

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA

TRABAJO FIN DE GRADO

TRANSMISIÓN DE SEÑALES ANALÓGICAS CON XBEE EN EL ENTORNO LABVIEW

DOCUMENTO 1- MEMORIA

Alumno/Alumna: Bengoetxea Abasolo Aitor
Director/Directora (1): Oleagordia Aguirre Iñigo Javier

Curso: 2018-2019

Fecha: 30/12/2018

RESUMEN

Este trabajo fin de grado está dividido en dos partes.

En la primera de ellas, se representa el comportamiento de un sensor RTD, concretamente una PT100. Se realiza la recogida de datos y se introduce en una plantilla Excel para comprobar que realmente este tipo de sensores se comporta de una manera lineal. Para poder sacar el máximo rendimiento a este monográfico se ha realizado la recogida de datos mediante un Arduino, concretamente el modelo "UNO". También se ha realizado la implementación del programa ".hex" de Arduino a la plataforma LabView, donde se ha podido observar el funcionamiento de la PT100 de una manera más visual.

En la segunda parte, se implementa una red inalámbrica mediante un sistema basado en dispositivos XBee. Se explicará detalladamente los modos de funcionamiento más comunes con estos dispositivos y cual ha sido la configuración utilizada para llevar a cabo este proyecto.

Cabe destacar que este TFG se ha hecho en el sistema operativo OS X 10.10 Yosemite en su totalidad. Con lo que no funcionará en el sistema operativo de Windows.

Palabras clave: XBee, PT100, RTD, LabView.

ABSTRACT

This final grade is divided in two parts.

First of all, we represent the behaviour of an RTD sensor (a PT100). We recollect data and we introduce it on an Excel project so that we can create a graphic. That graphic is going to show us how the sensor has a linear behaviour. We have used an Arduino “UNO” for the data acquisition. Also, we have made the implementation with the Arduino and LabView.

Later, using the XBee modules, we have created a wireless network. The explanation of how this modules must be configured is done.

It is important to say that this project has been made on OS X 10.10 Yosemite. So, if you try to do the same project on Windows it is not going to work.

Keywords: XBee, PT100, RTD, LabView.

LABURPENA

Gradu amaierako lana bi ataletan banatu egin da.

Lehenengo partean, RTD sentsore baten jokabidea aztertu egin da. Datuak bildu ditugu geroago Excel batean zahartu ahal izateko. Excel horren bidez, grafika batzuk lortuko ditugu. Grafika horretan ikuziko dugu nola RTD sentsorea linealki jokatuko den. Datuak hartzeko Arduino “UNO” erabili dugu. Gero Arduinoa LabViewrekin batera erabili dugu.

Bigarren partean, XBee moduloak erabili ditugu kablerik gabeko sare bat sohortu ahal izateko. Erabili dugun konfigurazioa azalduta dago.

Garrantzitsua da jakitea projectu ahu OS X 10.10 Yosemitean eginda dagoela. Ez du Windowsen funtzionatu.

Hitz klabeak: XBee, PT100, RTD, LabView.

ÍNDICE

INTRODUCCIÓN.....	7
OBJETIVOS.....	8
BENEFICIOS QUE APORTA EL TRABAJO.....	9
DESCRIPCIÓN DE REQUERIMIENTOS	10
ANÁLISIS DE ALTERNATIVAS	11
DESCRIPCIÓN GENERAL	13
DIAGRAMA DE BLOQUES GENERAL.....	13
SENSOR O TRANSDUCTOR	15
ACONDICIONAMIENTO DE LA SEÑAL.....	16
CONVERTIDOR A/D.....	18
ZIGBEE.....	21
TOPOLIGÍAS ZIGBEE	23
MODOS DE OPERACIÓN DE LOS MÓDULOS XBEE	24
XCTU Y SU CONFIGURACIÓN	32
ESQUEMA DE LABVIEW	33
PRESENTACIÓN	34
DIAGRAMA DE GANTT	35
PROBLEMAS A LA HORA DE REALIZAR EL PROYECTO	36
RESULTADOS	38
BIBLIOGRAFÍA	40

ÍNDICE DE FIGURAS

<i>Figura 6. 1 Diagrama de bloques del sistema.....</i>	<i>13</i>
<i>Figura 6. 2 PT100 con la configuración a 3 hilos</i>	<i>15</i>
<i>Figura 6. 3 Acondicionamiento de la señal.....</i>	<i>16</i>
<i>Figura 6. 4 Esquema del Arduino UNO</i>	<i>18</i>
<i>Figura 6. 5 SparkFun Xbee Shield</i>	<i>19</i>
<i>Figura 6. 6 El Arduino acoplado a la SparkFun.....</i>	<i>19</i>
<i>Figura 6. 7 El Arduino acoplado al XBee Explorer USB</i>	<i>20</i>
<i>Figura 6. 8 Topologías ZigBee.....</i>	<i>23</i>
<i>Figura 6. 9 XBee en modo transparente.....</i>	<i>25</i>
<i>Figura 6. 10 Estructura de un frame en conexión API</i>	<i>26</i>
<i>Figura 6. 11 Tres redes PAN diferentes</i>	<i>27</i>
<i>Figura 6. 12 Vista superior de los pines de un XBee</i>	<i>28</i>
<i>Figura 6. 13 Red de Broadcast.....</i>	<i>29</i>
<i>Figura 6. 14 Frecuencias de una conexión punto a multipunto.....</i>	<i>30</i>
<i>Figura 6. 15 Sintaxis de un comando AT</i>	<i>31</i>
<i>Figura 6. 16 Panel de configuración de XBee</i>	<i>32</i>
<i>Figura 6. 17 Diagrama de bloques LabView</i>	<i>33</i>
<i>Figura 7. 1 Organización del Diagrama de Gantt.....</i>	<i>35</i>
<i>Figura 7. 2 Diagrama de Gantt.....</i>	<i>35</i>
<i>Figura 8. 1 Aplicación de Wineskin</i>	<i>36</i>
<i>Figura 8. 2 X-CTU en el punto de reconocer los puertos</i>	<i>37</i>
<i>Figura 9. 1 X-CTU en el punto de reconocer los puertos</i>	<i>38</i>
<i>Figura 9. 2 Tabla de los resultados en Excel.....</i>	<i>38</i>
<i>Figura 9. 3 Gráfica del resultado obtenido con el Excel</i>	<i>39</i>

INTRODUCCIÓN

“Las tecnologías de redes inalámbricas han tenido un rápido desarrollo en los últimos años. Desde las comunicaciones punto a punto por infrarrojos a las redes inalámbricas de área personal de corto alcance (WPAN), pasando por multipuntos como Bluetooth, Wi-Fi para redes de área local, GPRS para telefonía móvil y otras como WIMAX, ANT, etc.”

La tendencia está siendo la de eliminar el máximo número de cables posibles a la hora de realizar una comunicación. El hecho de eliminar cables y recurrir a otro tipo de tecnologías tiene inconvenientes, por ejemplo; la longitud máxima que la tecnología nos proporciona o la calidad con la que se transmite la información. Eso quiere decir que habrá casos en los que no se podrá eliminar los cables.

Un ejemplo claro es el de los cascos inalámbricos, un experto en sonido siempre notará la diferencia entre unos que funcionen mediante cable y unos que utilicen la tecnología Bluetooth. La calidad con la que transmite el cable la información siempre será mayor.

En este trabajo de fin de grado se utiliza una tecnología llamada XBee que son dispositivos integrados capaces de comunicarse entre sí. Se calcula la temperatura en un punto y mediante los XBee se pasa el valor del voltaje de un punto a otro.

Como veremos más adelante, dichos dispositivos se comunican por radiofrecuencia y están basados en el estándar IEEE 802.15.4.

Para finalizar, se utiliza el software de LabView para la representación gráfica de los resultados obtenidos.

OBJETIVOS

- Desarrollar e implementar de una red inalámbrica de sensores ZigBee mediante el modulo XBee.
- Representar el comportamiento de un sensor RTD y comprobar su comportamiento lineal.
- Analizar los diferentes modos de funcionamiento de los XBee.
- Configurar el Arduino.
- Elaborar una plataforma en la que se trabajen el Arduino y el XBee en conjunto.
- Profundizar en aspectos teóricos sobre el funcionamiento de los XBee.

BENEFICIOS QUE APORTA EL TRABAJO

Mediante este sistema se tiene la opción de separar la parte en la que se recoge la temperatura, y la parte en la que se realizan los cálculos y se representan los resultados. El hecho de no necesitar cable que una las dos partes ofrece una gran variedad de posibilidades a la hora de organizar el proyecto.

Este sistema sin cables puede ser útil para aislar la parte peligrosa del proyecto/proceso. Lógicamente se tiene que tener en cuenta el alcance que tienen los dispositivos XBee y sus limitaciones (Se explicarán en apartados siguientes).

DESCRIPCIÓN DE REQUERIMIENTOS

- Utilizar un ordenador con el sistema operativo OS X 10.10 Yosemite.
- Tener instalado LabView y así poder visualizar los resultados.
- Utilizar la aplicación de X-CTU para la programación de los XBee.
- Hacer uso de SparkFun XBee Wireless Kit.
- Realizar este montaje teniendo en cuenta los límites de distancia de los XBee.

ANÁLISIS DE ALTERNATIVAS

Este proyecto es un primer paso que puede desembocar en distintos tipos de proyectos. La tecnología XBee utilizada puede ser investigada más profundamente y centrar el proyecto únicamente en este apartado. En caso de que se quieran realizar otros proyectos basándose en el contenido de éste, aquí se escriben unas cuantas ideas:

- En el circuito de acondicionamiento se puede utilizar un amplificador de aislamiento. Con ello se logrará aislar galvánicamente el circuito de control del circuito de potencia. Hay casos en los que utilizar la misma masa para ambos puntos puede provocar cortocircuitos. Un ejemplo de amplificador de aislamiento que se puede utilizar es el C784.
- Se pueden utilizar diferentes modos de comunicación de los XBee, en este proyecto solo se utiliza el “Modo transparente” y la conexión “Punto a punto”. Como aparece en el documento de “Cálculos Software” hay infinidad de modos de configuración que se pueden realizar con los módulos. Un ejemplo podría ser que una vez LabView haya recogido la información del XBee, mande una señal a otro XBee que active algún tipo de actuador. Con lo que se podría realizar un proyecto en el que, una vez alcanzada una temperatura establecida, se encienda un LED. Hay un ejemplo de un actuador en el libro de “Programming Arduino with LabView” al que se hace referencia en la bibliografía.
- En vez de calcular la temperatura se puede calcular una magnitud diferente. Por ejemplo, se puede hacer un voltímetro o un amperímetro.
- Otro proyecto sería centrarse únicamente en la parte de XBee e investigar todas las configuraciones que tiene. El programa de X-CTU ofrece tal variedad, que únicamente investigando las opciones que proporciona, sería un TFG.
- Otra cosa que se podría variar respecto a este TFG sería el de implementar el montaje en el software de Windows. El proceso de instalación y de configuración de software es diferente.

- Se puede utilizar el software de Matlab en vez de el de LabView, son softwares que tienen capacidades parecidas con lo que sería interesante ver como se integra XBee con Matlab (también tienen compatibilidad).
- Durante el proyecto se verá como el programa de XCTU es muy inestable con lo que sería interesante realizar este mismo proyecto utilizando el software de RealTerm.
- Por último, cambiar de tarjeta de adquisición de datos. En este caso se ha utilizado Arduino UNO. Hay infinidad de opciones en el mercado con las que sería interesante trabajar.

DESCRIPCIÓN GENERAL

DIAGRAMA DE BLOQUES GENERAL

En el siguiente diagrama de bloques y flujo de información se da una solución a los requerimientos del sistema de una manera general, tratando de buscar de alcanzar los objetivos del proyecto.



Figura 6. 1 Diagrama de bloques del sistema

Cabe destacar, que un XBee irá acoplado al “Convertidor AD controlador” donde se realizará la conversión de voltios a temperatura y el otro será quien envíe la información al ordenador.

A continuación, se realiza una breve explicación de las partes que componen el sistema:

- Sensor o transductor: Se hace uso de una RTD, concretamente se utiliza el modelo PT100, en la que una de las características más importantes, es que varía su resistencia en función de la temperatura.
- Acondicionamiento de la señal: El acondicionamiento de la señal está formado por un puente de Wheastone que tiene las salidas conectadas al amplificador LM358, del cual obtendremos la señal amplificada.

- Convertidor A/D: Para realizar la función de convertidor analógico-digital se utiliza el Arduino Uno, se hace la conversión de voltios a temperatura.

- Comunicación entre los XBee: Los XBee se comunican punto a punto mediante la configuración establecida en el programa XCTU.

- Presentación de los resultados: Se muestran los resultados de una forma gráfica en LabView.

A continuación, se procede a describir detalladamente cada uno de los bloques en lo referente a su utilidad práctica.

SENSOR O TRANSDUCTOR

Como ya se ha explicado se utiliza una RTD. Una RTD es un sensor de temperatura que varía su resistencia interna en función de la temperatura y que esté percibiendo. En este caso se utiliza el modelo PT100, en el que una de las características más importantes, es que están fabricadas con platino. Las PT100 tienen una resistencia eléctrica de 100Ω a una temperatura de 0°C . Para obtener una mejor transmisión de datos se utiliza una configuración a 3 hilos trenzados entre sí.

“El cable de par trenzado consiste en grupos de hilos de cobre entrelazados en pares en forma helicoidal. Esto se hace porque dos alambres paralelos constituyen una antena simple. Cuando se entrelazan los alambres helicoidalmente, las ondas se cancelan, por lo que la interferencia producida por los mismos es reducida lo que permite una mejor transmisión de datos.”



Figura 6. 2 PT100 con la configuración a 3 hilos

La ecuación que caracteriza al sensor es la siguiente:

$$R_T = R_0 (1 + \alpha T)$$
$$\alpha = 0,00385 (\Omega / \Omega) / ^{\circ}\text{C}$$

Donde,

- R_0 es la resistencia a la temperatura de referencia T_0 .
- T es la desviación de temperatura respecto a T_0 .
- α es el coeficiente de temperatura del conductor especificado a 0°C .

ACONDICIONAMIENTO DE LA SEÑAL

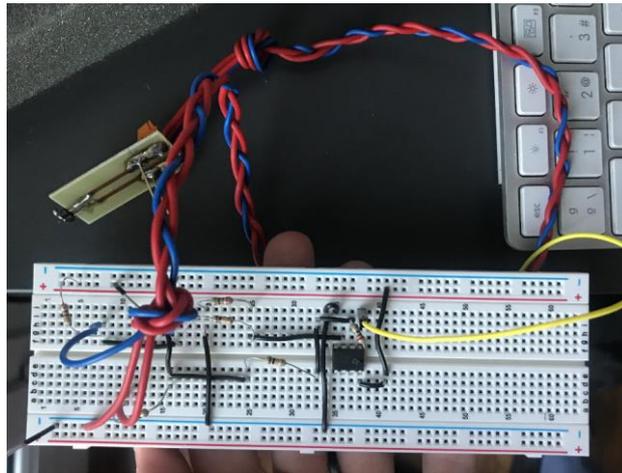


Figura 6. 3 Acondicionamiento de la señal (Montaje real)

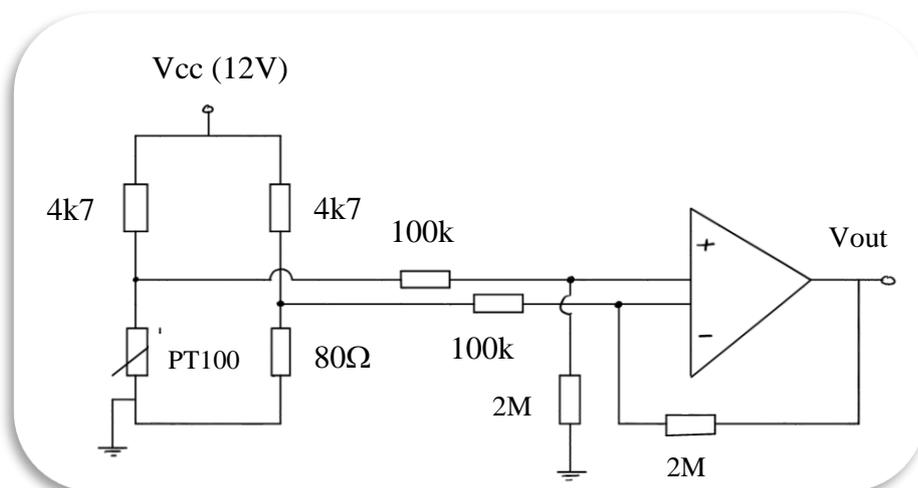


Figura 6. 4 Acondicionamiento de la señal (Esquema)

La señal de salida de un sensor de un sistema de medición se debe procesar adecuadamente para luego poder tratarla. En este caso, el sensor es una PT100 y como se puede observar en la imagen, se dispone de un puente de Wheastone y un amplificador LM358.

En este proyecto no se va a profundizar en aspectos teóricos del puente de Wheastone y el amplificador LM358. Simplemente saber que a través del puente de Wheastone se pueden detectar pequeñas variaciones en la salida de los transductores conectados a él.

A medida que la temperatura varíe cambiará la tensión a la salida del amplificador. La tensión de salida (V_{out}) será con la que se realizarán los cálculos posteriores.

Todos los cálculos realizados para este acondicionamiento de señal están en el documento de “Cálculos Software” ya que se han utilizado para la programación del Arduino.

CONVERTIDOR A/D

En este caso, se utiliza el Arduino UNO para realizar la conversión analógico-digital, que dispone de un entorno de desarrollo propio y está pensado para fomentar y facilitar el uso de la electrónica.

Al tratarse de un microcontrolador con un lenguaje basado en C / C++, existen funciones de entrada/salida analógica y digital, funciones aritméticas, manejo de interrupciones, comunicaciones por el puerto serie, etc.

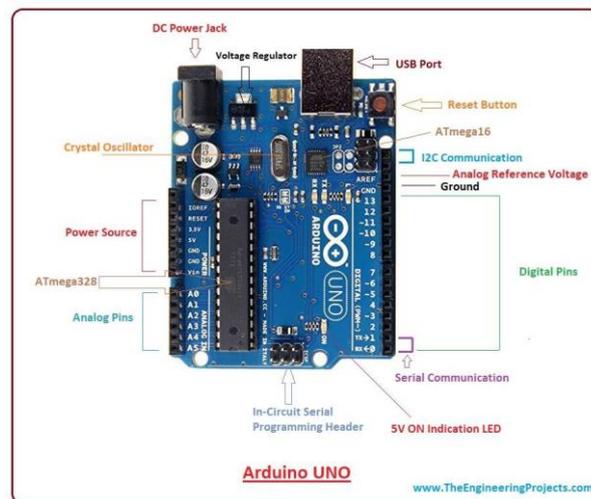


Figura 6. 5 Esquema del Arduino UNO

El microcontrolador se le acopla el primero de los XBee. Para ello se utiliza una placa *SparkFun XBee Shield*.

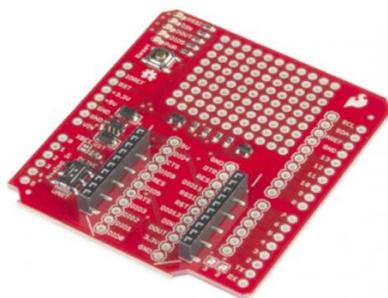


Figura 6. 6 SparkFun Xbee Shield

Dicha placa está creada para que encaje perfectamente con el Arduino, tal y como se puede ver en la siguiente Figura:



Figura 6. 7 El Arduino acoplado a la SparkFun

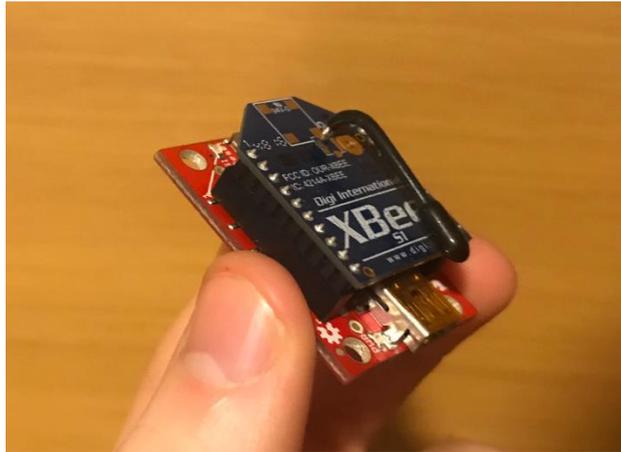


Figura 6. 8 El Arduino acoplado al XBee Explorer USB

Por otro lado, está el adaptador de XBee Explorer USB, será el que conecte con el ordenador.

Como es lógico hay que utilizar un código específico para que puedan trabajar conjuntamente el XBee y el Arduino. El código que se ha utilizado se explica en el documento de “*Cálculos Software*”.

ZIGBEE

Para entender lo que son los módulos XBee y su funcionamiento se debe explicar lo que es ZigBee. ZigBee es una tecnología creada para transferir datos de forma inalámbrica a un precio económico. Un conjunto de empresas llamado ZigBee Alliance se une con el objetivo de crear un estándar de comunicaciones; dichas comunicaciones implementarán tanto Wi-Fi como Bluetooth. Es por ello que en mayo de 2003 surge ZigBee.

El propósito de ZigBee es que sea usado en aplicaciones que requieren:

- Comunicación segura.
- Tasa de envío baja.
- Bajo consumo

Todos los módulos ZigBee operan en la banda ISM. Todas las bandas ISM están reservadas para uso no comercial de radiofrecuencia electromagnética en áreas como la industrial, científica o médica. Las bandas que se utilizan son:

- 868 MHz
- 915 MHz
- 2.4 GHz

Siendo la de 2.4 GHz la más extendida y utilizada.

También conviene saber que ZigBee se basa en el estándar IEE 802.15.4 es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos. En este trabajo de fin de grado no se profundiza en dicho estándar.

En cuanto a los diferentes tipos de dispositivos que define ZigBee están los siguientes:

- Coordinador: Su función es la de controlar la red y los dispositivos que van a realizar la comunicación.
- Router: Interconecta dispositivos separados diferentes mediante direccionamiento.
- Dispositivo final: Elemento pasivo que responde ante peticiones de otros dispositivos. Posee la funcionalidad necesaria para comunicarse con el coordinador o Router.

TOPOLIGÍAS ZIGBEE

Los módulos pueden trabajar de diferentes maneras conectarse de multitud de formas, en este punto se hace referencia a las diferentes topologías que tiene ZigBee.

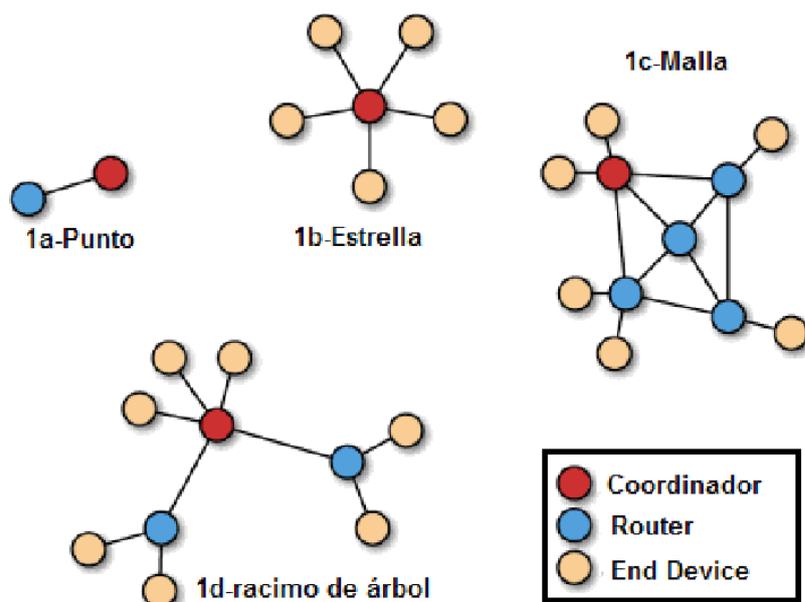


Figura 6. 9 Topologías ZigBee

- Estrella: Hay multitud de módulos. Todos ellos pueden realizar la comunicación con el dispositivo coordinador, pero no entre dispositivos que no lo son.
- Punto a punto: Es la topología utilizada en este proyecto. Existe un coordinador y un dispositivo final. Uno envía la información y el otro la recibe. Es lo más parecido a un cable.
- Malla: Es parecido a una comunicación punto a punto en el que cualquier dispositivo puede comunicarse con otro, pero hay un coordinador que se encarga de la gestión de la comunicación.
- Árbol: Hay un coordinador que es quien establece la red inicial. Después están los Router (transmiten el mensaje) que forman unas ramas con dispositivos finales a sus extremos llamadas hojas.

MODOS DE OPERACIÓN DE LOS MÓDULOS XBEE

Como se ha explicado en el punto de “Análisis de alternativas” los módulos pueden operar de diferentes maneras y tener diferentes configuraciones, tal y como se va a ver en este punto.

Los módulos XBee tienen diferentes modos de operación, dependiendo del uso que se le quiera dar y de la importancia de los datos que se quieran transmitir se utilizará un modo de operación u otro.

Este proyecto utiliza el modo de funcionamiento transparente y la conexión punto a punto:

Modo Transparente: Tiene como función una comunicación punto a punto, ya que no es necesario ningún tipo de control. Es la que traen los XBee por defecto. Este tipo de funcionamiento se puede utilizar para sustituir un cable serial, es la configuración más simple que hay y no requiere de una configuración compleja.

Por un lado, todo lo que entra por el pin de *Data In*, se guarda en el *buffer* de entrada para luego ser enviado. Por otro lado, todo lo que entra como paquete *RF*, se guarda en el *buffer* de salida y luego enviado por el pin de *Data Out*.

Este modo puede ser configurado para que envíe la información cada cierto tiempo (haya recibido información o no) o para que la envíe según reciba un carácter. En el momento en el que el XBee vaya a mandar la información, se recoge lo que se tenga en el *buffer* de entrada, se empaqueta (es lo mismo que integrar en un paquete *RF*) y después se transmite.

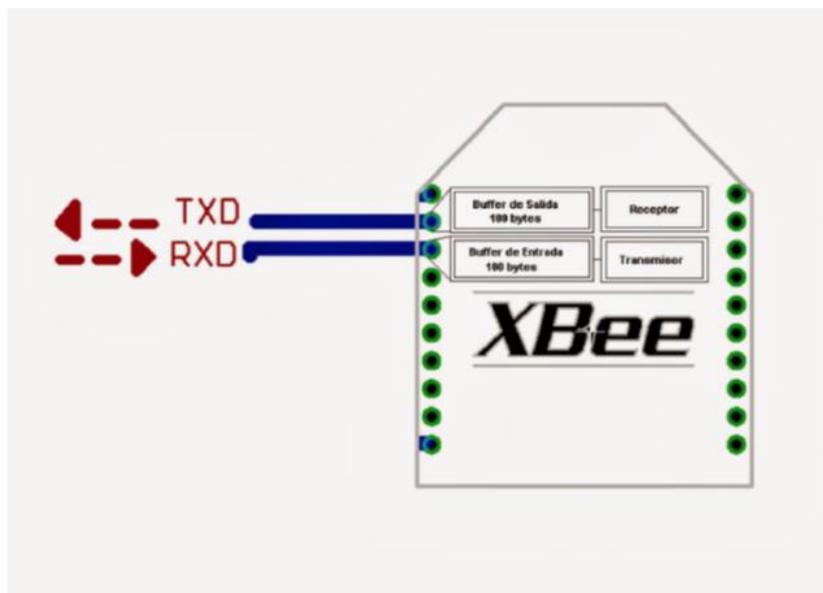


Figura 6. 10 XBee en modo transparente

Conexión Punto a Punto: No requiere de ningún tipo de configuración compleja con lo que es idóneo para sustituir a un cable. Se define la dirección del módulo destino y del módulo que va a enviar la información y se crea este tipo de conexión.

En cuanto al resto de tipos de conexiones y modos de operaciones que tienen los módulos XBee:

1. **Conexión API:** Este tipo de conexión es una conexión más robusta que la transparente. Los paquetes de datos se almacenan dentro de un *frame* (paquete de datos de longitud fija o variable) que tiene una estructura definida, lo cual permite determinar el origen de los paquetes recibidos dentro de la red.

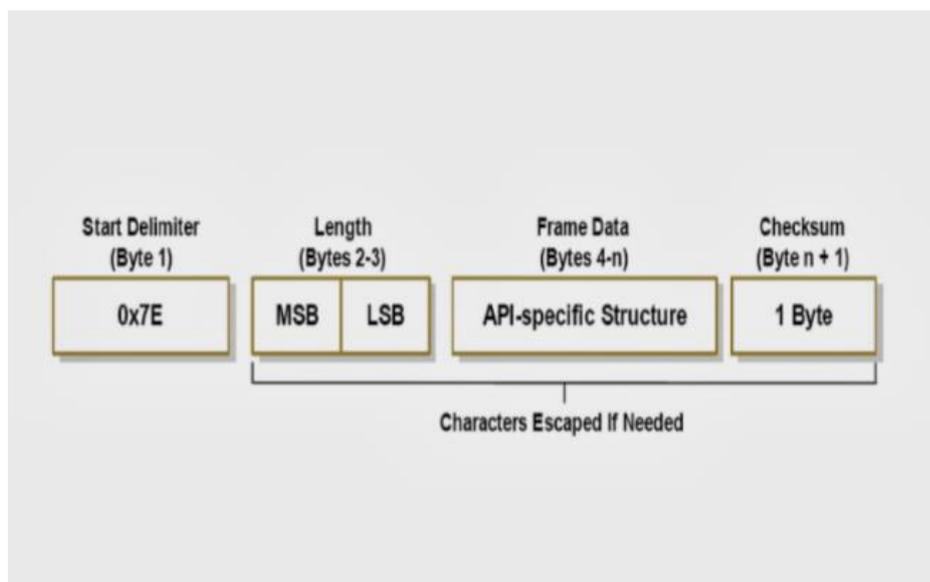


Figura 6. 11 Estructura de un frame en conexión API

Dentro de este modo de configuración tenemos 3 tipos de conexiones diferentes:

- AP = 0, se deshabilita el frame API, con lo que el XBee trabaja en el modo transparente.
- AP = 1, el XBee trabaja en modo API.
- AP = 2, el XBee trabaja en modo API, pero con una diferencia; trabaja con carácter de escape esto quiere decir que el paquete de datos es más grande, pero se evita que la cabecera del paquete de datos se confunda con los datos enviados.

2. Conexión NonBeacon con Coordinador: Este tipo de configuración se utiliza cuando se quiere lograr que la unidad central mande mensajes o reciba información de varios módulos. El coordinador es el que mantiene el control de los dispositivos terminales. Una red de datos RF consta de un coordinador y uno o varios dispositivos terminales, todos ellos forman un *Personal Area Network*. Todos los dispositivos pertenecientes a un PAN tienen un identificador que debe ser igual para todos los módulos dentro de un mismo PAN.

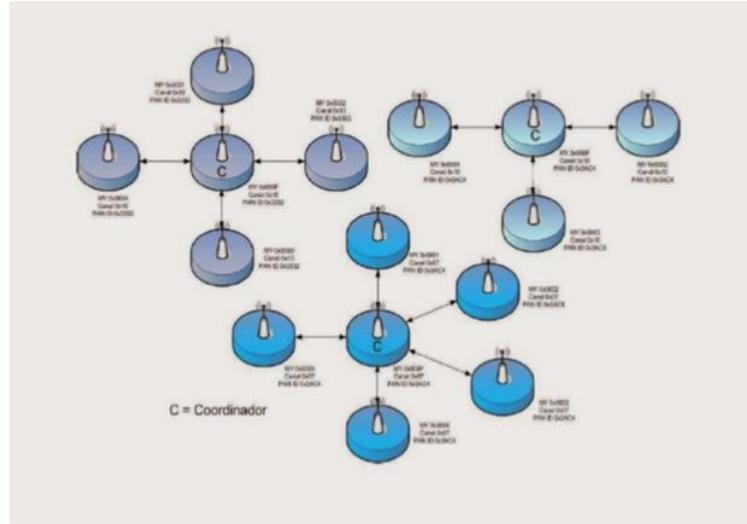


Figura 6. 12 Tres redes PAN diferentes

En la imagen anterior vemos tres redes PAN diferentes, se observa como no es necesario que coincidan el número de módulos. Dentro de cada PAN se aprecia un único coordinador, es el que controla la comunicación tal y como se ha explicado anteriormente.

3. Conexión NonBeacon Peer to Peer: En la conexión peer o peer se crea una conexión con cada uno de los módulos de la red que permite que se puedan comunicar entre ellos. Todos se comunican con todos. En las conexiones NonBeacon los nodos siempre están despiertos, de ese modo, los módulos se conectan al él y así poder entrar en modo de ahorro de energía (Sleep). Solo se despertarán cuando tengan que enviar datos.
4. Conexión Cable Virtual I/O: En este tipo de conexión se crea una comunicación de manera transparente entre los pines de cada módulo. Para crear dicha conexión, es suficiente con configurar los pines de I/O para estén asociados entre sí (vienen en pares). Es por ello, que solo se pueden crear cables virtuales entre pares de pines de distintos módulos.

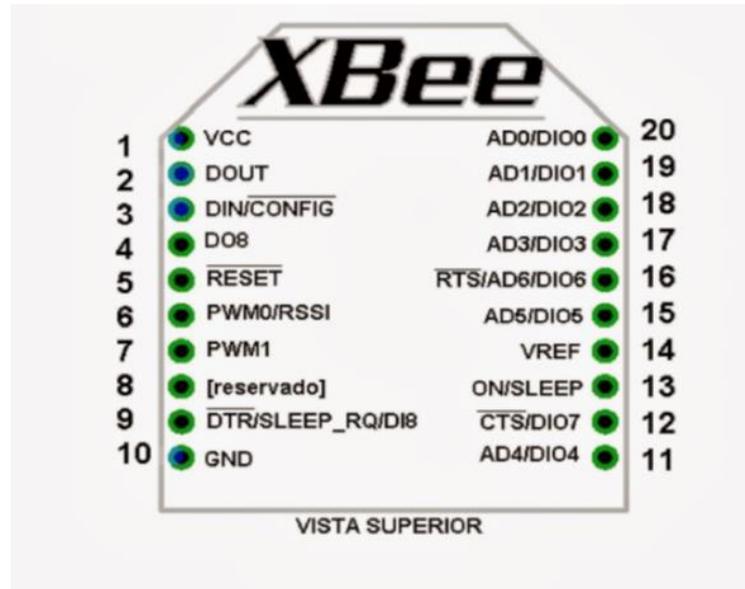


Figura 6. 13 Vista superior de los pines de un XBee

Como se ve en la figura, cada pin de entrada tiene su propio pin de salida. Esto permite que sea muy sencillo enviar y recibir información. Vienen configurados de par en par:

DIO5 = DI5 (entrada digital) y DO5 (salida digital)

5. Conexión Broadcast: En este otro tipo de conexión la información se envía desde un nodo a varios nodos dentro de una misma red. Todos los nodos reciben la misma información. Para su configuración hay que ajustar los módulos con la dirección de Broadcast. Siempre que alguno de los módulos reciba un paquete de datos que contenga una dirección de destino Broadcast, se aceptará. Un ejemplo de dirección Broadcast:

DL = 0x0000FFFF

DH = 0x00000000

Dicha dirección hay que configurarla en todos los nodos de la red.

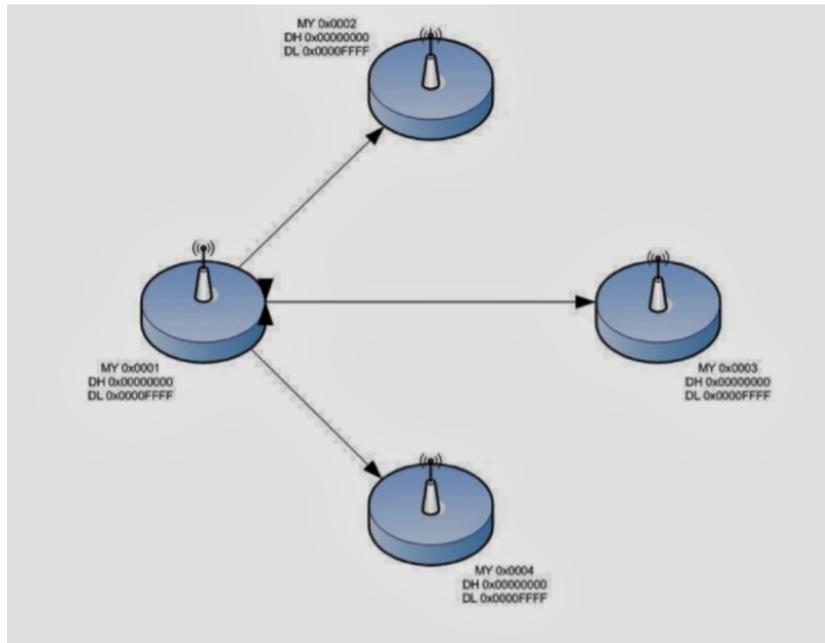


Figura 6. 14 Red de Broadcast

En esta imagen se observa como a la hora de enviar un dato por el módulo 0x0001 la información la recibirán los módulos restantes tal y como se ha explicado anteriormente.

6. Conexión Punto a Multipunto: A diferencia de la conexión Broadcast, este tipo de conexión es más seguro. Se necesitan las direcciones de los otros módulos, con lo que permite transmitir información desde una entrada serial de uno de los módulos a uno o varios módulos conectados a la misma red de una manera más controlada y segura. Según la norma IEEE 802.15.4. disponemos de 16 canales entre los que debe haber una diferencia de 5 MHz. Partiendo de una frecuencia base de 2.405 GHz y llegando a una frecuencia de 2.480 GHz tal y como veremos en la siguiente figura:

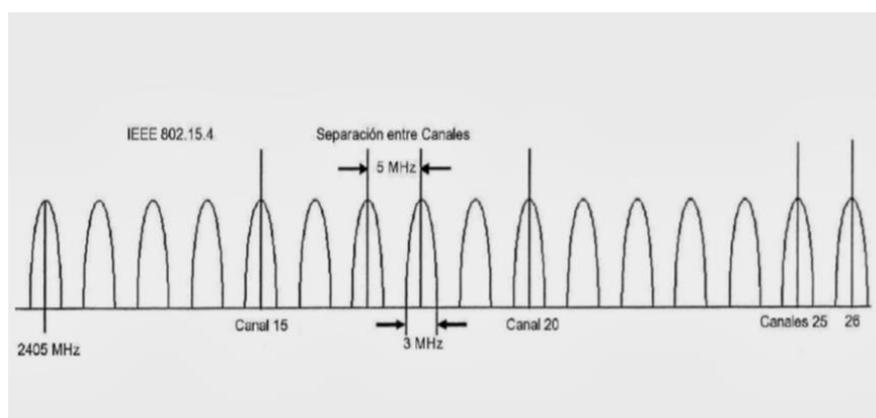


Figura 6. 15 Frecuencias de una conexión punto a multipunto

7. Modo IDLE: Siempre que el módulo no esté en ninguno de los modos de conexión anteriores estará en modo IDLE.
8. Modo Recibir/Transmitir: Siempre que al módulo le llegue o tenga que enviar un paquete estará en este modo. En modo directo la información se envía inmediatamente y en modo indirecto se envía cada un cierto tiempo estipulado.
9. Modo de Bajo Consumo: Cumpliendo dos condiciones el módulo entrará en este modo:
- Sleep_RQ (pin9) está en alto y el módulo está en pin Sleep mode (SM = 1,2 o5).
 - Una vez transcurrido un tiempo definido (ST) quiere decir que el módulo está en reposo.

10. Modo de Comando: Este tipo de modo permite ingresar comandos AT al módulo Xbee. AT es un lenguaje ideado para la programación de módems. Permite ajustar numerosos parámetros una vez ingresado este tipo de modo.

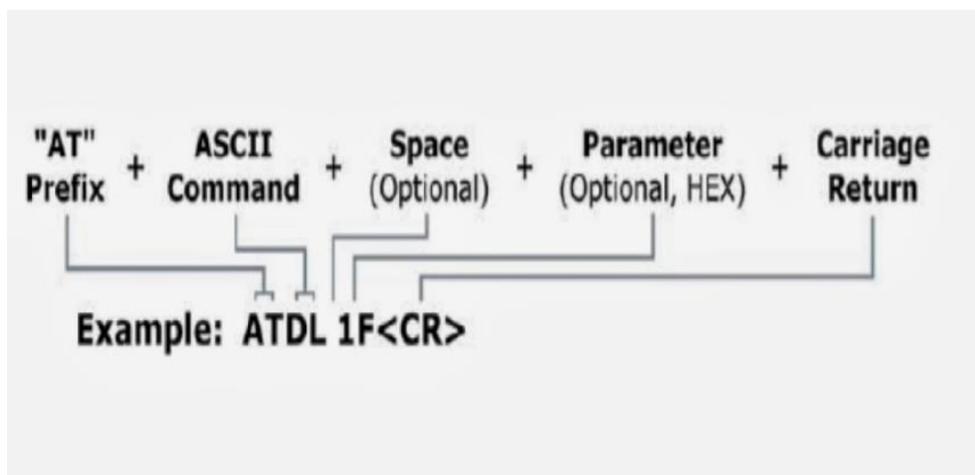


Figura 6. 16 Sintaxis de un comando AT

XCTU Y SU CONFIGURACIÓN

En XCTU se encuentran multitud de modos de configuración:

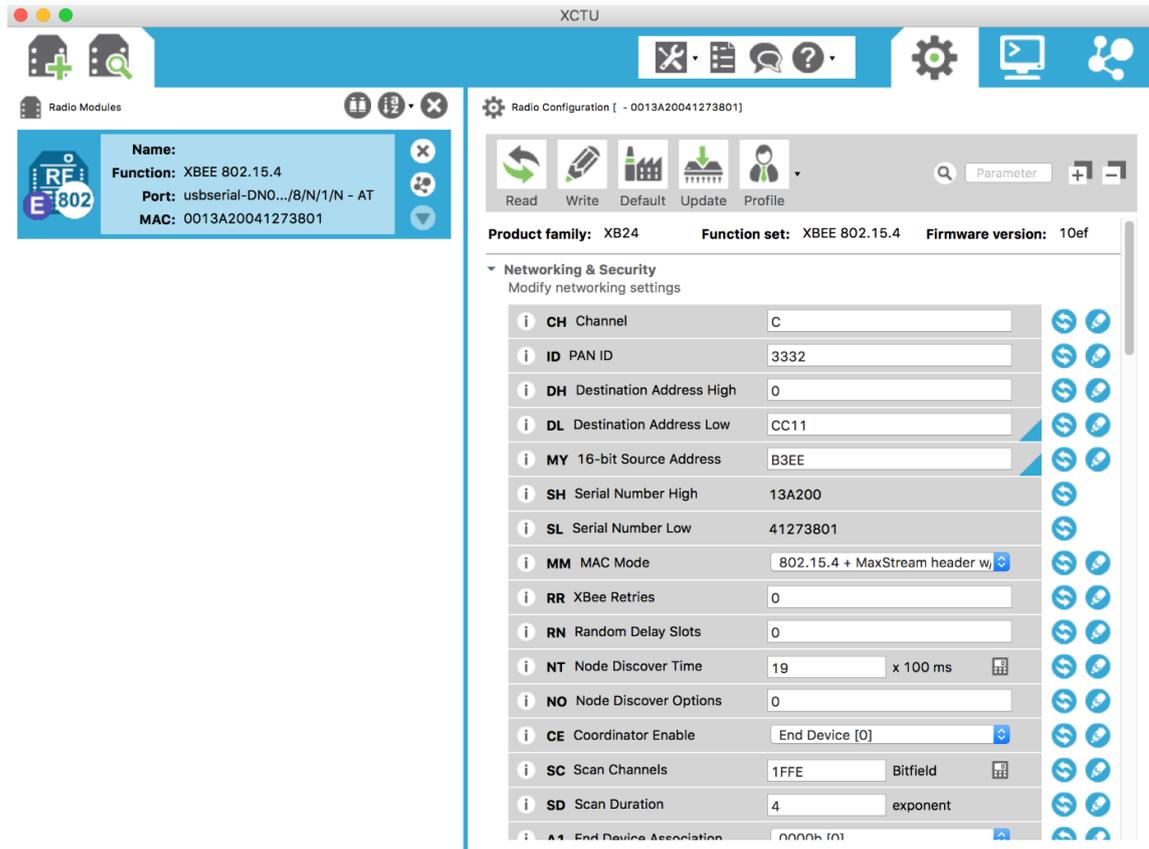


Figura 6. 17 Panel de configuración de XBee

En esta imagen se puede ver como hay muchos campos que se pueden modificar para lograr diferentes tipos de conexiones. En el documento de “Cálculos Software” se explica detalladamente como se realiza la conexión en el caso de este proyecto.

ESQUEMA DE LABVIEW

Para poder visualizar los resultados de una manera gráfica se configura un diagrama de bloques en LabView que tiene el siguiente aspecto:

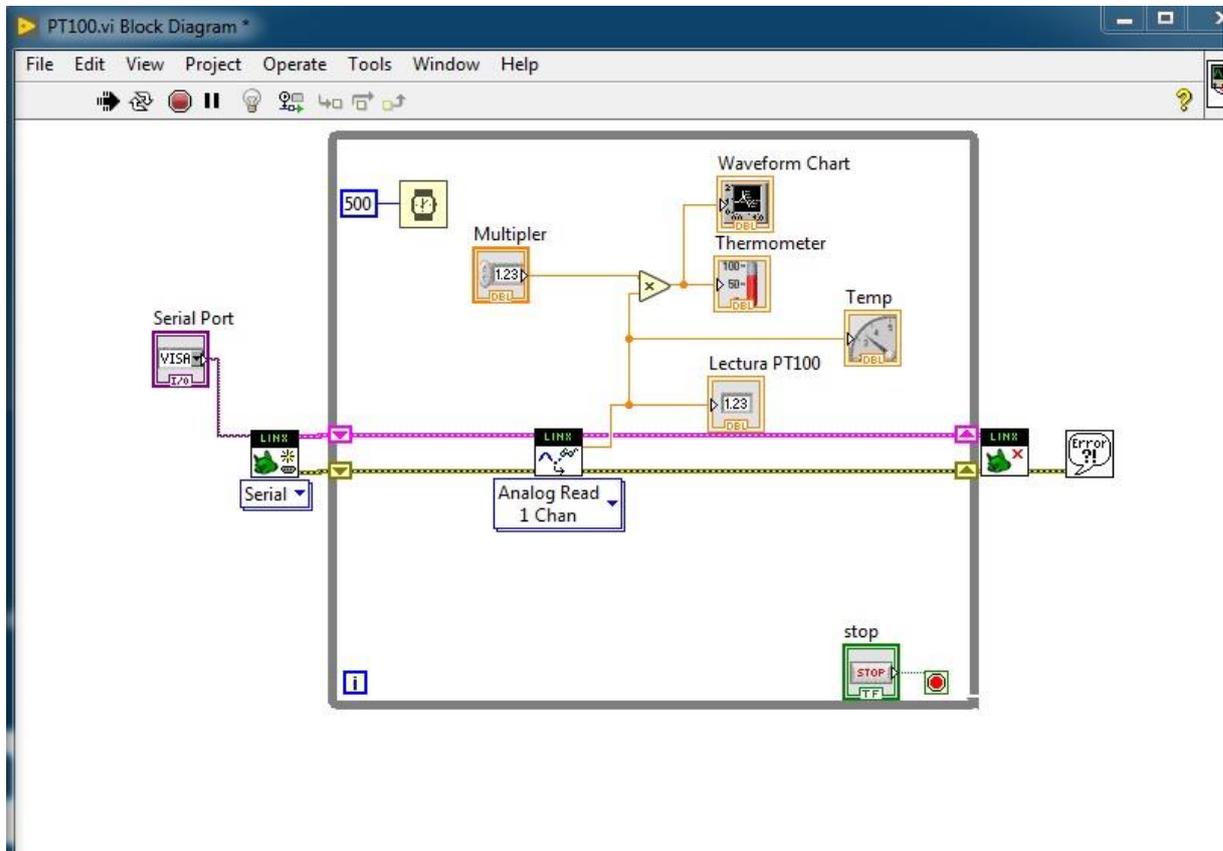


Figura 6. 18 Diagrama de bloques LabView

La explicación del diagrama de bloques y su configuración se hace en el documento de “Cálculos Software”. Ahí se explica detalladamente cual es la función de cada bloque del diagrama y que configuración se le ha dado para realizar la conexión con el XBee y a su vez Arduino.

PRESENTACIÓN

Los resultados se presentan en un Excel, en el que se ve claramente la linealidad de la RTD. También se puede ver en LabView en tiempo real como va variando la temperatura.

DIAGRAMA DE GANTT

El diagrama de Gantt es un sistema de coordenadas con dos ejes: en el eje vertical se ubican las tareas a realizar desde el inicio hasta el final del proyecto y en el eje horizontal se ponen los tiempos. Los beneficios de este tipo de diagramas son los siguientes:

- Se puede ver de una manera gráfica todos los pasos que tiene un proyecto.
- Ayuda a administrar proyectos y de esa manera reducir los problemas de programación.
- Se puede realizar con el programa de Excel de una manera rápida.

Nombre Actividad	Fecha inicio	Duración en días	Fecha fin
Recopilación de información	01-oct	31	01-nov
Comprar el material	03-nov	7	10-nov
Instalación de programas software y búsqueda de drivers	04-nov	24	28-nov
Montaje de acondicionamiento de señal	28-nov	4	02-dic
Conexión entre los dispositivos Xbee	02-dic	20	22-dic
Conexión entre Xbee y LabView	22-dic	5	27-dic
Visualización de los resultados en LabView	27-dic	3	30-dic
Realizar la memoria	30-dic	15	14-ene
Realizar los cálculos software	14-ene	7	21-ene
Realizar los presupuestos	21-ene	1	22-ene

Figura 7. 1 Organización del Diagrama de Gantt

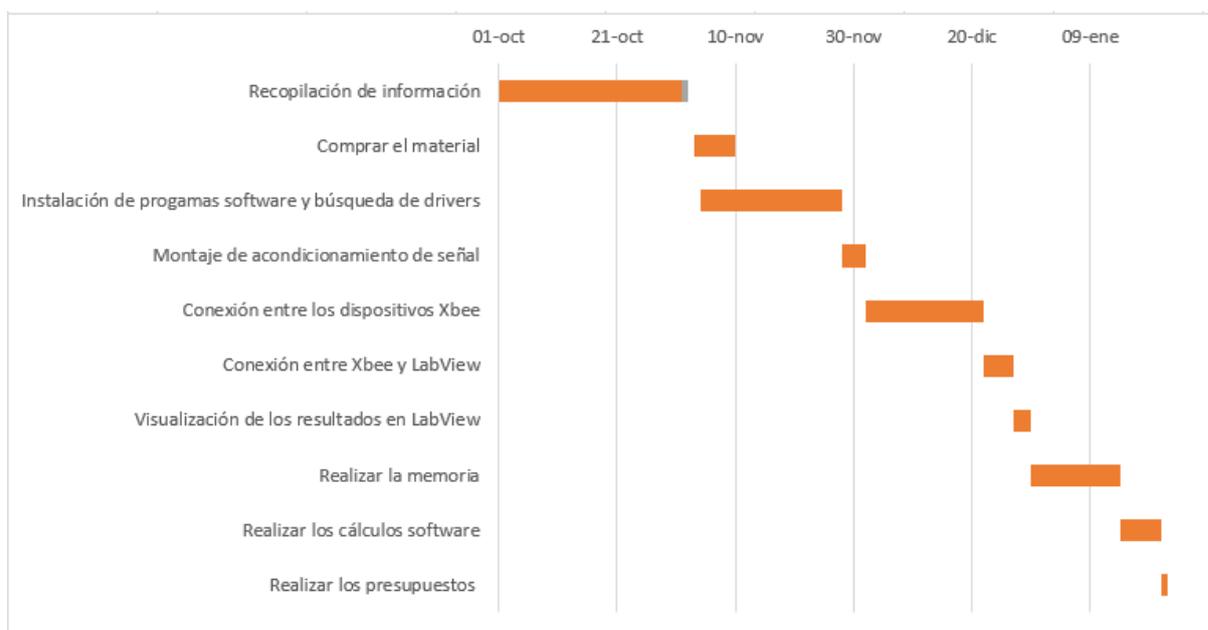


Figura 7. 2 Diagrama de Gantt

PROBLEMAS A LA HORA DE REALIZAR EL PROYECTO

Para poder implementar el proyecto en el sistema operativo de Apple se tuvo que seguir un procedimiento. El software de Arduino y el de XBee (X-CTU) si son compatibles con el Mac, pero para poder ejecutar la aplicación de LabView se tuvo que hacer uso de una aplicación llamada Wineskin Winery.

Dicha aplicación se utiliza para ejecutar programas de Windows en los Mac que tengan los siguientes sistemas operativos:

- OS X 10.6 Snow Leopard
- OS X 10.7 Lion
- OS X 10.8 Mountain Lion
- OS X 10.9 Mavericks
- OS X 10.10 Yosemite



Figura 8. 1 Aplicación de Wineskin

El proceso a seguir para poder ejecutar LabView no es sencillo, es un proceso largo y que requiere de paciencia. En la bibliografía se hace referencia al vídeo de YouTube utilizado para poder lograrlo. La decisión de utilizar Mac se tomó porque era bastante más estable a la hora de reconocer los XBee en el X-CTU. Sin ninguna razón aparente, había momentos en el que a la hora de seleccionar los puertos no se obtenía ningún resultado como podemos ver en la siguiente imagen:

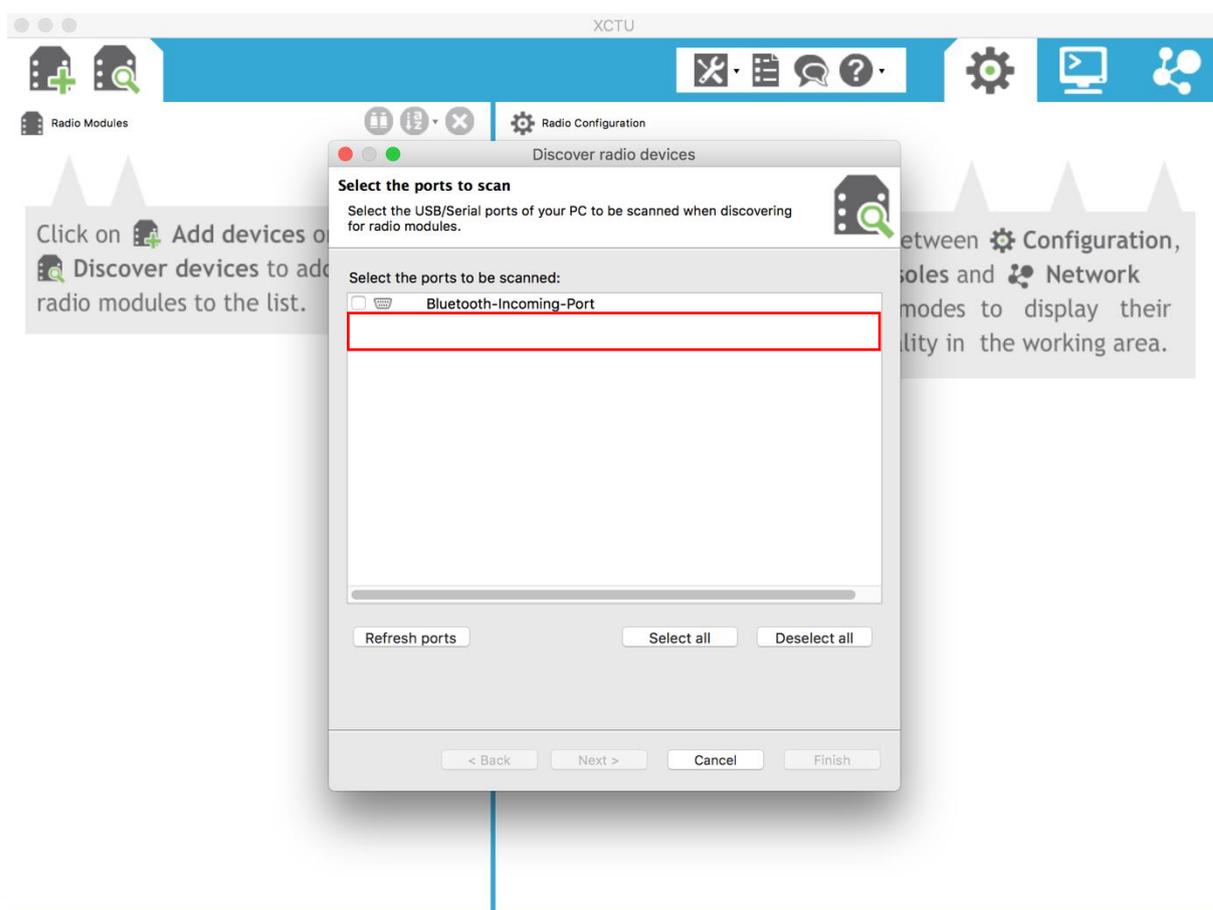


Figura 8. 2 X-CTU en el punto de reconocer los puertos

La foto se ha tomado en un Mac, pero el resultado es idéntico. En el cuadrado de rojo debería aparecer el puerto del XBee para luego poder pasar a configurarlo. Como se ha mencionado anteriormente, no hay ninguna razón aparente para que no aparezcan. De hecho, cambiando entre puertos llega un momento en el que se llega a realizar la conexión entre ordenador y dispositivo. Es mejor opción utilizar el programa de RealTerm. En el equipo de *Formula Student* de la Escuela de Ingeniería de Bilbao, no fueron capaces de utilizar el software de XCTU de una manera estable, con lo que se vieron obligados a pasar al de RealTerm.

Otro de los problemas que ha aparecido ha sido a la hora de realizar la conexión entre los XBee, había momentos en los que no se conectaban entre sí.

RESULTADOS

Uno de los objetivos de este TFG era comprobar la linealidad de la RTD. Para ir viendo en tiempo real la temperatura que se está calculando se utiliza el programa de LabView. En el documento de “Cálculos Software” se explica detalladamente como se configura el diagrama de bloques para obtener el siguiente resultado:

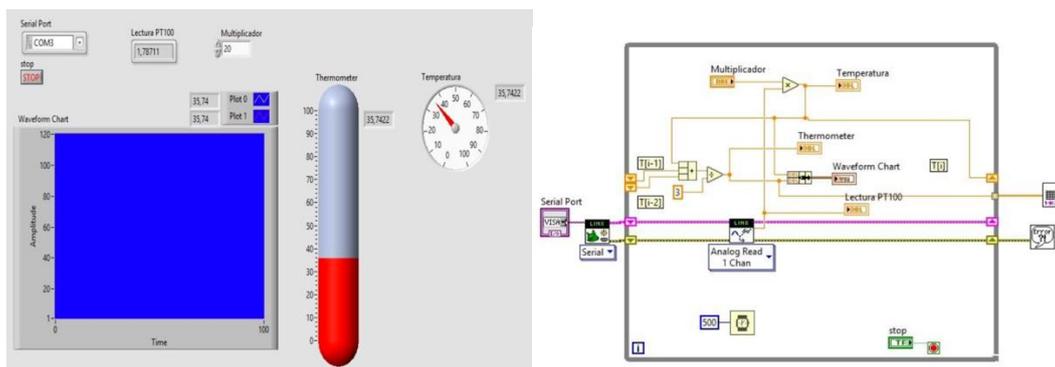


Figura 9. 1 X-CTU en el punto de reconocer los puertos

Pero para poder analizar todos los resultados obtenidos se creó un Excel en el que se obtiene un gráfico donde se ve claramente que el sensor se comporta de manera lineal:

Temperatura real	Voltaje medido	RTD calculada	Temperatura calculada
36	0,853	113,30	34,56
67	1,19	126,59	69,07
82	1,341	132,57	84,60
120	1,727	147,92	124,47
174	2,2	166,87	173,68
179	2,3	170,89	184,14
225	2.765	189.70	232.98

Figura 9. 2 Tabla de los resultados en Excel

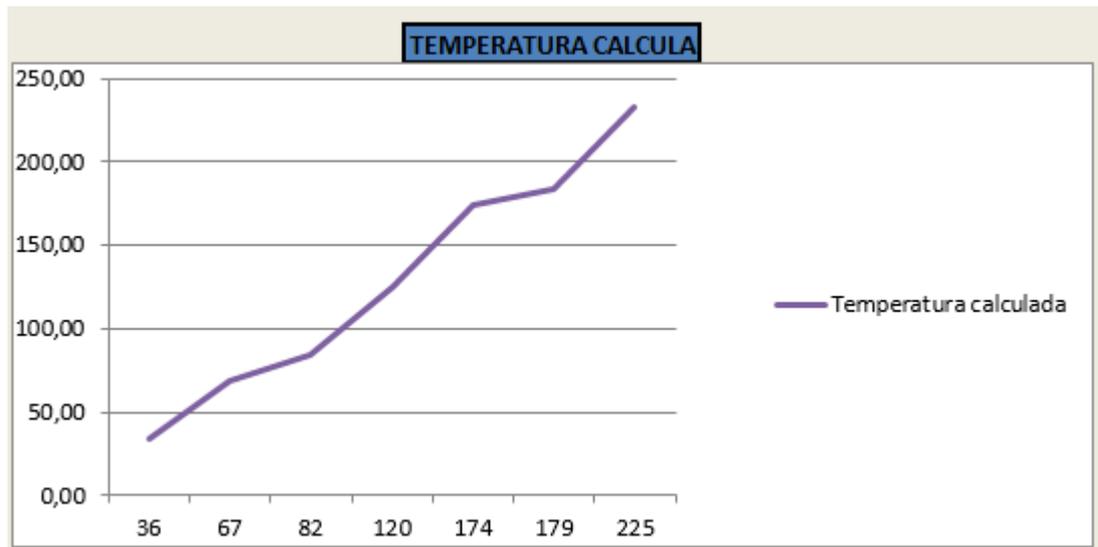


Figura 9. 3Gráfica del resultado obtenido con el Excel

Con esta gráfica se puede ver claramente como la PT100 se comporta de una manera lineal.

BIBLIOGRAFÍA

- [1] <https://learn.sparkfun.com/tutorials/xbee-shield-hookup-guide/all>
- [2] Programming Arduino with LabVIEW
- [3] <https://www.youtube.com/watch?v=vP7L3A8Lt8M>
- [4] <https://www.arduino.cc/>
- [9] <https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>