

COMUNICACIÓN XBEE DE TIPO BIDIRECCIONAL PARA SISTEMAS EMBEBIDOS.

Didier Vera

e-mail: dvera@unicomfacauc.edu.co

James Plaza

e-mail: jamesplaza@unicomfacauc.edu.co

Resumen El presente artículo muestra el desarrollo y las pruebas de comunicación realizadas para controlar un motor tipo Brushless de forma remota, incluyendo la medición de temperatura a partir de sensórica y contador de revoluciones para el motor en forma digital, para lo cual se hace uso de una comunicación inalámbrica full dúplex o en doble sentido mediante tecnología de tipo XBee con protocolo de comunicaciones Zigbee.

Palabras Clave: Comunicación Full Dúplex, Tecnología XBee; sistemas embebidos

Abstract: This Article presents the development and communication tests carried out to control a brushless motor remotely, including temperature measurement from sensor technology and rev counter for the engine in digital form, for which use of a done full duplex wireless communication or two-way by technology type with XBee Zigbee communication protocol.

Keywords: Full Duplex communication; XBee Technology; embedded systems

INTRODUCCIÓN

Los módulos XBee y XBee-PRO son módulos de radiofrecuencia que trabajan en la banda de 2.4 GHz con protocolo de comunicación ZigBee basada en estándares IEEE, fabricados por Maxstream, a diferencia del Bluetooth, este protocolo no utiliza FHSS (Frequency hopping), si no que realiza las comunicaciones a través de una única frecuencia, es decir un canal, su funcionamiento es basado en 16 canales situados en el rango de frecuencias de 2.4GHz por lo que se puede escoger un canal entre 16 posibles y con un ancho de banda para cada uno de 5MHz. El protocolo CSMA/CA se utiliza para evitar colisiones durante la transmisión, una transmisión basada en la norma IEEE802.15.4, de redes inalámbricas de áreas personales (wireless personal area network, WPAN) este conjunto de protocolos permite minimizar el tiempo de actividad y así lograr un bajo consumo, la línea XBee se puede encontrar en diversas aplicaciones industriales y comerciales, generalmente utilizado en la domótica, control y manipulación de prototipos, control de equipos y automatización. [1] [2]

Las posibilidades con los módulos XBee son gigantes, pues permiten la sinergia de las operaciones en cuanto a transmisión y recepción de datos se refiere, cuando los módulos son usados para interconectar sensórica en forma remota se hace necesario que el consumo de

corriente sea bajo, y más aún cuando se usan mediante baterías, la comprensión de cómo funcionan estos módulos es “bastante simple” sin embargo se deben tener en consideración algunos aspectos que para algunos no resultan no ser tan obvios, como por ejemplo; los tipos de comunicación que se pueden realizar y la diferencia entre ellos.

Aunque los módulos XBEE tienen homologación ZigBee y son compatibles con casi con cualquier red de datos dentro del rango de frecuencia en el que trabajan, el presente artículo está enfocado principalmente en el uso de los mismos para sistemas embebidos, y más específicamente la plataforma de desarrollo ARDUINO, logrando establecer un puente entre dos de estos sistemas haciendo uso de los módulos ya mencionados.

MATERIALES Y MÉTODO

El presente sistema de comunicación será de tipo bidireccional en donde se incluye un sistema embebido Arduino para el envío, adquisición, y trasmisión (Tx) de datos hacia otro sistema embebido al cual se le denominara su homólogo, la interfaz de usuario para la verificación de las señales o datos se realiza mediante el Monitor Serial de Arduino, puesto que la finalidad no es la interfaz si no el principio de funcionamiento y la programación XBee con el sistema embebido, las señales podrán ser visualizadas en tiempo real.

MATERIALES:

Hardware: Los componentes a utilizar se mencionan a continuación:

- 2 Módulos de comunicación inalámbrica XBee; módulo XBee 2,4 GHz, potencia de salida 60 mW (+18 dBm), opera con 3.3V 215 mA, 1 milla (1.500 metros) Rango, cifrado de 128 bits.
- 2 Shield de XBee para Arduino (opcional, creados para reducir las conexiones cableadas, incluye regulación de voltaje a 3.3volt).
- 2 Fuentes de voltaje; 1 Batería Lipo de 3 celdas, 11.1 volts, la otra fuente de voltaje la proporciona la conexión USB con el PC, 5 volts.
- 1 Motor Brushless; NTM Prop Drive Series 2836 2200 rpm/v, Max current: 50A Max Power: 310W, 11.1V (3S) / 696W, 15V (4S), ESC: 40~60A, Cell count: 3s~4s Lipoly.
- 1 Servomotor; Volt de operación entre 4.8~6 Volts, torque entre 2.7 Kg-cm.
- 1 Sensor de Temperatura LM35; Calibrado directamente en grados Celsius, La tensión de salida es proporcional a la temperatura, Tiene una precisión garantizada de 0.5 °C a 25 °C, Baja corriente de alimentación (60 µA), Bajo coste, y facilidad de adquisición.
- 1 XBee Explorer Dongle; Conexión directa por puerto USB, sirve de puerta de enlace (Gateway) entre el ordenador y el Xbee.
- 1 Protoboard; tablero con orificios conectados eléctricamente entre sí, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares.
- 2 Joystick PSP; Tres ejes (X, Y, Z + (botón)).
- 1 Controlador de Motor Brushless, HobbyKing 60A ESC 4A UBEC; corriente constante 60amp, corriente pico Max 80amp.
- 2 Sistemas embebidos **Arduino uno**; procesador Atmega328, volt de operación 5volt,14 pines digitales, 6 pines pwm, corriente continua por pin 40ma, memoria flash de 32KB, SRAM 2KB, EEPROM 1KB, 6 entradas analógicas, frecuencia de reloj 16Mhz, **Arduino Mega**; procesador Atmega2560, volt de operación 5volt, 54 pines digitales,12 pines pwm, frecuencia de reloj 16Mhz, 16 entradas analógicas, corriente continua por pin 40ma,memoria flash de 256KB, SRAM 8KB, EEPROM 4KB,

En la **Figura. 1** se presenta el diagrama en bloques de la transmisión, recepción y visualización de las señales. La **Figura. 2** representa el homólogo en donde ejecuta

procesos de control y adquisición de señales (temperatura, rpm), aunque es una comunicación full Dúplex se entenderá que uno de los diagramas representará al transmisor y el otro será la representación del receptor.

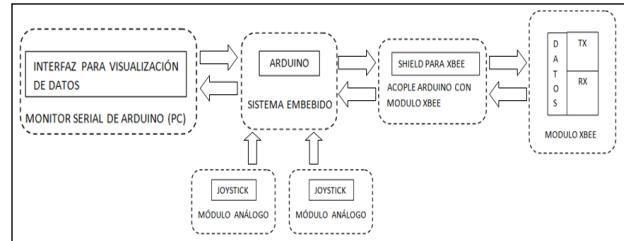


Figura. 1 Diagrama de bloques sistema Tx

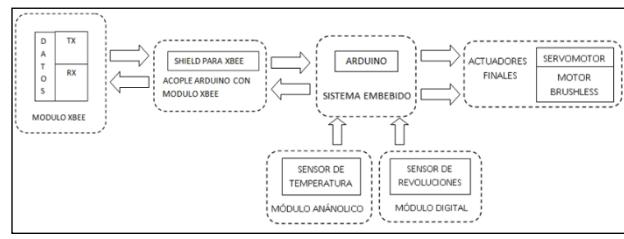


Figura. 2 Diagrama de bloques sistema Rx

Software: Programas necesarios para la configuración y programación de los módulos XBee y los Sistemas Embebidos.

ARDUINO: Software de distribución libre en donde se ofrece una amplia variedad de código abierto, el código fuente para el entorno Java es liberado bajo la GPL y la C/C++, las bibliotecas y microcontroladores están bajo la LGPL.

X-CTU: Software necesario para configurar los módulos Xbee, desarrollados solo para plataformas Windows.

MÉTODO

Módulos de comunicación: De cada tipo de módulo XBee existen dos versiones (PRO y Regular) que se diferencian en la potencia de transmisión, los módulos PRO (Serie 1 y 2) tienen mayor capacidad de alcance, permitiendo incluso doblar la distancia de transmisión, sin embargo poseen un mayor consumo de corriente. [3] Independientemente de su serie, cada módulo XBee permite emplear diferentes tipos de antenas para transmitir y recibir las señales, de los cuales se destacan los mostrados en la **Figura. 3**

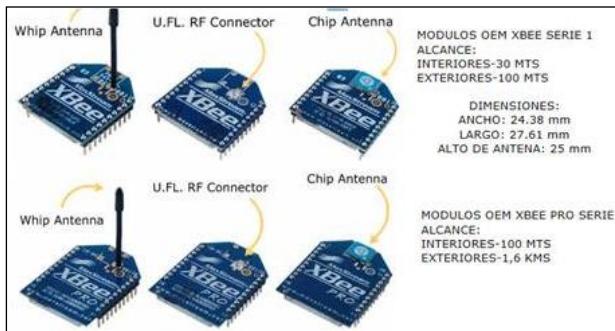


Figura. 3 Tipos de módulos Xbee

Para el desarrollo del proyecto se utilizan módulos XBee Pro Serie 1 con antena wire (dipolo de cable flexible) este módulo se destaca por su amplio alcance de Tx de hasta 1.5 km y una tasa de transferencia de 256Kbps, los valores de voltaje permitidos para su alimentación se encuentran en el rango de 2.8V a 3.4V, el ideal es de 3.3 voltios.

Configuración del módulo XBee: Existen dos modos de configuración inalámbrica para los módulos XBee, el Modo Transparente y modo API, dependiendo de su aplicación se selecciona el más conveniente. [3] A continuación en la **Tabla 1** se muestran los modos de comunicación y sus principales diferencias.

Tabla 1 Modos de comunicación XBee

Modos de comunicación	Sentido de la Comunicación	Tipo de comando	Tipo de Configuración	Comunicación serial
Modo Transparente	Unidireccional	Indiferente	Punto a Punto / Multipunto	Si
Modo API	- Bidireccional	API		Si

Teniendo en cuenta que el objetivo del proyecto es usar una comunicación bidireccional con dos sistemas embebidos se opta por utilizar el modo de comunicación Transparente, ya que su finalidad será enviar y recibir una estructura de bytes previamente programadas desde estos dos sistemas. El modo de comunicación API se diferencia por poder programar directamente líneas de código dentro de sus módulos, con lo cual no es necesario la adición de sistemas adicionales para la obtención de los datos, sin embargo su principal limitación radica en la cantidad de puertos de I/O que poseen además de su escasa memoria de procesamiento.

Configuración de Módulos: Para la configuración de los módulos será necesario un programador que permita grabar la información dentro de los mismos para lo cual se hace uso de un programador llamado Explorer Dongle, a la derecha de la **Figura. 4** se puede visualizar este programador.



Figura. 4 Módulos Xbee y Explorer Dongle

El primer paso de configuración es establecer qué tipo de conexión va a ser utilizado, como se mencionó anteriormente será Modo Transparente punto a punto, para esto se precisa en utilizar el software *XCTU* provisto por la empresa *DIGI*.

Antes de iniciar con el proceso se deben tener en cuenta ciertos parámetros necesarios antes de su aplicación, a continuación en la **Tabla 2** se muestran las especificaciones de estos parámetros. [5]

Tabla 2 Parámetros de configuración.

Parámetros	
Puertos COM	Será el puerto que establecerá la comunicación con el Explorer dongle
Baud Rate	Es la velocidad del puerto para Transmitir y recibir la información, ambos módulos deben ser configurados con parámetros similares
Modo Transparente	Para Habilitar en modo transparente lo único que se necesita es que ambos módulos se encuentren conectados a la misma red
PAN ID	Es el identificador de la red, puede ser cualquier número iniciando desde 0 hasta 9999, Los dos módulos deberán contener la misma información
DH	Será la primera dirección de Hardware de cada módulo XBee, dirección de destino, Ver Figura. 5
DL	Será la segunda dirección de Hardware de cada módulo XBee, dirección 2 de destino, Ver Figura. 5
MY	Es la dirección propia del módulo
CH	Es el canal de comunicación de los módulos, 16 canales a disposición, inicia con la letra C indicando que es el canal cero



Figura. 5 Identificadores de los módulos XBee DH y DL

Sabiendo el tipo de parámetros necesarios para establecer una comunicación exitosa en modo transparente, se proceden a seleccionar los valores tal y como se muestra en la **Tabla 3**.

Tabla 3 Parámetros de configuración para los módulos XBee.

Parámetros de Configuración		
	Transmisor	Receptor
Baud Rate	9600	9600
PAN ID	1990	1990
DH	13A200	13A200
DL	40C2EAD4	40C2EC4D
MY	0	0
CH	C	C

Con los valores elegidos se procede a conectar y configurar los módulos XBee mediante el software XCTU-Digi, el primer paso en la configuración es establecer los Baudios o velocidad de transmisión en el cual trabajarán los módulos. Ver la **Figura. 6**

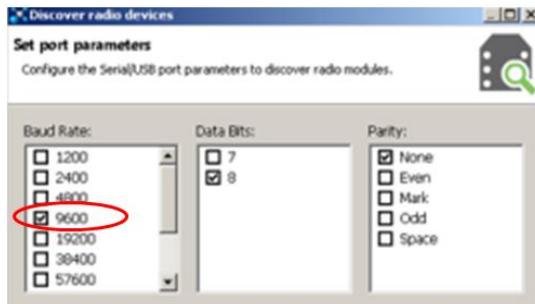


Figura. 6 Configuración de Baudios

Paso seguido se ingresan los datos previamente establecidos en la **Tabla 3**, y se proceden a grabar los parámetros en los módulos, así como se ilustra en la **Figura. 7**

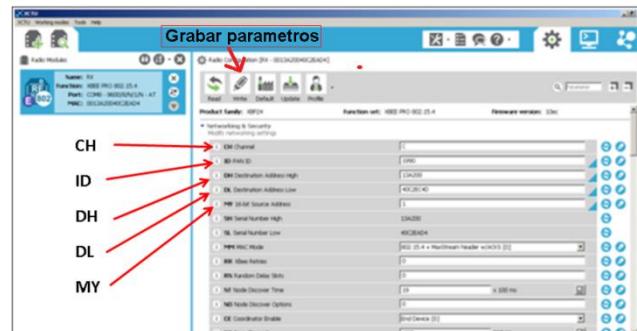


Figura. 7 Configuración CH, PAN ID, DH, DL, MY

Una vez finalizada la configuración de ambos módulos se procede a realizar dos diagramas de conexión básica de sus componentes a fin de verificar el correcto acoplamiento inalámbrico de sus sistemas y módulos, en la **Figura. 8** y **Figura. 9** se pueden observar los diagramas utilizados, después de verificados y ensamblados físicamente sus conexiones se realiza un código de programación básico para llevar a cabo las pruebas preliminares.

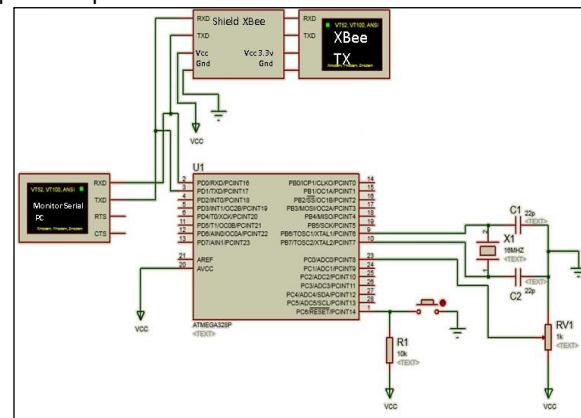


Figura. 8 Diagrama de conexiones TX y RX

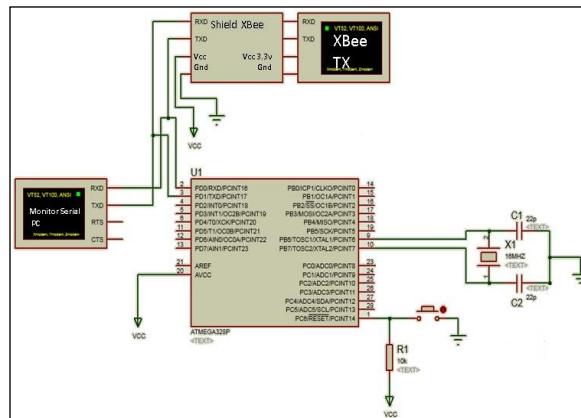


Figura. 9 Diagrama de conexiones TX y RX

Código de programación: El código de programación será necesario para saber si los módulos XBee fueron bien configurados y tienen comunicación con los dos sistemas embebidos. A continuación se muestran los códigos propuestos.

Código del TX

```
int potenciometro=A0;  
void setup()  
{ Serial.begin(9600);}  
void loop() { int datos = analogRead(potenciometro);  
delay(5);  
Serial.println("Lectura del POT: ");  
Serial.println(datos); //Datos enviados}
```

Código del RX

```
void setup()
{ Serial.begin(9600); }
```

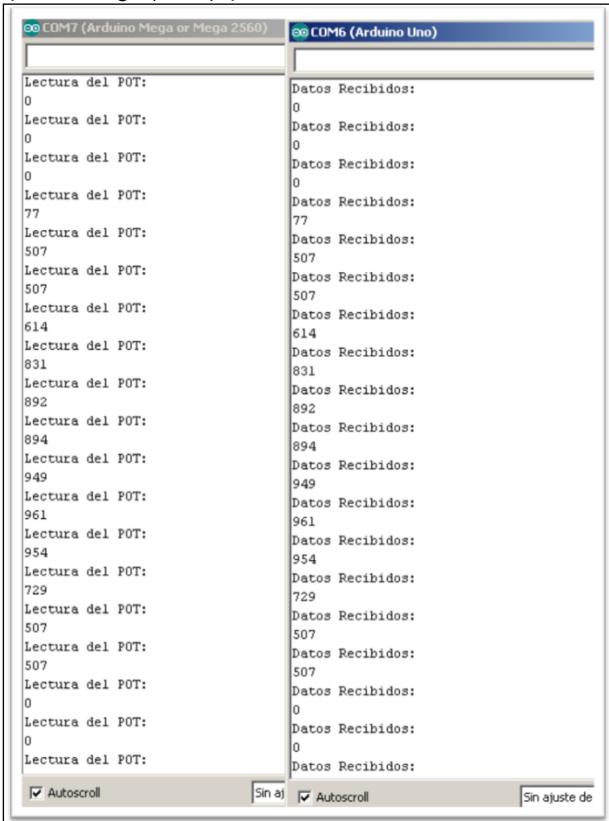


Figura. 10 Pruebas de comunicación, envío y recepción de datos

```
void loop() { if (Serial.available() > 0 ){  
    int Entrada= Serial.parseInt();  
    Serial.println("Datos Recibidos: ");  
    Serial.println(Entrada); } }
```

Finalmente se realizan dichas pruebas, en la **Figura. 10** se pueden ver los resultados obtenidos, cabe recalcar que esta prueba se realiza solo con el fin de verificar la comunicación en ambos sistemas por lo cual solo es una comunicación punto a punto en un solo sentido.

Después de verificar que la comunicación entre los dos sistemas es adecuada, se procede a realizar la conexión de los componentes faltantes, y establecer finalmente la comunicación en doble sentido o Full Duplex. A continuación en la **Figura. 11** y **Figura. 12** se pueden detallar los diagramas finales.

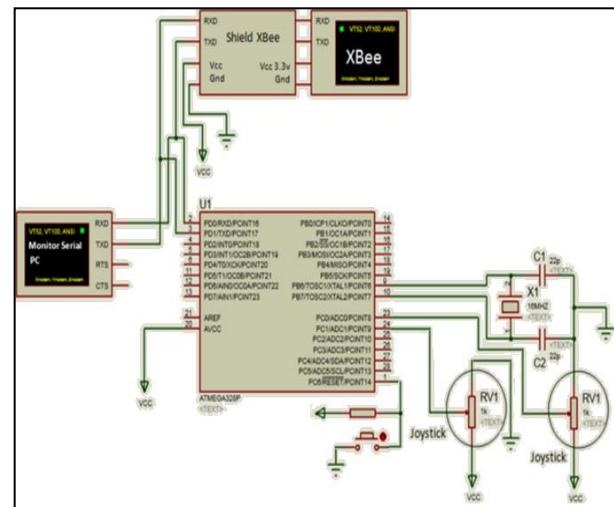


Figura. 11 Diagrama de conexiones (Tx)

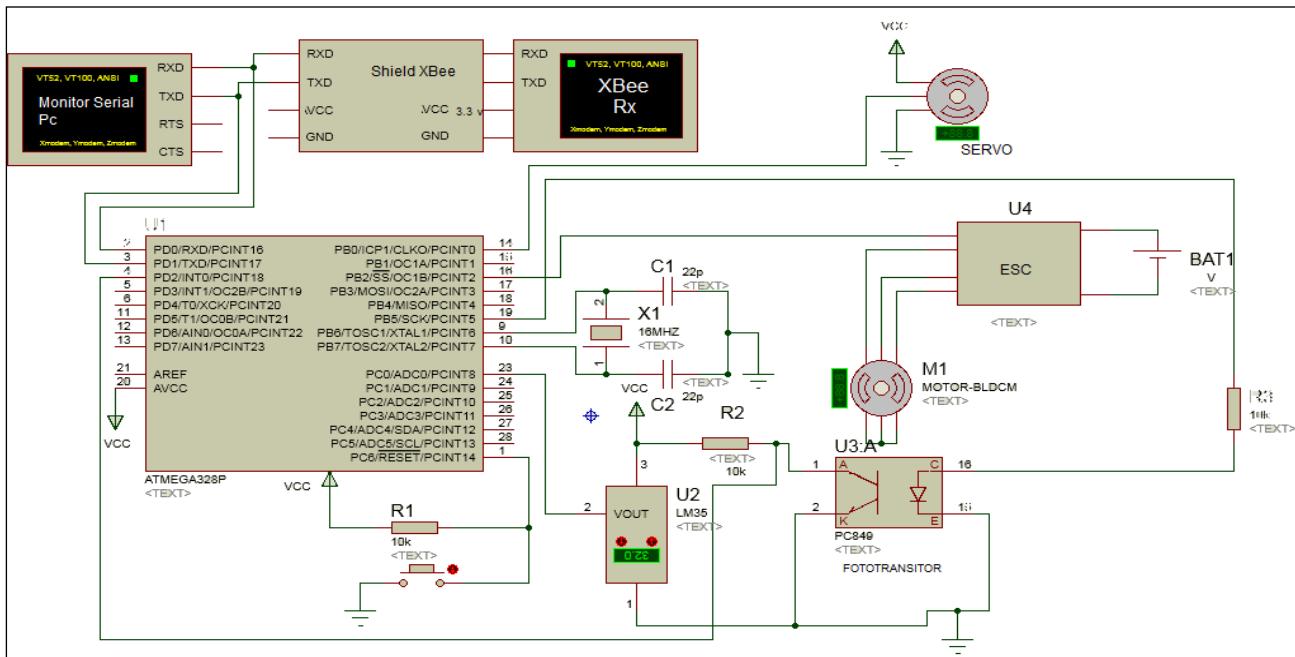


Figura. 12 Diagrama de conexiones (Rx)

Debido a la complejidad que se necesita para lograr una comunicación en doble sentido de manera casi simultánea se hace necesario la elaboración de un diagrama de flujo que permita determinar una lógica de programación acorde con su finalidad, A continuación en la Figura. 13 y Figura. 14 se muestran los dos principales diagramas de flujo, mientras que en la Figura. 15 y Figura. 16 se muestran las interrupciones que necesita el receptor para una correcta adquisición de su información.

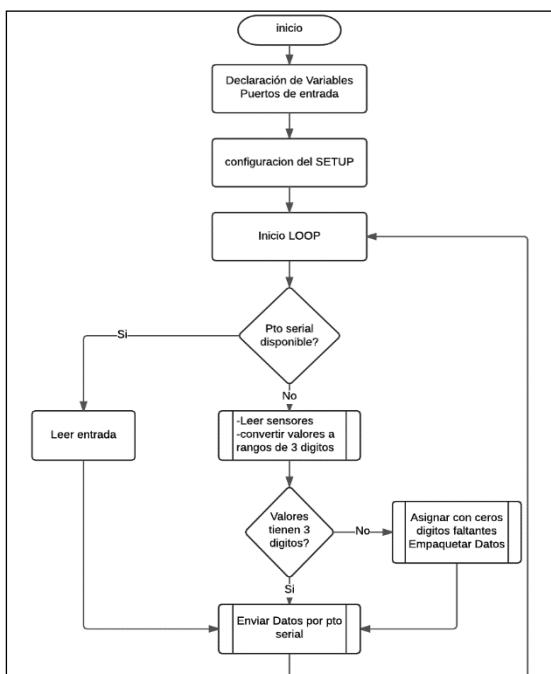


Figura. 13 Diagrama de flujo (Tx)

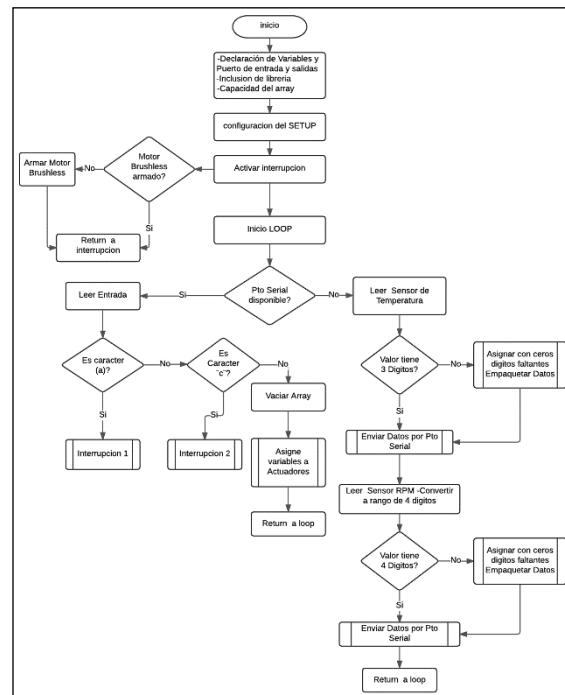


Figura. 14 Diagrama de flujo (Rx)

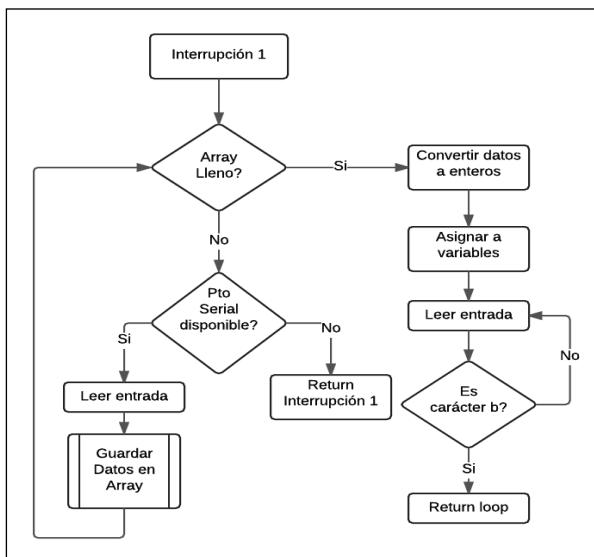


Figura. 15 Diagrama de flujo Interrupción 1

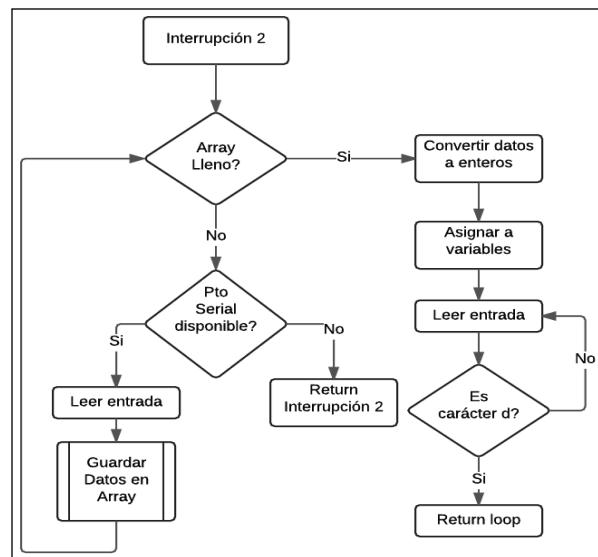


Figura. 16 Diagrama de flujo Interrupción 2

En la Figura. 17 se muestra el montaje físico de todos los componentes.

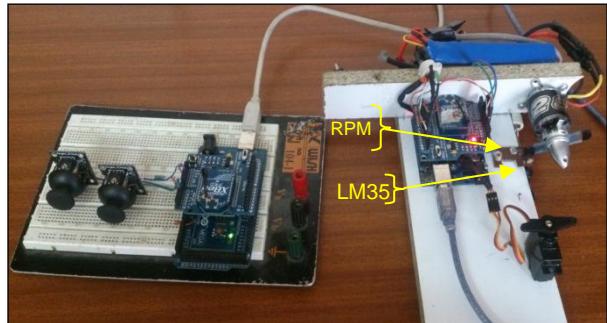


Figura. 17 Montaje real de los dos sistemas embebidos

Datos de transmisión y recepción: A continuación se expone la información enviada y recibida por los puertos seriales (Tx y Rx).

La información enviada y recibida, aunque pueda parecer caótica cumple con los requerimientos de empaquetado para telecomunicaciones, puesto que una de ellas y más importante es enviar la información con un bit de inicio y un bit final para cada trama, con esto el desempaquetado será 100% fiable para el receptor, adicionalmente se agregan algunos dígitos (ceros) para completar las tramas faltantes, generando así armonía en el envío de la información, de esta manera se logra separar las tramas con un arreglo de vectores, matriz o zonas de almacenamiento de un modo sencillo, el resultado de dichas pruebas se pueden observar en la Figura. 19

La imagen muestra dos ventanas de Arduino IDE. La ventana izquierda, titulada 'RPM_TX', muestra el código para el sistema de transmisión. La ventana derecha, titulada 'RPM_RX', muestra el código para el sistema de recepción. Ambos códigos utilizan el puerto serial para comunicarse.

```

//RPM_TX Arduino 1.6.0
//RPM_RX Arduino 1.6.0

```

Figura. 18 Carga de códigos en los sistemas embebidos

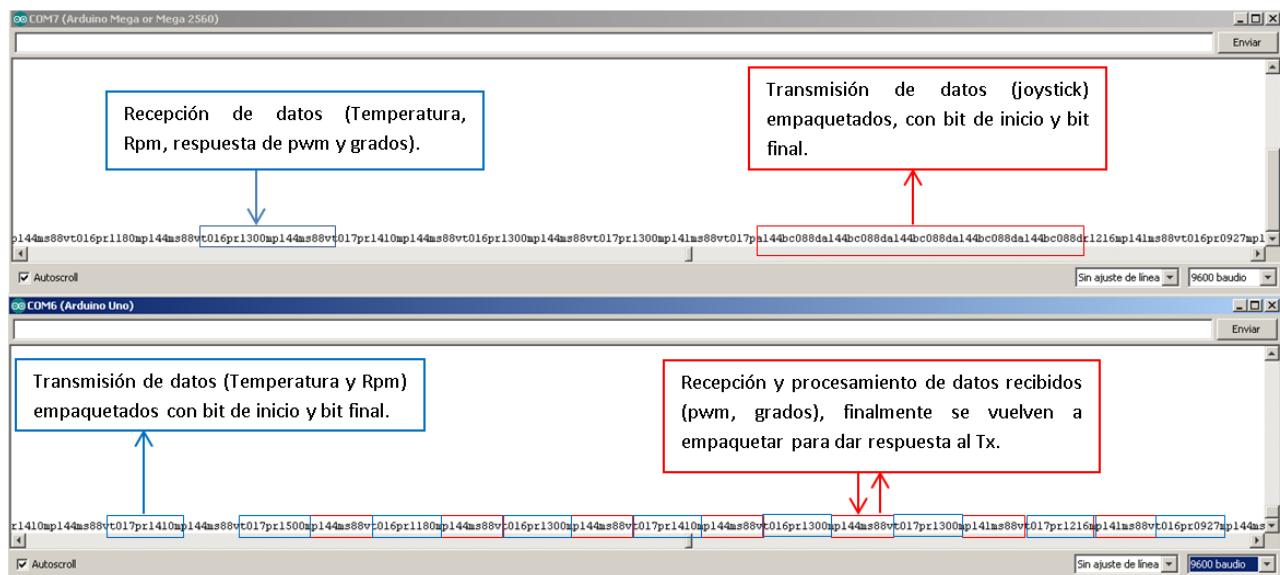


Figura. 19 Datos enviados y recibidos por los sistemas embebidos

Por otro lado se realizan pruebas a diferentes distancias para establecer el comportamiento que se obtiene y la fiabilidad en la entrega de la información

Las pruebas se realizan de la siguiente manera, se establece un número de N muros y se busca el límite de los mismos en donde sea posible traspasarlos, y recibir la información de manera confiable. Estas pruebas se realizan en un área cerrada limitando la potencia de irradiación en la señal.

A campo abierto también se realizan pruebas para determinar el comportamiento que se obtiene con esta tecnología. Así pues los resultados de dichas pruebas se resumen a continuación en la **Tabla 4**.

Tabla 4 Resultados a diferentes distancias

Número de Muros	Distancias de funcionamiento				
	Con Muros		A Campo Abierto		
	Distancia en metros	Observaciones	Distancia en metros	Observaciones	
1	23mts	Control y envío de datos confiables, no se evidencia fallas en la comunicación	10 mts	Control y envío de datos confiables, no se evidencia fallas en la comunicación	
3	10mts	Control y envío de datos confiables, no se evidencia fallas en la comunicación	30mts	Control y envío de datos confiables, no se evidencia fallas en la comunicación	
4	23mts	Control y envío de datos con retardo, se evidencian retardos en la comunicación, los muros de ladrillo y cemento tienen un grosor de 14,5 cm cada uno.	60mts	Control y envío de datos confiables, no se evidencia fallas en la comunicación	
6	24mts	Control y envío de datos con retardo, se evidencian retardos en la comunicación, los muros de ladrillo y cemento tienen un grosor de 14,5 cm cada uno.	100mts	Control y envío de datos confiables, no se evidencia fallas en la comunicación	
			160mts	Control y envío de datos, se evidencia un mínimo de retardo, no se afecta el número de paquetes entregados	

La comunicación presenta excelentes resultados de control, envío y recepción de datos a diferentes

distancias, puesto que la confiabilidad en la entrega de paquetes fue un 100% de fiabilidad.

Se evidencia un retardo en la entrega de los paquetes a distancias prolongadas o superiores a 100mts a campo abierto, donde la comunicación presenta una demora mínima en la entrega de la información, pero finalmente no afecta la llegada de los datos.

Los muros por su parte retrasan la comunicación de los dos sistemas en un porcentaje mayor a menor distancia, evidenciando esto en los datos obtenidos.

El criterio de calificación usado para definir la calidad en la comunicación depende en gran medida si se evidencia la perdida de paquetes a diferentes rangos de distancia:

- 1.0 =<2; " Envío y recepción de datos – Estado con deficiencias"
- 2 =<3.9; " Envío y recepción de datos – estado regular";
- 4< 5.0 "Envío y recepción de datos – Estado confiable"

La importancia de esta asignación de puntaje es tener un valor de referencia para la calificación de confiabilidad y así tener una idea clara de los distintos aspectos involucrados que pueden afectar la comunicación entre los módulos, la evaluación resultante se expone a continuación en la **Tabla 5** y



, en la

Tabla 6 Distancia de funcionamiento a campo abierto

Distancia de Funcionamiento comunicación XBee a campo Abierto

Distancia (Mts)	Aspectos individuales considerados	Valor	Calificación	
			Valor Promedio Global= 4,8	Aspecto bien implementado
10	Control y envío de Datos a 10mts	5,0	Envío y recepción de datos-Estado confiable	
30	Control y envío de Datos a 30mts	5,0	Envío y recepción de datos-Estado confiable	
60	Control y envío de Datos a 60mts	5,0	Envío y recepción de datos-Estado confiable	
100	Control y envío de Datos a 100mts	5,0	Envío y recepción de datos-Estado confiable	
160	Control y envío de Datos a 160mts	4,0	Envío y recepción de datos-Estado confiable	

y **Grafica 2** se resumen los criterios de evaluación.

Tabla 5 Distancia de funcionamiento con muro

Distancia de Funcionamiento comunicación XBee con muro

Distancia (Mts)	Obstáculo N° Muros	Aspectos individuales considerados	Valor	Calificación	
				Valor Promedio Global= 4,8	Aspecto bien implementado
10	1	Control y envío de Datos 1 Muro	5,0	Envío y recepción de datos-Estado confiable	
23	3	Control y envío de Datos 3 Muro	5,0	Envío y recepción de datos-Estado confiable	
23	4	Control y envío de Datos 4 Muro	5,0	Envío y recepción de datos-Estado confiable	
24	6	Control y envío de Datos 6 Muro	4,0	Envío y recepción de datos-Estado confiable	

Tabla 6 Distancia de funcionamiento a campo abierto

Distancia de Funcionamiento comunicación XBee a campo Abierto

Distancia (Mts)	Aspectos individuales considerados	Valor	Calificación	
			Valor Promedio Global= 4,8	Aspecto bien implementado
10	Control y envío de Datos a 10mts	5,0	Envío y recepción de datos-Estado confiable	
30	Control y envío de Datos a 30mts	5,0	Envío y recepción de datos-Estado confiable	
60	Control y envío de Datos a 60mts	5,0	Envío y recepción de datos-Estado confiable	
100	Control y envío de Datos a 100mts	5,0	Envío y recepción de datos-Estado confiable	
160	Control y envío de Datos a 160mts	4,0	Envío y recepción de datos-Estado confiable	



Grafica 1 Resultado XBee con muros



Grafica 2 Resultado XBee a campo abierto

RESULTADOS

Con la implementación de la tecnología XBee se obtienen resultados confiables en la recepción y envío de información como se pudo observar en las gráficas anteriormente ilustradas.

La comunicación en doble sentido mediante los módulos XBee permite desarrollar nuevas aplicaciones, un ejemplo claro de esto es el manejo y realimentación de la información usando diversos sensores y actuadores.

Gracias al uso de la comunicación serial con los que vienen provistos los módulos XBee se pueden usar en gran cantidad de sistemas embebidos, dando flexibilidad para sus aplicaciones.

CONCLUSIONES

Con la tecnología XBee es posible controlar diversas aplicaciones para uso industrial y comercial, el proyecto solo pretende demostrar la flexibilidad con la que cuentan este tipo de sistemas, su funcionamiento básico permite manipular y acoplar señales provenientes de diversos sensores en módulos embebidos y ejecutar acciones de control y adquisición de datos.

La capacidad y flujo de información cumplió con las expectativas previstas en el comienzo, puesto que no se evidenciaron perdidas de información a mayores distancias y obstáculos, solo pequeños retrasos en la llegada de los mismos.

REFERENCIAS

- [1] Zaradnik J. Ignacio, "Módulos XBEE de Digi International," Electrocomponentes 2014.
- [2] Posada A and gomez L, "XBee y arduino," electronics CapsuLab08 2012.
- [3] Carballal J., "Wi-Fi Cómo construir una red inalámbrica," 2005.
- [4] VALVERDE J, "El Estándar Inalámbrico ZigBee," Universidad Nacional de Trujillo, Perú, 2007.
- [5] Unal A., "Módulo XBee parte 4," Universidad Nacional, Colombia, 2011.