

Reprogramación de firmware de Arduino a través de transceptor XBEE

Rodrigo E. Russo

*Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp*

Mar del Plata, Argentina
rodrigo.e.russo@fi.mdp.edu.ar

Melisa G. Kuzman

*Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp*

Mar del Plata, Argentina
melisakuzman@fi.mdp.edu.ar

Walter A. Gemin

*Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp*

Mar del Plata, Argentina
agemin@fi.mdp.edu.ar

Juan M. López

*Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp*

Mar del Plata, Argentina
juanml@fi.mdp.edu.ar

Esteban A. Zapirain

*Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp*

Mar del Plata, Argentina
estebanzapirain@fi.mdp.edu.ar

Resumen—Este trabajo muestra el proceso de actualización de firmware de placas de desarrollo Arduino Nano, conectadas a través de una red inalámbrica XBEE con topología BUS. Se detallan las configuraciones necesarias en los módulos XBEE, las modificaciones en el bootloader y código que es necesario agregar al fuente de Arduino, para su correcto funcionamiento.

Index Terms—Arduino, XBEE, IoT, RF, bootloader

I. INTRODUCCIÓN

Muchas veces es necesario actualizar el firmware de microcontroladores que están ubicados en lugares de difícil acceso y como en este caso particular, pertenecen a una red de dispositivos que se comunican a través de mensajes en formato ASCII. Este trabajo presenta una solución a este problema, a través de una red inalámbrica conformada por módulos comerciales XBEE PRO S2C como el de la Fig. 1. Estos operan en la banda de 2.4 GHz y poseen un alcance de 3200 metros en espacios abiertos [1].

El esquema utilizado en este trabajo consiste en varias placas de desarrollo Arduino Nano con microcontrolador ATmega328p (Fig. 2), conectadas a un módulo XBEE por medio de su interface serie, posibilitando la comunicación inalámbrica entre ellos y una computadora conectada a un módulo XBEE que funciona como programador. Cada conjunto conformado por un módulo XBEE y un Arduino Nano o una computadora lo denominamos Endpoint. Esto se observa en la Fig. 3.

La idea es que se pueda reprogramar a cada uno de los Arduino Nano reemplazando el cable USB utilizado tradicionalmente, por un enlace inalámbrico realizado



Fig. 1. Módulo XBEE PRO S2C

por los módulos XBEE como se observa en la Fig. 4. También es necesario una manera de direccionar el módulo Arduino a reprogramar. Esto se analizará en detalle más adelante en este artículo.

II. CONFIGURACIÓN DE MÓDULOS XBEE

En este caso se utilizaron módulos XBEE PRO S2C XBP24CZTSIT-004 y por medio de la aplicación Digi XCTU [2] se carga el firmware 802.15.4 TH PRO, Fig. 5, que permite armar una red tipo BUS para enviar y recibir mensajes serie tipo caracteres ASCII.

Además, permite encriptar los mensajes para evitar la intromisión de terceros, que con un módulo XBEE, podrían enviar mensajes con intenciones maliciosas.

Los parámetros mínimos de configuración necesarios para este trabajo se muestran en la Tabla I.

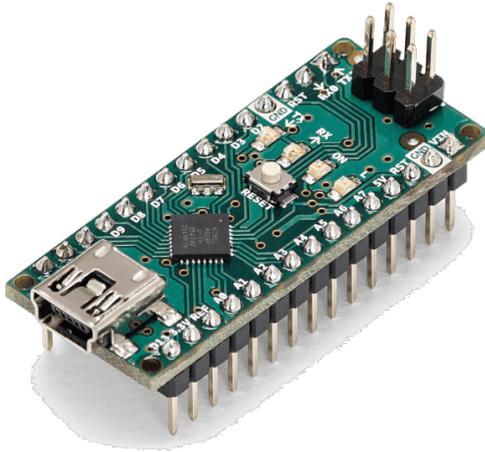


Fig. 2. Módulo Arduino Nano

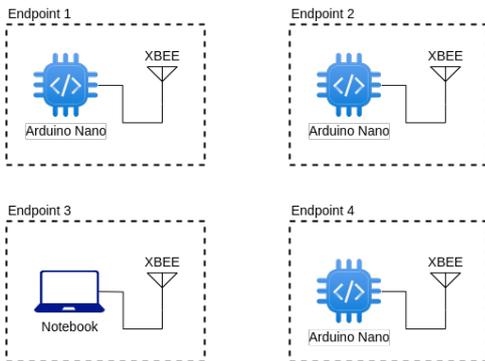


Fig. 3. Topología de la red

Aplicada esta configuración a cada uno de los módulos que van a intervenir en la red, se obtiene una topología BUS inalámbrica. Los caracteres que se envían a través de uno de ellos, llegan al resto de los módulos que conforman la red, dentro de su alcance.

De las pruebas realizadas surge que la configuración mostrada en la Fig. 3 para reprogramación de los Arduino Nano, solo funciona cuando hay únicamente dos módulos XBEE activos. Es decir que al momento de reprogramar, la red inalámbrica debe estar compuesta por el programador con su correspondiente módulo XBEE y el Arduino Nano conectado a otro módulo XBEE.

III. CONFIGURACIÓN DE MÓDULOS ARDUINO

III-A. Bootloader

Las placas de desarrollo Arduino Nano cuentan con un bootloader que permite la carga del firmware a través de un puerto UART utilizando un protocolo denominado stk500 [3].

El proceso de carga del firmware consiste en reiniciar el microcontrolador y enviar el código compilado palabra por palabra. Para este proceso se cuenta con una ventana de tiempo después del reinicio para comenzar la transferencia.

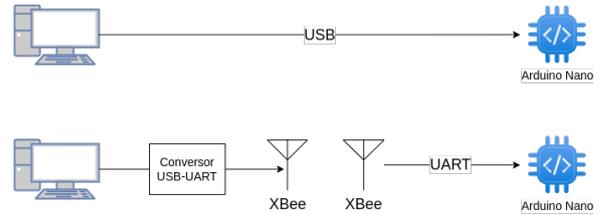


Fig. 4. Reemplazo de cable USB por enlace inalámbrico

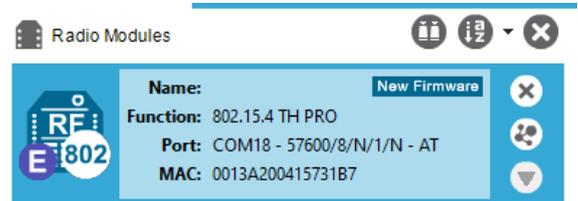


Fig. 5. Aplicación XCTU – Configuración de firmware

El bootloader utilizado debido a su simplicidad y disponibilidad de código fuente es el Optiboot [4]. En el mismo es posible cambiar la velocidad de carga del firmware a través de la interface serie, donde se debe hacer coincidir con la velocidad de puerto serie del XBEE de 57600 baudios, para que la comunicación entre ambos sea exitosa.

Una vez compilado el bootloader, el archivo con extensión hex generado, debe ser grabado en el microcontrolador por medio de un programador como el USBasp [5].

La velocidad de 57600 baudios resultó ser la más exitosa logrando un balance entre máxima velocidad y confiabilidad de transmisión, a diferencia de la velocidad por defecto que es de 115200 baudios.

III-B. Hardware

Las conexiones entre los módulos XBEE y Arduino Nano que conforman un Endpoint, se observan en la Fig. 6.

El microcontrolador ATmega328P, no tiene una instrucción para hacer un RESET por software, por lo que se implementa a través de una salida digital que mantiene por un tiempo en bajo la entrada de RESET por hardware. Con el objetivo de generar este tiempo es que se incorpora la red RC entre la salida D2 y RESET.

El Dip switch conectado entre D9 y D12 proporciona la dirección en binario del Endpoint.

III-C. Firmware

Para reprogramar el microcontrolador, se implementan cuatro comandos que se describen en la Tabla II. En primer lugar, los comandos PROG y NORM consisten en alternar entre los modos de programación y operación normal del equipo. En el modo PROG, el XBEE

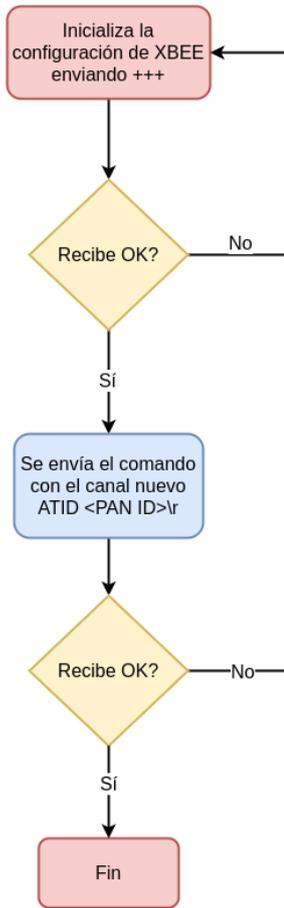


Fig. 7. Diagrama de flujo de reconfiguración de PAN ID de XBEE

Tabla II
COMANDOS NECESARIOS PARA INICIAR LA REPROGRAMACIÓN DE ARDUINO

Comando	Descripción
PROG<ID>	Cambia el PAN ID del XBEE a uno específico para la reprogramación (en este caso, PAN ID 9999)
NORM<ID>	Vuelve al PAN ID original para volver al funcionamiento normal del equipo
RESE<ID>	Reinicia el Arduino para comenzar el proceso de reprogramación
STRT<ID>	Comando que envía el Arduino al inicializarse para comunicarse correctamente

Tabla I
PARÁMETROS DE CONFIGURACIÓN DEL FIRMWARE DE XBEE

Parámetro	Valor	Descripción
+++		Ingreso al modo programación (comandos AT)
ID	0x2345	Identificador de Red de 4 caracteres hexadecimales
EE	Enable [1]	Habilitar encriptación AES
KY	0x1234...	Clave de encriptación AES de 32 caracteres hexadecimales
BD	57600	Velocidad en baudios de interfaz de serie.

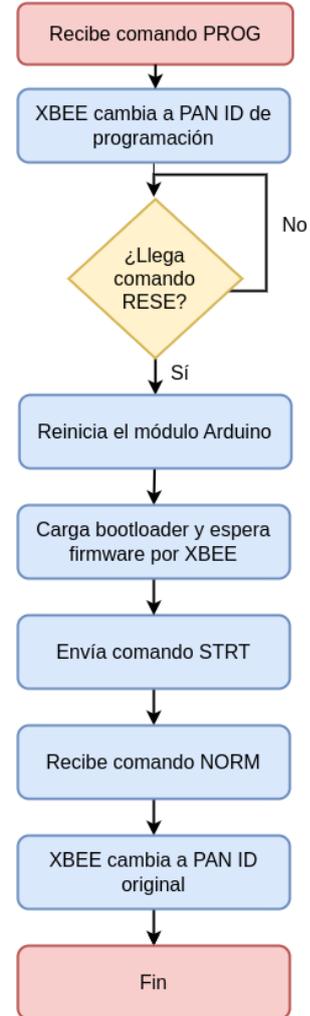


Fig. 8. Diagrama de flujo de reconfiguración de PAN ID de XBEE

