# Configuración del Border Router

## Requerimientos

Para la instalación inicial se necesitarán los siguientes componentes o elementos:

* Un dispositivo Border Router KTBRN1.
* Una tarjeta micro SD, de al menos 2GB y de clase 10 A1.
* Cable USB de tipo A a micro USB tipo B.
* Cable Ethernet.
* Un ordenador personal

## Guía de Instalación

Para la instalación del Software del Border Router debemos seguir los siguientes pasos.

### Descarga del Software Requerido

Deberemos descargar la imagen, basada en Debian, en su última versión para el KTBRN1. Este software incluye el software de KiBRA. Esta imagen la podemos encontrar en el siguiente enlace:

* [KTBRN1 + KiBRA image file](https://www.kirale.com/products/ktbrn1/#resources)

Una vez descargada esta imagen, descargaremos el fichero [KiBRA-v2.x.x.zip](https://www.kirale.com/products/ktbrn1/#resources) para un uso posterior.

Una vez descargada la imagen y el fichero .ZIP, necesitaremos un software para actualizar la imagen guardada en la SD o flashear una nueva. Un ejemplo de este software sería:

* [Balena Etcher](https://www.balena.io/etcher/)

Por otro lado necesitaremos una terminal Serie y/o un cliente SSH para la conexión con el dispositivo KTBRN1.

* [MobaXterm free](https://mobaxterm.mobatek.net/): Para el cliente SSH y la terminal Serie. Admite ambos tipos de sesiones a la vez, pero puede usarse cualquier otro.
* [Zadig](https://zadig.akeo.ie/): Se usará en caso de necesitar instalar los drivers de USB Serie.

### Flashear la imagen en la tarjeta SD

En caso de necesitar actualizar o flashear una imagen en una tarjeta SD, deberemos seguir las siguientes instrucciones:

* Instalar y abrir Balena Etcher.
* Seleccionar el fichero con extensión *.gz* (la imagen) en Etcher.
* Introducir la SD en el lector de tarjetas del ordenador y seleccionarla en Etcher. Recordemos que es recomendado el uso de una micro SD clase 10 de al menos 2 GB de capacidad.
* Seleccionaremos Flash y esperaremos a que termine.
* Expulsaremos la tarjeta SD y la introduciremos en la ranura para micro SD del módulo KTBRN1.

Captura de pantalla de un celular

Descripción generada automáticamente

### Primera Instalación

Conectaremos con un cable USB el dispositivo KTBRN1 al PC. La primera vez que encendamos el dispositivo, tardará unos minutos en estar listo para aceptar conexiones. **No apagar** el sistema hasta que el proceso de primera instalación haya terminado.

Una vez terminado, seguiremos los pasos descritos debajo para acceder al dispositivo KTBRN1 a través de puerto USB Serie.

#### Conexión vía puerto USB Serie

Al conectarse, deberá detectarse y listarse un nuevo dispositivo Serie (USB a Serie), dependiendo del sistema operativo del ordenador. Quizás se requiera que instalemos el driver para el puerto USB a Serie, para ello comprobaremos si nuestro ordenador lo reconoce.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Si sale como en la imagen anterior, instalaremos los driver usando la herramienta Zadig.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamenteUna vez tengamos acceso al KTBRN1 vía USB Serie, abriremos MobaXterm con una nueva sesión Serie. Seleccionaremos el puerto asignado al KTBRN1 con un BaudRate de 115200. Tras realizar esto aparecerá una consola de inició de sesión, en la cual deberemos iniciar sesión con el usuario *root* y la contraseña *kirale123*.

La pantalla de bienvenida mostrará información sobre las direcciones IP configuradas en el KTBRN1 y que versión de KiBRA está utilizando. Por defecto, la imagen instalada viene configurada con una dirección IPv4 estática para la interfaz Ethernet.

**Dirección IPv4 por defecto: 192.168.75.84/24**

Está dirección deberemos cambiarla a una dirección IPv4 que esté dentro de la red local nuestra, para poder ser visible al resto de los equipos.

Además, KTBRN1 viene con el protocolo IPv6 habilitado para la interfaz Ethernet, por lo que es posible acceder tanto a la Administración Web como a puerto SSH usando las direcciones IPv4 e IPv6.

## Panel de Administración Web

Para acceder al Panel de Administración Web, está habilitado el puerto 8000 del KTBRN1. Accederemos introduciendo <http://[IPv4]:8000> o <http://[IPv6]:8000> en el navegador (preferiblemente Google Chrome o Mozilla Firefox debido a razones de compatibilidad). Una vez introducida la dirección web, aparecerá la página de acceso / login.

*Nota: El ordenador deberá estar en la misma red que el dispositivo KTBRN1.*

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Las credenciales son las mismas a las mencionadas anteriormente. Acceder con usuario *root* y contraseña *kirale123.*

### Cambiar la configuración de red

El administrador puede querer cambiar y permitir una configuración automática de DHCPv4 o cambiar la dirección IPv4 por otra. Esto podrá realizarse accediendo al menú “Network” en la administración Web.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Como se muestra en la imagen de arriba, la nueva dirección IPv4 del DHCP es 192.168.0.102. Ahora será posible acceder por cliente SSH a esta dirección e incluso por enlace local de dirección IPv6 si el ordenador está conectado a la misma red. KiBRA también usará esta dirección IPv4 externa para su funcionalidad NAT64.

*Nota: Se necesitará un reinicio del dispositivo KTBRN1 para asegurar que se aplican correctamente las nuevas configuraciones.*

### Actualizar KiBRA

Para actualizar la versión de KiBRA, iremos al menú de KiBRA y pincharemos en el icono de “Upgrade”, que es el que se sitúa al lado de la imagen del KTBRN1 en la sección de “System”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Después seleccionaremos el fichero KiBRA-v2.x.x.zip descargado anteriormente en nuestro ordenador, pincharemos en el botón “Install” y seguiremos las instrucciones que nos aparecerán en pantalla.

### Configurar Border Router

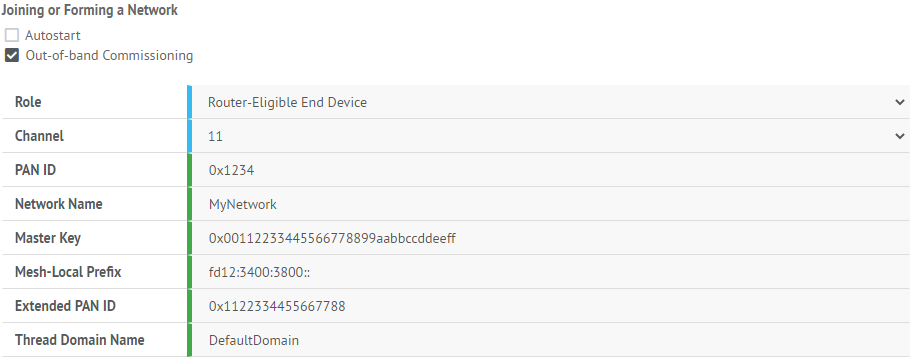
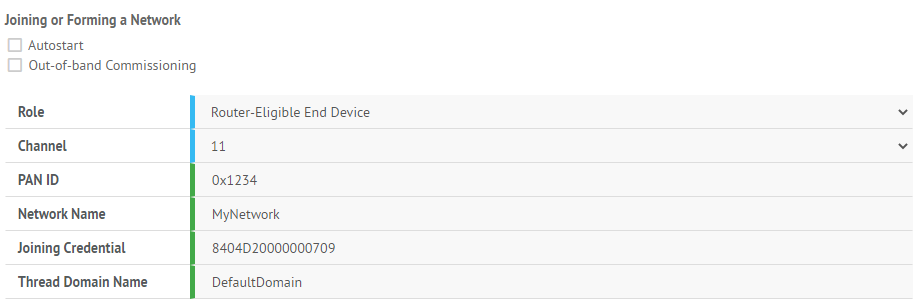
Pincharemos en el submenú “Settings” por debajo del menú KiBRA, para acceder a la página de configuración. En esta pestaña, será donde se configurará los diferentes parámetros para unirse a la red.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Unirse o Formar una Red Thread

En esta sección se mostrará la configuración para que un dispositivo pueda unirse a una red y los parámetros que necesitarán para el proceso.

1. **Autostart:** Si esta opción está activada, el BR se intentará unir a la red Thread después del siguiente reinicio.
2. **Out-of-band Commisioning:**  Permite seleccionar o desactivar este tipo de unión a la red cuando el sistema arranca.
   1. **Activando**  este modo los parámetros a configurar de la red serán los mostrados en la siguiente imagen:
   2. **Desactivando** esta opción, los parámetros a configurar serán:

#### Backbone Router Server (BBR).

Da la posibilidad de habilitar o deshabilitar la función BBR. De igual manera, el administrador de la red podrá configurar los parámetros específicos que usará el Servidor del Border Router.

#### Prefijo de Red (Network Prefix)

Esta opción activa la configuración manual del prefijo de la red, el cuál será usado en la red Thread. El usuario puede decidir entre diferentes opciones como será el direccionamiento IPv6 de los nodos dentro de la red, como DHCP o SLAAC. Si el BBR está activado, la opción de prefijo DUA será activada.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Una vez configurado el Border Router con los parámetros deseados, **“Guardar”** los cambios.

### Inicio del Border Router (Start-up).

Iremos a la pestaña “KiBRA” en el menú para encender el “Border Router Engine”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Después de clickear en el botón de Start, el Border Router se unirá a la red seleccionada o formará una nueva, según los ajustes configurados por el administrador.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Si volvemos al submenú de “Settings” (Ajustes), es posible ver el resto de parámetros de la configuración de la red Thread que han sido configurados, ya sea por el administrador o automáticamente en el arranque.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En esta pestaña, podemos coger la información necesaria para que otros dispositivos puedan unirse a la misma red. Esto se podrá gracias al botón situado en la esquina inferior derecha llamado “Export Settings”, el cuál permite copiar la información de “commissioning”, requerida para la configuración del nuevo dispositivo e introducirlo en la red, utilizando comandos KiNOS.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Servicios

El administrador de la red podrá ver que servicios están siendo provistos por el Border Router en cada momento y su estado en el submenú “Services” por debajo del menú “KiBRA”. Hay cuatro posibles servicios que el Border Router es capaz de proveer.

#### Servidor Backbone Router

Cuando la opción “Backbone Router Server” esté habilitada en el menú de “Settings”, los datos relacionados aparecerán en esta página.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### DHCP

Si la opción DHCP se activa al configurar el prefijo de red, la siguiente página mostrará una lista de los nodos que han adquirido una dirección IPv6 vía DHCP y cuál es.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

#### NAT64

Siempre que haya una dirección IPv4 configurada en la interfaz externa, esta será usada para realizar una función de NAT64 en el Border Router. La tabla de la sesión NAT se mostrará en esta pestaña.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Commissioner

El Border Router puede hacer también de comisario dentro de la red Thread. Una vez esta función está activada, el administrador de la red, puede activar la dirección de datos para permitir que se puedan unir nuevos dispositivos a la red.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

### Visual Networtk

El mapa de topología de red es un mapa que permite al administrador de la red ver el Layout físico de los dispositivos conectados. Este mapa con la topología de la red es muy útil para entender cómo se han conectado los dispositivos unos a otros y así entender las mejores técnicas para los problemas.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

### Logs

El administrador puede ver dentro de los logs del sistema en el submenú “Logs” por debajo del menú “KiBRA”. Se permite filtrar logs por nivel de gravedad/severidad y categoría.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Breve resumen

De cara a tener un mayor conocimiento sobre cómo funciona KTBRN1, a continuación se dará una detallada descripción del sistema, que herramientas hay disponibles y algunos consejos para solucionar problemas.

### Sistema de ficheros avanzado

El dispositivo KTBRN1 es un sistema basado en Linux, el cuál tiene la peculiaridad que su sistema de ficheros está corriendo desde una tarjeta SD. Esto supone un gran desafío a la hora de garantizar la fiabilidad y el rendimiento del sistema.

Las tarjetas SD son propensas a dañarse o corromperse, implicando a la perdida de ficheros almacenados y otros datos. Además, tienen una vida útil, tiempo a partir del cual pueden dañarse.

Kirale Technologies ha diseñado un avanzado sistema de ficheros para superar estos inconvenientes de manera eficaz. Este diseño monta la partición de “solo lectura” y todos los ficheros que son escritos no son realmente escritos en el disco pero permanecen en la RAM. De esta manera el sistema de ficheros no se corromperá porque al reiniciarse vuelve a tener la imagen antigua. Por otro lado, una sincronización software escribirá estos ficheros que deben de permanecer actualizados y reflejar así los cambios después del reinicio.

### Servicios críticos

Hay dos servicios críticos corriendo en el dispositivo KTBRN1. Por un lado el servicio “kibra”, el cual se encarga de todas las funcionalidades de Border Router, mientras que por otro lado está el servicio de “ajenti”, el cual gestiona el Panel de Administración Web. Ambas son aplicaciones Python instaladas en entorno virtual.

Usando comandos comunes de Linux, el administrador podrá saber el estado de ambos servicios y reiniciarlos si es necesario.

|  |
| --- |
| *root@KTBRN1:~# service kibra (status | start | stop | restart)*  *root@KTBRN1:~# service ajenti (status | start | stop | restart)* |

El administrador puede iniciar manualmente la aplicación “kibra” usando los siguientes comandos:

|  |
| --- |
| *root@KTBRN1:~# service kibra stop*  *root@KTBRN1:~# source /opt/kirale/py3env/bin/activate*  *(py3env) root@KTBRN1:~# python -m kibra –-log debug* |

En el caso de la aplicación “ajenti”, los comandos a utilizar son:

|  |
| --- |
| *root@KTBRN1:~# service ajenti stop*  *root@KTBRN1:~# source /opt/kirale/py2env/bin/activate*  *(py2env) root@KTBRN1:~# ajenti-panel –dev* |

### Comunicación entre Procesos

El Panel de Administración Web y KiBRA se comunican constantemente entre ellos a través de un puerto local TCP.

# Configuración Inicial módulo KTWM102

En este capítulo hablaremos sobre como configurar los módulos KTWM102, tanto la configuración para poder acceder a él desde el ordenador como los parámetros a configurar para poder conectar el dispositivo a una red. Esto último se explicará tanto para configurarlo a través del PC por conexión USB, como a través de un microcontrolador vía UART.

***Nota:*** *Este procedimiento es para los dispositivos KTDG102 Evaluation Dongles, los cuales añaden al KTWM102 el circuito necesario para poder conectarlos a un ordenador vía USB. En caso de usar un módulo KTWM102, se deberá diseñar un circuito con conector USB para poder conectarlo. Una vez realizado el circuito, seguir el mismo procedimiento explicado a continuación.*

## Instalación de Drivers USB y del Bootloader.

Conectar el dispositivo a un puerto USB disponible del ordenador. Si aún no hay una imagen firmware valida grabada en el dispositivo, el Led del Dongle, empezará a parpadear rápidamente. Esto indica que el dispositivo ha entrado en modo DFU y está esperando una actualización de firmware. El administrador de dispositivos mostrará el dispositivo como “*KiNOS Boot DFU”* en la pestaña de “*Otros dispositivos*”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

### Windows

Los sistemas Windows requieren instalar manualmente los drivers USB para los KTDG102 (solo será necesario la primera vez). En algunos casos, el dispositivo KiNOS Boot DFU puede instalarse automáticamente con los drivers genéricos de Windows, pero se necesitará reemplazarlos.

Para instalar o reemplazar los drivers, necesitaremos una herramienta gratuita llamada **Zadig**. Esta herramienta puede descargarse de la página de Zadig <https://zadig.akeo.ie/>. Esta aplicación es para instalar una “libusb” compatible con el dispositivo.

Abrir Zadig (no necesita de instalación). En caso de que salte el aviso de una ventana UAC (User Account Control), seleccionar “Yes” o “Sí”.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez esté Zadig corriendo, deberán aparecer las interfaces KiNOS en la lista desplegable. Es posible conectar el dispositivo incluso después de haber abierto Zadig, la lista se actualizará automáticamente. En caso de que no aparezca, probablemente sea que ya haya algún driver instalado. Para verlo, ir al menú de “***Options***” y seleccionar “***List All Devices***”.

Seleccionar KiNOS Boot DFU en la lista despegable, y seleccionar el driver “liusbK” y pinchar en “***Install / Replace Driver***”.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

El proceso tomará alrededor de un segundo y el resultado será un mensaje de éxito:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora, en el *Administrador de Dispositivos*, el KiNOS Boot DFU deberá salir de la siguiente manera:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Linux / Mac OS

No se necesita instalación de ningún driver específico para sistemas basados en Linux con versión de Kernel superior a 2.6.22 ni para sistemas MAC OS X desde la versión 10.4 (Tiger).

## Install “dfu-util”

El protocolo usado para cargar el Firmware a los dispositivos Kirale KTDG USB Dongle a través de interfaz USB es el estándar **DFU 1.1.** Descargar “dfu-util” e instalarlo.

1. En Windos: descargar de <http://dfu-util.sourceforge.net/> y extraerlo en la carpeta deseada.
2. En MAC OS:

$ brew install dfu-util

([Get Brew](https://brew.sh/))

1. En Linux:

$ sudo apt install dfu-util

## Actualización de Firmware

Abrir dfu-util desde la ventana de comandos y listar los dispositivos conectados para encontrar el dispositivo deseado. El USB Product ID par un dispositivo KiNOS DFU en modo bootloader es **0000**.

|  |
| --- |
| $ dfu-util –-list  Found DFU: [2def:0000] ver=0100, devnum=8, cfg=1, intf=0, path=”1-1.4.3″, alt=0, name=”KiNOS DFU”, serial=”8404D2000000045B”  Found Runtime: [2def:0102] ver=0100, devnum=9, cfg=1, intf=0, path=”1-1.4.4″, alt=0, name=”KiNOS DFU”, serial=”8404D2000000045C” |

Flashear el fichero del firmware al dispositivo deseado (especificando el número de serie). Esta transferencia del archivo puede tardar varios segundos.

|  |
| --- |
| $ dfu-util --download KiNOS-GEN-KTWM102-1.1.6533.62822.dfu --serial 8404D2000000045B  Match vendor ID from file: 2def  Match product ID from file: 0000  Opening DFU capable USB device...  ID 2def:0000  Run-time device DFU version 0110  Claiming USB DFU Interface...  Setting Alternate Setting #0 ...  Determining device status: state = dfuIDLE, status = 0  dfuIDLE, continuing  DFU mode device DFU version 0110  Device returned transfer size 64  Copying data from PC to DFU device  Download [=========================] 100% 245628 bytes  Download done.  state(6) = dfuMANIFEST-SYNC, status(0) = No error condition is present  unable to read DFU status after completion |

Una vez dfu-util ha terminado la transferencia del firmware, el KTDG USB Dongle se reiniciará y empezará a aplicar el nuevo firmware en la memoria flash interna (parpadeo rápido del led). Esto puede tardar varios segundos. Cuando el led empiece a parpadear lentamente, de manera estable, el flasheo del firmware ha terminado y el firmware KiNOS empieza a operar en el modo de runtime.

El fichero DFU puede conseguirse desde el siguiente link:

<https://www.kirale.com/support/#downloads>

## Runtime – Instalación de drivers USB.

En modo run-time el KTDG102 Dongle es un dispositivo USB Compuesto que combina tres tipos de interfaces USB:

* Device Firmware Upgrade (DFU).
* Virtual Serial (CDC – ACM).
* Ethernet over USB (CDC-ECM).

Windows no soporta el modelo USB-ECM de manera nativa, por lo que se requiere un driver de terceros que está fuera del alcance de Kirale Technologies.

Para los otros dos interfaces USB, se necesitará la instalación de los drivers para sistemas Windows siguiendo las siguientes instrucciones:

### Windows

En algunos casos el dispositivo KiNOS DFU puede instalar automáticamente un driver genérico de Windows y aparecerá por debajo de “Virtual COM Ports”. En caso de haberse instalado el driver genérico, se necesitará reemplazar. Se usará Zadig para instalar o reemplazar los drivers USB.

Seleccionar *KiNOS DFU (Interface 0)*  en la lista despegable, seleccionar el driver “*libusbK”* y seleccionar “*Install/Replace Driver”*.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El proceso tardará alrededor de un segundo y saldrá un mensaje de éxito.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Después, se selecciona *KiNOS Virtual COM (Interface 1)* en la lista despegable, seleccionar el dirver “***USB Serial (CDC)”*** y pinchar “*Install/Replace Driver”*.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El proceso tardará alrededor de un segundo y saldrá un mensaje de éxito.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora, en *Administrador de Dispositivos*, la interfaz KiNOS debería aparecer de la siguiente manera:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Linux

No se necesita ninguna instalación de un driver en específico para sistemas basados en Linux con versiones de Kernel superiores a 2.6.

***Nota:*** *La interfaz USB CDC-ECM está deshabilitada por defecto. Puede habilitarse por línea de comandos. Para más información detallada, mirar la* ***KSH Reference Guide.***

Para encontrar Puertos de Serie e interfaces Ethernet, conectar el Dongle a un USB del ordenador y después abrir una terminal e introducir:

|  |
| --- |
| ~$ dmesg | tail |

### Mac OS

No se necesita ninguna instalación de un driver en específico para sistemas Mac OS X desde la versión 10.4 (Tiger).

***Nota:*** *La interfaz USB CDC-ECM está deshabilitada por defecto. Puede habilitarse por línea de comandos. Para más información detallada, mirar la* ***KSH Reference Guide.***

Para encontrar Puertos de Serie e interfaces Ethernet, conectar el Dongle a un USB del ordenador y después abrir una terminal e introducir:

|  |
| --- |
| ~$ networksetup –listallhardwareports |

## Configuración de Terminal COM

### Windows

Hay gran variedad de terminales serie COM disponibles para sistemas Windows. En esta guía se usa una genérica llamada “Termite”, pero valdría cualquier otra terminal.

Abrir la terminal serie y configurar con las siguientes configuraciones para comunicaciones serie con dispositivos de Kirale:

**Configuración Estándar para puerto serie USB: 9600 bauds, 8 bits, 1stop bit, no parity.**

***Nota:*** *La terminal USB serie, debe configurarse para añadir un carácter “CR” cuando se pulsa la tecla Enter.*

Una vez configurado, seleccionar el puerto serie que aparece en el *Administrador de dispositivos* para el KiNOS Virtual COM. Después, para testear si la interfaz de USB de Kirale está activa y corriendo, presionar “Enter” y deberá devolverse el indicador de KiNOS.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

***Nota:*** *Recomendado la activación del Echo Local para la lectura de los comandos enviados.*

### Linux / Mac Os

Si no se tiene ninguna terminal serie para puerto serie COM, se deberá instalar una, como puede ser “*Picocom”.*

La configuración es la misma necesaria para la comunicación serie con los Sistemas Windows.

|  |
| --- |
| ~$ picocom -c --omap lfcr /dev/ttyACM0 |

-c 🡪 Activación Local Echo.

--omap lfcr 🡪 Asignar el avance de línea de salida al retorno de carro

/dev/ttyACM0 🡪 El dispositivo serie asignado por Linux. Usar ***dmesg* | *tail*** cuando se haya conectado el dispositivo para comprobar el nombre del dispositivo.

Para irse del programa utilizar ***Ctrl+a*** *y* ***Ctrl+x*** .

# Configuración de red del módulo KTWM102

A la hora de enviar comandos tanto para configurar los parámetros del módulo KTWM102 como para ver los parámetros ya configurados tenemos dos posibles vías para ello.

1. La primera vía es comandos KSH, los cuáles son con una sintaxis amigable para una ejecución manual y sencilla para el usuario desde un PC. Se conecta el módulo por USB.
2. La segunda vía es por comandos KBI. Estos comandos se enviarán por vía UART al módulo.

## Kirale Command-Line Shell Refence Guide – Comandos KSH

### Sintaxis de los comandos

Los comandos KSH tienen una sintaxis amigable para las personas. Esta sintaxis está basada en palabras ASCII separadas, las cuales especifican cada una las palabras clave o parámetros del comando. Se representan de la siguiente manera:

|  |
| --- |
| kinos@local:~$ command < arg > < key > < subkey > [ param 1 ] … [ param N ] |

Donde **Command** es el nombre del comando a ejecutarse de primer nivel. Este puede tener un argumento a ejecutar a bajo nivel e incluso claves y subclaves y otros parámetros que sean datos variables requeridos para la correcta ejecución del comando.

Para el envío de estos comandos se necesitará descargar la herramienta KiTools a su última versión de la siguiente dirección:

<https://www.kirale.com/support/#downloads>

Una vez descargado, conectar el módulo KTWM102 al PC por conexión USB abrir el ejecutable descargado y se abrirá una ventana de comandos similar a la de Windows.

Texto

Descripción generada automáticamente

Para ver más en detalle la sintaxis de cada comando, descargar y ver la [KSH Reference Guide](https://www.kirale.com/products/ktwm102/#083596d3193c172fa).

### Sintaxis de los parámetros

Hay seis tipos de formatos posibles de parámetros:

* **Hexadecimal:** Representado en la guía como [hex]. Estos parámetros deberán ser completamente en notación hexadecimal, como “*0xABCD”* o *“0xabcd”*, siempre con el prefijo “*0x”*.
* **Decimal:** Representado en la guía como [dec]. Estos parámetros deberán ser con notación decimal, como “*64”*.
* **Cadena de caracteres:** Representado en la guía como [str]. Estos parámetros deberán ir entre comillas dobles y pueden estar limitados en tamaño. Este límite de tamaño está especificado como número de caracteres admitidos.
* **Dirección IP:** Este parámetro se usa para especificar direcciones IPv6, rutas, o prefijos y debe ir con el formato especificado en la  **RFC 4291 section 2** (16 bytes).

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* **Direcciones MAC:** Representado en la guía como [mac]. Este parámetro se usa para especificar direcciones de 64-bits.
* **Cadena / Array:** Se representa en la guía como [arr]. Este parámetro se usa para especificar payloads que deben enviarse sin ningún cambio. Están hechos de cadena de bytes en notación hexadecimal sin el prefijo “0x”.

### Mensajes de Respuesta

Cuando se ejecuta un comando en la Kirale Command-Line Shell, se pueden reportar algunos errores como respuesta a su ejecución.

1. ***Invalid Syntax***: Tanto el comando como el argumento son reconocidos pero alguna de las palabras claves o subclaves son erróneas, o hay más parámetros de lo esperado para ese comando.
2. ***Command not found:*** El comando o el argumento no existe.
3. ***Bad parameter:*** Alguno de los parámetros que se han introducido es erróneo o su valor está fuera de rango.
4. ***Command not allowed:*** El comando no puede ejecutarse en ese momento. Quizás se requiera que el dispositivo esté en un específico estado u otra configuración no permita la ejecución del comando.
5. ***Configuration setting missing:*** El comando no puede ejecutarse debido a que se requieren otras configuraciones antes de ejecutarse.
6. ***Processing – please wait:*** El dispositivo está ejecutando otro proceso de mayor prioridad. Solo se podrá usar el comando “***show status***”. Otros comandos no podrán usarse hasta que el proceso prioritario termine.

Si no hay mensaje de respuesta, significa que el comando se ha procesado correctamente.

## Kirale Binary Interface Refence Guide – Comandos KBI

### Operación de interfaz

Antes de entrar en el funcionamiento de como usar los comandos KBI por puerto UART, en la siguiente imagen se muestra un esquema de un Host externo usando KBI para comunicarse con el módulo KTWM102 y el sistema KiNOS a través del puerto UART:

Diagrama

Descripción generada automáticamente

Una vez conectado el sistema como en el esquema, deberemos configurar el puerto serie de nuestro Host de la siguiente manera:

***Configuración Serie:*** *115200 bps, 8 data bits, no parity, 1 stop bit, no flow control.*

***Nota:*** Comprobar que los pines de comunicación serie cumplen con las características eléctricas.

Como se detalla en la imagen, entre la interfaz UART y el KBI hay un nivel intermedio de codificación/decodificación. Esto es necesario para transmitir un paquete con un carácter delimitador de inicio ***0x00***. Cada vez que se transmite un delimitador, se indica el comienzo de un nuevo paquete. Por esta razón, otros caracteres 0x00 del propio paquete necesitarán ser codificados para no confundirlos con un nuevo delimitador. Si se detecta un error en la recepción del paquete UART y el recibidor es incapaz de decodificar el mensaje, este enviará una notificación de error (codificada como [0x00 0xFF]) automáticamente al remitente.

Este procedimiento es llevado a cabo por el sistema de codificación ***Consisten Overhead Byte Stuffing (COBS)*** implementado en KiNOS. Este sistema se explica con más detalle en la [KBI Reference Guide](https://www.kirale.com/products/ktwm102/#083596d3193c172fa) y en [draft-ietf-pppext-cobs-00](https://datatracker.ietf.org/doc/html/draft-ietf-pppext-cobs-00). Por lo que el Host externo deberá implementar este sistema de codificación COBS para poder interactuar con KiNOS vía UART.

### Formato del paquete

En esta sección se explicará el formato de los paquetes a transmitir o a recibir desde un host externo sin la codificación COBS. Este formato es el siguiente:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Header | | | | | Payload |
| L0 | L1 | TYPE | CMD | CKS | (Optional ≤ 1268 Bytes) |

En primer lugar, hay una cabecera de cinco bytes de largo donde:

* **L0:** El byte más significativo del largo de la payload.
* **L1:** El byte menos significativo del largo de la payload.
* **TYPE:** Descriptor de tipo.
* **CMD:** Descriptor de Comando
* **CKS:** Byte de Checksum. Se calcula con el XOR de todo el resto de bytes del paquete.

Después de la cabecera puede haber una payload de tamaño variable, el cuál puede presentarse como información opcional para un comando o respuesta específico. El tamaño máximo de la payload es de 1268 bytes. Todos los subcampos de la payload deben ser big endian.

El byte **TYPE** tiene los siguientes bits:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TYPE (1 byte) | | | | | | | |
| FT | FT | FT | FT | FC | FC | FC | FC |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **FT: FRAME TYPE** | | | | |
| **FC: FRAME CODE** | **0 -Reserved** | **1 - Command** | **2 - Response** | **3 - Notification** | **4 to 15 - Reserved** |
| **0** | - | Write / Execute | OK | Ping Reply | - |
| **1** | - | Read | Value | Socket Received Data | - |
| **2** | - | Delete | Bad parameter | Named Ping Reply | - |
| **3** | - | - | Bad Command | Named Socket Received Data | - |
| **4** | - | - | Operation not allowed | Destination Unreachable | - |
| **5** | - | - | Memory Allocation Error | - | - |
| **6** | - | - | Config. Settings Misssing | - | - |
| **7** | - | - | Firmware Update Error | - | - |
| **8** | - | - | Busy | - | - |
| **9 to 15** | - | - | - | - | - |

### Representación de datos

La payload de un comando, la respuesta o una notificación, puede consistir en uno o más parámetros separados, cada uno con diferente representación. Los principales tipos son:

* **HEX (n):** Valor hexadecimal genérico de un tamaño variable de hasta n bytes.
* **HEXN (n):** Valor hexadecimal genérico de un tamaño fijo de n bytes.
* **DEC (n):** Valor decimal (entero sin signo) representado en n bytes (tamaño fijo).
* **ENU:** Caso específico de DEC(1) en el cual solo estarán permitidos una enumeración de valores definidos (diferentes para cada comando).
* **STR (a, b):** Cadena ASCII de un tamaño fijado entre a y b caracteres. Se omite el EOS para comandos y se incluye para respuestas o notificaciones.
* **STRN (n):** Cadena ASCII con un tamaño fijado de n caracteres, incluido EOS. En caso de que la cadena sea más corta de lo requerido, está permitido añadir más bytes EOS como padding.
* **MAC:** Caso específico de HEXN(8), representando un identificador de interfaz (dirección MAC), con un tamaño de 8 bytes.
* **ADDR (n):** Caso específico de HEXN(8) o HEXN(16), representando un prefijo IPv6 de tamaño 64 bits o una dirección IPv6 con 128 bits de tamaño.
* **LIST (X):** Se usa para indicar que la payload consiste en la repetición de cierto patrón.

### Comandos y Respuestas

Los dispositivos KiNOS pueden realizar dos tipos de comunicaciones serie vía UART:

1. La primera es la repuesta ante la recepción de un comando desde un host externo y su posterior ejecución. Esto implica, que cada comando recibido, genera una respuesta y esta es transmitida de vuelta al host.
2. La segunda opción de transmisión desde el dispositivo KiNOS son las notificaciones, o mensajes para informar de eventos asíncronos. (Ver [3.2.5. Notificaciones](#_Notificaciones))

Será necesaria una rutina “*Comando – Esperar respuesta”* en el host externo cuando se transmite un comando al dispositivo KiNOS. Esto implica que el host debe esperar a una respuesta justo después cada comando enviado al dispositivo KiNOS. Se recomienda que en el lado del host haya una coherencia entre el comando enviado y la respuesta recibida.

En algunos casos, puede pasar que el host externo reciba una notificación asíncrona mientras está esperando una respuesta a un comando, por lo que el host tendrá que decidir si guardarla y procesarla después o rechazarla y continuar esperando la respuesta del comando.

Un ejemplo de esta rutina de “Comando – Esperar respuesta” sería:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Notificaciones

Un dispositivo KiNOS manda una notificación a un host externo para informar de eventos asíncronos. Estos eventos asíncronos pueden ser eventos como la recepción de datos por trafico serie por UDP, la cual puede darse en cualquier momento por radio y se transmite la notificación por UART.

En estas notificaciones, el byte de CMD tiene valor nulo, 0x00.

Para más detalle de los comandos ver en la [KBI Reference Guide](https://www.kirale.com/products/ktwm102/#083596d3193c172fa)

## Configuración de red

Una vez explicadas las diferentes maneras de enviar los comandos al módulo KTWM102, se procederá a explicar el procedimiento para poder configurar los parámetros para que el nodo se una a la red correctamente.

Antes de introducir los parámetros de la red, deberemos tener en cuenta una cosa: si deseamos o no realizar el proceso de “In Band” o no. Si queremos unirnos a una red en específico, deberemos activar el modo de Commisioning “Out of Band”, y posteriormente introducir todos los parámetros necesarios.

Para ver la sintaxis de los comandos a enviar para las diferentes configuraciones, ver las guías KBH y KBI anteriormente mencionadas.

***Nota:*** *En ambos casos, siempre se debe ejecutar el comando* ***Clear*** *para borrar posibles configuraciones anteriores. Solo es conveniente* ***no***  *ejecutarlo cuando la configuración guardada es la misma que se vaya a utilizar en ese momento.*

### Modo Out-of-Band Commissioning Desactivado

Este modo es el modo por defecto, pero aún así es recomendable ejecutar el comando de desactivar el modo “Out-of-Band Commisioning”.

Una vez se aseguré que este modo esté desactivado, la configuración requerirá de pocos parámetros obligatorios.

Antes de la ejecución del comando **Ifup**, deberá configurarse el Role que tendrá el nodo. En caso contrario, se generará el error “*Configuration settings missing”***.** Otros parámetros como el *Canal, PAN ID, Nombre de Red y la Credencial de Commisioning,* serán opcionales, y se generarán automáticamente en caso de no especificarles.

Una vez enviado el comando IfUp, el módulo tardará unos pocos segundos el realizar el proceso de unirse/crear la red. Una vez unido a la red, se guardará su configuración como configuración válida y ya no podrá modificarse salvo realizar el comando Clear.

### Modo Out-of-Band Commissioning Activado

Este modo no está activado por defecto, por lo que se deberá ejecutar el respectivo comando de activación del modo “Out-of-Band Commissioning”. Este modo convendrá activarlo cuando se quiera unir los módulos a una red en concreto, como puede ser una red “privada”.

En este modo se deberá configurar más parámetros para poder especificar correctamente la red a la que se quiera unir o que se quiera crear. Los parámetros a configurar son : *Rol, Canal, PAN ID, Nombre de Red, Prefijo Local, Master Key, PAN ID Extendida y la Credencial de Commisioning.*

Sin uno de esos parámetros, se generará el error “*Configuration settings missing”***.**

# Pruebas Realizadas

En este capítulo describiremos las diferentes pruebas realizadas. En estas pruebas se han utilizado los siguientes utensilios:

|  |  |
| --- | --- |
|  |  |
| 1 x Kit de Desarrollo STM  STM32F407G-DISC1 | 2 x KTDG102 Evaluation Dongle |
|  |  |
| 1 x KTWM102 integrado en la PCB  Coockie Thread | 1 x Border Router |

## Primera interacción con Dongle USB

Imagen que contiene circuito

Descripción generada automáticamenteEsta primera prueba consiste en una primera interacción con el módulo, tanto desde el PC usando la herramienta proporcionada por Kirale de KiTools y usando el puerto USB, como desde un microcontrolador por vía UART. El diagrama de conexión usado ha sido el siguiente:

Esta primera prueba se divide en dos partes:

1. La primera parte se ejecutan los diferentes comandos desde el PC usando la herramienta KiTools con los comandos KSH, comprobando que se ejecutan correctamente según lo esperado.
2. Una vez comprobado los comandos KSH, se usa el microcontrolador para el envío de comandos KBI vía puerto UART. Para la comprobación del correcto envío se usan dos maneras:
   1. La primera es la activación de los modos de debug del módulo en la herramienta KiTools en el PC, pidiéndole al módulo que informe de la actividad en las diferentes capas. Cuando le lleguen mensajes y realice las respectivas respuestas, saltará un log de mensaje por rx y por tx con los bytes recibidos/transmitidos. A su vez, si tenemos un error de Checksum, esta herramienta nos avisará del error.
   2. Una vez enviada el comando KBI, recogemos la respuesta enviada por el módulo KTDG102 y analizamos su respuesta.

Tras hacerse al envío de comandos de manera correcta, se elabora la secuencia de comandos para la configuración completa de la red para ese módulo KTDG102 con el Rol de LEADER.

## Red de dos Nodos

El siguiente paso ha consistido en la creación de una red con un nodo LEADER y la unión de un nodo MED a dicha red. Para esto se han usado los dos KTDG Evaluation Dongles conectados al PC con la herramienta KiTools por USB y al microcontrolador ARM por vía UART.

Imagen de la pantalla de un celular con texto e imágenes

Descripción generada automáticamente con confianza mediaSe han realizado varias pruebas para las cuales el esquema de montaje ha sido el siguiente:

### Creación de la Red

La primera prueba realizada ha sido la creación correcta de la red (de la manera vista en [Primera interacción con Dongle USB](#_Primera_interacción_con) ) y la posterior unión del segundo nodo con el rol de MED. La red, para ambos nodos se ha realizado con el modo “Out-of-Band Commissioning” desactivado, pero configurando en ambos casos el PAN ID, el Canal y la Commisioning Credential.

Para esto, se ha usado la secuencia de creación de red con Rol Leader usada anteriormente para el primer nodo, y una vez creada la red, se volvía a usar esta secuencia modificando el Rol a configurar en el nodo para el segundo nodo.

Tras ejecutar el primer nodo su secuencia el comando IfUp, se debe dejar un tiempo de alrededor a 7 segundos para que pueda crearse la red definida. A su vez, se comprueba que debe dejarse entre 7 y 9 segundos en el nodo MED después del comando IfUp para que este se una a la red, ya que al ser en modo “Out-of-Band Commissioning” desactivado, y no tener todos los parámetros, este nodo buscará una red de entre las cercanas que tenga los mismos parámetros a los configurados. Cuando este encuentre la red, saltará un aviso en el nodo LEADER en la herramienta de KiTools, de que se está uniendo un nodo.

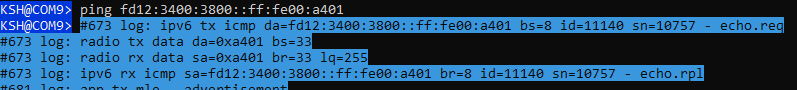
Por otro lado, cuando se ejecutaban los comandos y se enviaba respuesta al microcontrolador, este las imprimía por pantalla en el PC, pudiendo así detectar posibles fallos en alguno de los comandos para poder arreglarlos.

### Ping entre nodos

Una vez tenemos los nodos en la misma red, una primera comprobación de la comunicación entre ambos módulos. En este caso, al ser una prueba rápida y sencilla de comprobación, se ha usado la herramienta KiTools, abriendo dos sesiones, una por cada módulo. Una vez en red, se ejecuta el comando *show netconfig* desde una de las dos sesiones de KiTools. Con esto podemos ver la dirección IP de los módulos.

Tras ver la dirección de uno de los módulos Dongle, se ejecutará el comando *ping <arg>* en el módulo KTDG102 restante, siendo el argumento una de las direcciones IP que hemos visto del otro módulo KTDG102.

En el lado que se realiza el ping saldrán los siguientes Logs en KiTools:



En cambio en el KiTools del lado que recibimos el ping, saldrán los siguientes Logs:

Texto

Descripción generada automáticamente con confianza media

### Envío de mensajes UDP a través de Sockets entre ambos nodos

Una vez se ha comprobado que los dos nodos han tenido una primera comunicación básica mediante el comando ping, el siguiente paso es la prueba del envío de mensajes entre ellos vía Sockets UDP.

Antes de poder enviar y/o recibir un socket, se deberán tener en cuenta dos cosas:

1. A priori, no se puede saber la dirección IP que tendrá el nodo una vez conectado a la red. Por lo que convendrá ejecutar el comando de asignación de IP (en nuestro código llamado como WriteIP) con una dirección IP a elegida arbitrariamente. Este comando debe realizarse una vez se haya unido el nodo a la red. Esto convendrá de cara al posterior envío de los mensajes.
2. En esta prueba, al no haber un nodo router como tal, deberá ejecutarse desde el nodo LEADER el comando ROUTE con dirección al nodo MED. Este creará el enlace para los propios mensajes. Debe ejecutarse antes del envío de los mensajes. Como se ha visto en la prueba [Ping entre nodos](#_Ping_entre_nodos), en este caso no hizo falta realizar el comando route, pero a la hora del envío de sockets no se consiguió recibir los mensajes sin este comando previo.

Para enviar un mensaje a través de sockets UDP, se debe:

* Abrir un socket en un puerto determinado (por cada nodo). El puerto asociado al socket puede ser diferente en cada nodo.
* Enviar mensaje a través del puerto asignado al socket del propio nodo, al puerto y dirección IPv6 del nodo receptor.

Para recibir un mensaje a través de sockets UDP, bastará con tan solo abrir un socket en un puerto determinado, en caso de no haberlo abierto para un anterior envío.

Para la lectura de los mensajes, se recogerá la notificación enviada por puerto UART al microcontrolador, ya que dispondremos de la información del remitente y del mensaje recibido, mientras que por la herramienta KiTools solo dispondremos de la dirección de origen o de destino del mensaje, según veamos la sesión de un nodo u otro.