



Bizkaiko Foru Aldundia
Diputación Foral de Bizkaia
Berrikuntza eta Ekonomi
Sustapen Saila
Departamento de Innovación
y Promoción Económica

Programa Ekinberri 2007

SmartMotes

Nodos inalámbricos de redes de
sensores con inteligencia
semántica

D2.3 Evolución OTAP
(Over-the-air Programming)



Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa



RESUMEN

El presente documento se fundamentará en la presentación de las pruebas para aplicar la técnica OTAP (*Over-the-air Programming*) que se han realizado sobre la plataforma MicaZ.

La estructura del mismo comenzará por una explicación de los conceptos fundamentales, especialmente los de funcionamiento, que ayuden a situarse y a comprender la técnica. Después se procederá a describir de forma minuciosa las pruebas que se han realizado, paso a paso, hasta el momento en que los problemas superan a las soluciones. Se presentan también las alternativas tomadas para intentar superarlos, y los focos de ayuda para ello. Finalmente, cierran el documento unas conclusiones que en realidad sirven a modo de recordatorio de aspectos críticos de cara a nuevos intentos por parte de terceros.

Cabe mencionar que la finalidad del documento es la de servir de utilidad para futuros intentos de utilizaciones de la técnica OTAP, de cara a que no sea necesario comenzar de nuevo desde el punto de partida, ya que quedarán aquí recogidos los pasos avanzados.

HISTORIAL DE CAMBIOS

Versión	Descripción	Autor	Fecha	Comentarios
V0.9	Definición de la plantilla	Iñaki Vázquez	2/10/2007	
V1.0	Elaboración documento	Asier Arruti	14/3/2008	

TABLA DE CONTENIDOS

Resumen	3
Historial de cambios	4
Tabla de contenidos	5
1 Introducción.....	7
2 FUNCIONAMIENTO DE OTAP	9
2.1 Concepto	9
2.2 Funcionamiento	10
2.2.1 Distribución de la memoria	10
2.2.2 Método de carga	11
2.2.3 Requisitos	11
2.3 XOTAPLiteM.....	12
3 PRUEBAS DOCUMENTADAS.....	14
3.1 Consideraciones previas.....	14
3.2 OTAP con MoteConfig.....	14
3.2.1 Configuración de las conexiones del equipo.....	15
3.2.2 Configuración del software	16
3.2.3 Grabación local con habilitación OTAP.....	16
3.2.4 Grabación remota mediante OTAP.....	19
3.3 Pruebas para buscar soluciones.....	20
3.3.1 MoteView	21
3.3.2 Búsqueda por la red	23
3.3.3 Foros	24
3.4 Vías futuras	24
4 Conclusión	25
5 Referencias	26

ÍNDICE DE FIGURAS

Figura 1. Arquitectura <i>XOTAP</i>	10
Figura 2. Transferencia de la imagen OTAP.....	11
Figura 3. Puertos (COM & LPT)	15
Figura 4. Interface Board Settings	16
Figura 5. Configuración MoteConfig	17
Figura 6. Proceso de carga con éxito	18
Figura 7. Carga local de la base.....	18
Figura 8. Remote program	19
Figura 9. Mensaje de error	20
Figura 10. Connect to WSN.....	21
Figura 11. Opciones Gateway en MoteView.....	22
Figura 12. Opciones Sensor Board en MoteView	22
Figura 13. MoteView	23

1 INTRODUCCIÓN

La plataforma de WSN (*Wireless Sensor Networks*) empleada en el desarrollo del RFIDGlove, aplicación perteneciente al proyecto SmartMotes, es la MicaZ de Crossbow.

Esta compañía ha desarrollado diversas plataformas, de manera evolutiva gran parte de ellas, que permiten implementar redes inalámbricas destinadas principalmente a la sensorización del entorno y a obtener información para un nodo central de mayor capacidad computacional, para desarrollar aplicaciones domóticas, de inteligencia ambiental, o similares.

Alternativa a la tecnología Zigbee (aunque los últimos modelos incorporan 802.15.4 para la transmisión RF), estas plataformas se centran asimismo en la búsqueda de la máxima reducción posible de consumo energético y de dimensiones, de forma equilibrada con una potente capacidad de computación, procesado y transmisión de datos a una velocidad interesante (dado el escaso volumen del tráfico que intercambiarán).

De entre todas las que Crossbow pone a disposición de sus clientes, como ya se ha comentado, la seleccionada para llevar a la práctica la aplicación de RFIDGlove es la plataforma MicaZ, cuyas especificaciones pueden hallarse, además de en su *datasheet*, en el documento comparativo “D2.1_SmartMotes_WSN_ComparativeAnalysis”, disponible en la misma localización que el presente archivo. Su funcionamiento se sustenta en el sistema operativo TinyOS, y el lenguaje de programación a manejar es NesC, una variante de C.

Se conoce así de forma genérica a las motas de prototipado, las cuales permiten un mayor rango de acciones mediante diversas funcionalidades para cada conexión posible, así como la reiteración de pruebas hasta obtener el programa adecuado para el producto deseado. Sin embargo, una vez se ha definido correctamente tal programa, existe una variante empastillada de las MicaZ, las MicaZ OEM, las cuales se hallan en un formato compacto de soldado en superficie, diseñadas en forma para adecuarse a las necesidades de máxima reducción dimensional de los productos comerciales. Sin embargo, el manejo precario de las pruebas propias de la investigación puede provocar en ocasiones la necesidad de reprogramar también este segundo modelo de motas.

Para ello, existen dos alternativas. Por un lado, se plantea la posibilidad de incorporar conectores a los pines cuyo empleo se requiere para la reprogramación descrita, sobre la misma placa de la ubicación final de la mota compacta (o al menos en su primera versión

hasta estar completamente seguros de su completo y correcto estado). La segunda alternativa viene dada por una de las opciones que incorpora la propia tecnología de las motas, denominada como OTAP (*Over-the-air Programming*), y que será objeto de estudio de este documento, centrándose en las pruebas y el avance realizado en este campo. Esta funcionalidad viene definida con objeto de permitir volcar programas desde una estación base que hace de pasarela entre la plataforma y el ordenador y una mota situada en una ubicación remota, de forma totalmente inalámbrica. La utilidad de esto resulta de sumo interés; sin embargo, la definición que incorpora es limitada, y consta de múltiples restricciones y limitaciones que complican su uso, aspectos que también se incluirán en el documento, más adelante, así como todo el proceso seguido y a seguir.

A pesar de todas las posibles dificultades, no puede obviarse el hecho de que toda esta funcionalidad se irá simplificando según los usuarios y la gente que trabaja con esta plataforma la aplique y descubra nuevas vías para resolver los problemas, como ha sucedido con multitud de posibilidades que ofrecen estas motas. El mejor recurso de aprendizaje disponible on-line es la lista de distribución que Crossbow ha puesto a disposición de sus clientes para que compartan sus conocimientos, en la que algunos responsables de la propia empresa colaboran resolviendo algunas de las cuestiones que surgen a los usuarios.

2 FUNCIONAMIENTO DE OTAP

El presente apartado recogerá algunas nociones fundamentales de la programación *Over-the-air* (a través del aire, OTAP) para algunas de las plataformas de Crossbow, información que, a pesar de hallarse ya descrita en la documentación oficial junto a otros aspectos de esta técnica de programación, se incluye aquí por representar un alto interés para la comprensión global de nuevos usuarios que deseen acercarse a ella.

2.1 Concepto

La capacidad de programación OTAP que presenta la plataforma MicaZ (así como sus hermanas Mica2...) aporta la posibilidad a quienes trabajan con ella de reprogramar cualquiera de las motas que forman parte de una aplicación, como es fácilmente intuible por su nombre, sin necesidad de ningún tipo de cableado, siendo únicamente necesaria una conexión física inalámbrica, disponible mediante la capacidad de comunicación por radiofrecuencia que forman parte de la razón de ser de estos equipos para redes de sensores inalámbricas (WSN, *Wireless Sensor Networks*).

La condición indispensable para poder aplicar este tipo de programación es que las motas a reprogramar, así como la estación base que hará las funciones de pasarela para ello, formen parte de una misma red *Mesh*.

Así, OTAP permite a una o más motas recibir nuevas imágenes de programas que han de cargar en sus memorias y ejecutar como rutina habitual hasta nueva reprogramación, mediante la aplicación software XServe, que asumirá el papel de servidor en la comunicación inalámbrica.

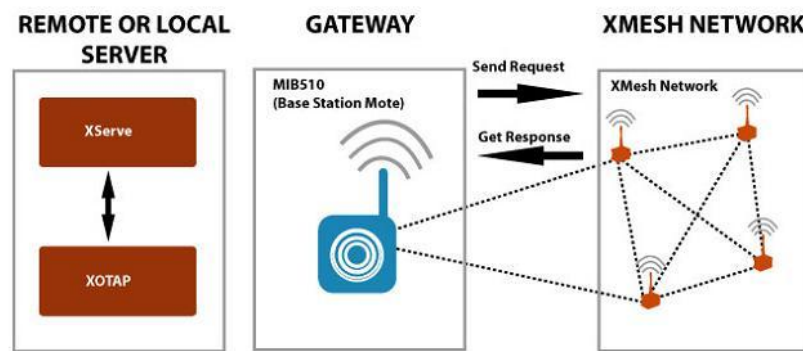


Figura 1. Arquitectura XOTAP

2.2 Funcionamiento

2.2.1 Distribución de la memoria

Cada mota de Crossbow dispone de una memoria serie Flash externa comunicada con el microprocesador. Esta memoria puede almacenar hasta 512 kB de datos y es no-volátil; es decir, que mantiene su contenido cuando se desconecta la alimentación y se apaga el dispositivo. La técnica OTAP emplea esta memoria para almacenar el código. Dicha memoria se divide en 4 ranuras o espacios de memoria (nombradas de 0 a 3) para almacenar las diferentes imágenes de código o datos. Cada ranura puede ser un número de páginas configurado por el usuario en tramos de 2 kB. Por defecto, cada ranura es de 128 kB (y por lo tanto 64 páginas). Las ranuras sin usar quedan disponibles para el usuario para que pueda registrar datos.

La ranura 0 siempre almacena la imagen OTAP. Es el software de OTAP completo que debe invocarse para descargar por radio un nuevo programa y cargarlo e instalarlo en la mota. Los usuarios pueden conectar los componentes OTAP directamente en sus aplicaciones, aunque estos componentes consumirán varios cientos de cientos de bytes de RAM. Para evitar eso, Crossbow promueve entre sus usuarios un pequeño componente, *XOTAPLiteM*.

Este componente emplea tan sólo 30 bytes extra. Si una aplicación es cargada en un componente *XMesh*, *XOTAPLiteM* queda automáticamente disponible.

2.2.2 Método de carga

El servidor puede enviar un comando a *XOTAPLiteM* que provoque que la mota se reinicie desde la imagen de la ranura 0 para procesar la descarga del nuevo código completo. Las otras ranuras en la memoria están destinadas para el código del usuario o datos. Antes de reprogramar la mota, los usuarios deben estar seguros de que la imagen OTAP reside en la ranura 0 de la memoria Flash externa.

OTAP trabaja transmitiendo pequeños fragmentos de código, descargándolos a la mota seleccionada. La imagen del código completo es segmentada en esos fragmentos. En cuanto OTAP reciba los fragmentos, los almacenará en la memoria flash. La secuencia de descarga es independiente de los procesos de reprogramación y reinicio. Una vez que la imagen del código haya sido almacenada en la memoria Flash, los usuarios podrán reiniciar cualquier imagen en cualquier momento.

A partir de ahí, tal y como se muestra en la figura 2, en la mota se carga el programa desde la Flash externa hasta la zona de memoria adecuada en la Flash de programa que incorpora el microprocesador ATmega128L.

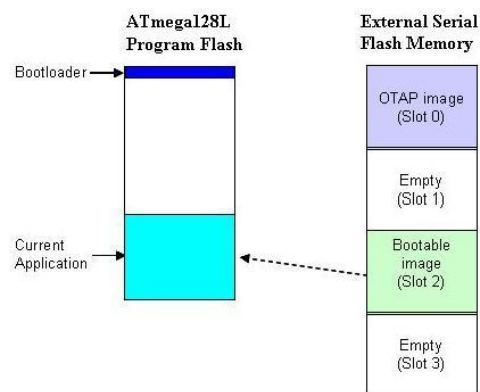


Figura 2. Transferencia de la imagen OTAP

2.2.3 Requisitos

Reprogramar el microprocesador desde la memoria Flash requiere que el código de arranque resida en la parte superior de la memoria del microprocesador ATmega. Este código se carga una vez durante la programación inicial de la mota, mediante UISP.

Hay múltiples componentes requeridos para que la técnica OTAP funcione correctamente:

- *Bootloader* o cargador de arranque: Este bloque independiente de código garantiza

la ejecución después de cada reinstalación de la mota, independiente de la aplicación TinyOS. Es responsable de la carga de la imagen del programa en la memoria flash del procesador, si la consulta ordenada a la reinstalación se hace mediante un mensaje de la herramienta *XOtap*.

- *XOTAPLiteM*: se trata de un sub-componente conectado en la aplicación del usuario que incluye la funcionalidad de *XMesh* en la mota.
- Imagen del código OTAP, en la ranura 0 de la memoria flash serie.
- La aplicación *XServe* ejecutándose en el servidor.
- La aplicación *XOtap* ejecutándose en el servidor.

La funcionalidad OTAP es implementado dentro de TinyOS por un componente denominado como *XMgmt*. Este componente es responsable del almacenamiento de información sobre el estado de las imágenes en la flash serie, respondiendo a cualquier cuestión, pregunta o solicitud que le haga la herramienta *XOtap* y de-fragmentando en programa o las imágenes de datos enviados por el servidor.

2.3 XOTAPLiteM

El *XOtapLoader* programado puede ser implementarse de forma vinculada con la aplicación del usuario de la forma en que sigue.

En el archivo de componente de la aplicación, ha de añadirse la siguiente sentencia.

```
implementation {  
  
    ...  
  
    MyAppM.XOtapLoader -> MULTIHOPROUTER;  
  
}
```

Asimismo, en el archivo de componente, ha de añadirse el siguiente interface y el comando:

```
Module MyAppM {  
  
    Provides {  
  
        ...;  
  
    } uses {  
  
        ...  
  
    Interface X0tapLoader;  
  
        ...  
  
    }  
}
```

Adicionalmente, añádase el evento para comprobar conjuntamente con la aplicación si puede ser interrumpida después de cargar y reiniciar con otra imagen. Si la aplicación está en medio de otra acción, ésta completará sus acciones y dejará la mota en un estado conocido antes del reinicio.

```
Event result_t X0tapLoader.boot_request (uint8_t imgId) {  
  
    call X0tapLoader.boot (imgId);  
  
    return SUCCESS;  
  
}
```

Una vez que la mota recibe el comando de arranque y la aplicación da luz verde al reinicio, se programará el tiempo de reinicio en base a la distancia a la que se halla de la base; es decir, a la más cercana a la base se le programará un retardo mayor para el reinicio, para compensar que las motas dependientes de otras no pierdan mensajes de la descarga que aún tengan pendientes, debido al reinicio de sus padres. Algunos valores típicos de estos retardos se presentan en la página 84 de [\[Crossbow07\]](#), así como más información adicional sobre el funcionamiento de OTAP en las páginas que lo rodean.

3 PRUEBAS DOCUMENTADAS

A diferencia del anterior, en el cual se presentaban los conceptos fundamentales de partida para comprender como funciona el sistema OTAP (necesarios para asumir la base y poder desarrollar las pruebas), este apartado se centrará en describir las pruebas de experimentación con esta técnica hasta el punto en que el avance a su éxito de funcionamiento se vio frenado por problemas técnicos.

3.1 Consideraciones previas

Las pruebas que se van documentar a continuación se han desarrollado siguiendo los pasos descritos en los manuales pertinentes. Para conocer este progreso teórico de forma completa (aquí sólo se presentarán los puntos alcanzados con éxito), deben consultarse los documentos [Crossbow06] y [Crossbow07], disponibles ambos en el website del fabricante, www.xbow.com, en el apartado *support*.

Por otro lado, se presupondrá que los programas a instalar son correctos tanto semántica como lógicamente, y asimismo, que se hallan correctamente compilados para la plataforma con la que se trabaja, MicaZ en este caso, y el grabador empleado.

3.2 OTAP con MoteConfig

Crossbow pone a disposición de sus clientes una aplicación software que simplifica lo máximo posible la programación de las motas con la posibilidad de habilitar la técnica OTAP para futuras reprogramaciones, conocida como *MoteConfig*.

Dicho de otra forma, antes de que el usuario pueda programar las motas de la red por el aire, es necesario prepararlas con el cargador de arranque y la imagen OTAP. Para ello es para lo que se utiliza *MoteConfig*.

Partiendo de la base de una correcta instalación de la aplicación, los pasos seguidos se van detallando a continuación.

3.2.1 Configuración de las conexiones del equipo

Lo primero de todo es asegurarse de la correcta configuración del equipo conectado. En el laboratorio de investigación MoreLab se dispone tanto del grabador MIB510 como del MIB520, y será éste último el empleado por ser el más nuevo de ambos, y cómodo al prescindir de alimentación (la cual toma del propio conector USB). Es importante cerciorarse del número de puerto COM al que se conecta el MIB520, siendo preferible que el número del mismo sea de una sola cifra, ya que según la aplicación empleada, esto da problemas en ocasiones. La comprobación del número de puerto COM puede hacerse siguiendo la ruta:

Panel de control > Sistema > Hardware > Administrador de dispositivos > Puertos (COM & LPT)

En el caso del MIB520 aparecerán 2 números de COM para él. El de valor inferior es el que se corresponde con la salida de datos, y el de valor superior el que funciona para la entrada de mensajes por el puerto serie. Para la configuración de las motas, deberá configurarse para el de menor valor. Como se aprecia en la siguiente figura, para el caso de ejemplo, se configurará el COM6.



Figura 3. Puertos (COM & LPT)

En el caso de que el número de COM sea de 2 cifras, se recomienda modificarlo para evitar posibles errores en el establecimiento de la conexión entre el PC y la base de la plataforma. Estos errores se han dado en ocasiones debido a la conversión de los puertos USB a puertos COM virtuales, y suceden frecuentemente al trabajar con las pantallas uOLED, que se emplean para trabajar de forma integrada con las motas. Aunque no sea éste el preciso caso, también se ha dado alguna situación conflictiva, por lo que el cambio es recomendable. Esto puede hacerse fácilmente (a partir de la ruta anterior), seleccionando **Propiedades** mediante el menú desplegable del puerto a modificar, y en la pestaña **Port Settings**, la opción **Advanced Settings for COMX** lo permite mediante el desplegable **COM Port Number** (debe emplearse uno que se halle libre de uso).

3.2.2 Configuración del software

El primer paso será arrancar la aplicación de MoteConfig, disponible por defecto tras la instalación en:

Inicio > Programas > Crossbow > MoteConfig 2.0

O bien como icono en el escritorio si así se ha elegido.

Para configurar las conexiones, se accede al menú **Settings**, en su opción **Interface Board**. Así, debería quedar de la manera que se aprecia en la imagen.

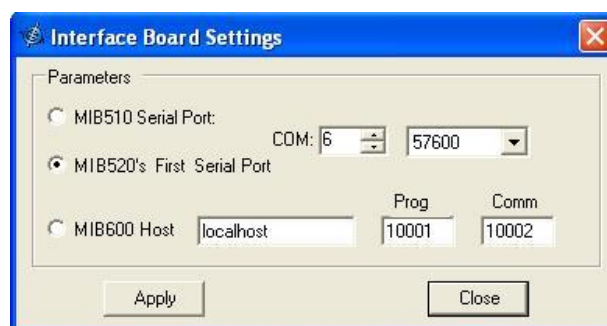


Figura 4. Interface Board Settings

3.2.3 Grabación local con habilitación OTAP

Una vez dentro del programa, en la pestaña **Local Program**, para programar las motas que se reprogramarán de forma remota, ha de seleccionarse mediante el botón **Select** el archivo .exe que se crea al compilar el programa, en **build > micaz > ...**. Hay que tener en cuenta que debe tratarse siempre de una aplicación *XMesh*; es decir, que la topología de red para que OTAP funcione siempre debe ser en forma *Mesh*. Esto resulta ser una restricción relativa, ya que en realidad este tipo de red suele resultar el más adecuado para el tipo de aplicaciones en que se emplean estas plataformas de *Wireless Sensor Networks*.

En caso de utilizar OTAP para cargar a una mota una aplicación que no sea *XMesh*, no se podrá volver a cargar de nuevo ningún tipo de aplicación, “nunca” según parece. Esto lo hace en parte inapropiado para ciertos tipos de aplicación en que se programen las motas para funciones muy distintas de aquellas para las que han sido diseñadas.

Una vez seleccionado el archivo, **MoteConfig** hace un breve análisis del archivo y

preconfigura algunos parámetros que después se pueden modificar (por ejemplo, si reconoce el archivo como el propio de la base, asigna 0 a su *Node ID*, o en caso contrario, lo preconfigura a 1).

El siguiente paso es seleccionar y asignar los valores deseados a las diversas características configurables que permite el programa, mediante los campos que presenta habilitados, como *Node ID* o *RF Channel* (campos que habitualmente se configuran mediante el archivo *MakeXBowLocal*).

La acción más importante en este momento es activar con un *tick* la opción **OTAP Enable**. Es muy importante tener siempre presente que esta opción sólo debe configurarse para las motas que se reprogramarán remotamente, nunca para la que actuará como estación base, de pasarela entre la plataforma y el ordenador que procesará los datos.

Así pues, la aplicación debe presentar una apariencia similar a la de la siguiente imagen.

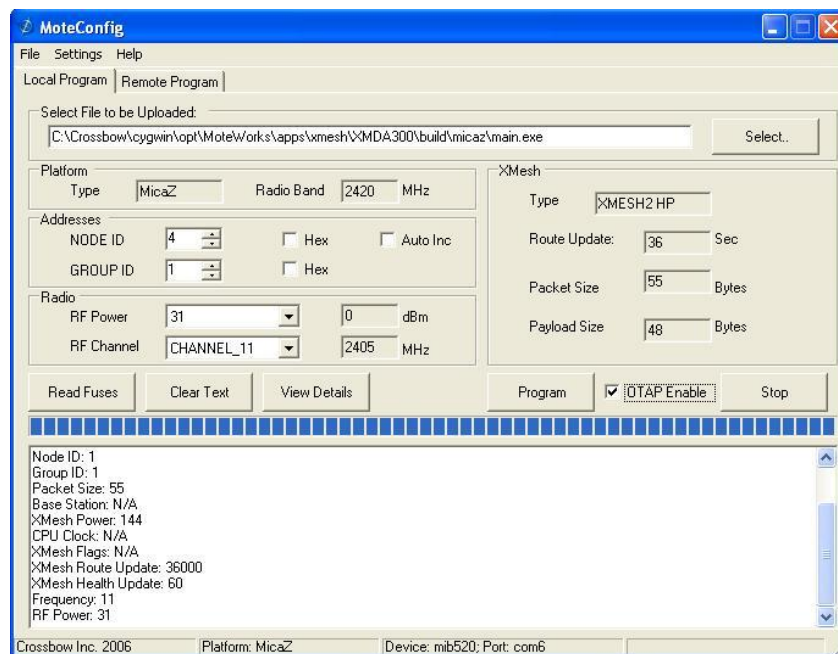


Figura 5. Configuración MoteConfig

La carga en la mota conectada en cada momento comenzará al pulsar el botón **Program**. El proceso finaliza de forma exitosa cuando la consola inferior presenta el mensaje "Uploading SUCCESSFUL!", como se ve en la imagen inferior.



```

Setting fuse using mib520
uisp -dprog=mib520 -dserial=/dev/ttyS5 -dpart=ATMega128 --wr_fuse_h=0xd0
installing bootloader using mib520

installing binary using mib520
uisp -dprog=mib520 -dserial=/dev/ttyS5 -dpart=ATMega128 --upload if="C:\Crossbow\MoteConfig\TOSBootLoader\bl_micaz.srec"

Setting fuse using mib520
uisp -dprog=mib520 -dserial=/dev/ttyS5 -dpart=ATMega128 --wr_fuse_h=0xd8
Uploading SUCCESSFUL!

```

Crossbow Inc. 2006 Platform: MicaZ Device: mib520; Port: com6

Figura 6. Proceso de carga con éxito

Cuando ya se han grabado todas las motas que se programarán de forma remota, queda configurar la base. Para ésta, mediante el botón **Select** debe seleccionarse el archivo .exe de la aplicación *XMeshBase*. Configurándolo de forma análoga a las otras motas, pero con las características propias de la base, la principal diferencia sólo ha de ser que la opción de **OTAP Enable** ha de estar deshabilitada, y esto es de suma importancia. Así pues, después de programar la base, la apariencia final de MoteConfig será como la que presenta la siguiente imagen.

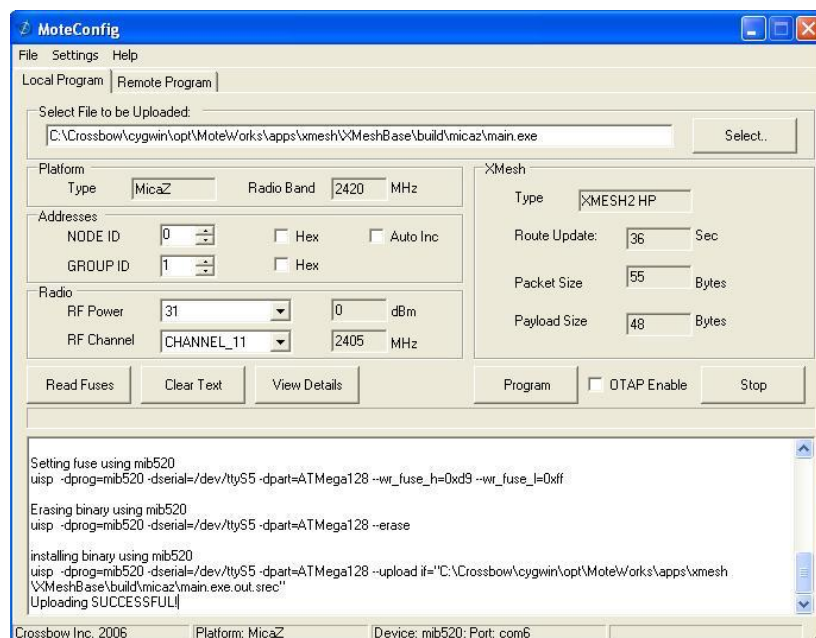


Figura 7. Carga local de la base

3.2.4 Grabación remota mediante OTAP

MoteConfig permite programar las motas de forma remota (el fin de OTAP) mediante las opciones presentadas en la pestaña **Remote Program**, cuya presencia puede apreciarse en la figura 8.

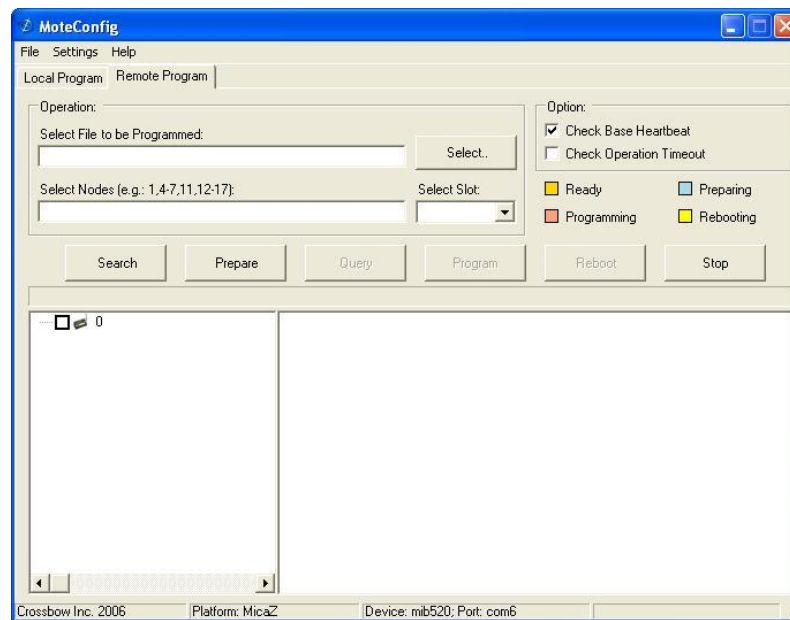


Figura 8. Remote program

La primera tarea ha de ser cerciorarse de la correcta predisposición del hardware; es decir, la mota base debe estar conectada al grabador que actuará de pasarela, y cada una de las motas remotas que se pretende reprogramar inalámbricamente (y que previamente se cargaron con aplicaciones *XMesh*) debe estar encendida.

Una vez todas están en su estado adecuado, el siguiente paso es pulsar en el botón **Search**, que arrancará el *XServe* y escuchará la presencia de los nodos remotos. Las motas encontradas dentro de la red serán representadas en la pantalla, en el diagrama de árbol de la zona inferior izquierda del MoteConfig, mostrándose su dependencia respecto a la estación base.

Sin embargo, es aquí donde surge el primer problema. Tras hacer la búsqueda, MoteConfig no halla ninguna mota mediante la base, y presenta el siguiente mensaje.

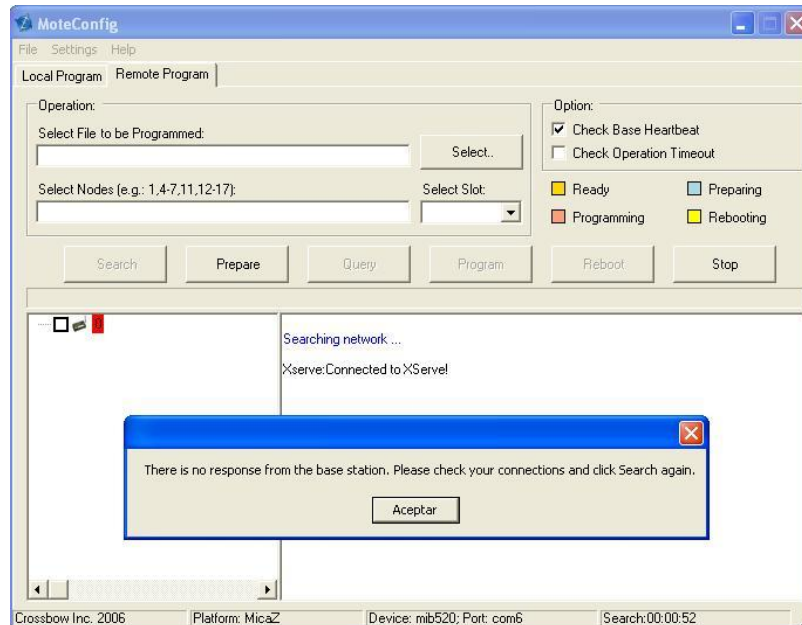


Figura 9. Mensaje de error

Como se ve en la imagen, XServe ha sido arrancado correctamente, pero las conexiones con la estación base se han establecido de forma incorrecta. La recomendación de la aplicación es comprobar las conexiones y pulsar **Search** de nuevo.

Sin embargo, tras probar con también con el MIB510, y con diversos cables en cada uno de los casos (convertidores USB-serie para éste y cables USB para MIB520), el resultado ha sido siempre el mismo mensaje de error, por lo que se descartan los fallos físicos en estos elementos.

3.3 Pruebas para buscar soluciones

Llegados a esta situación, el primer paso a dar (además de la mencionada comprobación de cables y grabadores para el descarte de errores físicos) es la comprobación del puerto COM, dado que en ocasiones, al perderse temporalmente la conexión, resulta modificado.

Después, se comprueban si realmente se llega a leer algo, mediante la aplicación **MoteView**.

3.3.1 MoteView

Esta aplicación está localizable, al igual que **MoteConfig**, en la siguiente ubicación.

Inicio > Programas > Crossbow > MoteView 2.0F

O bien como icono en el escritorio si así se ha elegido.

3.3.1.1 Configuración de la conexión

Al igual que con el anterior programa, el primer paso será configurar correctamente los parámetros de la conexión que va a establecerse.

Esto puede hacerse desde el menú **File**, seleccionando la opción **Connect to WSN**. Esto abrirá la ventana que se presenta en la siguiente imagen, que permite diversas opciones.

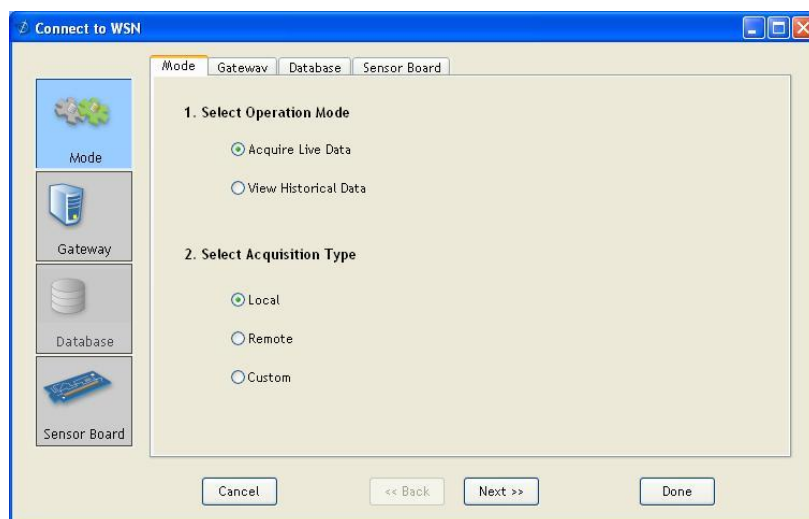


Figura 10. Connect to WSN

Pulsando el botón **Next**, es posible desplazarse por las diversas opciones. En la pestaña **Gateway** deben modificarse los parámetros clásicos para que se adecuen a la conexión deseada.

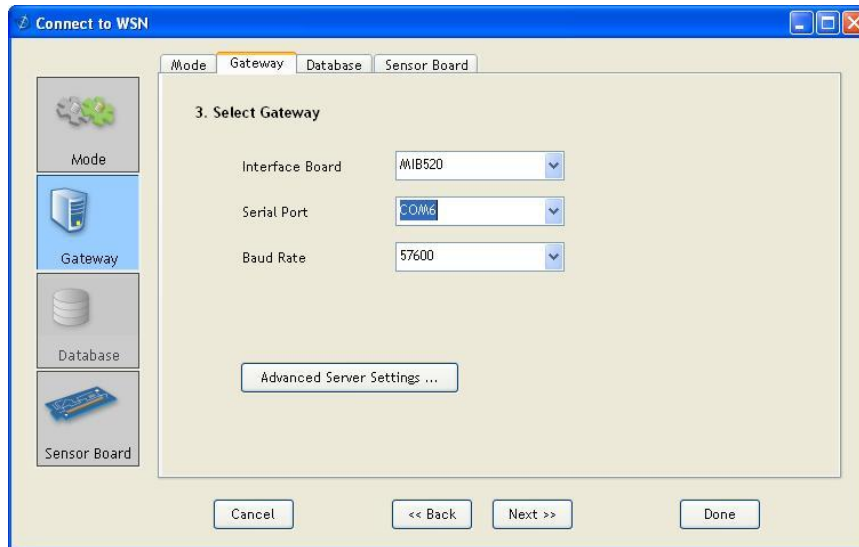


Figura 11. Opciones Gateway en MoteView

Adicionalmente, en la pestaña **SensorBoard** hay que seleccionar aquel modelo de placa con la que se va a trabajar, que se adecuará en función de la aplicación que se ha cargado sobre la mota.

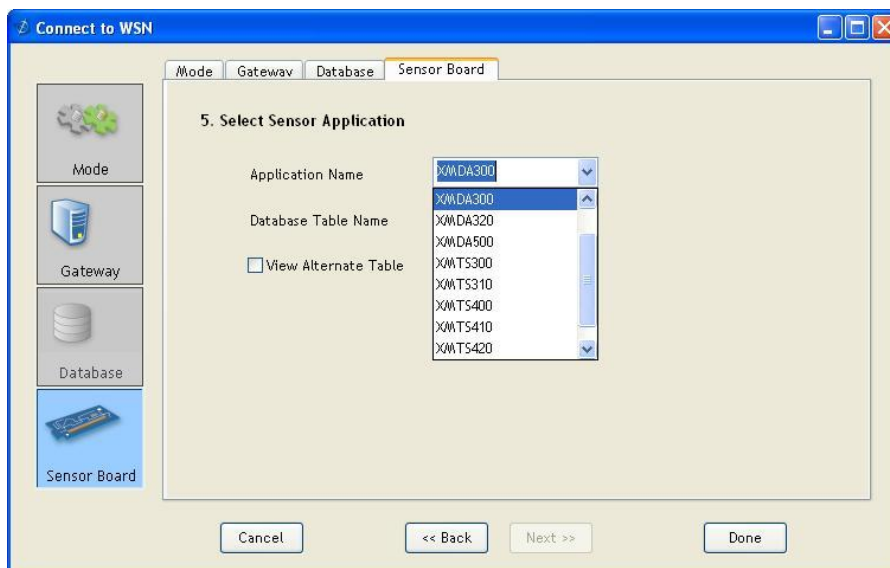


Figura 12. Opciones Sensor Board en MoteView

Pulsando sobre el botón **Done**, se establecerá la pertinente conexión, arrancándose de nuevo XServe. Sin embargo, como se aprecia en la siguiente imagen, el resultado ha sido indiferente, continúa sin recibirse ningún paquete de las motas, y sin siquiera detectarse

ninguna de ellas.

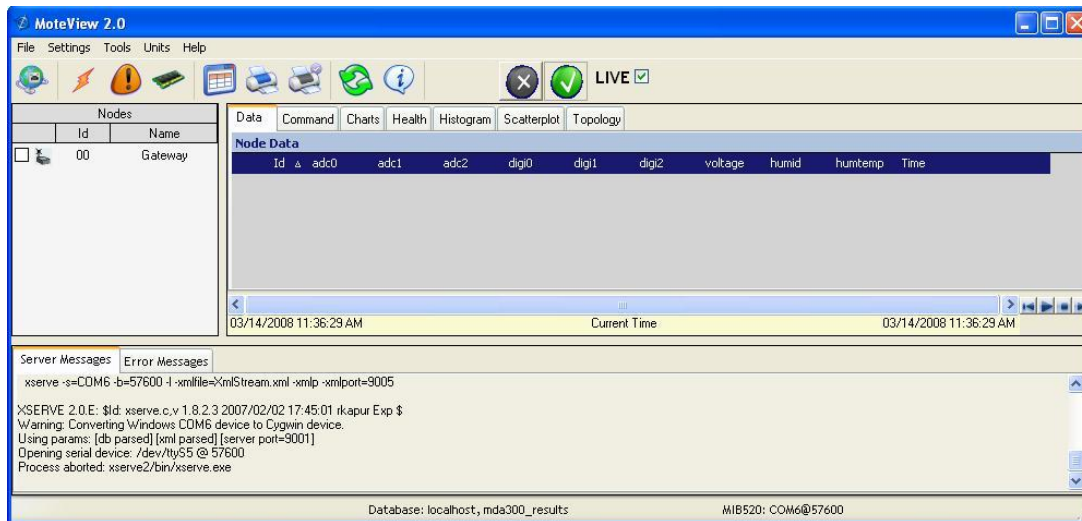


Figura 13. MoteView

La aplicación permite detener su funcionamiento y/o arrancarlo de nuevo con los botones correspondientes del cuadrante superior derecho de la pantalla.

3.3.2 Búsqueda por la red

Cabe hacer mención en estos momentos de forma genérica a la búsqueda en la red. Y el comentario que corresponde es acerca de la escasa información sobre la técnica OTAP. Salvo la información y documentación que proporciona la propia web del fabricante Crossbow, y los foros de ayuda existentes (que se comentarán más adelante), el resto de entradas aparecidos hoy por hoy en los buscadores más frecuentes se reducen a menciones de algunas aplicaciones reales en las que se dice haber empleado esta técnica, y en las que sólo se alaba brevemente la utilidad de la misma, sin entrar en ningún caso a detallar el proceso de su uso. Un ejemplo de su empleo se sucedió en un gimnasio en que se monitorizaba la sensorización del estado de algunas máquinas de ejercitación.

Esto complica la solución de consultar otras experiencias por este medio. Sin embargo, se solventará probablemente según pase el tiempo y más gente desarrolle experiencias satisfactorias con OTAP.

3.3.3 Foros

Para solucionar los inconvenientes y problemas que surgen, los foros siempre son buenas herramientas. Respecto a éstos, los comentarios útiles se hallan en el foro / lista de distribución que gestiona la propia empresa Crossbow. El historial de archivos se halla accesible en [\[TinyOSHelp\]](#), se actualiza constantemente y permite acceder a todos los mensajes que se han mandado a la lista desde que se creó.

La consulta directamente a lista, donde otra gente que también trabaja con la plataforma trata de ayudarse entre sí, también da sus frutos.

En ambos casos, las propuestas remiten a volver a comprobar las configuraciones de grabador y puerto COM. También se habla de múltiples problemas con los adaptadores USB-Serie, por lo que se recomienda trabajar con el MIB520 de entre los disponibles en el laboratorio MoreLab. Adicionalmente, se insiste en la necesidad de que los programas con los que se trabaje sean siempre aplicaciones *XMesh*, y de comprobar si con otras aplicaciones que exclusivamente se diseñaron para capturar todos los mensajes que se hayan enviado por radio, se reciben los correspondientes.

Así pues, mediante la vía del foro no resulta posible una mayor ayuda que todas estas sugerencias, en múltiples ocasiones de gran ayuda. Mención especial merecen por su ayuda Giri Baleri (de Crossbow) y Jamie Coates.

3.4 Vías futuras

En [\[Crossbow07\]](#) se comenta la posibilidad de utilizar OTAP desde el interfaz *Cygwin* mediante comandos *bash*, pero se presenta como alternativa para programadores con ciertos conocimientos de la materia.

4 CONCLUSIÓN

La técnica de programación remota OTAP supone una gran ventaja a primera vista. Sin embargo, es importante tener siempre en cuenta las múltiples restricciones que las intenciones de su uso implican, como se han ido describiendo a lo largo del documento. Hoy por hoy, y para la plataforma con la que se trabaja (MicaZ), se permite en teoría su uso para unas aplicaciones bastante concretas.

Además, debe decirse que, como ya se citó, dispone de escasa documentación al respecto, dado que su uso no parece estar aún demasiado extendido. Es por ello que aún no ha sido posible emplear esta técnica, dados los múltiples problemas que han surgido, y que no se hayan recogidos ni ellos ni sus soluciones en ningún manual ni documento de otro tipo.

Esas dificultades no son barreras infranqueables, sino que dadas las circunstancias, el tiempo dedicable en estos momentos a investigar esta vía de programación ha expirado, pero precisamente la razón de este documento es facilitar el camino en la medida de lo posible a la persona o personas que más adelante retomen esta labor.

Por otro lado, debe tenerse presente la restricción que ya se mencionó anteriormente, la obligación de que se trabaje siempre con aplicaciones, y en caso de no hacerlo la posibilidad de inhabilitar el dispositivo para otras aplicaciones. Todo esto restringe la funcionalidad de la técnica OTAP.

En resumen, OTAP puede ser una buena alternativa según su desarrollo y uso mejore o se incremente, pero actualmente se halla en un estado algo precario para dedicarle un período de tiempo más extenso.

5 REFERENCIAS

[Crossbow06] Crossbow. "MoteConfig User's Manual". California, USA, November 2006.

[Crossbow07] Crossbow. "XMesh User's Manual". California, USA, May 2007.

[TinyOSHelp] "The TinyOS-help Archives",

<http://mail.millennium.berkeley.edu/pipermail/tinyos-help/>, February 2008