]-[12]. Los transceptores de este tipo de tecnologías presentan las siguientes características generales: bajo consumo de energía, maximizando la vida útil de sus baterías; tamaño reducido; precio bajo; uso de señales de baja potencia, lo cual no impide que se usen para el control de dispositivos a través de paredes y otros obstáculos; y además pueden configurarse para formar redes flexibles de sensores y actuadores llamadas Redes de Área Personal Inalámbrica (Wireless Personal Área Networks-WPAN) o Redes Inalámbricas de Sensores (Wireless Sensor Network-WSN) [17][16].

Actualmente los principales fabricantes se están inclinando por el uso de ZigBee. ZigBee porque su velocidad y alcance es adecuada para la transmisión de información de redes de sensores y actuadores [14]-[15]. En el Anexo A se presentan las especificaciones técnicas se la tecnología ZigBee. Las redes de monitoreo inalámbricas son aún una propuesta de los fabricantes la cual ya empezó a desarrollarse, donde se ha identificado una oportunidad de competencia de investigación y desarrollo tecnológico.

En lo que respecta a sensores, existe una gran variedad con diferentes tecnologías y fabricantes. Una de las tecnologías que ha tenido grandes avances y desarrollos, existen sensores de luz, humedad, temperatura, presión, velocidad infinidad y cada uno con características específicas, actualmente son utilizadas en muchos ambientes junto con la tecnología ZigBee.

2. Antecedentes

Actualmente la investigación en cuanto a la medición de humedad y temperatura en diversas universidades es demasiada amplia, entre las investigaciones que se han realizado destacan sistemas inalámbricos para supervisión de variables ambientales en invernaderos [20] donde la temperatura y humedad son primordiales para el cultivo de plantas que se desarrollan en otros países , centros de investigación que monitorean y controlan la temperatura y humedad también en invernaderos mediante un sistema [21], en cuanto a la UAM-Azcapotzalco también se ha destacado por diversas investigaciones realizadas en la medición de variables , es por ello que en este documento se plantea de menara especifica la extensión de la creación de un sistema que también pueda contribuir al desarrollo e investigación de control de variables.

3. Justificación

En el Área de Sistemas Digitales del Departamento de Electrónica de la UAM-Azcapotzalco se han realizado durante los últimos años algunos trabajos de monitoreo remoto de variables [11], con los cuales se ha adquirido experiencia en el uso básico de las tecnologías RFiD, ZigBee y Bluetooth con microcontroladores [8]-[13]. Se realizaron trabajos de monitoreo de presión [9]; monitoreo de cantidad de agua consumida [9]; monitoreo de temperatura y humedad [11]; identificación de artículos con RFiD [6]; transmisión de datos con dispositivos ZigBee [12] y monitoreo de ozono [13]. En esta Área se tiene planteado para este año trabajar con la tecnología ZigBee orientadas al monitoreo, control y automatización de actividades y procesos propios de centros de datos, de industrias y del sector agrícola. Centros de datos en particular, ya que cada vez más son evaluados y auditados para poder ser certificados como centros de datos del siglo 21, donde el reto principal es que sean confiables y que hagan uso eficiente de la energía que consumen. En los centros de datos hay una gran cantidad de variables que necesitan ser monitoreadas y controladas con mayor precisión de manera remota. Particularmente, se contempla limitar a dos las variables a monitorear inalámbricamente: temperatura y humedad, ya que son las que están presentes no solo en centros de datos y sector agrícola, sino también en muchos otros ambientes de trabajo son de bajo costo, bajo consumo, y con una conectividad sofisticada en cuanto a otros sistemas inalámbricos lo que se pretende es poder ampliar las aplicaciones.

4. Objetivos

4.1 Objetivo general

✓ Diseñar y construir una red inalámbrica de sensores usando un microcontrolador y transceptores ZigBee.

4.2 Objetivos específicos

- Diseñar una red inalámbrica de monitoreo remoto de temperatura y humedad usando el protocolo ZigBee.
- Configurar un transceptor ZigBee como concentrador para recibir comandos por su antena y transmitirlos a sensores conectados al mismo transceptor.
- Configurar un transceptor ZigBee como concentrador para recibir información por su línea de entrada serie y transmitirla inalámbricamente a otro transceptor ZigBee configurado como coordinador.
- Configurar un transceptor ZigBee como coordinador para recibir información de transceptores del mismo tipo configurados como concentradores.
- Configurar un transceptor ZigBee como coordinador para enviar información a un transceptor del mismo tipo configurado como ruteador.
- Configurar un transceptor ZigBee como ruteador para recibir información de un transceptor del mismo tipo configurado como coordinador.
- Configurar un transceptor ZigBee como ruteador para enviar y recibir información a una computadora personal conectada a la Internet.
- Realizar la programación en una computadora personal para: recibir periódicamente, por medio de la Internet, información enviada de sensores usando transceptores ZigBee, mostrarla en pantalla y almacenarla en disco.

5. Marco Teórico.

Conforme la tecnología se va desarrollando y se realizan investigación al respeto cada ve se puede interpretar de manera más eficiente la información, para ello se quiere de procesos de recopilación de información, dispositivos que permiten adquirir esta información así como, software que sea capaz de ser programado para la interpretación de dicha información.

Existen diversas técnicas de recolección de información se basan en determinados principios, normas y procedimientos cuentan con herramientas en este caso para el desarrollo del proyecto Software y hardware con las características adecuadas y adaptadas al sistema que se desea implementar.

- ### Variables de medición

  El clima cuenta con diversos factores que actúan conjuntamente, existen muchos grupos de elementos con características distintas que son parte del medio ambiente entre estos elementos podemos encontrarla humedad, la temperatura, la presión, etc.,variables que pueden ser medidas respeto a un sistema internacional de unidades, por ejemplo la unidad de medición de la temperatura es el Kelvin (K), desde tiempo atrás se han realizado distintas investigacionespara verificar el comportamiento de estos elementos en nuestro entorno , como nos afectan directamente a nosotros así como a las plantas y animales que nos rodean.

- ### Dispositivos de medición.

  Los dispositivos de medición son aparatos que se utilizan para comparar magnitudes físicasmediante un proceso de medición. Como unidades de medida se utilizan objetos y sucesos previamente establecidos como estándares o patrones y de la medición resulta un número que es la relación entre el objeto de estudio y la unidad de referencia. Los instrumentos de medición son el medio por el que se hace esta lógica conversión. Cuentan con diversas características en las cuales podemos destacar:

  - Precisión: es la capacidad de un instrumento de dar el mismo resultado en mediciones diferentes realizadas en las mismas condiciones.

- Exactitud: es la capacidad de un instrumento de medir un valor cercano al valor de la magnitud real.
- Apreciación: es la medida más pequeña que es perceptible en un instrumento de medida.
- Sensibilidad: es la relación de desplazamiento entre el indicador de la medida y la medida real.

- Microcontroladores

Un microcontrolador es un circuito integrado es una pastilla de silicio en la que se implementan ciertos componentes electrónicos para que juntos puedan llevar a cabo una función determinada.

Este tipo de circuitos tiene múltiples funciones: amplificación, regulación de tensión, conversión analógico/digital y digital/analógico, temporización, etc., Son fáciles de programar lo que permite ejecutar las instrucciones grabadas en su memoria. De esta forma, se convierte en unpequeño ordenador encargado de realizar las necesidades del programador. Además de la flexibilidad que esto aporta, los microcontroladores suelen contar con otros módulos que añaden nuevas funcionalidades que analizaremos más adelante, como temporizadores, conversores analógico digital, módulos CCP.

- Trasmisión de información.

El envío y recepción la información con el paso del tiempo se ha desarrollado con mayor apogeo, el desarrollo de las redes informáticas posibilito su conexión mutua y, finalmente, la existencia de Internet, una red de redes gracias a la cual una computadora puede intercambiar fácilmente información
La información a la que se accede a través de Internet combina el texto con la imagen y el sonido, es decir, se trata de una información multimedia, una forma de comunicación que está conociendo un enorme desarrollo gracias a la generalización de computadores personales dotadas del hardware y software necesarios.
Existen 3 principales formas de la trasmisión de información que a continuación se enumeran:

1. Conexión directa: A este tipo de conexión se le llama transferencia de datos on – line. Las informaciones digitales codificadas fluyen directamente desde una computadora hacia otra, sin ser transferidas a ningún soporte intermedio.
2. Los datos pueden viajar a través de una interfaz serie o paralelo, formada simplemente por una conexión física adecuada, como por ejemplo un cable.

3. Conexión a media distancia: Es conocida como conexión off-line. La información digital codificada se graba en un soporte magnético o en una ficha perforada y se envía al centro de proceso de datos, donde será tratada por una unidad central u host.
4. Conexión a gran distancia: Con redes de transferencia de datos, de interfaces serie y módems se consiguen transferencia de información a grandes distancias.

Es por ello que la trasmisión de información vía inalámbrica también cuenta con sus ventajas y es donde más se está empleando un mejor desarrollo que permita que la conexión de datos sea de menor costo, flexibilidad y facilidad conexión si necesidad de tanto cableado , así como las mejoras tecnológicas constantes introducidas en enlaces de mayor velocidad, longitud, etc.

- Lenguajes de programación.

Concretamente lenguaje de programación se entiende como un sistema de comunicación que posee una determinada estructura, está diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras de manera que se llevan a cabo acciones que deben ser concretadas.

✓ Arduino

Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares. Se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador (por ejemplo: Macromedia Flash, Procesing, Max/MSP, Pure Data). Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente. Al ser open-hardware, tanto su diseño como su distribución libre. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

Cuenta con estructuras definidas, se pueden asignar variables, tipos de datos, aritmética, constantes, control de flujos, etc.

✓ PHP

Es un lenguaje de alto nivel que se ejecuta en un servidor, la ejecución en un servidor es aquel que se ejecuta en un servidor donde están alojadas paginas al contrario de otro lenguajes que son ejecutados en un mismo navegador, esto permite que todas las paginas puedan ser vistas en cualquier ordenador, independientemente del navegador con el que se cuente.

Por mencionar algunas características de este lenguaje podemos destacar:

- Está Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con Mysql y PosgreSQL.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

6. Desarrollo

6.1 Estructura general del sistema de monitoreo.

En el **Figura 6.1** se muestra de forma general la estructura general de las secciones en la que se dividió a desarrollar para la implementación del proyecto.

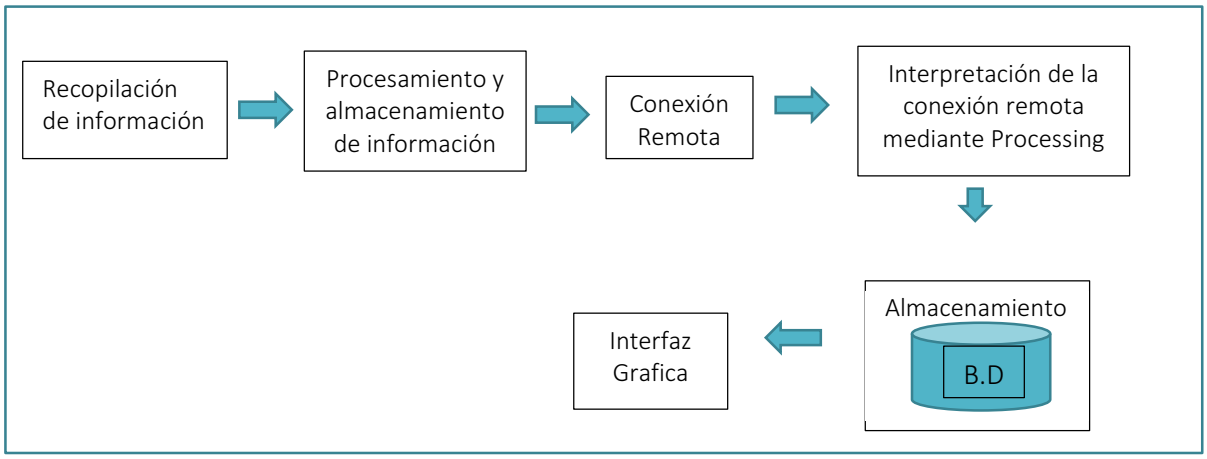


Figura 6.1 Estructura General del sistema mediante bloques

6.2 Lista de Hardware y Software

De manera breve se enlista el Hardware y Software utilizado para realización de este proyecto

Hardware

- 3 Módulos Xbee modelo Xb24-z7wit004
- 2 Arduinos R3
- 1 sensor AM2302
- 1 sensor HMZ-433A1
- 1 XBEE EXPLORER USB SPARKFUN
- 1 cable USB CABLE Tipo A/B ARDUINO
- 1 cable USB MINI CABLE ARDUINO
- 2 Connector de Bateria 9v A Jack 2.1v
- 2 resistencias de 1 00K ohm
- 1 capacitor 0.1u

Software

- Arduino 1.0.5-r2 (para programar los Arduino uno R3)
- Procesing 2.2.1
- X-CTU 40003002_C
- Mysql 5.6.15
- El servidor para revisar mi página fue Apache 2.4.7

6.3 Sección recopilación de la información.

Para la obtención de la información se requirió un dispositivo capaz de detectar magnitudes físicas en este caso que son las variables de instrumentación son la temperatura y humedad, el dispositivo utilizado fue el Sensor HMZ-433A1 en la Tabla 6.3.1podemos apreciar las características con las que cuenta, con estas características permite implementar sistemas de monitoreo para recolectar los datos e información para posteriormente la creación de sistemas o procesos de control automático.

Sensor HMZ-433A1

• Entrega una salida lineal de voltaje para la lectura de humedad que va de 0 a 3.3 volts
• Tiene un termistor integrado del tipo NTC para medir la temperatura.
• Tamaño reducido.
• Se alimenta con 5 voltios.
• Bajo consumo de corriente.

Tabla 6.3.1 Características del Sensor HMZ-433A1.

Para tener una medida fiable de temperatura adquirida por el termistor es necesario hacer los ajustes, para tal ajuste es necesario identificar el modelo que rige el comportamiento del

sensor; ese modelo es: la *curva característica* de un termistor individual, que puede ser aproximada a través del uso de la Ecuación 6.3.2de Steinhart-Hart:

$$\frac{1}{T} = A + B\ln 160.56 + (C\ln 160.56)^3$$

Ecuación 6.3.2de Steinhart-Hart

Dónde:
T = Temperatura en °Kelvin
R = Resistencia del Termistor en Kohm
A, B, C = Constante de la curva de aproximación

Las constantes A, B, y C se calculan seleccionando tres puntos de la Tabla 6.3.4 o curva que acompaña el termistor y resolviendo un sistema de ecuaciones simultáneas de tres incógnitas.

T °C	R KΩ
0	160.56
10	98.714
20	62.328
25	50
30	40.356
40	26.756
50	18.138
60	12.554

Tabla 6.3.4 Constantes definidas por el fabricante.

Para el cálculo de las constantes, se propone el sistema de ecuaciones simultáneas de tres incógnitas para las temperaturas 0°C, 25°C y 50°C. Observe que °Kelvin = °C + 273.15.

Para la obtención de las constantes se resolvieron las ecuaciones con las incógnitas: a, b, c, para ello se utiliza la Ecuación 6.3.2Steinhart_Hart la cual dará como resultado la temperatura en las unidades correspondientes (grados Kelvin) y también se utilizan los datos de la Tabla 6.3.4 la cual sirve como referencia ya que expresa los valores en grados Centígrados

Para resolverlas ecuaciones se sumó 273 a la ecuación de Steinhart_Hart para hacer la conversión de grados Kelvin a grados Centígrados.

$$0 + 273 = \frac{1}{a + b * \ln(160.56) + c * ln^3(160.56)}$$

$$30 + 273 = \frac{1}{a + b * \ln(40.356) + c * ln^3(40.356)}$$

$$60 + 273 = \frac{1}{a + b * \ln(12.554) + c * ln^3(12.554)}$$

Resolviendo la ecuación de 3 incógnitas se obtuvieron los siguientes valores:

$\boldsymbol{a} = 0.002375311945203 \boldsymbol{b} = 0.000246322089759 \qquad \boldsymbol{c} = 0.000000280194860$

Para realizar el cálculo de la humedad se utilizó la Tabla 6.3.5 que ya está definida por el fabricante donde se indica cuanta corriente proporcional suministra el sensor al grado de humedad desde in rango de 0 v hasta 3.3 v aproximadamente.

Humedad (%RH)	Voltaje de salida (v)
20	0.66
30	0.99
40	1.32
50	1.65
60	1.98
70	2.31
80	2.64
90	2.97

Tabla 6.3.5Valores de referencia de la humedad del Sensor HMZ-433A1 proporcionados por el fabricante.

El Sensor HMZ-43A1 cuenta con cuatro terminales para su conexión Tabla 6.3.6.

Terminal	Content.
Pin 1	5 V DC
Pin 2	Salida Humedad
Pin 3	Tierra
Pin 4	Salida Temperatura 50KΩ (@ 25° C)

Tabla 6.3.6Terminales del Sensor HMZ-43A1.

En la Figura 6.3.1.1se muestra el diseño de la conexión del circuito de como de debe conectar el sensor de temperatura y humedad

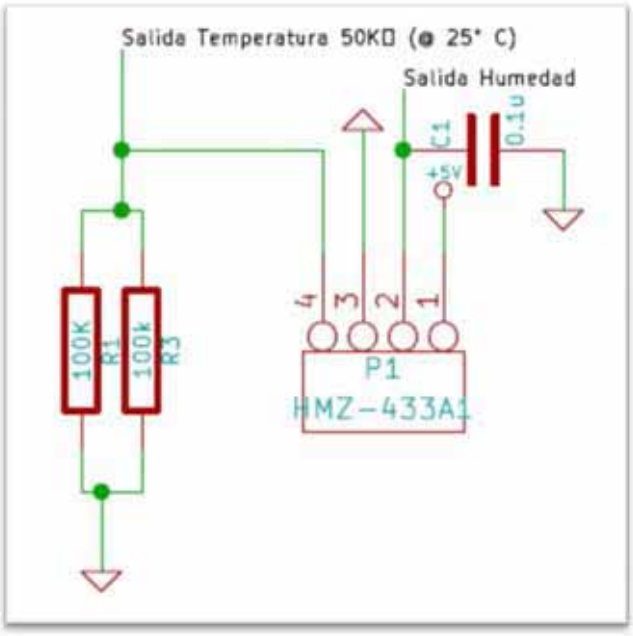


Figura 6.3.1.1 Circuito esquemático del Sensor HMZ-43A1.

Otro sensor a utilizar es el sensor AM2302 de temperatura y la humedad, el cual es de bajo costo, permite medir el aire circundante, y regresa una señal digital en el pin de datos.

Es simple de usar ya que cuenta con diversas características Tabla 6.3.7, pero requiere de una cuidadosa sincronización para tomar datos. El único inconveniente de este sensor es que sólo se puede obtener nuevos datos de una vez cada 2 segundos, por lo se requiere utilizar una biblioteca, ya que las lecturas del sensor puede ser de hasta 2 segundos.

Sensor AM2302	
	
Alimentación	3.3 v ≤ Vcc ≤ 6 V
Señal de salida	Digital
Rango Temperatura	De -40°C a 80°C
Precisión Temperatura	<± 0.5°C
Resolución de temperatura	0.1°C
Rango Humedad	De 0 a 100% RH
Precisión humedad	2% RH
Resolución humedad	0.1 % RH
Tiempo de censado	2s
Tamaño	14x18x5.5mm

Tabla 6.3.7. Características del Sensor AM2302

El Sensor AM2302 cuenta con cuatro terminales para su conexión Tabla 6.3.8.

Terminales	Conexiones
Pin 1	5 V DC
Pin 2	Salida Datos
Pin 3	Tierra (NC)
Pin 4	Tierra

Tabla 6.3.8Terminales del sensor AM2302.

A continuación en la Tabla 6.3.9se muestran los datos que son enviados del Dato enviado del sensor al microcontrolador

40 datos son recibidos

0000 0010	1001 0010	0000 0001	00001101	10100010
H hum	L hum	H temp	L temp	bit par

Tabla 6.3.9Tabla de datos recibidos

Posteriormente se muestras los cálculos de la tabla mostrada con anterioridad

- *Calculo del Bit par:*

0000 0010 + 1001 0010 + 0000 0001 + 00001101 = 10100010 (bit par)

- *Datos son corrector:*

Hum : 0000 0010 1001 0010 = 0292H (Hexadecimal) = 2 * 256 + 9 * 16 + 2 = 658

Hum = 65.8 % RH

Temp: 0000 0001 00001101 = 10DH (Hexadecimal) = 1 * 256 + 0 * 16 + 13 = 269

Temp = 26.9°C

En la Figura 6.3.1.2se muestra el diseño del diagramapara conectarlo a un microcontrolador.

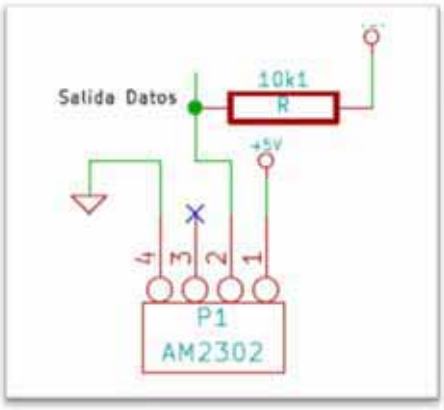


Figura 6.3.1.2Circuito esquemático del Sensor AM2302

6.4 Sección de Procesamiento y almacenamiento de información.

Para que los datos obtenidos de los sensores mencionados anteriormente se requiere el uso de microcontroladores capases de monitorear la información del sensor para posteriormente ser tramitada, para ello se utilizó una placa Arduino la cual denominaremos Uno que identifica que está conectada al sensor HMZ-433A1, Puede tomar información del entorno a través de sus pines de entrada. El microcontrolador en la placa se programa mediante el lenguaje de programación Arduino y el entorno de desarrollo Arduino, los proyectos hechos con esta tarjeta pueden ejecutarse sin necesidad de conectar a un ordenador en la Tabla 6.4.1 podemos ver las característica principales.

Características de Arduino Uno
o Es una placa electrónica basada en el ATmega328
o Cuenta con 14 entradas / salidas digitales pines (de los cuales 6 pueden ser utilizados como salidas PWM
o 6 entradas analógicas
o un oscilador de cristal de 16 MHz
o una conexión USB
o un conector de alimentación
o una cabecera de ICSP
o un botón de reinicio

Tabla 6.4.1Descripción de características de Arduino.

Contiene todo lo necesario para apoyar al microcontrolador, sólo tiene que conectarlo a un ordenador con un cable USB o el poder con un adaptador AC-DC o la batería para empezar.

En la Tabla 6.4.2 se muestran las características de esta tarjeta y las razones por las que se escogió para este proyecto.

Microcontrolador	ATmega328

Voltaje de funcionamiento	5 V
Voltaje de estrada (recomendado)	7 V – 12V
Voltaje de entrada limites	6V – 20V
Digital (IN / OUT)	14 (de los cuales 6 proporcionan PWM)
Pines de entrada analógicos	6
Corriente continua para las IN / OUT Pin	40mA
Corriente de la CC para Pin 3.3 v	50mA
Memoria Flash	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad de reloj	16 MHz

Tabla 6.4.2. Características de la Tarjeta Arduino Uno R3.

En la Figura 6.4.1.1. Se muestra puede observar el diseño de la conexión entre una Tarjeta Arduino Uno R3 y el Sensor HMZ-433A1.

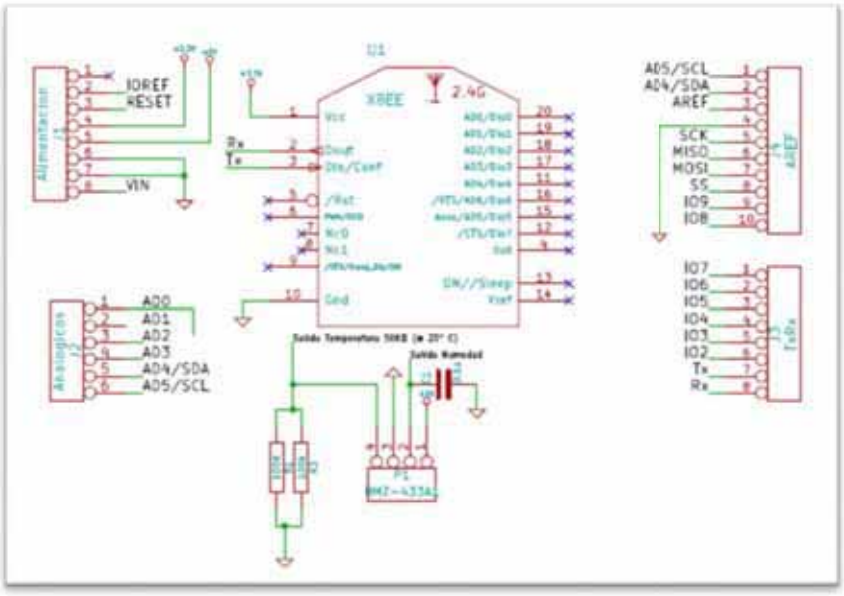


Figura 6.4.1.1. Circuito esquemático para la conexión del sensorHMZ-433A1 y la tarjeta Arduino Uno R3

En el siguiente segmento se muestra el desarrollo de como ejecutar una rutina de lectura del sensor y escritura de las variables en el puerto serial.

```
//Rutina de como leer los datos del sensor HMZ-433A1.

void setup()
{

  Serial.begin(57600);
  pinMode(A0, INPUT);//humedad
  pinMode(A1, INPUT);//temperatura




  var = 1024 - antem;
  resist = (50*var)/(1024-var);
  tempActual = log(resist);
  tempActual2 = ((1)/ ((0.0023753119452
03) + (0.000246322089759 * tempActual)
 + (0.000000280194860 * tempActual * t
empActual * tempActual)));
tempActual2 = tempActual2 - 273.15 + 60
0;
humidActual = ((anhum*(5.0/3.3))/10.24);

  humidActual = humidActual + 400;
```

```
}
void loop()
{
  Leersensor();
}
else
{
   delay (2000);
   Serial.print(humidActual);
   Serial.print("\n");
   Serial.print(tempActual2);
   Serial.print("\n");
   delay (2000);
}}
```

En este caso observamos que se trabajó con 2 tipos de sensores es por ello que se utilizó la Tarjeta de Arduino para conectar el sensor DM2302, conectándolo a una terminal digital. En la Figura 6.4.1.2 se muestra el diseño del diagrama de conexión utilizado.

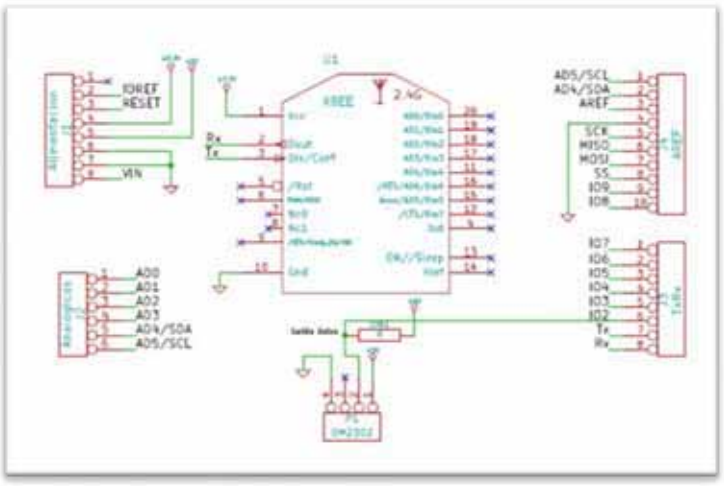


Figura 6.4.1.2 Circuito esquemático para la conexión del sensorDM2302 y la tarjeta Arduino Uno R3.

En el siguiente segmento se muestra el desarrollo de como ejecutar una rutina de lectura del sensor y escritura de las variables en el puerto serial.

//Rutina de como leer los datos del sensor DM2302.

```
void setup()
{
  iniciar();
  Serial.begin(57600);
  delay(1000);
}

void loop()
{
  Leer();
  switch (Error)
  {
     case 0:
        h= (((float((Datos[0]))*256) + float((Datos[1])))/10) + 100;
        t=(((float((Datos[2]))*256) + float((Datos[3])))/10) + 300;
        delay(50);
    Serial.print(h);
    Serial.print("\n");
    Serial.print(t);
    Serial.print("\n");
    break;
    default:
    break;
  }
  delay(2000);
}
void iniciar(){
       pinMode(Pinard,OUTPUT);
       digitalWrite(Pinard,HIGH);
}
void Leer(){
  Error=0;
  byte dht_in;
  byte i;
  digitalWrite(Pinard,LOW);
  delay(18);
  delay(5);
  digitalWrite(Pinard,HIGH);
```

```
delayMicroseconds(40);
  pinMode(Pinard,INPUT);
  delayMicroseconds(40);
  dht_in=digitalRead(Pinard);
  if(dht_in)
  {
    Error=1;
    return;
  }

  delayMicroseconds(80);
  dht_in=digitalRead(Pinard);
  if(!dht_in){
    Error=2;
    return;
  }
  delayMicroseconds(75);
  for (i=0; i<5; i++)
    Datos[i] = read_dht_dat();
  pinMode(Pinard,OUTPUT);
  digitalWrite(Pinard,HIGH);
  byte DHTCHECKSUM = Datos[0]+Datos[1]+Datos[2]+Datos[3];
  if(Datos[4]!= DHTCHECKSUM)
    Error=3;
  };
byte read_dht_dat()
{
  byte i = 0;
  byte result=0;
  for(i=0; i< 8; i++)
  {
    while(digitalRead(Pinard)==LOW);
    delayMicroseconds(30);
    if (digitalRead(Pinard)==HIGH)
      result |=(1<<(7-i));
    while (digitalRead(Pinard)==HIGH);
  }
  return result;
}
```

6.5 Sección de Configuración de comunicación remota.

Como ya se había mencionado con anterioridad el enfoque principal en este proyecto es la automatización utilizando tecnologías inalámbricas es por ello que se utilizó un protocoló de comunicaciones llamado ZigBee el cual es un protocolo de comunicaciones inalámbricas basado en el estándar IEEE 802.15.4podemos resaltar dos capas: la Capa Física y la Subcapa de Control de Acceso al Medio de la Capa de Enlace de Datos, la cual se encarga de aislar los detalles de las tecnologías físicas a la capa de control de acceso al medio. Estas capas son utilizadas por ZigBee para crear un marco de trabajo para las aplicaciones cuya función es solucionar los problemas de interoperabilidad, duración de la batería y costos de los protocolos propietarios, etc.

Los módulos que se utilizaron fueron los XBeeFigura 6.5.1,1 estos son dispositivos que integran una transmisión – receptor (transceptores) de ZigBee y además de un procesador que se encuentra en el mismo modulo. Estos módulos proveen 2 formas amigables de comunicación: Transmisión serial transparente (modo AT) y el modo API que provee muchas ventajas. Los módulos XBeese configuraron desde una PC utilizando un microcontrolador.

Figura 6.5.1.1 Módulo XBee serie

Debido a que los módulos XBee tienen una separación de pines de 2mm se utilizó una tarjeta adaptadora, la cual permitió conectar los módulos XBeeen una protoboard con una separación de separación de 0.1 pulgadas. Además del uso de del adaptador USB que te va a permitió la configuración del módulo fácilmente.

Los módulos XBee son económicos, poderosos y fáciles de utilizar. Algunas sus principales características son:

- Buen Alcance: hasta 300ft (100 mts) en línea vista para los módulos XBee y hasta 1 milla (1.6 Km) para los módulos XBee Pro.
- 9 entradas/salidas con entradas analógicas y digitales.
- Bajo consumo <50mA cuando están en funcionamiento y <10uA cuando están en modo sleep.
- Interfaz serial.
- 65,000 direcciones para cada uno de los 16 canales disponibles. Se pueden tener muchos de estos dispositivos en una misma red.
- Fáciles de integrar.

Existen 2 series de estos módulos. La serie 1 y la serie 2 o también conocida como 2.5. Los módulos de la Serie 1 y la Serie 2 tienen el mismo pin-out, sin embargo, NO son compatibles entre sí ya que utilizan distintos chipset y trabajan con protocolos diferentes.

La serie 1 está basada en el chipset Freescale y está pensado para ser utilizado en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están basados en el chipset de Ember y están diseñados para ser utilizados en aplicaciones que requieren repetidores o una red mesh. Ambos módulos pueden ser utilizados en los modos AT y API.

Una red ZigBee la forman básicamente 3 tipos de elementos. Un único dispositivo coordinador, dispositivo Routers y dispositivos finales (end points).

El coordinador: Es el nodo de la red que tiene la única función de formar la red. Es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el coordinador puede formar la red

Los Routers: Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para enrutar un paquete de información.

End Device: Los dispositivos finales no tienen la capacidad de enrutar paquetes. Deben interactuar siempre atreves del nodo Padre, ya se esté un Coordinador o un Router, es decir no puede enviar información directamente a otro end device.

Modo comando permite ingresar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otras cosas. Para poder ingresar los comandos AT es necesario utilizar la Hyperterminal de Windows, el programa X-CTU [17] o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria.

Para la realización de este proyecto se utilizaron 2 módulos XBee de Digi [17]de la serie con la configuración de Routers.

Para realizar el proceso de configuración se programaron cada uno delos módulos XBee haciendo uso de la tarjeta XBee Explorer, dicha tarjeta basa su funcionamiento en la conversión USB a serial, donde las líneas seriales van conectadas a las del Xbee. Funciona con los módulos los módulos Xbee Standar y Xbee Pro. Solo necesita conectarse el módulo en su base y conectar el cable USB, de esta manera ya se tendrá acceso a los pines seriales y de programación del módulo. Las funciones que cambiaremos se describen en la Tabla 6.5.1.

Tarjeta XBee Explorer	
	
Funcionamiento	Parámetro
Funcion Set	ZIGBEE ROUTER AT
PAN ID	222
MY Address	Asignado (identificador)
Dirección de destino (DH)	0
Dirección de destino (DL)	0
NI- Node Identifier	Nombre del sensor
BD- Baud Rate	6 (57600)

Tabla 6.5.1. Características de configuración para la tarjeta Xbee Router.

Para hacer uso de la tarjeta XBee Explorer y los módulos Xbee se utilizó el Software X-CTU de Digi Figura 6.5.1.2, el cual se descargó directamente de la página[17].

Figura 6.5.1.2 Software X-CTU

Dicho software mencionado con anterioridad se utilizó para la configuración de los módulos XBee en modo Router y coordinador se modificaron diverso parámetros que a continuación de explicaran de manera detalla.

A) Configuración Xbee (Router)

Para la configuración en modo Router se realizaron los siguientes pasos.

1. En primera instancia se utilizó la terminal X-CTU
2. Se seleccionó el puerto en donde tienes conectado el XBee
3. Modificación en el Baut rate a 9600 (Baut rate ajustado de fábrica)

En la Figura 6.5.1.3 podemos observar de manera más grafica como es la interfaz y los parámetros que se configuraron.

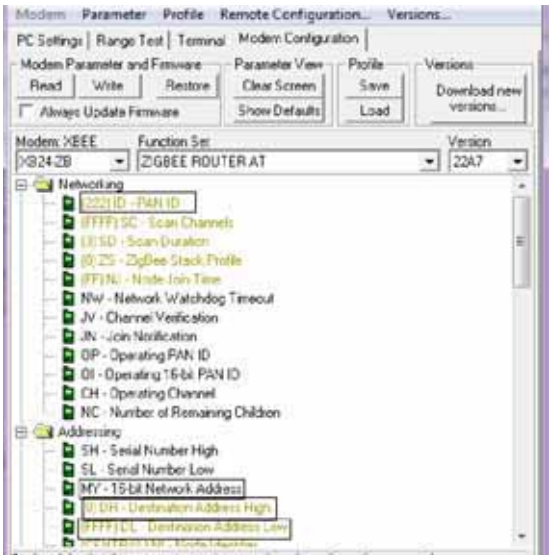


Figura 6.5.1.3. Interfaz de la pestaña PC Settings del programa X-CTU

4. Posteriormente se le dio clic en el botón Test/ Query para probar la comunicación con el módulo XBee.

5. En la pestaña Modem Configuración se realizó la modificación en el Read para leer los parámetros del módulo XBee.

Para diferenciar cada módulo Xbee Router se le asignaron distintos nombres en el parámetro NI- Node Identifier con el nombre sensor 1 y sensor 2 para que no hubiera ninguna confusión en la Figura 6.5.1.4. Se pueden observar los cambios realizados.

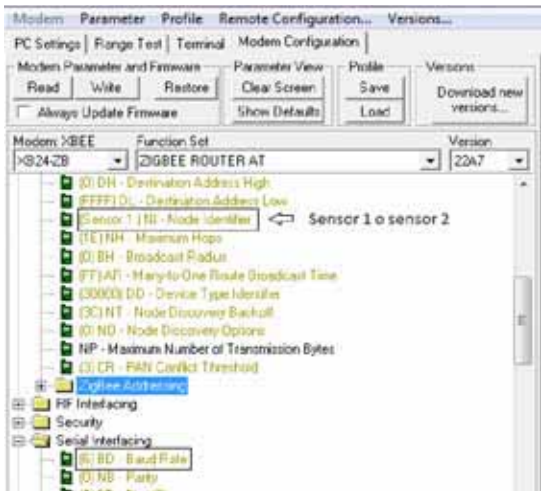


Figura 6.5.1.4.Descripción de los parámetros que se configuran en el módulo XBee.

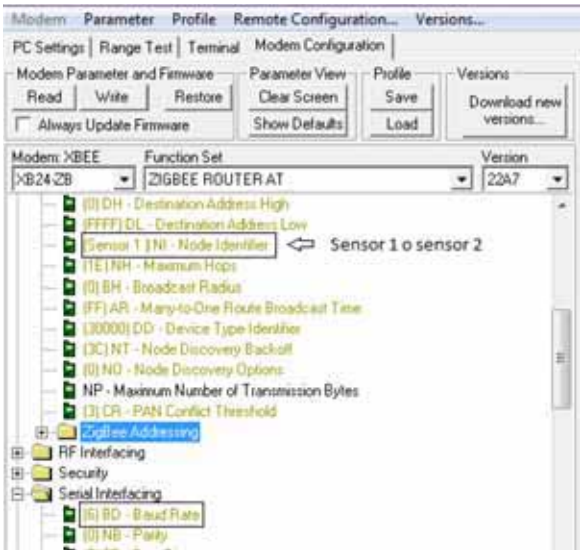


Figura 6.5.1.5. Descripción de los parámetros que se configuran en el módulo XBee.

En este caso la configuración del módulo XBee se realiza como ZIGBEE ROUTER AT. Por último se da clic en el botón Write para guardar la configuración de Router en el XBee.

B) Configuración Xbee (Coordinador)

Para la configuración de módulo XBee como coordinador se utilizó de igual manera Tarjeta XBee Explorer, cambando algunas propiedades como se muestra en la Tabla 6.5.2.

Tarjeta XBee Explorer	
	
Funcionamiento	**Parámetro**
Funcion Set	ZIGBEE COORDINADOR AT
PAN ID	222
MY Address	0
Dirección de destino (DH)	0
Dirección de destino (DL)	FFFF
NI- Node Identifier	CENTRAL
BD- Baud Rate	6 (57600)

Tabla 6.5.2. Parámetros de configuración para la tarjeta Xbee Router.

De igual manera que en la configuración del módulo XBee como Router se realizaron paso a paso.

Para la configuración en modo Coordinador se realizaron los siguientes pasos.

1. En primera instancia se utilizó la terminal X-CTU

2. Se seleccionó el puerto en donde tienes conectado el XBee

3. Modificación en el Baut rate a 9600 (Baut rate ajustado de fábrica)

En la Figura 6.5.1.6. Podemos observar de manera más grafica como es la interfaz y los parámetros que se modificaron.

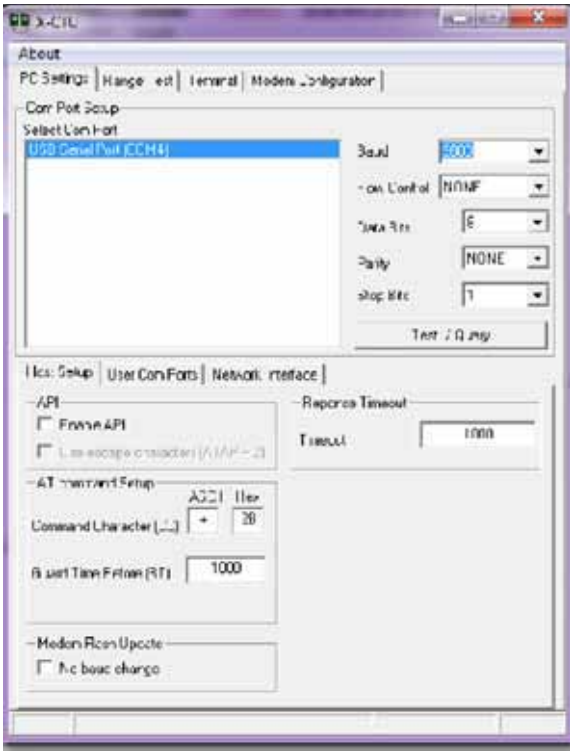


Figura 6.5.1.6. Interfaz de la pestaña PC Settings del programa X-CTU

4. Posteriormente se le dio clic en el botón Test/ Query para probar la comunicación con el módulo XBee.

5. En la pestaña Modem Configuración se realizó la modificación en el Read para leer los parámetros del módulo XBee.

Para diferenciar cada módulo Xbee Router se le asignaron distintos nombres en el parámetro NI- Node Identifier con el nombre sensor 1 y sensor 2 para que no hubiera ninguna confusión en la Figura 6.5.1.7y Figura 6.5.1.8 se pueden observar los cambios realizados.

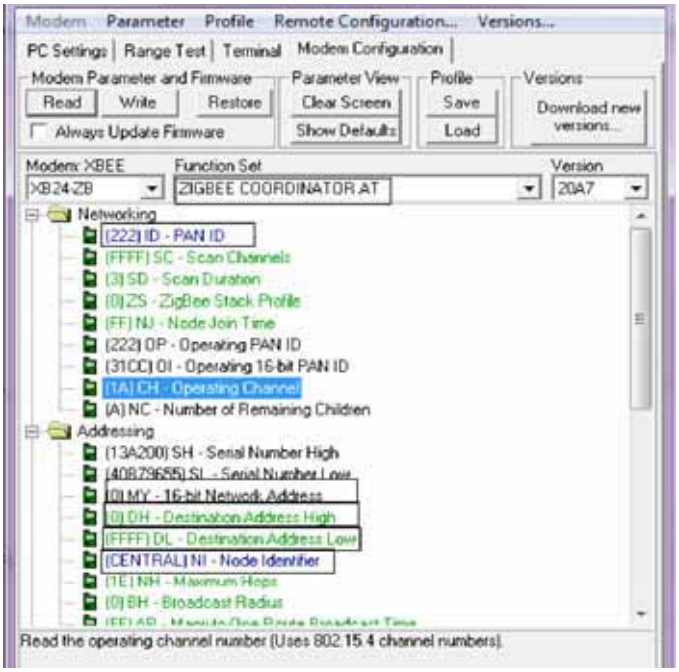


Figura 6.5.1.7Descripción de los parámetros que se configuran en el módulo XBee.

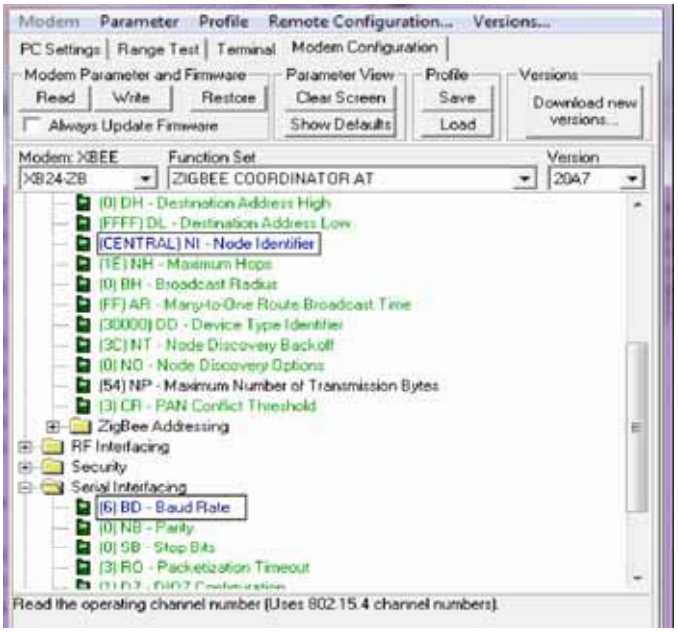


Figura 6.5.1.8. Descripción de los parámetros que se configuran en el módulo XBee.

Para este módulo se realizó como ZIGBEE COORDINADOR AT. Por último se da clic en el botón Write para guardar la configuración de coordinador en el XBee.

6.6 Interpretación de información mediante Processing

Como se comentó anteriormente Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo y puede ser conectado a software del ordenador en este caso se utilizó Procesing ya que fácil de utilizar gratuito. Está basado en java y debido a ello es multiplataforma.

El objetivo de este proyecto es tratar que Procesing pueda interactuar con un puerto serie, como recibir datos, y guardarlos en un 4 ficheros.

Para instalar ×Processing Figura 6.6.1 se descargó la versión correspondiente a nuestro sistema operativo desde la página oficial oficial [30] posteriormente se descargó el fichero zip y descomprimirlo en alguna carpeta en nuestro ordenador. La carpeta extraída contiene el fichero processing-algo.exe y simplemente haciendo doble clic en el icono arranca el Entorno de desarrollo de Processing.

Figura 6.6.1Software Processing

Posteriormente se cargó la librería correspondiente en Arduino. El fichero de la librería Arduino para Processing está en el archivo processing-arduino que se encuentra en la página de Arduino. Dentro de ellos hay una carpeta que se llama Arduino y contiene la librería [31].

Processing necesita trabajar con la librería Arduino se deberá insertar la carpeta Arduino dentro de la librería del IDE de Processing en este caso la carpeta se encuentra ubicada en la ruta "C:\Users\nombre\Documents\Processing"

Ahora que tanto el IDE de Arduino y Processing están Configurados para que se establezca una comunicación por el puerto serial. Se abre una ventana del IDE de Processing para escribir el programa que leerá y guardara los datos enviados de los sensores y recibidos por la tarjeta Xbee configurada como coordinador.

Como Processing puede usar librerías al estilo de java, hay una que necesitaremos para la programación del puerto serie, esta es *processing.serial* y se puede importar desde un sketch de la siguiente manera:

```
import processing.serial.*;
```

Después hay que instanciarlo en el método *Setup*

```
puerto=new Serial(this,Serial.list()[0],57600);
```

Se observa que la velocidad a la que se quiere enviar y recibir datos es la misma con la que se configuraron los dispositivos XBee y las tarjetas Arduino.

Sino sabemos que puerto serie tenemos en nuestro ordenador, podemos imprimir un listado de los disponibles.

```
print(Serial.list());
```

El programa cumple con la función de guardar las diferentes variables en archivos llamados temperatura1.txt, humedad1.txt, temperatura2.txt y humedad2.txt. De esta manera, luego con la creación de una base de datos podremos graficar la humedad y la temperatura de los sensores.

EL siguiente segmentó muestra parte del cogido que se realizó el sketch en Processing.

Procedimiento para ejecutar una rutina de lectura del puerto serie y escritura de los valores en los ficheros.

```
//Rutina de lectura del puerto serie y escritura de los valores en los ficheros.

import processing.serial.*;
Serial puerto;
int colorin;
int t2= 0;
int t1= 0;
int h2= 0;
int h1= 0;
PrintWriter salidah1;
PrintWriter salidat1;
PrintWriter salidah2;
PrintWriter salidat2;
float temperatura1=0;
float humedad1=0;
```

```
float temperatura2=0;
float humedad2=0;
void setup(){
size(400,300);
print(Serial.list());
colorin=0;
puerto=new Serial(this,Serial.list()[0],57600);
salidat1 = createWriter("temperatura1.txt");
salidah1 = createWriter("humedad1.txt");
salidah2 = createWriter("humedad2.txt");
salidat2 = createWriter("temperatura2.txt");
}
void serialEvent (Serial puerto){
  String inString = puerto.readStringUntil('\n');
 if (inString != null) {
     float comp = float (inString);
  inString = trim(inString);
  if(comp >= 100 && comp <= 200){
  humedad1 = float(inString) - 100;
  h1= h1+1;
  salidah1.print(h1);
  salidah1.print(TAB+ " "+humedad1);
  salidah1.print(TAB+"    "+year()+"-"+month()+"-"+day()+"
"+hour()+":"+minute()+":"+second());
  salidah1.println("");
 }
 salidah1.flush();

    if(comp >200 && comp <=400){
    temperatura1 = float(inString)-300;
    t1 = t1 +1;
    salidat1.print(t1);
    salidat1.print(TAB+" "+temperatura1);
    salidat1.print(TAB+"    "+year()+"-"+month()+"-"+day()+"
"+hour()+":"+minute()+":"+second());
    salidat1.println("");
   }
      salidat1.flush();

    if(comp >400 && comp <=500){
    humedad2 = float(inString)-400;
    h2 = h2 + 1;
    salidah2.print(h2);
    salidah2.print(TAB+" "+humedad2);
```

```
   salidah2.print(TAB+"    "+year()+"-"+month()+"-"+day()+"
"+hour()+":"+minute()+":"+second());
   salidah2.println("");
  }
     salidah2.flush();

   if(comp >500 && comp <=700){
   temperatura2 = float(inString)-600;
   t2 = t2 + 1;
   salidat2.print(t2);
   salidat2.print(TAB+" "+temperatura2);
   salidat2.print(TAB+"    "+year()+"-"+month()+"-"+day()+"
"+hour()+":"+minute()+":"+second());
   salidat2.println("");
  }
     salidat2.flush();
}}
void draw(){
if (puerto.available()>0){
  colorin=puerto.read();
}
background(0,0,colorin);


  salidah1.println("");
 }
 salidah1.flush();

   if(comp >200 && comp <=400){
   temperatura1 = float(inString)-300;
   t1 = t1 +1;
   salidat1.print(t1);
   salidat1.print(TAB+" "+temperatura1);
   salidat1.print(TAB+"    "+year()+"-"+month()+"-"+day()+"
"+hour()+":"+minute()+":"+second());
   salidat1.println("");
  }
     salidat1.flush();

   if(comp >400 && comp <=500){
   humedad2 = float(inString)-400;
   h2 = h2 + 1;
   salidah2.print(h2);
   salidah2.print(TAB+" "+humedad2);
```

```
   salidah2.print(TAB+"    "+year()+"-"+month()+"-"+day()+"
"+hour()+":"+minute()+":"+second());
   salidah2.println("");
  }
     salidah2.flush();

   if(comp >500 && comp <=700){
   temperatura2 = float(inString)-600;
   t2 = t2 + 1;
   salidat2.print(t2);
   salidat2.print(TAB+" "+temperatura2);
   salidat2.print(TAB+"    "+year()+"-"+month()+"-"+day()+"
"+hour()+":"+minute()+":"+second());
   salidat2.println("");
  }
     salidat2.flush();
}}
void draw(){
if (puerto.available()>0){
 colorin=puerto.read();
}
background(0,0,colorin);
}
```

6.7. Sección almacenamiento de información

Para interpretar la información se requiere almacenarla en una base de datos por ello e eligió MYSQLFigura 6.7.1.1 la cual soporta un lenguaje SQL y la conexión de varios usuarios ,permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas,etc., resumiendo: administrar bases de datos. Se descarga el instalador de Mysql. En este caso el MySQL Installer for Windows [20].

Figura 6.7.1.1Software MYSQL versión5.6.15

Una vez instalado MySQL, vamos a crear y trabajar en una base de datos que se llama monitoreo_remoto, con cuatro tablas y cada una de ellas con tres campos. En la Tabla 6.7.1 se muestran la base de datos que se l agino el nombre de monitoreo_remoto con 4 tablas donde se permitió el almacenamiento de la información de los archivos con extensión Txt. mencionados anteriormente

Tablas	shum1	stem1	shum2	stem2
1 campo	Id	Id	id	id
2 campo	h1	tem1	h2	tem2
3 campo	t1	t1	t2	t2

Tabla 6.7.1Especificaciones de la base de datos.

Para crear las tablas es necesario realizar algunas instrucciones mediante código a continuación se describen dichas instrucciones que se requirieron.

1. Para crear la base de datos se tecleo la siguiente instrucción
   mysql > créate database monitoreo_remoto;

2. Para usar esta base de datos que acabamos de crear se utilizó la instrucción :

mysql > USE monitoreo_remoto;

3. Como ya se tenía un diseño de lo que se requería en la Tabla 6.7.1 es necesario utilizar instrucciones que correspondan a lo que se necesita es por eso que las siguientes instrucciones muestran la forma de cómo realizar una tabla en mysql.

```
CREATE TABLE shum1 (
        Id INT (11) NOT NULL,
        h1 VARCHAR (11) NOT NULL,
        t1 VARCHAR (22) NOT NULL ,
        PRIMARY KEY (id)
        ENGINE=InnoDB;
```

```
CREATE TABLE stem1 (
        Id INT (11) NOT NULL,
        tem1 VARCHAR (11) NOT NULL,
        t1 VARCHAR (22) NOT NULL ,
        PRIMARY KEY (id)
        ENGINE=InnoDB;
```

```
CREATE TABLE shum2 (
        Id INT (11) NOT NULL,
        h2 VARCHAR (11) NOT NULL,
        t2 VARCHAR (22) NOT NULL ,
        PRIMARY KEY (id)
        ENGINE=InnoDB;
```

```
CREATE TABLE stem2 (
        Id INT (11) NOT NULL,
        tem2 VARCHAR (11) NOT NULL,
        t2 VARCHAR (22) NOT NULL ,
        PRIMARY KEY (id)
        ENGINE=InnoDB;
```

En la Tabla 6.7.2 se muestra los parámetros tabla shum1 que fue creada.

Columna	Tipo	Nulo
Id	int(11)	No
h1	varchar(11)	No
t1	varchar(22)	No

Tabla 6.7.2. Estructura interna de la tabla shum1.

En la Tabla 6.7.3 se muestra como se visualiza la tabla después de realizar una consulta donde se van almacenando los datos que fueron extraídos por los archivos con extensión Txt.

Tabla	Filas	Tipo	Tamaño
shum1	0	InnoDB	16 KB
shum2	0	InnoDB	16 KB
stem1	0	InnoDB	16 KB
stem2	0	InnoDB	16 KB
4 tablas	0	--	64 KB

Tabla 6.7.3. Representación de las tablas creadas en la base de datos

6.8 Diseño de interfaz Grafica

El tipo de servidor a utilizar apache**Figura 8.1** en su versión 2.4.7 la cual se descarga desde su página oficial [21], a continuación se describe su instalación y configuración.

Figura 6.8.1 Software Apache

Descripción de instalación y configuración de Apache.

1. Descargar la última versión de Apache
2. Seleccionar la carpeta directory
3. Instalar y seleccionar el fichero Apache.1.0-RC1-win32-etup.exe
4. Cuando se tenga el fichero en el ordenar configurar el servidor Apache
   Donde se aginaran los siguientesvalores
   -Network Domain: localhost.
   -Server Name: root.
   -Password:// sin ninguna contraseña lo que permite.
   Modificar la opción: for All Users, on Port 80n, as a Servece-Recommended

5. Posteriormente se configurara el fichero httpd.conf , donde se cargara un módulo de PHP , se buscara la cadena Loadme y añadir la siguiente línea
   Loadmode php5module *C:/php /php5apache2.dll*
   La dirección agregada indica que es donde se guardará el módulo php5, donde se instalara phploque permitirá que se almacenen allí los ficheros.

   Al buscar DocumentRoot se realizara el cambo de ruta por DocumentRoot"*C:/ficheros* para ello se debe crear en la raíz C:/ un directorio llamado ficheros que será donde se guardaran los ficheros

6. Posteriormente se buscara la cadena de texto: DirectoriIndex donde se modificara la línea
   DirectoryIndex index.htm index.php. index. Php3. Index. Php.4 index.phtml index.html.var .

   De ésta manera se crean los directorios, al acceder a alguno de ellos, se ejecutara el índex predeterminado si existe de lo contrario se enviara un mensaje de error.

PHP

El lenguaje de programación es de tiene una estructura que permite la visualización de páginas web en un servidor, es por ello que se utilizó para la realización de este proyecto ya que es de fácil manejo y la estructura para programar no es tan compleja en la Figura 8.2 podemos observar la interacción de un servido, un navegador, y un navegador.

Figura 6.8.2Interacción de Php

Para la configuración e instalación de PHP

Se requiere descargar la última versión de la página oficial [22] donde se seleccionara un el fichero PHP 5.1.6 (tar.bz2), cuando s e tenga se descargara la versión y en la unidad C:/ se creara un directorio llamado php , donde se descomprimirán todos los ficheros .

También se requiere la configuración de fichero php.ini. Para ello se accesara a la carpeta C:/php donde se encontrara el fichero que se desea modificar.

Se buscara la cadena: register_globals, cuyo valor es Off y se debe modificar a On con esto se permitiría el uso de variables globales, también se activara la extensión necesaria que permite a php el manejo de funciones relacionada con Mysql. Para ello se buscara la cadena de texto. Windows Extensions, en esta sección existirán extensiones desactivada pero se modificara para que puedan ser utilizadas en esta

sección se modificara la cadena y pondremos; extensión= php_mysql.dll para poder activarla.

Para realizar la interfaz era necesario el uso de tener instalado un servidor así como un fichero donde mediante el lenguaje de programación a utilizar fue PHP para poder realizar la interfaz y se pudiera realizar la interpretación de los resultados de sistema en cuestión, como los valores ya se encontraban almacenados en Mysql.

Resultados

Para verificar el funcionamiento correcto del proyecto tecnológico se realizaron pruebas después del desarrollo descrito anteriormente las siguientes imágenes describen las lecturas adecuadas de la información.

Como primera prueba fue la verificación de la lectura de los sensores **Figura 7.1** esto por medio del puerto serie con el sensor DM2302.

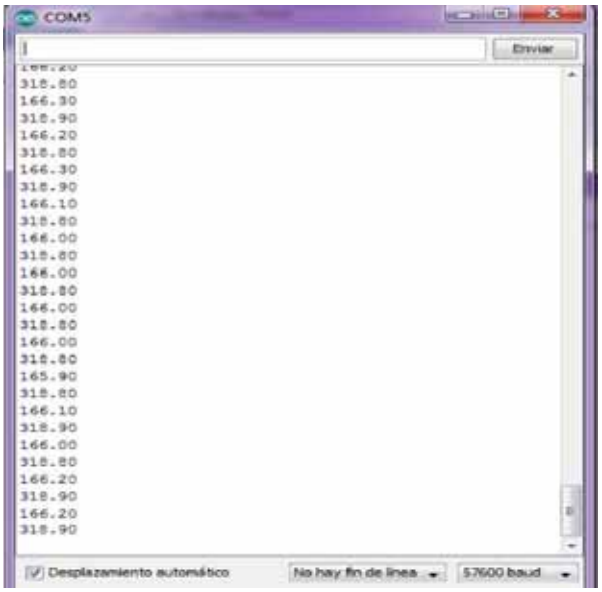


Figura 7.1 Lecturas que recibe el puerto serie por el sensor DM2302.

De igual manera se muestra en la **Figura 7.2** la lectura del puerto serie con el sensor HMZ-433A1. Se usa el monitor serie que tiene el IDE de Arduino

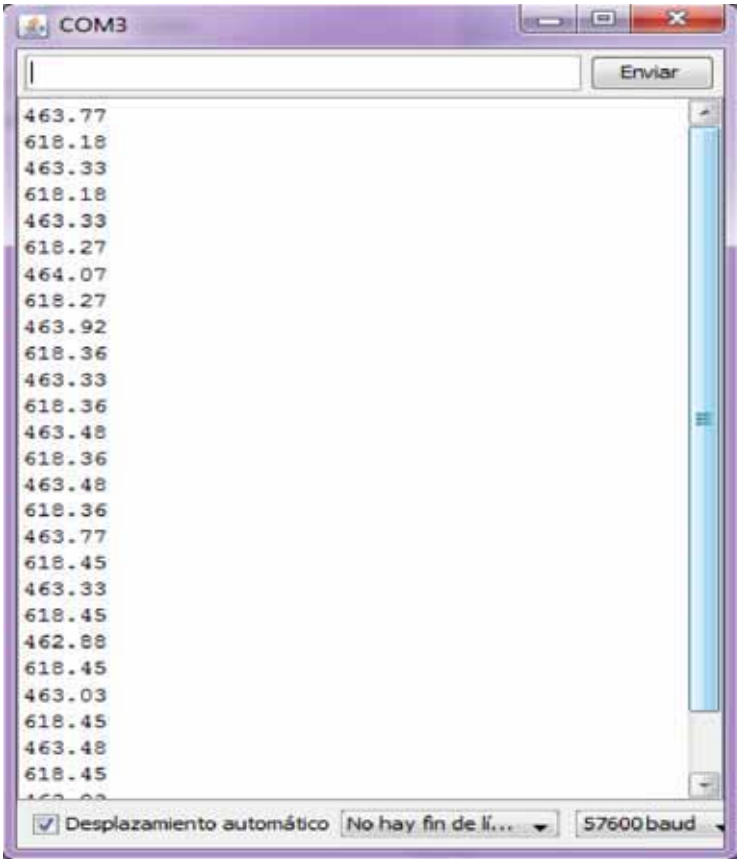


Figura 7.2. Lecturas que recibe el puerto serie por el sensor HMZ-433A1.

Cada 20 segundos en el dispositivo Xbeen envía la información y se actualiza lo que permite que se almacenen fácilmente los datos, para los 2 sensores se realiza el mismo procedimiento.

Posteriormente de la verificación de la obtención dela información se verifico la conexión de los sensores y los dispositivos XBee Figura 7.3para la comunicación, esto mediante el puerto serie en el cual está conectado el Xbee configurado como coordinador.

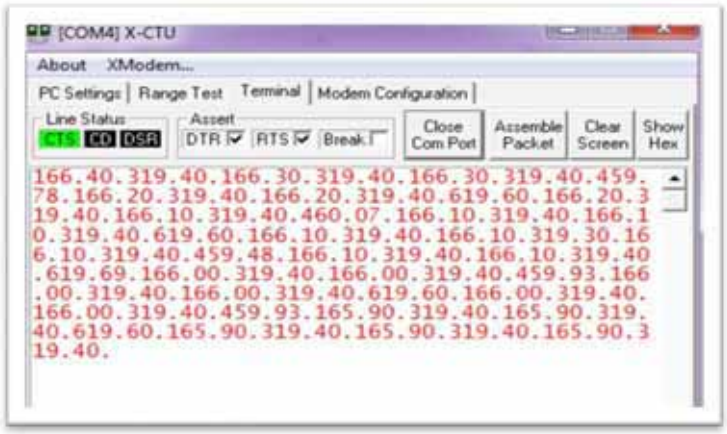


Figura 7.3. Monitoreo del puerto serie del Xbee coordinador, con el programa X-CTU

En la Figura 7.4 se muestra la ejecución de programa realizado en el software denominado Processing donde se realizó la creación de fichero para almacenar la información.

Figura 7.4. Implementación de Processing para leer el puerto serie y almacenar los datos.

Continuación **Figura 7.5** se muestran los ficheros que se crearon después de ejecutar Processing

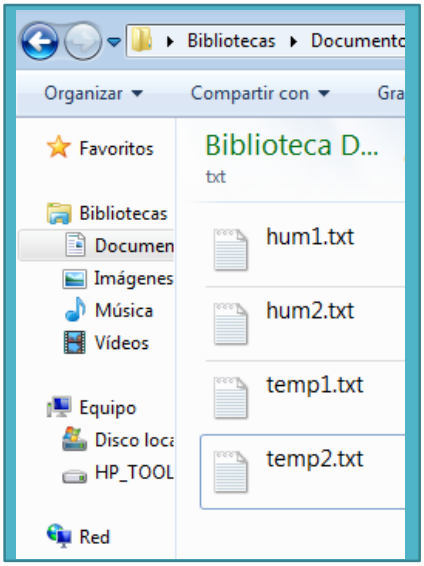


Figura 7.5. Creación de los ficheros donde se almacenaran datos, fecha y hora de las lecturas.

Para verificar que la información delos ficherosesté almacenada en la base de datos Mysql se realizaron algunas consultas en las tablas **Tabla 7.2, Tabla 7. 3 Tabla 7.4, y Tabla 7.5** se pueden observar que en efecto los resultados se encuentran almacenados.

Se muestra información adicional como es la dirección ip del servidor con el que se está trabajando, el nombre de la base de datos, como se realizó la consulta y el número de filas que aparecen en las consultas en todas las tablas que se mencionaron con anterioridad.

Servidor: 127.0.0.1
Base de datos: monitoreo_remoto
consulta SQL: SELECT * FROM `shum1` LIMIT 0, 10;
Filas: 10

id	h1	t1
1	60.600006	2014-8-24 12:18:49
2	60.600006	2014-8-24 12:18:58
3	60.699997	2014-8-24 12:19:16
4	60.600006	2014-8-24 12:19:26
5	60.399994	2014-8-24 12:19:35
6	60.399994	2014-8-24 12:19:44
7	60.5	2014-8-24 12:19:54
8	60.5	2014-8-24 12:20:3
9	60.600006	2014-8-24 12:20:12
10	60.600006	2014-8-24 12:20:21

Tabla 7.2. Datos de la tabla shum1

Servidor: 127.0.0.1
Base de datos: monitoreo_remoto
consulta SQL: SELECT * FROM `shum2` LIMIT 0, 10;
Filas: 10

id	h2	t2
1	55.78	2014-8-24 12:18:45
2	55.630005	2014-8-24 12:18:54
3	55.630005	2014-8-24 12:19:3
4	55.630005	2014-8-24 12:19:12
5	55.04001	2014-8-24 12:19:21
6	55.48999	2014-8-24 12:19:30
7	55.48999	2014-8-24 12:19:39
8	55.339996	2014-8-24 12:19:48
9	55.190002	2014-8-24 12:19:57
10	54.75	2014-8-24 12:20:6

Tabla 7.3. Datos de la tabla shum2.

Servidor: 127.0.0.1
Base de datos: monitoreo_remoto
consulta SQL: SELECT * FROM `stem1` LIMIT 0, 10;
Filas: 10

id	tem1	t1
1	20.399994	2014-8-24 12:18:49
2	20.399994	2014-8-24 12:18:58
3	20.399994	2014-8-24 12:19:7
4	20.399994	2014-8-24 12:19:16
5	20.399994	2014-8-24 12:19:26
6	20.299988	2014-8-24 12:19:35
7	20.299988	2014-8-24 12:19:44
8	20.299988	2014-8-24 12:19:54
9	20.299988	2014-8-24 12:20:3
10	20.299988	2014-8-24 12:20:12

Tabla 7.4. Datos de la tabla stem1.

Servidor: 127.0.0.1
Base de datos: monitoreo_remoto
consulta SQL: SELECT * FROM `stem2` LIMIT 0, 10;
Filas: 10

id	tem2	t2
1	20.75	2014-8-24 12:18:45
2	20.75	2014-8-24 12:18:54
3	20.840027	2014-8-24 12:19:3
4	20.75	2014-8-24 12:19:12
5	20.75	2014-8-24 12:19:21
6	20.75	2014-8-24 12:19:30
7	20.75	2014-8-24 12:19:39
8	20.75	2014-8-24 12:19:48
9	20.75	2014-8-24 12:19:57
10	20.75	2014-8-24 12:20:6

Tabla 7.5 Datos de la tabla stem2.

Mediante el uso de un explorador web se puede ver la interfaz que interpreta lo datos que están almacenados en la base de datos mediante graficas que permiten la interpretación de la información en la Figura 7.6se muestra la lectura del sensor HMZ-433A1

Figura 7.6Grafica que se obtiene del sensor HMZ-433A1 con respecto a la humedad.

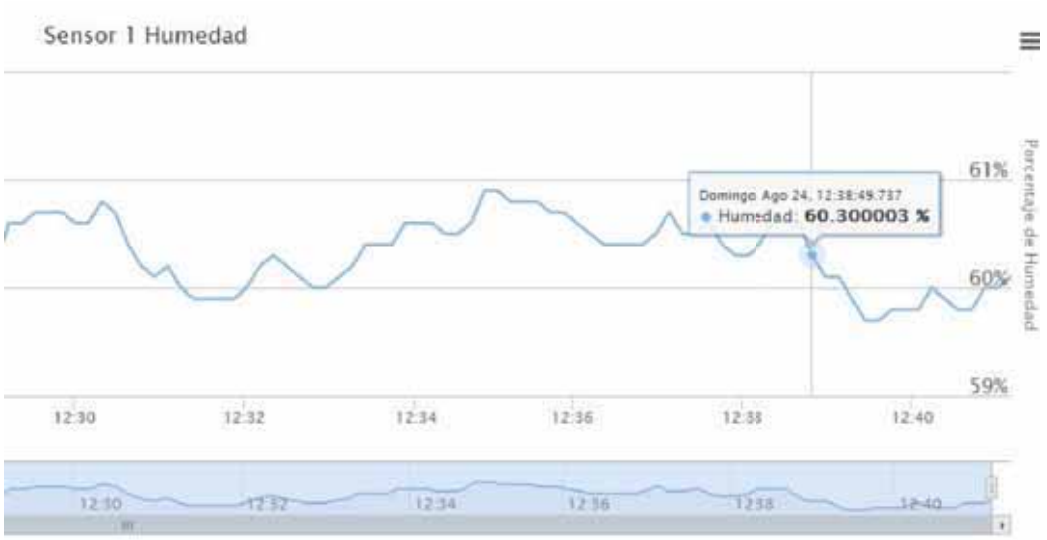


Figura 7.7. Grafica que se obtiene del sensor AM2302 con respecto a la humedad.

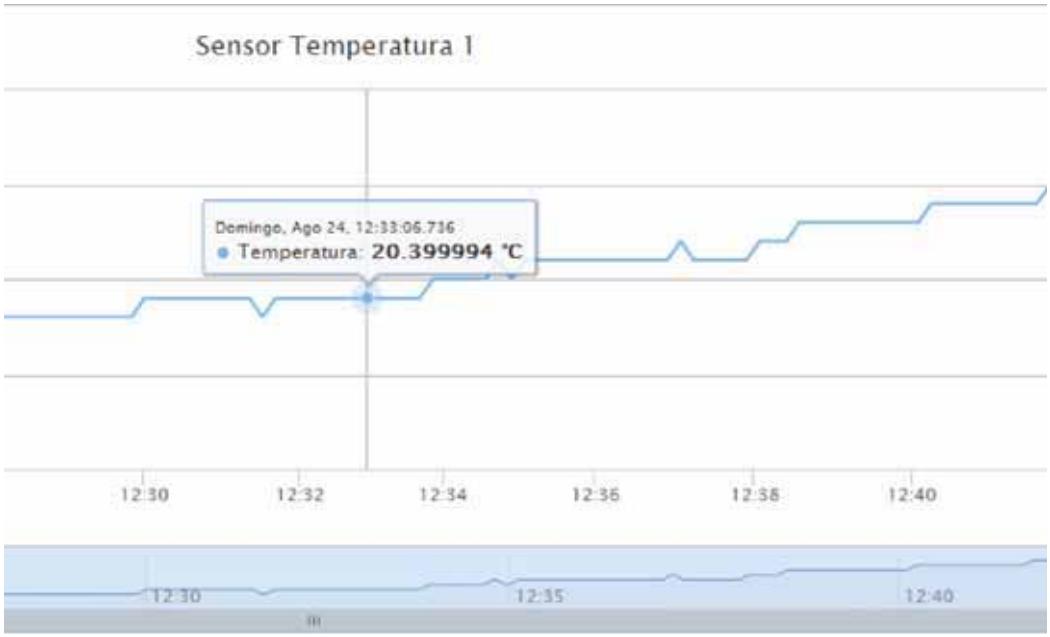


Figura 7.8. Grafica que se obtiene del sensor AM2302 con respecto a la humedad

Figura 7.9. Grafica que se obtiene del sensor HMZ-433A1 con respecto a la Temperatura.

En la Figura 77.10 se observa la red de monitoreo donde del lado izquierdo se encuentran los sensores (HMZ-433A1 y AM2302) mediante los Xbeen (modelo Xb24-z7wit004) que están programado como routers se puede envía la información, el otro xbeen (modelo Xb24-z7wit004) configurado como coordinador recibe la información y la envía a un servidor remoto que es la PC que se puede observar.

Figura 7.10 Red de monitoreo usando tecnología Zigbee.

Conclusiones

Se diseñó una red de monitoreo remoto de forma inalámbrica con el uso de sensores capaces de medir temperatura y humedad además del uso de dispositivos zigbee, el desarrollo que se llevó acabo fue exitoso ya que los objetivos planteados al principio de este documento fueron cumplidos, para ello se utilizaron dispositivos programables capaces de llevar a cabo la recopilación de información permitiendo la recepción y trasmisión de información.

De manera concreta este proyecto cumplió con los objetivos que fueron desarrollados de manera adecuada, aplicando conocimientos de dos ingenierías (en computación y en electrónica) que fueron combinadas para poder llegar a cumplir las expectativas correspondientes, además de investigaciones realizadas sobre el tema y comprendiendo proyectos que anteriormente ya se habían realizado, se enfrentó a varios retos que se pudieron solucionar de manera eficiente y sin complejidades, como ya se había menando existen distintas investigaciones relacionadas en cuanto al tema de monitoreo de variables de medición así como el uso de tecnologías inalámbricas que cada vez son más fáciles de desarrollar y obtener.

Bibliografía

[1] C.U. Romo Guerrero, N. Jiménez López, R.E Hurtado Torres "Enlace de comunicación de vehículos de exploración de una estación", Tesis*, Instituto Politécnico Nacional, Escuela Superior de ingeniería Mecánica y Eléctrica*, México, 2009.

[2] O.D. Durán Robles, "Geolocalización usando tecnología WiFi (Wireless fidelity), Tesis*, Universidad Nacional Autónoma de México, Faculta de ingeniería, México*, 2009.

[3] I. Barneda Faudot, " Zigbee aplicado a la trasmisión de datos de sensores biomédicos, Tesis*, Universidad Autónoma de Barcelona , Ingeniería Técnica en Telecomunicaciones-especialidad en Sistemas Electrónicos* , 2008.

[4] R. Rodríguez Martínez," "Monitoreo remoto de temperatura usando sensores 1-wire",Proyecto terminal, Ingeniería en Electrónica, Universidad Autónoma Metropolitana unidad Azcapotzalco, México, 2013.

[5] J.L. Ortigosa Flores, “Software colaborativo para dispositivos móviles con comunicación Bluetooth”, Proyecto terminal, Ingeniería en Computación, Universidad Autónoma Metropolitana unidad Azcapotzalco, México, 2013.

[6] B. López Pérez, R. Hernández Jiménez. “Transmisión y registro de posiciones geográficas de dispositivos móviles, usando el protocolo Bluetooth”, Proyecto terminal, Ingeniería en Computación, Universidad Autónoma Metropolitana unidad Azcapotzalco, México, 2013.

[7] Vega, J. I.; Salgado G.; Lagos, M. A. (2012). Informe del Proyecto de Investigación: Aplicaciones con Embedded Microcontroladores, Periodo 2002-2012. Departamento de Electrónica. Universidad Autónoma Metropolitana-Azcapotzalco.

[8] Vega, J. I.; Salgado, G.; Lagos, M. A. (2010). Sistema de monitoreo de presión con microcontrolador. Vigésima primera Reunión de Otoño de Comunicaciones, computación, Electrónica y Exposición Industrial ROC&C 2010. Acapulco, Guerrero, México. 28 de Noviembre-4 de Diciembre de 2010.

[9] Vega, J. I.; Salgado, G.; Lagos, M. A. (2011). Monitor y generador de alertas por variaciones de temperatura y humedad en un centro de datos. 10º Congreso Interamericano de Computación Aplicada a la Industria de Procesos (CAIP’2011). Girona, España. 30 de Mayo-3 de Junio de 2011.

[10] Vega, J. I.; Salgado, G.; Lagos, M. A. (2012). Control digital de temperatura de un centro de datos usando el protocolo X10. Vigésima tercera Reunión de Otoño de Comunicaciones, computación, Electrónica y Exposición Industrial ROC&C 2012. Acapulco, Guerrero, México. 11-17 de Noviembre de 2012.

[11] Wang, W.; He, G. (2011). Research on ZigBee wireless communication technology. 2011 International Conference on Electrical and Control Engineering (ICECE). 16-18 Sept. 2011. Page(s): 1245-1249.

[12] Vişan, D.A.; Jurian, M.; Cioc, I.B. (2010). Wireless Measurement System Based on ZigBee Transmission Technology. 2010 33rd International Spring Seminar on Electronics Technology (ISSE). 12-16 May 2010. Page(s): 464-467.

[13] Jung, J.Y.; Lee, J.W. (2008). ZigBee Device Access Control and Reliable Data Transmission in ZigBee Based Health Monitoring System. 2008. ICACT 2008. 10th International Conference on Advanced Communication Technology. 17-20 Feb. 2008. Volume: 1. Page(s): 795-797.

[14] Li, J.; Zhu, X. ; Tang N.; Sui. (2010). Study on ZigBee network architecture and routing algorithm. 2010 2nd International Conference on (Volume:2) Signal Processing Systems (ICSPS). 5-7 July 2010. Page(s): V2-389-V2-393 E-ISBN: 978-1-4244-6893-5 Print ISBN: 978-1-4244-6892-8.

[15] Chen, C.; Lin, H.; Huang, Y. (2012). Power management system based on ZigBee. 2012 International Conference on Security and Identification (ASID). 24-26 Aug. 2012. Page(s): 1–5. ISSN: 2163-5048 E-ISBN: 978-1-4673-2143-3 Print ISBN: 978-1-4673-2144-0.

[16] Hirakata, Y.; Nakamura, A.; Ohno, K.; Itami, M. (2012). Navigation system using ZigBee wireless sensor network for parking. 2012 12th International Conference on ITS Telecommunications (ITST). Page(s): 605-609.

[17]Digi http://www.digi.com

[18] ZigBee Alliance http://www.zigbee.org

[19] IEEE 802.15 http://www.ieee802.org/15

[20]http://www.mysql.com/

[21]www.apache.org

[22] http://php.net/

Apéndices

A. Codificación de Processing

Es este apartado podemos encontrar la codificación que se realizó que permitió la creación de los ficheros txt, Esta información fue enviada de los sensores a los dispositivos Zigbee conectados como Router y posteriormente al Dispositivo Zigbee coordinador, el software Processing se encargó de decodificar la información y enviarla a los ficheros mencionados.

```
//Importamos la librerÃ-a serial
import processing.serial.*;

//Declaramos un objeto tipo Serial y un int llamado //colorin (color es palabra reservada)
Serial puerto;
int colorin;

/// inicialización de variablesque almacenran los valores obtenido del dispositivo coordinador
int t2= 0;
int t1= 0;
int h2= 0;
int h1= 0;

PrintWriter salidah1;
PrintWriter salidat1;
PrintWriter salidah2;
PrintWriter salidat2;

float temperatura1=0;
float humedad1=0;
float temperatura2=0;
float humedad2=0;
void setup(){
//Delimitamos la ventana del programa
size(400,300);
//Processing nos pasarÃ¡ por consola la lista de dispositivos USB que
//estÃ¡n conectados en este momento en el ordenador
print(Serial.list());
//Conectamos nuestro objeto puerto al primer dispositivo //de la lista (o al que corresponda)
colorin=0;
puerto=new Serial(this,Serial.list()[0],57600);
salidat1 = createWriter("temperatura1.txt");
salidah1 = createWriter("humedad1.txt");
salidah2 = createWriter("humedad2.txt");
salidat2 = createWriter("temperatura2.txt");
}
```

```
void serialEvent (Serial puerto){
  String inString = puerto.readStringUntil('\n');
 if (inString != null) {
    float comp = float (inString);

  inString = trim(inString);
  if(comp >= 100 && comp <= 200){
  humedad1 = float(inString) - 100;
  h1= h1+1;
  salidah1.print(h1);
  salidah1.print(TAB+ " "+humedad1);
  salidah1.print(TAB+"    "+year()+"-"+month()+"-"+day()+" "+hour()+":"+minute()+":"+second());
  salidah1.println("");
 }
 salidah1.flush();

   if(comp >200 && comp <=400){
   temperatura1 = float(inString)-300;
   t1 = t1 +1;
salidat1.print(t1);
   salidat1.print(TAB+" "+temperatura1); //Guardo en fichero el valor de temperatura
salidat1.print(TAB+"    "+year()+"-"+month()+"-"+day()+" "+hour()+":"+minute()+":"+second());
salidat1.println(""); //Salto de lÃ-nea
}
     salidat1.flush();

   if(comp >400 && comp <=500){
   humedad2 = float(inString)-400;
   h2 = h2 + 1;
salidah2.print(h2);
   salidah2.print(TAB+" "+humedad2); //Guardo en fichero el valor de temperatura
salidah2.print(TAB+"    "+year()+"-"+month()+"-"+day()+" "+hour()+":"+minute()+":"+second());
salidah2.println(""); //Salto de lÃ-nea
}
     salidah2.flush();

   if(comp >500 && comp <=700){
   temperatura2 = float(inString)-600;
   t2 = t2 + 1;
salidat2.print(t2);
   salidat2.print(TAB+" "+temperatura2); //Guardo en fichero el valor de temperatura
salidat2.print(TAB+"    "+year()+"-"+month()+"-"+day()+" "+hour()+":"+minute()+":"+second());
salidat2.println(""); //Salto de lÃ-nea
  }
     salidat2.flush();
}
}
```

```
void draw(){
//Comprobamos si hay datos en el puerto serie
if (puerto.available()>0){
  colorin=puerto.read();
}
//El color de fondo oscilarÃ¡ desde negro a azul completo
background(0,0,colorin);
```

B. Codificación de Php

En este apartado podemos encontrar los códigos que se realizaron para la implementación de este proyecto mediante el uso de Php, lo que implica la conexión a la base de datos y la realización de consultas.

Como las tablas que se realizaron en Mysql son similares para la temperatura y humedad solo cambia en la forma en que se le nombro en esta sección podemos encontrar los siguientes códigos de los archivos datoshum1, ghum1, datoshum2, ghum2, datostem1, gtem1, datostem2, gtem2.

Para los archivos datoshum1, datoshum2, datostem1, datostem2 básicamente se manda a llamar las tablas de las bases de datos para poder verificar si se está almacenando la información posteriormente se hace consultas para obtener la información es todo lo que hace en estos archivos, pero se relizarón de forma independiente para identificar las diferentes consultas y las distintas tablas.

Los archivos ghum1, ghum2, gtem1, gtem2 utilizan los archivos anteriormente mencionados (datoshum1, datoshum2, datostem1, datostem2), para poder realizar la interfaz grafica. Mediante el uso de algunas librerías ya definidas (`jquery.min.js, ighstock.js, exporting.js)`, que permiten que se grafique de forma más fácil.

❖ datoshum1

```
<?php
class ValoresTabla{// RandomTable{

    public $IDr = 0 ;
    //FunciÃ³n que crea y devuelve un objeto de conexiÃ³n a la
base de datos y chequea el estado de la misma.

function conectarBD(){
```

```
            $server = "localhost";
            $usuario = "root";
$pass = "";
            $BD = "monitoreo_remoto";
            //variable que guarda la conexiÃ³n de la base de datos
$conexion = mysqli_connect($server, $usuario, $pass, $BD);
//Comprobamos si la conexiÃ³n ha tenido exito
            if(!$conexion){
               echo 'Ha sucedido un error inexperado en la
conexion de la base de datos<br>';
            }
            //devolvemos el objeto de conexiÃ³n para usarlo en las
consultas
            return $conexion;
    }
    /*Desconectar la conexion a la base de datos*/
    function desconectarBD($conexion){
            //Cierra la conexiÃ³n y guarda el estado de la
operaciÃ³n en una variable
            $close = mysqli_close($conexion);
            //Comprobamos si se ha cerrado la conexiÃ³n
correctamente
            if(!$close){
               echo 'Ha sucedido un error inexperado en la
desconexion de la base de datos<br>';
            }
            //devuelve el estado del cierre de conexiÃ³n
            return $close;
    }

    //Devuelve un array multidimensional con el resultado de la
consulta
    function getArraySQL($sql){
        //Creamos la conexiÃ³n
        $conexion = $this->conectarBD();
        //generamos la consulta
if(!$result = mysqli_query($conexion, $sql)) die();

$rawdata = array();
        //guardamos en un array multidimensional todos los datos
de la consulta
$i=0;
        while($row = mysqli_fetch_array($result))
{
            //guardamos en rawdata todos los vectores/filas que
nos devuelve la consulta
            $rawdata[$i] = $row;
            $i++;
        }
        //Cerramos la base de datos
        $this->desconectarBD($conexion);
        //devolvemos rawdata
```

```
        return $rawdata;
    }
    //inserta en la base de datos un nuevo registro en la tabla
usuarios

    function incertardatos(){//insertRandom(){
        $conexion = $this->conectarBD();
       $lineas = file('humedad1.txt') or die ("problemas al
abrir");

foreach ($lineas as $linea_num => $linea)
{
     $datos = explode("\t",$linea);

        $id = trim($datos[0]);
     $h1 = trim($datos[1]);
     $t1 = trim($datos[2]);
$sql = "INSERT INTO shum1(id,h1,t1)
                 VALUES('$id','$h1','$t1')";
//hacemos la consulta y la comprobamos

        $consulta = mysqli_query($conexion,$sql);

       // if(!$consulta){
           // echo "No se ha podido insertar en la base de
datos<br><br>".mysqli_error($conexion);

        }
        //Desconectamos la base de datos
        $this->desconectarBD($conexion);

        //devolvemos el resultado de la consulta (true o false)
        return $consulta;

    }
    function getAllInfo(){
        //Creamos la consulta
        $sql = "SELECT * FROM shum1;";
        //obtenemos el array con toda la informaciÃ³n
return $this->getArraySQL($sql);
    }
}



?>
```

- **ghum1**

```
<HTML>
<BODY>
```

```
<meta charset="utf-8" content="20" http-equiv="REFRESH">
<?php
require_once("datoshum1.php");

//Creamos un objeto de la clase ValoresTabla
$rand = new ValoresTabla();
//insertamos un valores
$rand->incertardatos();
//obtenemos toda la informaciÃ³n de la tabla variables1
$rawdata = $rand->getAllInfo();

//nos creamos dos arrays para almacenar el tiempo y el valor
numÃ©rico
$valoresArray;
$timeArray;
//en un bucle for obtenemos en cada iteraciÃ³n el valor nÃºmerico
y
//el TIMESTAMP del tiempo y lo almacenamos en los arrays
for($i = 0 ;$i<count($rawdata);$i++){
$valoresArray[$i]= $rawdata[$i][1];
    //OBTENEMOS EL TIMESTAMP
$time= $rawdata[$i][2];

    $date = new DateTime($time);
//ALMACENAMOS EL TIMESTAMP EN EL ARRAY

$timeArray[$i] = $date->getTimestamp()*1000;
}

?>
<div id="contenedor"></div>

<script src= "jquery.min.js"></script>
<!-- Importo el archivo Javascript de Highcharts directamente
desde su servidor -->
<script src="highstock.js"></script>
<script src="exporting.js"></script>
<script type="text/javascript">
$(function () {
chartCPU = new Highcharts.StockChart({
    chart: {
        renderTo: 'contenedor'
        //defaultSeriesType: 'spline'

    },
    rangeSelector : {
        enabled: false
    },
    title: {
        text: 'Sensor 1 Humedad'
    },
    xAxis: {
```

```
        type: 'datetime'
        //tickPixelInterval: 150,
        //maxZoom: 20 * 1000
    },
    yAxis: {
        minPadding: 0.2,
        maxPadding: 0.2,
        title: {
            text: 'Porcentaje de Humedad',
margin: 10
        },
                    labels: {
                format: '{value}%',
                style: {
                    fontSize: '15px'
                },
                x: -3
            },
    },
    series: [{
        name: 'Humedad',
        tooltip: {
                valueSuffix: ' %'
            },
        data: (function() {
                // generate an array of random data
                var data = [];
<?php
                    for($i = 0 ;$i<count($rawdata);$i++){
                ?>
                data.push([<?php echo $timeArray[$i] *
1.000005119495;?>,<?php echo $valoresArray[$i] ;?>]);

<?php } ?>
                return data;
            })()
    }],
    credits: {
            enabled: false
}
});
});

</script>
</BODY>

</html>

//////////////////////
```

- **datoshum2**

```
<?php
class ValoresTabla{// RandomTable{

    public $IDr = 0 ;
    //FunciÃ³n que crea y devuelve un objeto de conexiÃ³n a la
base de datos y chequea el estado de la misma.

function conectarBD(){
            $server = "localhost";
            $usuario = "root";
$pass = "";
            $BD = "monitoreo_remoto";
            //variable que guarda la conexiÃ³n de la base de datos
$conexion = mysqli_connect($server, $usuario, $pass, $BD);
//Comprobamos si la conexiÃ³n ha tenido exito
            if(!$conexion){
               echo 'Ha sucedido un error inexperado en la
conexion de la base de datos<br>';
            }
            //devolvemos el objeto de conexiÃ³n para usarlo en las
consultas
            return $conexion;
    }
    /*Desconectar la conexion a la base de datos*/
    function desconectarBD($conexion){
            //Cierra la conexiÃ³n y guarda el estado de la
operaciÃ³n en una variable
            $close = mysqli_close($conexion);
            //Comprobamos si se ha cerrado la conexiÃ³n
correctamente
            if(!$close){
               echo 'Ha sucedido un error inexperado en la
desconexion de la base de datos<br>';
            }
            //devuelve el estado del cierre de conexiÃ³n
            return $close;
    }

    //Devuelve un array multidimensional con el resultado de la
consulta
    function getArraySQL($sql){
        //Creamos la conexiÃ³n
        $conexion = $this->conectarBD();
        //generamos la consulta
if(!$result = mysqli_query($conexion, $sql)) die();

$rawdata = array();
```

```
        //guardamos en un array multidimensional todos los datos
de la consulta
$i=0;
        while($row = mysqli_fetch_array($result))
{
            //guardamos en rawdata todos los vectores/filas que
nos devuelve la consulta
            $rawdata[$i] = $row;
            $i++;
        }
        //Cerramos la base de datos
        $this->desconectarBD($conexion);
        //devolvemos rawdata
        return $rawdata;
    }
    //inserta en la base de datos un nuevo registro en la tabla
usuarios

    function incertardatos(){//insertRandom(){
        $conexion = $this->conectarBD();
       $lineas = file('humedad2.txt') or die ("problemas al
abrir");

foreach ($lineas as $linea_num => $linea)
{
     $datos = explode("\t",$linea);

        $id = trim($datos[0]);
     $h2 = trim($datos[1]);
     $t2 = trim($datos[2]);
        $sql = "INSERT INTO shum2(id,h2,t2)
                  VALUES('$id','$h2','$t2')";
//hacemos la consulta y la comprobamos

        $consulta = mysqli_query($conexion,$sql);

       // if(!$consulta){
           // echo "No se ha podido insertar en la base de
datos<br><br>".mysqli_error($conexion);

        }
        //Desconectamos la base de datos
        $this->desconectarBD($conexion);

        //devolvemos el resultado de la consulta (true o false)
        return $consulta;

    }
    function getAllInfo(){
        //Creamos la consulta
        $sql = "SELECT * FROM shum2;";
        //obtenemos el array con toda la informaciÃ³n
```

```
return $this->getArraySQL($sql);
    }
}


?>
```

- **ghum2**

```
<HTML>
<BODY>
<meta charset="utf-8" content="20" http-equiv="REFRESH">
<?php
require_once("datoshum2.php");

//Creamos un objeto de la clase ValoresTabla
$rand = new ValoresTabla();
//insertamos un valores
$rand->incertardatos();
//obtenemos toda la informaciÃ³n de la tabla variables1
$rawdata = $rand->getAllInfo();

//nos creamos dos arrays para almacenar el tiempo y el valor
numÃ©rico
$valoresArray;
$timeArray;
//en un bucle for obtenemos en cada iteraciÃ³n el valor nÃºmerico
y
//el TIMESTAMP del tiempo y lo almacenamos en los arrays
for($i = 0 ;$i<count($rawdata);$i++){
$valoresArray[$i]= $rawdata[$i][1];
    //OBTENEMOS EL TIMESTAMP
$time= $rawdata[$i][2];

    $date = new DateTime($time);
//ALMACENAMOS EL TIMESTAMP EN EL ARRAY

$timeArray[$i] = $date->getTimestamp()*1000;
}

?>
<div id="contenedor"></div>

<script src= "jquery.min.js"></script>
<!-- Importo el archivo Javascript de Highcharts directamente
desde su servidor -->
<script src="highstock.js"></script>
<script src="exporting.js"></script>
<script type="text/javascript">
$(function () {
chartCPU = new Highcharts.StockChart({
```

```
    chart: {
        renderTo: 'contenedor'
        //defaultSeriesType: 'spline'

    },
    rangeSelector : {
        enabled: false
    },
    title: {
        text: 'Sensor 2 Humedad'
    },
    xAxis: {
        type: 'datetime'
        //tickPixelInterval: 150,
        //maxZoom: 20 * 1000
},
    yAxis: {
        minPadding: 0.2,
        maxPadding: 0.2,
        title: {
            text: 'Porcentaje de Humedad',
            margin: 10
},
                    labels: {
                format: '{value}%',
                style: {
                    fontSize: '15px'
                },
                x: -3
            },
    },
    series: [{
        name: 'Humedad',
        tooltip: {
                valueSuffix: ' %'
            },
        data: (function() {
                // generate an array of random data
                var data = [];
<?php
                    for($i = 0 ;$i<count($rawdata);$i++){
                ?>
                data.push([<?php echo $timeArray[$i] *
1.000005119495;?>,<?php echo $valoresArray[$i] ;?>]);

<?php } ?>
                return data;
            })()
    }],
    credits: {
            enabled: false
    }
```

```
});
});

</script>
</BODY>

</html>
```

datostem1

```
<?php
class ValoresTabla{// RandomTable{

public $IDr = 0 ;
    //FunciÃ³n que crea y devuelve un objeto de conexiÃ³n a la
base de datos y chequea el estado de la misma.

function conectarBD(){
            $server = "localhost";
            $usuario = "root";
$pass = "";
            $BD = "monitoreo_remoto";
            //variable que guarda la conexiÃ³n de la base de datos
$conexion = mysqli_connect($server, $usuario, $pass, $BD);
//Comprobamos si la conexiÃ³n ha tenido exito
            if(!$conexion){
               echo 'Ha sucedido un error inexperado en la
conexion de la base de datos<br>';
            }
            //devolvemos el objeto de conexiÃ³n para usarlo en las
consultas
            return $conexion;
    }
    /*Desconectar la conexion a la base de datos*/
    function desconectarBD($conexion){
            //Cierra la conexiÃ³n y guarda el estado de la
operaciÃ³n en una variable
            $close = mysqli_close($conexion);
            //Comprobamos si se ha cerrado la conexiÃ³n
correctamente
            if(!$close){
               echo 'Ha sucedido un error inexperado en la
desconexion de la base de datos<br>';
            }
            //devuelve el estado del cierre de conexiÃ³n
            return $close;
    }

    //Devuelve un array multidimensional con el resultado de la
consulta
    function getArraySQL($sql){
```

```
        //Creamos la conexiÃ³n
        $conexion = $this->conectarBD();
        //generamos la consulta
if(!$result = mysqli_query($conexion, $sql)) die();

$rawdata = array();
        //guardamos en un array multidimensional todos los datos
de la consulta
$i=0;
        while($row = mysqli_fetch_array($result))
{
            //guardamos en rawdata todos los vectores/filas que
nos devuelve la consulta
            $rawdata[$i] = $row;
            $i++;
        }
        //Cerramos la base de datos
        $this->desconectarBD($conexion);
        //devolvemos rawdata
        return $rawdata;
    }
    //inserta en la base de datos un nuevo registro en la tabla
usuarios

    function incertardatos(){//insertRandom(){
        $conexion = $this->conectarBD();
       $lineas = file('temperatura1.txt') or die ("problemas al
abrir");

foreach ($lineas as $linea_num => $linea)
{
     $datos = explode("\t",$linea);

        $id = trim($datos[0]);
     $tem1 = trim($datos[1]);
     $t1 = trim($datos[2]);
$sql = "INSERT INTO stem1(id,tem1,t1)
                 VALUES('$id','$tem1','$t1')";
//hacemos la consulta y la comprobamos

        $consulta = mysqli_query($conexion,$sql);

       // if(!$consulta){
           // echo "No se ha podido insertar en la base de
datos<br><br>".mysqli_error($conexion);

        }
        //Desconectamos la base de datos
        $this->desconectarBD($conexion);

        //devolvemos el resultado de la consulta (true o false)
        return $consulta;
```

```
    }
    function getAllInfo(){
        //Creamos la consulta
        $sql = "SELECT * FROM stem1;";
        //obtenemos el array con toda la informaciÃ³n
return $this->getArraySQL($sql);
    }
}


?>
```

gtem1

```
<HTML>
<BODY>
<meta charset="utf-8" content="20" http-equiv="REFRESH">
<?php
require_once("datostem1.php");

//Creamos un objeto de la clase ValoresTabla
$rand = new ValoresTabla();
//insertamos un valores
$rand->incertardatos();
//obtenemos toda la informaciÃ³n de la tabla variables1
$rawdata = $rand->getAllInfo();

//nos creamos dos arrays para almacenar el tiempo y el valor
numÃ©rico
$valoresArray;
$timeArray;
//en un bucle for obtenemos en cada iteraciÃ³n el valor nÃºmerico
y
//el TIMESTAMP del tiempo y lo almacenamos en los arrays
for($i = 0 ;$i<count($rawdata);$i++){
$valoresArray[$i]= $rawdata[$i][1];
    //OBTENEMOS EL TIMESTAMP
$time= $rawdata[$i][2];

    $date = new DateTime($time);
//ALMACENAMOS EL TIMESTAMP EN EL ARRAY

$timeArray[$i] = $date->getTimestamp()*1000;
}

?>
<div id="contenedor"></div>

<script src= "jquery.min.js"></script>
```

```
<!-- Importo el archivo Javascript de Highcharts directamente
desde su servidor -->
<script src="highstock.js"></script>
<script src="exporting.js"></script>
<script type="text/javascript">
$(function () {
chartCPU = new Highcharts.StockChart({
    chart: {
        renderTo: 'contenedor'
        //defaultSeriesType: 'spline'

    },
    rangeSelector : {
        enabled: false
    },
    title: {
        text: 'Sensor Temperatura 1'
    },
    xAxis: {
        type: 'datetime'
        //tickPixelInterval: 150,
        //maxZoom: 20 * 1000
    },
    yAxis: {
        minPadding: 0.2,
        maxPadding: 0.2,
        title: {
            text: 'Grados Centigrados',
            margin: 10
        },
                    labels: {
                format: '{value}Â°',
                style: {
                    fontSize: '15px'
                },
                x: -3
            },
    },
    series: [{
        name: 'Temperatura',
        tooltip: {
                valueSuffix: ' Â°C'
            },
        data: (function() {
                // generate an array of random data
                var data = [];
<?php
                    for($i = 0 ;$i<count($rawdata);$i++){
                ?>
                data.push([<?php echo $timeArray[$i] *
1.000005119495;?>,<?php echo $valoresArray[$i] ;?>]);
```

```
<?php } ?>
                return data;
            })()
    }],
    credits: {
            enabled: false
}
});
});

</script>
</BODY>

</html>
```

datostem2

```
<?php
class ValoresTabla{// RandomTable{

    public $IDr = 0 ;
    //FunciÃ³n que crea y devuelve un objeto de conexiÃ³n a la
base de datos y chequea el estado de la misma.

function conectarBD(){
            $server = "localhost";
            $usuario = "root";
$pass = "";
            $BD = "monitoreo_remoto";
            //variable que guarda la conexiÃ³n de la base de datos
$conexion = mysqli_connect($server, $usuario, $pass, $BD);
//Comprobamos si la conexiÃ³n ha tenido exito
            if(!$conexion){
               echo 'Ha sucedido un error inexperado en la
conexion de la base de datos<br>';
            }
            //devolvemos el objeto de conexiÃ³n para usarlo en las
consultas
            return $conexion;
    }
    /*Desconectar la conexion a la base de datos*/
    function desconectarBD($conexion){
            //Cierra la conexiÃ³n y guarda el estado de la
operaciÃ³n en una variable
            $close = mysqli_close($conexion);
            //Comprobamos si se ha cerrado la conexiÃ³n
correctamente
            if(!$close){
               echo 'Ha sucedido un error inexperado en la
desconexion de la base de datos<br>';
            }
```

```
            //devuelve el estado del cierre de conexiÃ³n
            return $close;
    }

    //Devuelve un array multidimensional con el resultado de la
consulta
    function getArraySQL($sql){
        //Creamos la conexiÃ³n
        $conexion = $this->conectarBD();
        //generamos la consulta
if(!$result = mysqli_query($conexion, $sql)) die();

$rawdata = array();
        //guardamos en un array multidimensional todos los datos
de la consulta
$i=0;
        while($row = mysqli_fetch_array($result))
{
            //guardamos en rawdata todos los vectores/filas que
nos devuelve la consulta
            $rawdata[$i] = $row;
            $i++;
        }
        //Cerramos la base de datos
        $this->desconectarBD($conexion);
        //devolvemos rawdata
        return $rawdata;
    }
    //inserta en la base de datos un nuevo registro en la tabla
usuarios

    function incertardatos(){//insertRandom(){
        $conexion = $this->conectarBD();
       $lineas = file('temperatura2.txt') or die ("problemas al
abrir");

foreach ($lineas as $linea_num => $linea)
{
     $datos = explode("\t",$linea);

        $id = trim($datos[0]);
     $tem2 = trim($datos[1]);
     $t2 = trim($datos[2]);
$sql = "INSERT INTO stem2(id,tem2,t2)
                VALUES('$id','$tem2','$t2')";
//hacemos la consulta y la comprobamos

        $consulta = mysqli_query($conexion,$sql);

       // if(!$consulta){
           // echo "No se ha podido insertar en la base de
datos<br><br>".mysqli_error($conexion);
```

```
        }
        //Desconectamos la base de datos
        $this->desconectarBD($conexion);

        //devolvemos el resultado de la consulta (true o false)
        return $consulta;

    }
    function getAllInfo(){
        //Creamos la consulta
        $sql = "SELECT * FROM stem2;";
        //obtenemos el array con toda la informaciÃ³n
return $this->getArraySQL($sql);
    }
}


?>
```

gtem2

```
<HTML>
<BODY>
<meta charset="utf-8" content="20" http-equiv="REFRESH">
<?php
require_once("datostem2.php");

//Creamos un objeto de la clase ValoresTabla
$rand = new ValoresTabla();
//insertamos un valores
$rand->incertardatos();
//obtenemos toda la informaciÃ³n de la tabla variables1
$rawdata = $rand->getAllInfo();

//nos creamos dos arrays para almacenar el tiempo y el valor
numÃ©rico
$valoresArray;
$timeArray;
//en un bucle for obtenemos en cada iteraciÃ³n el valor nÃºmerico
y
//el TIMESTAMP del tiempo y lo almacenamos en los arrays
for($i = 0 ;$i<count($rawdata);$i++){
$valoresArray[$i]= $rawdata[$i][1];
    //OBTENEMOS EL TIMESTAMP
$time= $rawdata[$i][2];

    $date = new DateTime($time);
//ALMACENAMOS EL TIMESTAMP EN EL ARRAY

$timeArray[$i] = $date->getTimestamp()*1000;
```

```
}

?>
<div id="contenedor"></div>

<script src= "jquery.min.js"></script>
<!-- Importo el archivo Javascript de Highcharts directamente
desde su servidor -->
<script src="highstock.js"></script>
<script src="exporting.js"></script>
<script type="text/javascript">
$(function () {
chartCPU = new Highcharts.StockChart({
    chart: {
        renderTo: 'contenedor'
        //defaultSeriesType: 'spline'

    },
    rangeSelector : {
        enabled: false
    },
    title: {
        text: 'Sensor 1'
    },
    xAxis: {
        type: 'datetime'
        //tickPixelInterval: 150,
        //maxZoom: 20 * 1000
    },
    yAxis: {
        minPadding: 0.2,
        maxPadding: 0.2,
        title: {
            text: 'Grados Centigrados',
            margin: 10
        },
                    labels: {
                format: '{value}Â°',
                style: {
                    fontSize: '15px'
                },
                x: -3
            },
    },
    series: [{
        name: 'Temperatura',
        tooltip: {
                valueSuffix: ' Â°C'
            },
        data: (function() {
                // generate an array of random data
                var data = [];
```

```
<?php
                    for($i = 0 ;$i<count($rawdata);$i++){
                ?>
                data.push([<?php echo $timeArray[$i] *
1.000005119495;?>,<?php echo $valoresArray[$i] ;?>]);

<?php } ?>
                return data;
            })()
    }],
    credits: {
            enabled: false
}
});
});

</script>
</BODY>

</html>
```