

DESARROLLO MODULAR DE UN MODELO GENÉRICO PARA LA DOMÓTICA MEDIANTE COMUNICACIÓN INALÁMBRICA BASADA EN EL PROTOCOLO ZIGBEE

Víctor Manuel Jiménez Ramos

Instituto Tecnológico de Oaxaca
victorjimenezramos@gmail.com

Franco Gabriel Caballero Julián

Instituto Tecnológico de Oaxaca
francogcaballero@gmail.com

Roberto Tamar Castellanos Baltazar

Instituto Tecnológico de Oaxaca
rtcb.r21@gmail.com

Amalec Florean Pérez

Instituto Tecnológico de Oaxaca
amalecfp@hotmail.com

Resumen

En este artículo se presenta el desarrollo y los resultados de un proyecto de investigación aplicada, vigente en el Departamento de Ingeniería Eléctrica y Electrónica del Instituto Tecnológico de Oaxaca.

El proyecto consiste en integrar componentes de hardware y software para crear un sistema en domótica, que gestione los sensores y actuadores de manera inalámbrica mediante una red de equipos basados en el protocolo ZigBee perteneciente al grupo de Estándares del IEEE 802.15.4 en el área de redes inalámbricas personales (WPAN), utilizando la frecuencia de 2.4 GHz, en este se utilizan distintas herramientas tecnológicas para su implementación, digitalización, cuantificación, y transmisión; hasta llevar esos datos a un controlador que tiene la

capacidad de generar el monitoreo y seguridad en nuestros hogares. El modelo de la arquitectura del proyecto corresponde a una topología estrella, en esta se establece el coordinador de la red, el cual con la instalación de sensores configurados con Arduino se manipularán con los dispositivos XBee, permitiendo así la creación de los circuitos y programación de los sensores.

Palabras Claves: automatización, control, coordinador, monitoreo, señales.

Abstract

This article describes the development and the results of a research project applied force at the Department of Electrical and Electronic Engineering Institute of Technology of Oaxaca.

The project is to integrate hardware and software to create a system in home automation, which manages the sensors and actuators wirelessly through a network based on ZigBee protocol equipment belonging to the group of IEEE 802.15.4 standards in equipment wireless personal area networks (WPAN) using the 2.4 Ghz frequency in which various technological tools for implementation, digitization, quantification, and transmission are used; to bring that data to a controller that has the ability to generate monitoring and safety in our homes. The model of the project architecture corresponds to a star topology, where the network coordinator, which with installing configured with Arduino will be handled with XBee sensor devices is established, enabling the creation of circuits and programming sensors.

Keywords: automation, control, coordinator, monitoring, signals.

1. Introducción

El modo de vida de la sociedad moderna, la automatización de nuestras actividades, la forma de hacer negocios, la responsabilidad de salvaguardar los bienes, el cuidado del ambiente y la vida humana, han evolucionado con el único fin de proporcionar bienestar, comodidad, salud y alimentos a la gente; hace que tengamos que tomar medidas importantes para tener información de eventos en el instante en el que ocurren. Es ahí donde las tecnologías electrónicas, de

telecomunicaciones y de cómputo se hacen presente para observar el comportamiento en la empresa, en el hogar, en el estado, en una industria, en un hospital etc.

Un sistema domótico es capaz de recoger información proveniente de unos sensores o entradas, procesarla y emitir órdenes a unos actuadores o salidas. El sistema puede acceder a redes exteriores de comunicación en este caso se ocupa ZigBee.

La domótica permite dar respuesta a los requerimientos que plantean estos cambios sociales y las nuevas tendencias de nuestra forma de vida, facilitando el diseño de casas y hogares más humanos, más personales, polifuncionales y flexibles. Hoy en día, la domótica aporta soluciones dirigidas a todo tipo de viviendas. En definitiva, contribuye a aumentar la calidad de vida, hace más versátil la distribución de la casa, cambia las condiciones ambientales creando diferentes escenas predefinidas, y consigue que la vivienda sea más funcional al permitir desarrollar facetas domésticas, profesionales, y de ocio bajo un mismo techo.

En este trabajo se estudió el protocolo ZigBee que es la aplicación más difundida del IEEE 802.15.4 y que es una alianza de las más grandes empresas creadoras de hardware y software. Se define su arquitectura, alcances y limitaciones. Además se analizó la aplicabilidad de ZigBee a equipos alimentados a batería y se compara con otras tecnologías para mostrar que actualmente es la mejor alternativa para redes WPAN en donde el bajo consumo energético, el bajo costo y la simplicidad sean los compromisos de diseño. Las posibles aplicaciones son innumerables dadas las posibilidades de la domótica y las posibles necesidades de los propios usuarios, algunas son las siguientes:

- En el ámbito del ahorro energético:
 - ✓ Programación y zonificación de la climatización.
 - ✓ Racionalización de cargas eléctricas: desconexión de equipos de uso no prioritario en función del consumo eléctrico en un momento dado.
 - ✓ Reduce la potencia contratada.

- ✓ Gestión de tarifas, derivando el funcionamiento de algunos aparatos a horas de tarifa reducida.
- En el ámbito del nivel de confort:
 - ✓ Apagado general de todas las luces de la vivienda.
 - ✓ Automatización del apagado/encendido en cada punto de luz.
 - ✓ Regulación de la iluminación según el nivel de luminosidad ambiente.
 - ✓ Automatización de todos los distintos sistemas/instalaciones/equipos dotándolos de control eficiente y de fácil manejo.
 - ✓ Integración del portero al teléfono, o del video portero al televisor.
- En el ámbito de la protección personal y patrimonial:
 - ✓ Detección de un posible intruso.
 - ✓ Simulación de presencia.
 - ✓ Detección de conatos de incendio, fugas de gas, escapes de agua.
 - ✓ Alerta médica, teleasistencia.
 - ✓ Cerramiento de persianas puntual y seguro.
- En el ámbito de las comunicaciones:
 - ✓ Control remoto.
 - ✓ Transmisión de alarmas.
 - ✓ Intercomunicaciones.

2. Método

El proyecto se desarrolla mediante una metodología experimental de la convergencia de tecnologías, para la comunicación, control, del sistema confiable, portable y de fácil manejo para los usuarios.

Por lo cual se basa en una topología de estrella de red basada en el estándar IEEE 802.15.4 (ZigBee) ya que permite la comunicación inalámbrica estableciendo un dispositivo como Coordinador ZigBee (ZigBee Coordinator, ZC): con la función de controlar la red y los dispositivos para conectarse entre ellos, otros son dispositivos finales (ZigBee End Device, ZED) los cuales tienen conectados los sensores o actuadores y le notifican al coordinador si existe algún cambio de estado, esto con la finalidad de situar al coordinador en el centro, figura 1.

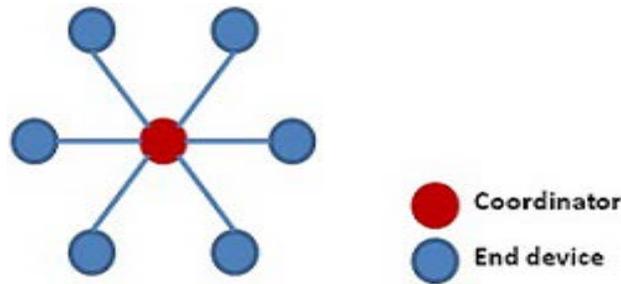


Figura 1 Estructura de la red ZigBee.

Programación de Red

La conexión entre los dispositivos Xbee se realiza con el programa X-CTU, configurando el coordinador y los dispositivos finales. Se inicia el programa X-CTU (figura 2) y veremos que efectivamente la tarjeta fue reconocida como COM3.

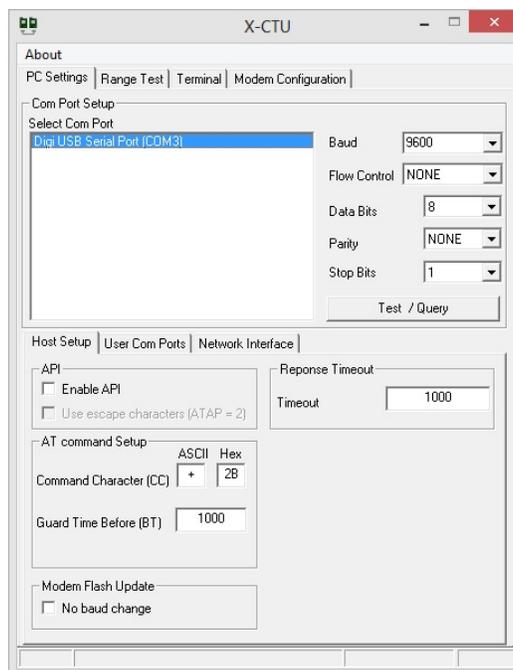


Figura 2 Programa X-CTU.

Para el coordinador se cambian los siguientes parámetros:

- PAN ID = 1234 (esta será la red a la cual se conectarán todos los dispositivos tanto coordinador como dispositivos finales).
- DL = FFFF (este parámetro se deja de fábrica para que el coordinador pueda comunicarse sin problemas con los dispositivos finales).

- CE = 1 Coordinator Enable (aquí tenemos 2 opciones 1 o 0 el primero es para poner el dispositivo como coordinador y el segundo como dispositivo final). Ya cambiados los parámetros anteriores se da click en el botón Write y esperamos a que termine de programarse el dispositivo Xbee, figura 3.

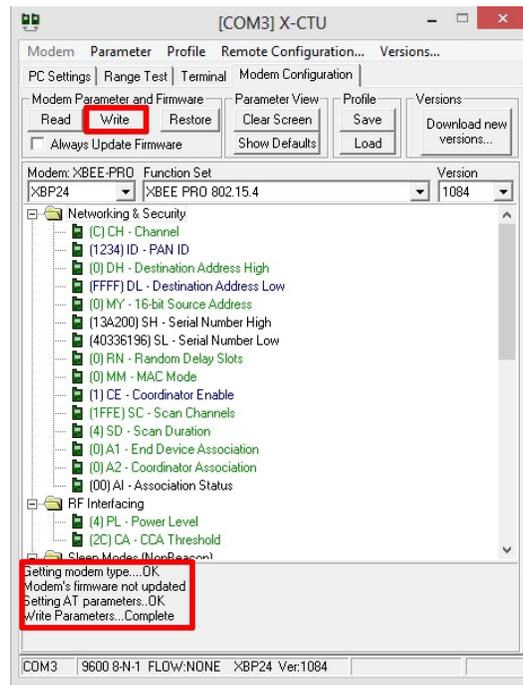


Figura 3 Click en Write.

Quedando el dispositivo programado como coordinador. En los dispositivos finales se configuran los siguientes parámetros:

- PAN ID = 1234 (esta será la red a la cual se conectarán todos los dispositivos tanto coordinador como dispositivos finales).
- MY = 1234 (esta será la dirección que contendrán todos los dispositivos finales y necesaria para que el coordinador pueda comunicarse con ellos, la dirección es igual para todos dispositivos finales, si se quiere cambiar el número no hay ningún problema).
- CE = 0 End Device (aquí tenemos 2 opciones 1 o 0 el segundo como dispositivo final). Configurados los parámetros anteriores se da click en el botón Write, quedando programado el dispositivo. Con esto quedan

configurados los dispositivos Xbee tanto coordinador como los dispositivos finales listos para ser usados en red.

Programación y configuración de sensores con Arduino.

El microcontrolador ocupado es el ATMEGA 328P el cual cuenta con 14 I/O Entradas/ salidas Digitales y 6 Entradas analógicas, se decide utilizar este dispositivo por sus lenguajes de programación C, C++. Para el sensor de Temperatura, se ocupan los siguientes materiales:

- 1 Tarjeta Arduino UNO
- 1 Microcontrolador ATMEGA 328P
- 1 Fuente de alimentación de 5 V
- 1 Resistencia 1 kohms
- 2 Resistencias de 330ohms
- 1 LM35
- 1 Cristal de 16 MHz
- 2 Capacitores de 22pF
- 2 LED's
- Cables de conexión

En Arduino se crea un sketch nuevo en el cual se ingresa la programación, figura 4. Esto hace que el LM35 que se ocupa como sensor de temperatura mande que temperatura se tiene en ese momento en grados centígrados, se visualiza a través de dos LED's los cuales se ubican en los pines 4 y 6 del microcontrolador (el pin 4 del microcontrolador es el pin 2 de la tarjeta Arduino UNO y el pin 6 del microcontrolador es el pin 4 de la tarjeta Arduino UNO estos fueron declarados en la programación), cuando el LM35 detecte más de 28 grados enviará a través del puerto serial la letra A la cual a su vez será enviada por el dispositivo Xbee (dispositivo final) una vez conectado y recibido por otro dispositivo Xbee (coordinador), se visualizara que el led del pin 4 se enciende y el led del pin 6 se apaga cuando se cumpla esta sentencia.

```

sketch_jul25a.cpp
float temperatura = 0; //Variable para la temperatura
int on=2; //El pin 2 contendra un led indicador
int off=4; //El pin 4 contendra un led indicador

void setup(){ //Configuración de los pin como entrada o salida
  Serial.begin (9600); //Se abre puerto serial para la comunicación con los dispositivos Xbee
  pinMode(2, OUTPUT); //Se declara on como una salida
  pinMode(4, OUTPUT); //Se declara off como una salida
}

void loop() //Configuración de los valores del sensor de temperatura
{
  temperatura = (5.0 * analogRead(A0)*100.0)/1023.0; //temperatura tendra el valor que sera mostrado visualmente a través de los LED's declarados

  if (temperatura >= 28){ //Si el valor de temperatura es mayor a 28 grados centigrados hara lo siguiente:
    Serial.println('A'); //Imprimir a través del puerto serial la letra A la cual sera enviada por el dispositivo Xbee y
    digitalWrite(2, HIGH); //que sera leída por el Coordinador, tambien se visualiza que se apaga el LED declarado en el
    digitalWrite(4, LOW); //pin 4 y se enciende el led declarado en el pin 2
  }
  if (temperatura < 28){ //Si el valor de temperatura es menor a 28 grados centigrados hara lo siguiente:
    Serial.println('B'); //Imprimir a través del puerto serial la letra B la cual sera enviada por el dispositivo Xbee y
    digitalWrite(2, LOW); //que sera leída por el Coordinador, tambien se visualiza que se enciende el LED declarado en el
    digitalWrite(4, HIGH); //pin 4 y se apaga el led declarado en el pin 2
  }
  delay(1000); //Ponemos 1 segundo para que no se saturate en la comunicación entre dispositivos Xbee
}

```

Figura 4 Programación de sensor de temperatura.

Cuando el LM35 tenga menos de 28 grados enviara a través del puerto serial la letra B la cual a su vez será enviada por el dispositivo Xbee (dispositivo final) y recibida por otro dispositivo Xbee (coordinador), se visualizará que el led del pin 4 se apaga y el led del pin 6 se enciende cuando se cumpla esta sentencia, todo esto con la finalidad de implementarlo en un *proto-board*, figura 5. Posteriormente se continúa con el sensor de presencia utilizando los siguientes materiales:

- 1 Tarjeta Arduino UNO
- 1 Microcontrolador ATMEGA 328P
- 1 Fuente de alimentación de 5V
- 1 Resistencia 1kohms
- 1 Resistencia de 330ohms
- 1 Sensor PIR
- 1 Cristal de 16MHz
- 2 Capacitores de 22pF
- 1 LED's
- Cables de conexión

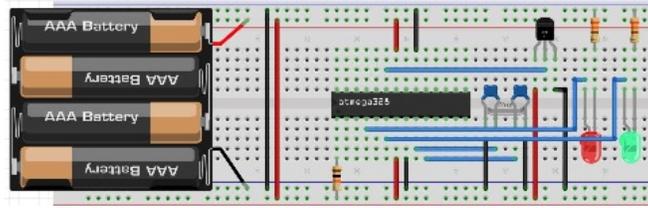


Figura 5 Diagrama de conexión del sensor de temperatura.

En Arduino se crea un sketch nuevo en el cual se ingresa la programación como se muestra en la figura 6. Este programa permite que el sensor PIR envíe una señal si detecta algún objeto en movimiento dentro de su rango de visibilidad, se puede visualizar a través de un led posicionado en el pin 19 del microcontrolador (el pin 19 del microcontrolador es el pin 13 de la tarjeta Arduino UNO el cual fue declarado en la programación), cuando el sensor PIR detecte movimiento emitirá a través del puerto serial la letra C la cual a su vez será enviada por el dispositivo Xbee (dispositivo final) una vez conectado y recibido por otro dispositivo Xbee (coordinador), se visualizara que el led se enciende cuando se cumpla esta sentencia. Cuando el sensor PIR no detecte a nadie enviará a través del puerto serial la letra D la cual permitirá visualizar que el led se apaga cuando se cumpla esta sentencia. Implementando el microcontrolador en un *protoboard* para obtener el diagrama del Sensor PIR, figura 7.

```

sketch_jul25a $
int pir=3;           //El pin 3 contendra el valor medido por el sensor PIR
int estado;         //Variable para el sensor PIR
int led=13;         //El pin 13 contendra un led indicador

void setup(){       //Configuración de los pin como entrada o salida
  Serial.begin(9600); //Se abre puerto serial para la comunicación con los dispositivos Xbee
  pinMode (3, INPUT); //Se declara pin 3 como una entrada
  pinMode(13, OUTPUT); //Se declara pin 13 como una salida
}

void loop(){        //Configuración de los valores del sensor de PIR
{
  estado=digitalRead(pir); //estado tendra el valor que sera mostrado visualmente a través del pin 13

if (estado==HIGH){  //Si el valor de estado es HIGH hara lo siguiente:
  Serial.println('C'); //Imprimir a través del puerto serial la letra C la cual sera enviada por el dispositivo Xbee y
  digitalWrite(13, HIGH); //que sera leida por el Coordinador, tambien se visualiza que se enciende el LED del pin 13
}
if (estado==LOW){   //Si el valor de estado es LOW hara lo siguiente:
  Serial.println('D'); //Imprimir a través del puerto serial la letra D la cual sera enviada por el dispositivo Xbee y
  digitalWrite(13, LOW); //que sera leida por el Coordinador, tambien se visualiza que se apaga el LED del pin 13
}
}
delay(1000);        //Ponemos 1 segundo para que no se sature en la comunicación entre dispositivos Xbee
}
    
```

Figura 6 Programación de sensor PIR.

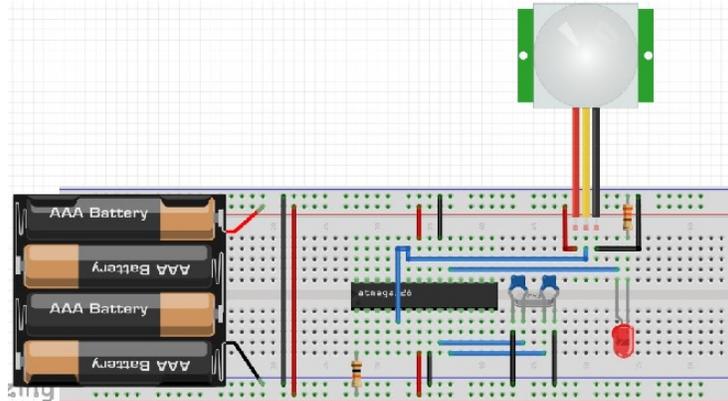


Figura 7 Diagrama de conexión de sensor PIR.

Posteriormente se continúa con el sensor de luminosidad utilizando los siguientes materiales:

- 1 Tarjeta Arduino UNO
- 1 Microcontrolador ATMEGA 328P
- 1 Fuente de alimentación de 5V
- 1 Resistencia 1 kohms
- 1 Resistencia de 470ohms
- 1 Resistencia de 330ohms
- 1 LDR
- 1 Cristal de 16MHz
- 2 Capacitores de 22pF
- 1 LED's
- Cables de conexión

En Arduino se crea un sketch nuevo en el cual se ingresa la programación como se muestra en la figura 8.

Esto permite que la LDR mande la resistencia que tiene cuando tiene luz y cuando no la tiene, será visualizado a través de un led que sale del pin 19 del microcontrolador (el pin 19 del microcontrolador es el pin 13 de la tarjeta Arduino UNO el cual fue declarado en la programación), cuando la LDR tenga menos de 490 ohms enviara a través del puerto serial la letra E la cual se visualizara que el

led se apaga cuando se cumpla esta sentencia. Cuando la LDR tenga más de 490 ohms se visualizará que el led se enciende cuando se cumpla esta sentencia. Permitiendo así determinar la luminosidad.

```
sketch_jul25a$
int LED = 13; //El LED esta conectado en el pin 13
int LDR = 0; //La LDR esta conectado en el pin A0
int luz = 0;
int valor_sensor = 0;

void setup() { //Configuración de los pin como entrada o salida
  Serial.begin(9600); //Se abre puerto serial para la comunicación con los dispositivos Xbee
  pinMode(LED,OUTPUT); //Se declara el LED como una salida
  pinMode(LDR,INPUT); //Se declara la LDR como una entrada
}

void loop () //Configuración de los valores del LDR
{
  valor_sensor = analogRead(LDR); //valor_sensor tendra el valor que nos marque la funcion analogRead * el valor que nos indique la LDR
  luz = (5.0 * valor_sensor * 100.0)/1024.0; //luz tendra el valor que sera mostrado visualmente a través del LED declarado en el pin 13

  if (luz <= 490){ //Si el valor de luz es menor a 490 ohms hara lo siguiente:
    Serial.println('E'); //Imprimir a través del puerto serial la letra E la cual sera enviada por el dispositivo Xbee y
    digitalWrite(13, LOW); //que sera leida por el Coordinador, tambien se visualiza que se apaga el LED declarado anteriormente
  }
  if (luz > 490){ //Si el valor de luz es mayor a 490 ohms hara lo siguiente:
    Serial.println('F'); //Imprimir a través del puerto serial la letra F la cual sera enviada por el dispositivo Xbee y
    digitalWrite(13, HIGH); //que sera leida por el Coordinador, tambien se visualiza que se enciende el LED declarado anteriormente
  }
  delay(1000); //Ponemos 1 segundo para que no se sature en la comunicación entre dispositivos Xbee
}
```

Figura 8 Programación de sensor de luminosidad.

Programación y configuración del coordinador con el microcontrolador ATMEGA328P

En esta etapa se realiza la configuración para el manejo y sincronización de los dispositivos, es por ello que se ocupan los siguientes materiales:

- 1 Tarjeta Arduino UNO
- 1 Microcontrolador ATMEGA 328P
- 1 Fuente de alimentación de 5 V
- 1 Fuente de alimentación de 3.3 V
- 1 Resistencia 1kohms
- 1 Resistencia de 330 ohms
- 1 Cristal de 16 MHz
- 2 Capacitores de 22pF
- 4 LED's
- Cables de conexión

En Arduino se crea un sketch nuevo en el cual se ingresa la programación, figura 9. Esto permite que el microcontrolador ATMEGA 328P sea el coordinador recibiendo toda la información de los sensores y mostrando su funcionamiento a través de LED's si están funcionando correctamente los sensores.

```

sketch_jul25a$
#include <SoftwareSerial.h> //Se llama a esta libreria para que podamos habilitar otros puerto para
//la comunicación con bluetooth
int temp=13; //Se declara el pin 13 que contendra una led indicador
int senpir=12; //Se declara el pin 12 que contendra una led indicador
int ldr=11; //Se declara el pin 11 que contendra una led indicador
int sonic=10; //Se declara el pin 10 que contendra una led indicador
int onoff=0; //Se declara onoff para hacer las comparaciones
int datain=0; //Se declara datain para hacer las comparaciones

void loop(){ //Configuración de los valores que ocupa el coordinador para poder comunicarse
data=Ssl.available(); //data sera igual al valor contenido por el software serial del bluetooth
datain=Serial.available(); //datain sera igual al valor contenido por el software serial del xbee
if(datain>0){ //Preguntamos si hay algun valor mayor a 0 en datain
onoff=Serial.read(); //onoff sera igual al valor contenido por software serial del xbee
}
if(data>0){ //Preguntamos si hay algun valor mayor a 0 en data
ONOFF=Ssl.read(); //ONOFF sera igual al valor contenido por software serial del bluetooth
}
if(onoff == 'A'){ //Si tenemos una letra A en el puerto serial hara lo siguiente:
digitalWrite(temp, HIGH); //El pin que contiene temp se encendera
Ssl.println("Temperatura Alta"); //Por el puerto serial del bluetooth veremos el mensaje Temperatura Alta
}
if(onoff == 'B'){ //Si tenemos una letra B en el puerto serial hara lo siguiente:
digitalWrite(temp, LOW); //El pin que contiene temp se apagara
}
if(onoff == 'C'){ //Si tenemos una letra C en el puerto serial hara lo siguiente:
digitalWrite(senpir, HIGH); //El pin que contiene senpir se encendera
Ssl.println("Presencia"); //Por el puerto serial del bluetooth veremos el mensaje Presencia
}
if(onoff == 'D'){ //Si tenemos una letra D en el puerto serial hara lo siguiente:
digitalWrite(senpir, LOW); //El pin que contiene senpir se apagara
}
if(onoff == 'E'){ //Si tenemos una letra E en el puerto serial hara lo siguiente:
digitalWrite(ldr, LOW); //El pin que contiene ldr se apagara
}
if(onoff == 'F'){ //Si tenemos una letra F en el puerto serial hara lo siguiente:
digitalWrite(ldr, HIGH); //El pin que contiene ldr se encendera
Ssl.println("Noche"); //Por el puerto serial del bluetooth veremos el mensaje Noche
}
if(onoff == 'H'){ //Si tenemos una letra H en el puerto serial hara lo siguiente:
digitalWrite(sonic, HIGH); //El pin que contiene sonic se encendera
Ssl.println("Obstaculo detectado"); //Por el puerto serial del bluetooth veremos el mensaje Obstaculo detectado
}
if(onoff == 'G'){ //Si tenemos una letra G en el puerto serial hara lo siguiente:
digitalWrite(sonic, LOW); //El pin que contiene sonic se apagara
}
delay(100); //Ponemos 100 ms para que no se sature en la comunicación.
}

```

Figura 9 Programación del Coordinador.

3. Resultados

Como resultados se obtiene el diseño de los sensores con comunicación ZigBee, permitiendo crear un modelo básico en domótica, que nos permite visualizar y manipular forma inalámbrica favoreciendo la portabilidad, figuras 10 a 15.

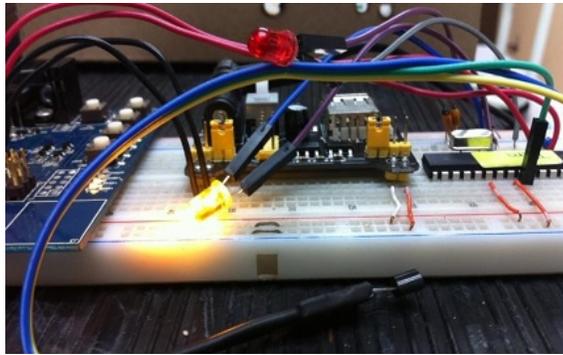


Figura 10 Sensor de temperatura.

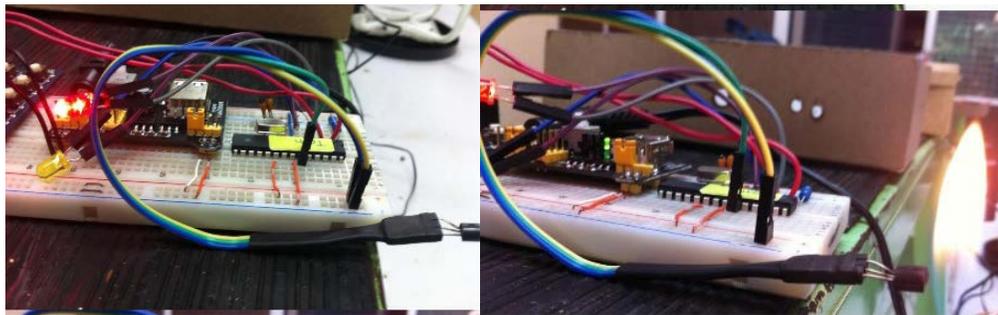


Figura 11 Incremento de temperatura.

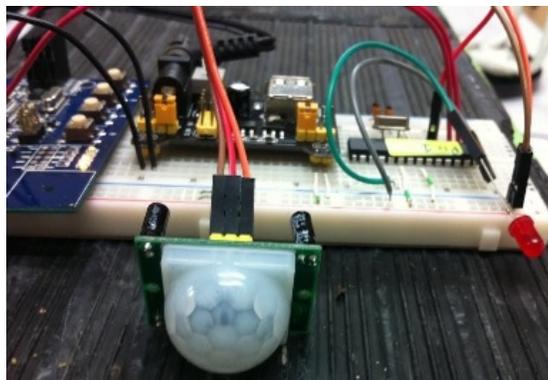


Figura 12 Sensor de presencia.

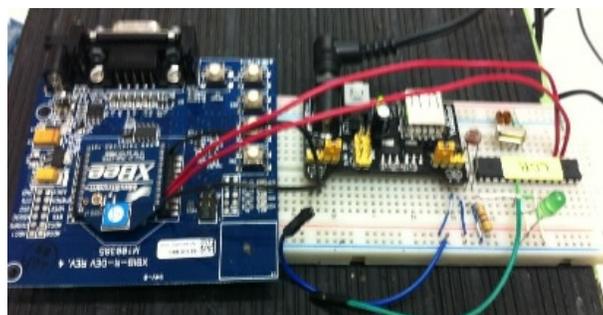


Figura 13 Sensor de luminosidad.

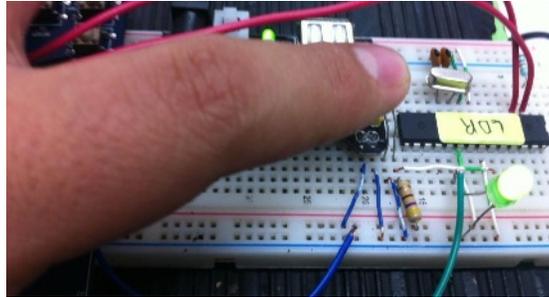


Figura 14 Led Indicador de luminosidad.

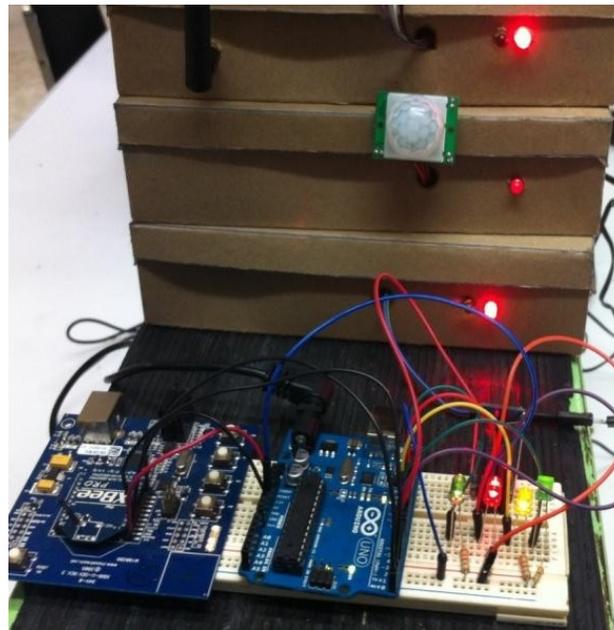


Figura 15 Activación y funcionamiento de sensores (LED's indicadores del coordinador).

Logrando la generación de circuitos y nuevas tecnologías. En estos tiempos las tecnologías de la información y la comunicación nos benefician a todos, de manera que tenemos la posibilidad de elegir dispositivos que se adapten a las necesidades facilitando nuestra vida diaria.

4. Discusión

El proyecto se origina por la necesidad de tener una mayor seguridad en nuestros hogares, mediante la creación y configuración de redes inalámbricas logrando tener un mejor monitoreo y seguridad de nuestros hogares mientras no estemos, esto se obtiene con la instalación de sensores los cuales configurados

de la manera correcta con Arduino se manipulan de forma inalámbrica con la ayuda de los dispositivos Xbee, permitiendo la portabilidad de los dispositivos, así como su fácil instalación, alimentados con baterías

El monitoreo en tiempo real permitirá que nosotros ahorremos tanto en dinero como en personal. En estos tiempos las tecnologías de la información y la comunicación nos benefician a todos, de manera que tenemos la posibilidad de elegir dispositivos que se adapten a nuestras necesidades para hacernos la vida más práctica.

5. Bibliografía y Referencias

- [1] Torrente, O. (2013). *Arduino Curso Práctico*. México: Alfaomega.
- [2] Olifer N y Olifer V. (2009). *Redes de Computadoras*. México: McGraw-Hill.
- [3] Tanenbaum A. (2004). *Redes de Computadoras*. Cuarta edición. México: Pearson.
- [4] Groover, M., Weiss, M. Nagel, R., y Odrey, N. (1990). *Robótica Industrial, Tecnología, programación y aplicaciones*. México: McGraw-Hill
- [5] Ceballos Sierra, F. J. (2008). *JAVA 2, Interfaces Gráficas y Aplicaciones para Internet*. México: RA-MA.
- [6] Stremier F.G. (1998). *Introducción a los Sistemas de Comunicación*. Tercera edición. México: Pearson-Addison Wesley
- [7] Herrera Pérez, E. (2003). *Tecnologías y redes de transmisión de datos*. México: Limusa.