

Tecnologías Celulares de Internet de las Cosas (IoT)

1 Laboratorios

Los laboratorios de la materia son demostrativos. Cada uno cuenta con una guía detallada para obtener los resultados que permitirán caracterizar ciertos parámetros de la capa física a partir de los cuales se podrá determinar la calidad de servicio (QoS) del operador.

Si bien cada laboratorio requiere de la programación adecuada de los kits de desarrollo a través de secuencias de comandos AT, no es necesario tener conocimiento previo en estos temas para desarrollar cada guía.

La materia cuenta con un total de cuatro laboratorios:

1. **Laboratorio 1:** Consiste en medir el desempeño del enlace NB a través de la herramienta 'Iperf3'. Mediante una conexión cliente – servidor configurada adecuadamente, la prueba permite realizar la medición de velocidad. La ubicación del cliente y el servidor determinarán el enlace analizado (uplink, downlink, ambos).
2. **Laboratorio 2:** Dado que NB-IoT está orientado a comunicaciones esporádicas, el retardo es mucho mayor que el existente en otras tecnologías, como por ejemplo LTE Cat-M1. Esta prueba consiste en determinar el retardo de transmisión de la red a través del envío de paquetes de datos de manera de poder establecer el retardo relativo existente. Siempre que sea posible, esta prueba se realizará tanto en uplink como en downlink.
3. **Laboratorio 3:** Sirve para determinar la distancia máxima de recepción para el ambiente en cuestión. En esta prueba se busca emular la máxima separación que puede existir entre el modem y la EB a través del uso de un atenuador variable con el cual se controla la potencia recibida en el modem. Este ensayo está íntimamente ligado con la prueba de consumo, dado que, a mayor distancia entre EB y modem, mayor será la potencia y número de repeticiones que deberá utilizar este último para lograr una transmisión exitosa. Asimismo, a través de la medición de los niveles potencia recibida, tanto RSRP (Reference Signal Received Power) como RSSI (Received Signal Strength Indicator) se podrán determinar parámetros de capa física como lo son el RSRQ (Reference Signal Received Quality) y el SINR (Signal-to-Interference and Noise Ratio).
4. **Laboratorio 4:** Esta prueba permite determinar la energía consumida por el modem tanto en el modo aislado como en el modo conectado. A partir de este ensayo, se podrá inferir la autonomía esperable en el caso de una aplicación alimentada por baterías como también el intervalo de transmisión mínimo en el caso de una aplicación alimentada por cosecha de energía. Se analizará la influencia del modo de funcionamiento tanto en la velocidad de transmisión como en la latencia. Por último, con la ayuda de un osciloscopio, se dispondrá de un esquema de medición que permita visualizar la gráfica del consumo en función del tiempo tratando de identificar las fases involucradas en la comunicación.

2 Descripción general de los módulos NB-IoT utilizados

La tecnología Narrow Band-Internet of Things (NB-IoT) es un estándar de comunicación inalámbrica de área extensa y baja potencia (LPWA, por sus siglas en inglés) que se caracteriza por utilizar dispositivos de muy bajo consumo y baja tasa de transmisión de datos. Hoy en día, es posible encontrar en el mercado un gran número de dispositivos multi-estándar que manejan esta tecnología y algunas otras también destinadas al despliegue de redes LPWA, como LTE-M, LoRa o Sigfox. Las aplicaciones donde podría utilizarse tecnología LPWA son: monitoreo remoto de pacientes, medición remota de sensores, control de iluminación inteligente, gestión de residuos y muchas más.

Para los laboratorios de la materia se podrá utilizar tanto el módulo Arduino MKR NB 1500 como el Quectel BG96. El primero permite conectividad Cat NB1 y LTE Cat M1, mientras que el segundo permite conectividad EGPRS, LTE Cat M1 y Cat NB1.

Para utilizar los módulos, es necesario contar con una tarjeta SIM habilitada con el servicio. También es necesario conectar una antena compatible con la banda a utilizar. En nuestro caso, utilizaremos una antena que cubre la banda 28 de LTE, ya que esta es la banda de frecuencia licenciada en Argentina para el servicio NB-IoT.

2.1 Módulo Arduino MKR NB 1500

El módulo Arduino ofrece gran cantidad de documentación y también facilidad de programación (se puede utilizar el mismo IDE y las mismas bibliotecas que se emplean para programar a otros dispositivos Arduino). La foto del módulo MKR NB 1500 se muestra en la Figura 2.1.

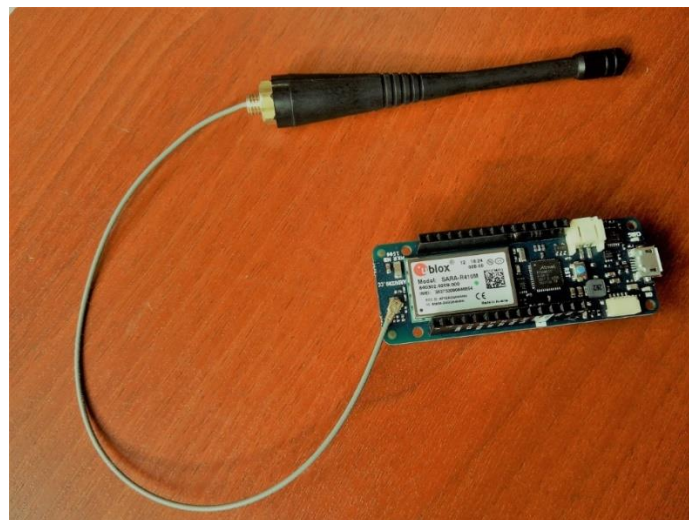


Figura 2.1. Arduino MKR NB 1500.

2.1.1 Descripción general

El MKR NB 1500 basa su conectividad de RF en un bloque transceptor U-Blox de la serie SARA-R4, concretamente, en la versión SARA-R410M-02B-00 que fue desarrollado para uso global, es decir, soporta todas las bandas. Como características principales se destacan los modos de ahorro de energía, la posibilidad de habilitar la recepción discontinua extendida y el manejo de varios protocolos que facilitan la programación y comunicación (TCP, UDP, HTTP, FTP, MQTT). Un resumen de las características del módulo se presenta en las Figuras 2.2 y 2.3.

	Rel. 13 Cat M1	Rel. 13 NB-IoT
Deployment	In-band LTE	In-band, guard band LTE, and standalone
Coverage	155.7 dB	164 dB
Downlink	OFDMA, 15 kHz tone spacing, Turbo code, 16 QAM, 1 Rx	OFDMA, 15 kHz tone spacing, TBCC, 1 Rx
Uplink	SC-FDMA 15 kHz tone spacing, Turbo code, 16QAM	Single tone, 15 kHz and 3.75 kHz spacing SC-FDMA, 15 kHz tone spacing Turbo code
Bandwidth	1.08 MHz (6 resource blocks)	180 kHz (1 resource block)
Duplexing	FD, HD (Type-B), FDD/TDD	HD only (Type-B), FDD-only (TDD for future releases)
Peak rate (UL/DL)	1 Mb/s DL and UL	DL – ~32 kb/s (in-band), 34 kb/s (standalone) UL – ~66 kb/s multitone, ~17 kb/s single tone
Power saving	PSM, eDRX, DRX	PSM, eDRX, DRX
Power class	Class 3 (23 dBm)	Class 3 (23 dBm)
Core network	Same as LTE	Simplified core network(optional), small data enhancements (data over NAS or user plane with security context in RAN) (additional non-IP data support)

Figura 2.2. Tecnologías de comunicación habilitadas en el módulo.

Microcontroller	SAMD21	Digital I/O Pins	22
Architecture	ARM Cortex-M0+ 32bit	Interfaces	I2C, I2S, SPI, UART
Operating Voltage	3.3V	PWM Output	12
Flash Memory	256 KB	Analog I/O Pins	7/1
SRAM	32 KB	Power Consumption	93 mA,
Clock Speed	32.768 kHz (RTC), 48 MHz		30mA (low power)
DC Current per I/O Pin	3 mA (I/O Pins)		

Figura 2.3. Características del módulo.

La alimentación del kit puede efectuarse de varias maneras:

1. Conectando una batería externa en el conector correspondiente.
2. Alimentando el dispositivo a través del puerto USB.
3. Alimentando el dispositivo a través de los pines VIN y GND.

La primera opción requiere que se conecte una batería de LiPo de 3.7 V al conector JST de la placa. La mínima capacidad recomendada para la batería es de 700 mAh. **Utilizar baterías de menor capacidad podría producir sobrecalentamiento y riesgo de explosión.**

La segunda opción es útil cuando se deben hacer pruebas sencillas en el kit, como por ejemplo controlar las interfaces I/O o enviar comandos AT al bloque SARA, siempre que no involucre la transmisión por RF. Esto se debe a que el kit produce picos de consumo durante la transmisión de RF que en general no son soportados por los puertos USB de las PC. La alimentación por USB también permite efectuar la carga de la batería.

La tercera opción es la que debe utilizarse cuando no se posee batería. La fuente debe ser de 5 V regulados.

2.1.2 Programación del dispositivo MKR NB 1500

El dispositivo se programa a través del puerto USB utilizando el IDE de Arduino (ver Figura 2.4). Para que el kit sea reconocido es necesario elegir el puerto de conexión correcto y tener instalada previamente la biblioteca MKRNB. Este procedimiento, al igual que la programación del kit es similar al que se realiza con otros productos Arduino, por lo que sólo se pondrá énfasis en los detalles de interés.

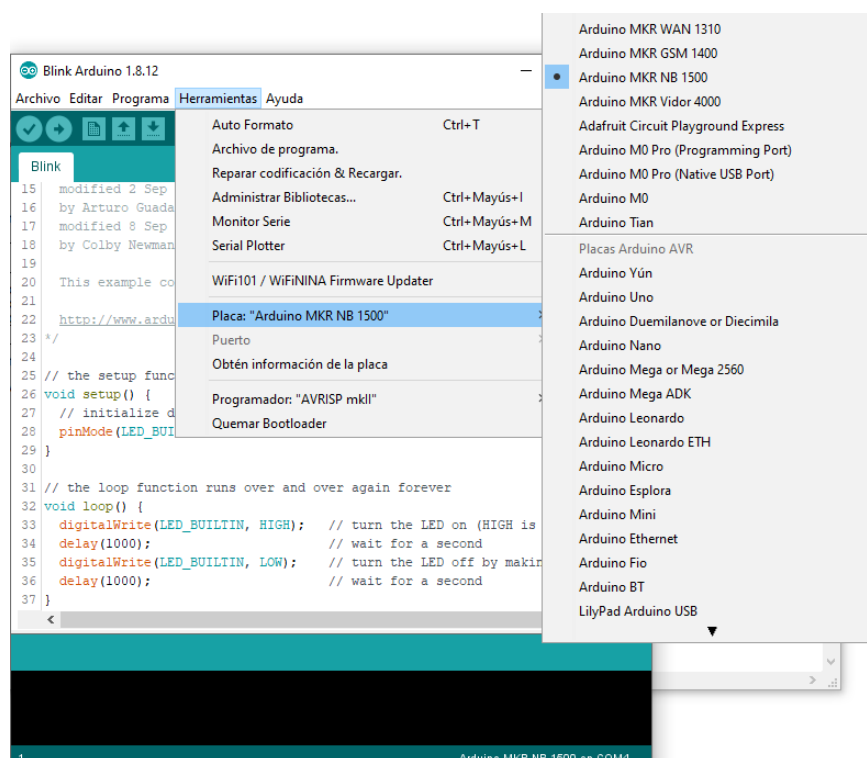


Figura 2.4. Ambiente de desarrollo integrado (IDE) de Arduino.

Según la metodología que utilicemos para programar del kit, podemos encontrarnos con escenarios diferentes:

Escenario 1: Una vez programado, el kit puede funcionar independientemente de la PC. Esta sería la forma más común de utilizarlo, por ejemplo, para alguna aplicación en particular.

Escenario 2: Utilizando una conexión directa con la PC través del puerto USB que nos permite enviar un comando AT al módem y recibir su respuesta a través del monitor serie del IDE. En los laboratorios utilizaremos esta forma de conexión para ajustar ciertas configuraciones del kit con ayuda del script “SerialSARAPassthrough.ino” (ver IDE Arduino: Archivo → Ejemplos → MKRNB → Tools).

Escenario 3: Este caso es similar al anterior, solo que se utiliza el IDE de Arduino para programar al kit con el script deseado. Luego, con ayuda de Python y su biblioteca PySerial se programa la rutina deseada.

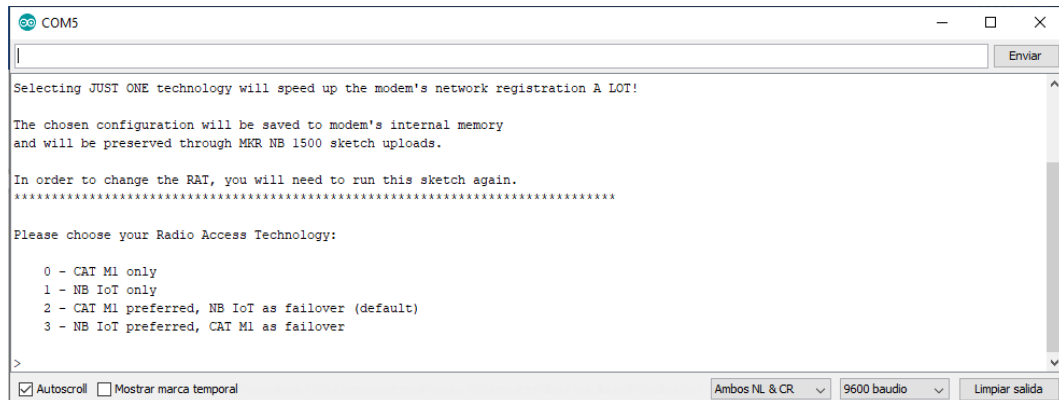
Escenario 4: Similar al Escenario 3, pero utilizando la biblioteca StandardFirmata. Con esta forma de programación se puede acceder a los pines de entrada y salida de la placa como si se lo hiciera desde el propio IDE de Arduino pero utilizando Python (ver IDE Arduino: Archivo → Ejemplos → Firmata).

2.1.3 Configuración básica del kit

Para la puesta en funcionamiento del kit MKR NB 1500 en cualquier aplicación, una parte del código de programación debe encargarse de realizar algunas configuraciones previas como, por ejemplo, seleccionar la tecnología de acceso de radio (RAT) o verificar el estado de los operadores de red. A continuación, se presentan algunos ejemplos básicos que muestran cómo hacerlo.

Ejemplo 1: Selección de RAT

A través del código “ChooseRadioAccessTechnology.ino” incorporado en el IDE (ver IDE Arduino: Archivo → Ejemplos → MKRNB → Tools) se puede seleccionar el tipo de tecnología de radio utilizada por el modem. Puede ser NB-IoT, LTE-M1 o ambas. Una vez configurado el modem, los cambios quedan guardados de manera permanente y el modem queda listo para ser programado con la rutina deseada. Cuando se ejecuta el código, el resultado en el monitor serie del IDE es el siguiente:



```
COM5
[Enviar]

Selecting JUST ONE technology will speed up the modem's network registration A LOT!

The chosen configuration will be saved to modem's internal memory
and will be preserved through MKR NB 1500 sketch uploads.

In order to change the RAT, you will need to run this sketch again.
*****

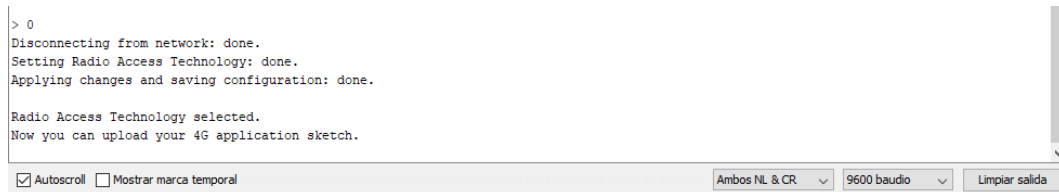
Please choose your Radio Access Technology:

0 - CAT M1 only
1 - NB IoT only
2 - CAT M1 preferred, NB IoT as failover (default)
3 - NB IoT preferred, CAT M1 as failover

>
[Autoscroll] [Mostrar marca temporal] [Ambos NL & CR] [9600 baudio] [Limpiar salida]
```

Figura 2.5. RAT Selection.

Eligiendo “0”, obtenemos:



```
> 0
Disconnecting from network: done.
Setting Radio Access Technology: done.
Applying changes and saving configuration: done.

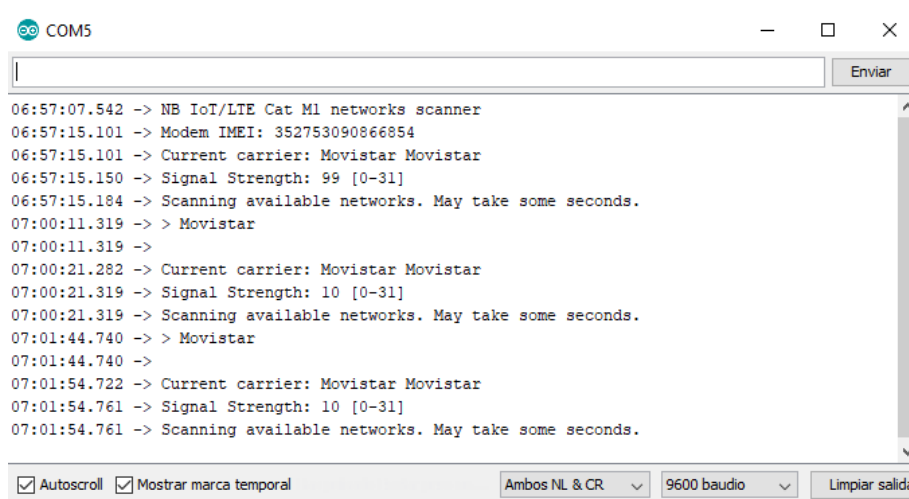
Radio Access Technology selected.
Now you can upload your 4G application sketch.

[Autoscroll] [Mostrar marca temporal] [Ambos NL & CR] [9600 baudio] [Limpiar salida]
```

Figura 2.6. Selección de la tecnología elegida.

Ejemplo 2: Network Scan

Permite buscar las redes NB-IoT, LTE-M1 o ambas, según la configuración realizada en el modem con Selección de RAT. Cuando se ejecuta el código “NBScanNetworks.ino” (ver IDE Arduino: Archivo → Ejemplos → MKRNB → Tools), el resultado en el monitor serie del IDE es el siguiente:



```
COM5
[Enviar]

06:57:07.542 -> NB IoT/LTE Cat M1 networks scanner
06:57:15.101 -> Modem IMEI: 352753090866854
06:57:15.101 -> Current carrier: Movistar Movistar
06:57:15.150 -> Signal Strength: 99 [0-31]
06:57:15.184 -> Scanning available networks. May take some seconds.
07:00:11.319 -> > Movistar
07:00:11.319 ->
07:00:21.282 -> Current carrier: Movistar Movistar
07:00:21.319 -> Signal Strength: 10 [0-31]
07:00:21.319 -> Scanning available networks. May take some seconds.
07:01:44.740 -> > Movistar
07:01:44.740 ->
07:01:54.722 -> Current carrier: Movistar Movistar
07:01:54.761 -> Signal Strength: 10 [0-31]
07:01:54.761 -> Scanning available networks. May take some seconds.

[Autoscroll] [Mostrar marca temporal] [Ambos NL & CR] [9600 baudio] [Limpiar salida]
```

Figura 2.7. Network Scan.

Ejemplo 3: Envío de comandos AT

Para enviar comandos AT al módem se debe cargar el código “SerialSARAPassthrough.ino” en el mismo (ver IDE Arduino: Archivo → Ejemplos → MKRNB → Tools). El envío de datos al módem a través del puerto serie se puede efectuar a través del monitor serie del IDE Arduino o empleando una consola como Putty (Windows) o Minicom (Linux). Los detalles de configuración y los resultados obtenidos para Putty se muestran a continuación:

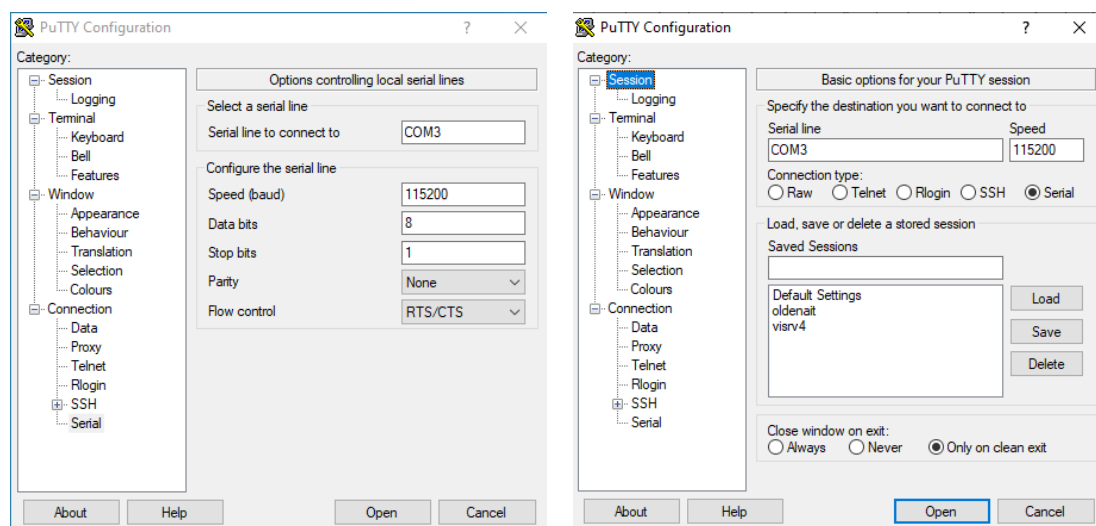


Figura 2.8. Configuración de Putty para el envío de datos a través del puerto serie.

Una vez lograda la conexión, aparece la siguiente ventana desde donde se envían los comandos AT:

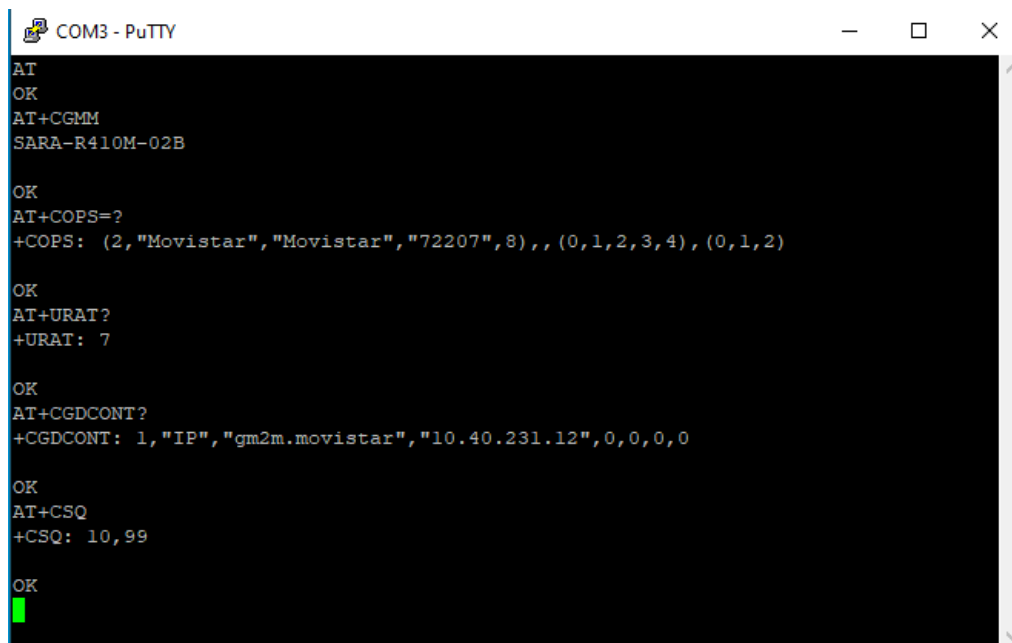


Figura. 2.9. Envío de comandos AT a través de Putty.

Notar que la respuesta del módem al comando +COPS contiene el nombre del operador y un código de identificación internacional. Los códigos para los operadores móviles de Argentina se dan en la tabla siguiente:

<i>Country or Geographical Area</i>	<i>Networks</i>	<i>MCC + MNC codes (Mobile Country Codes + Mobile Network Codes)</i>
Argentina	Compañía de Radiocomunicaciones Móviles S.A.	722 010
	Nextel Argentina srl	722 020
	Telefónica Comunicaciones Personales S.A.	722 070
	CTI PCS S.A.	722 310
	Compañía de Teléfonos del Interior Norte S.A.	722 320
	Compañía de Teléfonos del Interior S.A.	722 330
	Telecom Personal S.A.	722 341

TSB. TELECOMMUNICATION STANDARDIZATION BUREAU OF ITU. Mobile Network Codes (MNC) for the international identification plan for public networks and subscriptions (According to Recommendation ITU-T E.212 (09/2016)).

2.2 Módulo Quectel BG96

El módulo Quectel BG96 es para uso profesional y requiere montar el modem sobre una placa de desarrollo compatible. Una de las mayores ventajas que posee este módulo es la disponibilidad de drivers para diferentes Sistemas Operativos. La foto del dispositivo se muestra en la Figura 2.10.

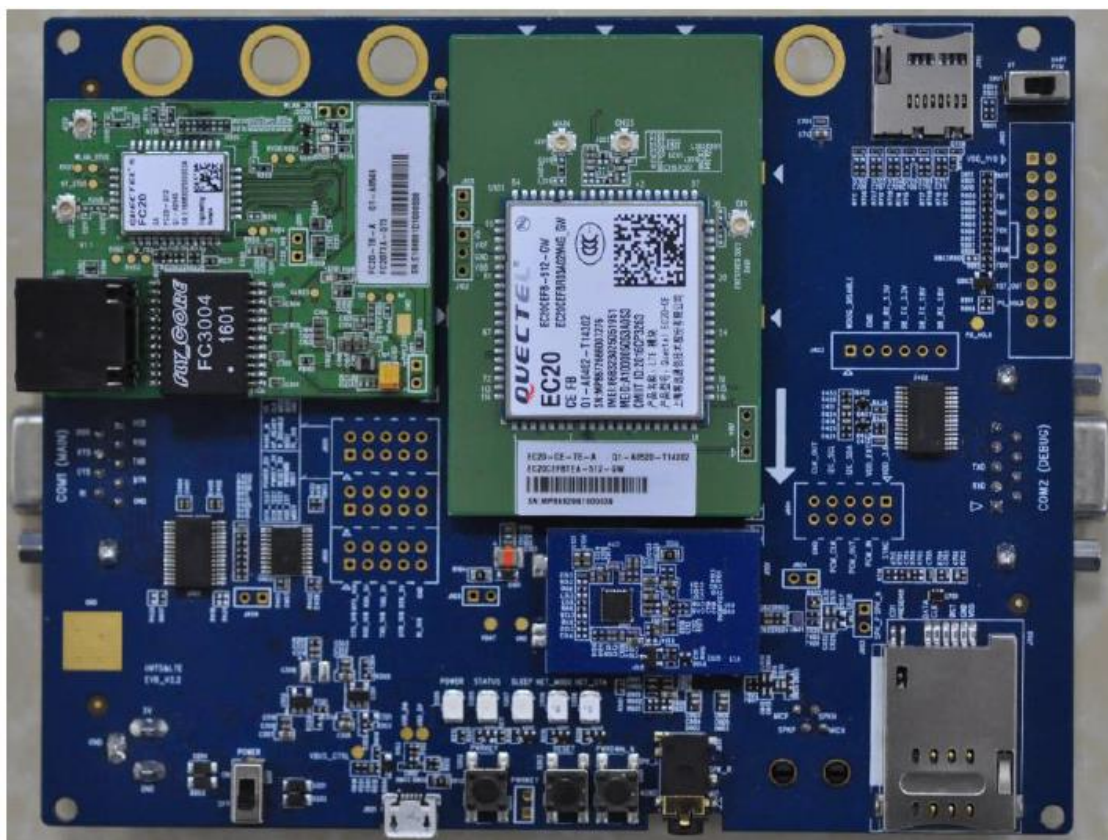


Figura 2.10. Placa de evaluación y módulo BG96.

2.2.1 Descripción general

Este dispositivo se controla a través del puerto USB incorporado en la placa de evaluación, el cual se utiliza para la comunicación mediante comandos AT, transmitir datos (utilizando el software QCOM), actualizar el firmware y recibir datos de GNSS. Las características generales del modem se presentan en la Figura 2.11.

Variant for the Global BG96 Cat M1/Cat NB1: LTE FDD: B1/B2/B3/B4/B5/B8/B12/B13/B18/B19/B20/B25 ^① /B26*/B28 LTE TDD: B39 (For Cat M1 Only) EGPRS: 850/900/1800/1900 MHz Data Cat M1: Max. 375 kbps (DL), Max. 375 kbps (UL) Cat NB1: Max. 32 kbps (DL), Max. 70 kbps (UL) EDGE: Max. 296 kbps (DL), Max. 236.8 kbps (UL) GPRS: Max. 107 kbps (DL), Max. 85.6 kbps (UL) Voice VoLTE (For Cat M1 Only. Support Realtek ALC 5616 /NAU8814 Codec by Default Firmware) SMS Point-to-point MO and MT SMS Cell Broadcast Text and PDU Mode Interfaces USB 2.0 × 1 (With High Speed up to 480 Mbps) UART × 3 PCM × 1 ADC × 2 (15 bits) GPIO × 2 (I2C and UART3 Can be Re-configured as Extra 4 GPIOs) (U)SIM × 1 NETLIGHT × 1 (For Network Status Indication) STATUS × 1 (For Power ON/OFF Indication) Main and GNSS Antenna Interfaces Enhanced Features GNSS (Optional): GPS, GLONASS, BeiDou/Compass, Galileo, QZSS Firmware Upgrade: via USB interface DFOTA: Delta Firmware upgrade Over the Air Processor: ARM A7 Processor, with 3 MB Flash and 3 MB RAM Available for Users	QuecLocator[®]: Support Cell ID Positioning Electrical Characteristics Output Power: Max. Power: 23 dBm Consumption @ LTE Cat M1 (Typical): Power Saving Mode: 10 µA Idle State: 15 mA @ DRX = 1.28 s 15 mA @ e-I-DRX = 40.96 s Sleep State: 1.5 mA @ DRX = 1.28 s 1.2 mA @ e-I-DRX = 40.96 s LTE Connected Mode (Avg.): 128 mA @ 0 dBm 140 mA @ 10 dBm 205 mA @ 23 dBm Consumption @ LTE Cat NB1 (Typical): Power Saving Mode: 10 µA Idle State: 15 mA @ DRX = 1.28 s 15 mA @ e-I-DRX = 40.96 s Sleep State: 1.96 mA @ DRX = 1.28 s 1.1 mA @ e-I-DRX = 40.96 s LTE Connected Mode (Avg.): 96 mA @ 0 dBm 110 mA @ 10 dBm 223 mA @ 23 dBm Sensitivity: -107 dBm @ Cat M1, 1.4 MHz Bandwidth, CE Mode A -113 dBm @ Cat NB1, CE Level 0 Software Features USB Serial Driver: Windows 7/8/8.1/10, Linux 2.6/3.x/4.1–4.14, Android 4.x–9.x GNSS/RIL Driver: Android 4.x–9.x NDIS Driver: Windows 7/8/8.1/10 GobiNet Driver: Linux 2.6–5.4 QMI_WWAN Driver: Linux 3.4–5.4 Protocols: PPP/TCP/UDP/SSL/TLS/FTP(S)/HTTP(S)/NITZ/ PING/MQTT	General Features 3GPP E-UTRA Release 13 Operation Temperature Range: -35 °C to +75 °C Extended Temperature Range: -40 °C to +85 °C Dimensions: 26.5 mm × 22.5 mm × 2.3 mm Approx. 3.1 g LGA Package Supply Voltage: 3.3–4.3 V, 3.8 V Typ. 3GPP TS27.007, 3GPP TS 27.005 and Quectel Enhanced AT Commands Approvals Carrier: Vodafone (Global) Deutsche Telekom/Telefónica (Europe) Verizon/AT&T/T-Mobile/Sprint/U.S. Cellular (North America) Rogers/Telus (Canada) SKT/LGU+ (South Korea) NTT DOCOMO/SoftBank/KDDI (Japan) Telstra (Australia) Regulatory: GCF (Global) CE (Europe) FCC/PTCRB (North America) IC (Canada) Anatel (Brazil) IFETEL (Mexico) CCC (China) KC (South Korea) NCC (Taiwan, China) JATE/TELEC (Japan) RCM (Australia) NBTC (Thailand) IMDA (Singapore) Others: RoHS ^① LTE B25 Supported on BG96 with R1.2 Hardware Version. * Under Development
--	--	---

Figura 2.11. Características generales del módulo BG96.

2.2.2 Configuración básica del kit

Para la puesta en funcionamiento del kit BG96 también se requieren algunas configuraciones previas. Se debe seleccionar la tecnología de acceso de radio (RAT) y verificar el estado de los operadores de red. La manera más fácil de hacerlo es a través de comandos AT.

Ejemplo 1: Envío de comandos AT

El envío de datos al módem a través del puerto serie se puede efectuar a través del monitor serie del IDE Arduino (debe seleccionarse el puerto correcto) o empleando una consola como Putty (Windows) o Minicom (Linux). Los detalles de configuración y los resultados obtenidos para Putty se muestran a continuación:

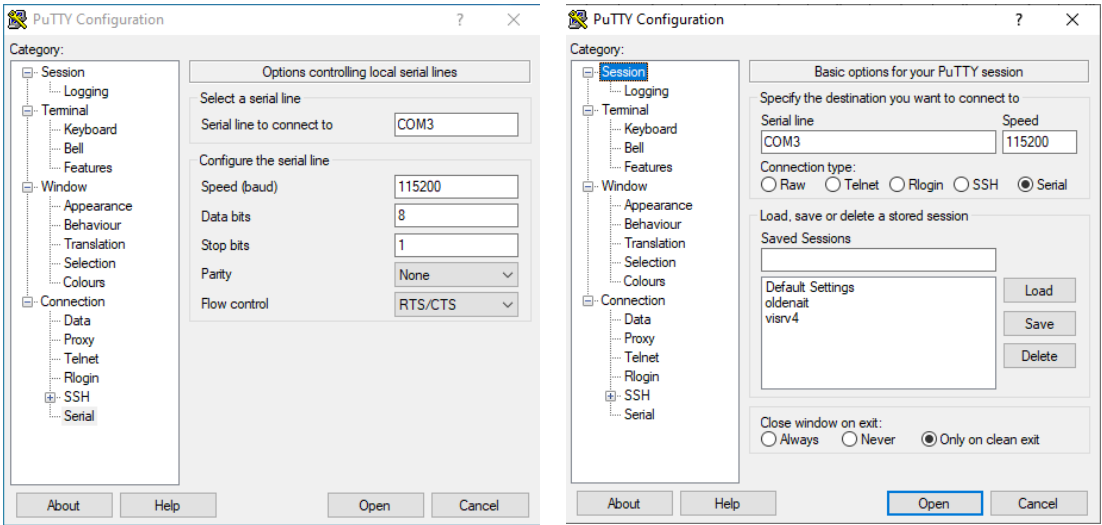
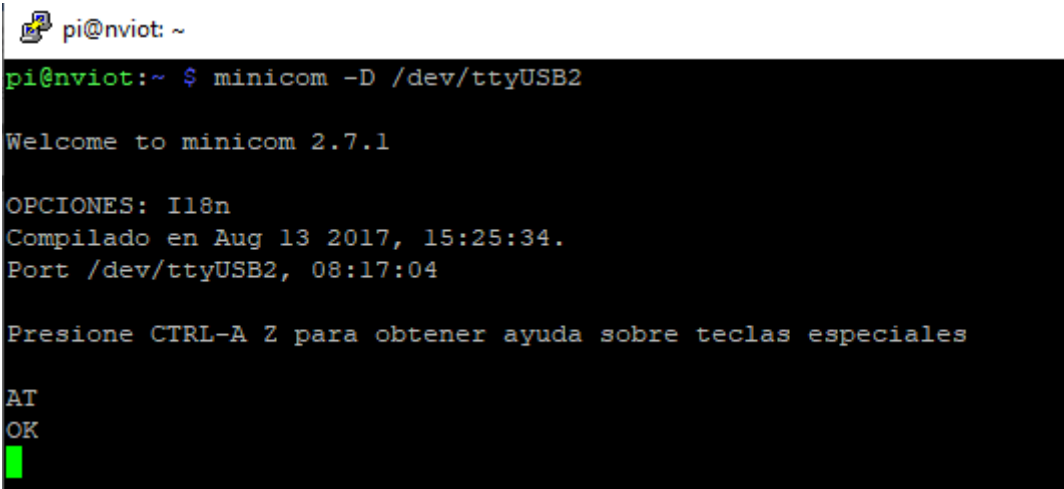


Figura 2.12. Configuración de Putty para el envío de datos a través del puerto serie.

Una vez lograda la conexión con la R-Pi, aparece la siguiente ventana desde donde se envían los comandos AT:



AT
AT+QCGF="nwscanmode",3
AT+QCFG="nwscanseq",03
AT+QCFG="iotopmode",1
AT+QNWINFO

Figura. 2.13. Envío de comandos AT a través de Putty.

```
AT
OK
AT+QCFG="nwscanmode",3
OK
AT+QCFG="nwscanseq",03
OK
AT+QCFG="iotopmode",1
OK
AT+QNWINFO
+QNWINFO: "CAT-NB1","72234","LTE BAND 28",9312
OK
```

2.2.3 Resumen de los pasos para la conexión con la red NB-IoT

```
AT
OK
AT+CGDCONT=1,"IP","nbiot.personal.com"
OK
AT+CEREG?
+CEREG: 0,1

OK
AT+CGATT=1
OK
AT+COPS?
+COPS: 0,0,"AR PERSONAL Personal",8

OK
AT+QIACT=1
OK
AT+QIACT=
ERROR
AT+QIACT?
+QIACT: 1,1,1,"181.9.64.1"

OK
```

3 Laboratorio 1: Prueba de velocidad

El objetivo de este ensayo consiste en determinar la calidad de la conexión con la red NB-IoT utilizando Iperf3. Iperf3 es una herramienta multiplataforma de código abierto que permite crear un flujo de datos con tramas TCP y UDP entre cliente y servidor. Esto permite cuantificar parámetros tales como ancho de banda, cantidad de paquetes perdidos y retardo de transmisión.

3.1 Introducción

El interés cada vez más reciente en NB-IoT reside principalmente en la conectividad confiable que deriva del estándar LTE como también del hecho de utilizar la propia infraestructura de la red móvil, combinadas con las características de bajo consumo específicas de la tecnología IoT. Si bien NB-IoT soporta conexiones TCP y UDP, el interés está centrado fundamentalmente en el protocolo UDP, dado que introduce menor sobrecarga en la trama de datos y es menos sensible a los retrasos de transmisión y pérdidas de paquetes asociados a esta tecnología, especialmente para aplicaciones basadas en MQTT.

Conocer la velocidad real del enlace, no solo nos permitirá dimensionar la capacidad del sistema que asegure la correcta transmisión de datos para una determinada aplicación, sino que además nos dará la posibilidad de comparar el servicio brindado por diferentes operadoras, dándonos la libertad de optar por aquella que brinde el más eficiente.

3.2 Procedimiento

La prueba consiste en establecer una conexión cliente – servidor a través de la herramienta Iperf3 utilizando las computadoras “PC1” y “PC2”. El esquema de conexiones es el que se muestra en la Figura 3.1.

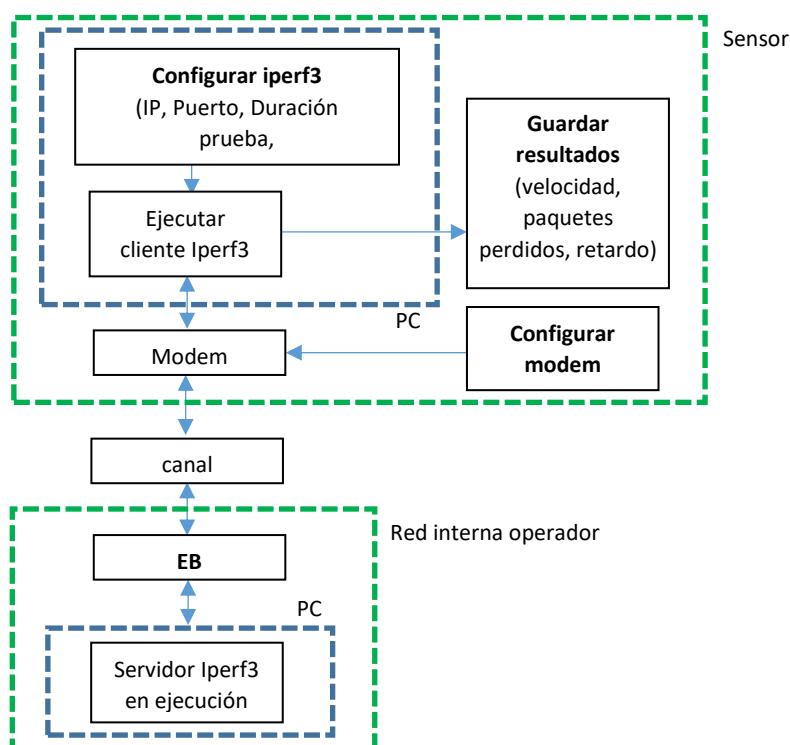


Figura 3.1. Esquema de conexiones para la prueba de medición de velocidad.

El servidor Iperf3 se ejecuta continuamente en la computadora “PC1”, en este caso, situada sobre la red del operador. El hecho de que el servidor Iperf3 se encuentre cercano al “core” del operador, permite independizarse de los retrasos aleatorios inherentes a internet, lo cual nos posibilita caracterizar el enlace con mayor precisión. Cuando esto no es posible, tanto la velocidad como el retardo del enlace pueden variar significativamente en diferentes momentos del día. En tales circunstancias, es aconsejable realizar las mediciones en diferentes horarios para obtener una estadística más completa del desempeño.

- a) Verificar que el kit está siendo detectado por la computadora “PC2”. A través de comandos AT, determinar la dirección IP que le fue asignada al modem y la tecnología de radio empleada. Esto puede realizarse desde el IDE Arduino o a través de una consola ejecutando:

Modem Arduino MKR NB1500	Modem Quectel BG96
AT	AT
AT+URAT?	AT+QNWINFO
AT+CREG?	
AT+CGDCONT?	

Para el caso del modem Quectel BG96, debería obtenerse una respuesta como la siguiente:

```
AT+QNWINFO
+QNWINFO: "CAT-NB1", "72234", "LTE BAND 28", 9312
```

- b) Iniciar el cliente Iperf3 en la computadora “PC2” a la que se conecta el kit NB. Para la configuración del cliente Iperf3, se debe tener en cuenta el tipo de enlace a analizar. En este caso, al tratarse de tecnología NB-IoT, se utilizarán tramas UDP. Comenzar con la configuración base utilizada en la primera fila de la Tabla 3.1.

Algunos ejemplos básicos de configuración y uso de Iperf3 se dan en el Apéndice A.1.

Comando Iperf3	Parámetro
iperf3 -c 181.9.64.1 -b 1m -u -t 10 -i 1	-c: modo cliente
	181.9.64.1 : dirección del servidor
	-b 1m : ajusta la velocidad a un máximo de 1 Mbps
	-u : establece una conexión UDP
	-t 10 : establece el tiempo de la transmisión a 10 seg.
	-i 1 : establece en 1 seg. el intervalo entre reportes
iperf3 -c 181.9.64.1 -b 10m -u -t 10 -i 1	...
	-b 10m : ajusta la velocidad a un máximo de 10 Mbps
	...
	-t 5 : establece el tiempo de la transmisión a 5 seg.
iperf3 -c 181.9.64.1 -b 10m -u --get-server-output -R	...
	--get-server-output : obtiene los resultados del servidor
	-R : efectúa la prueba en modo inverso.
iperf3 -c 181.9.64.1 -b 10m -u -l 8k --get-server-output -R	...
	-l : establece la longitud del buffer
	...

Tabla 3.1. Configuración del cliente Iperf3. Se debe tener en cuenta que la dirección IP del servidor Iperf3 puede variar, por lo que deberá verificarse la misma al inicio de las pruebas.

- c) Analizar los resultados obtenidos de la prueba en modo directo (cliente envía datos – servidor recibe), los cuales deberían ser similares a los siguientes:

[ID]	Interval	Transfer	Bandwidth	Jitter	Lost/Total Datagrams
[5]	2.01-3.01	sec 1.38 KBytes	11.3 Kbits/sec	629.304 ms	0/1 (0%)
[5]	3.01-4.01	sec 1.38 KBytes	11.3 Kbits/sec	663.063 ms	0/1 (0%)
[5]	4.01-5.00	sec 1.38 KBytes	11.4 Kbits/sec	712.531 ms	0/1 (0%)
[5]	5.00-6.01	sec 1.38 KBytes	11.2 Kbits/sec	742.949 ms	0/1 (0%)
[5]	7.01-8.01	sec 1.38 KBytes	11.3 Kbits/sec	772.537 ms	0/1 (0%)
[5]	8.01-9.00	sec 1.38 KBytes	11.3 Kbits/sec	802.336 ms	0/1 (0%)
[5]	9.00-10.01	sec 1.38 KBytes	11.2 Kbits/sec	843.119 ms	0/1 (0%)
[5]	10.01-11.01	sec 1.38 KBytes	11.3 Kbits/sec	859.496 ms	0/1 (0%)
[5]	12.01-13.01	sec 1.38 KBytes	11.3 Kbits/sec	885.813 ms	0/1 (0%)
[5]	13.01-14.01	sec 1.38 KBytes	11.3 Kbits/sec	910.182 ms	0/1 (0%)
[5]	14.01-15.01	sec 1.38 KBytes	11.3 Kbits/sec	944.375 ms	0/1 (0%)
[5]	16.00-17.01	sec 1.38 KBytes	11.2 Kbits/sec	954.457 ms	0/1 (0%)

- d) Repetir la prueba en modo inverso (servidor envía datos – cliente recibe) y analizar cómo cambian los resultados.
- e) Repita las pruebas cambiando el parámetro -l 8k a -l 1k (ver Apéndice A.1) que modifica la longitud del buffer de lectura y escritura.

Interpretación de los resultados

```
pi@enviot:~$ iperf3 -c telemetry-iperf3.cablevisionfibertel.com.ar -u -b 0 -k 8
Connecting to host telemetry-iperf3.cablevisionfibertel.com.ar, port 5201
[ 5] local 181.9.64.137 port 50057 connected to 181.10.172.66 port 5201
[ ID] Interval      Transfer    Bitrate      Total Datagrams
[ 5]  0.00-0.26    sec 13.7 KBytes  430 Kbits/sec  10
-----
[ ID] Interval      Transfer    Bitrate      Jitter      Lost/Total Datagrams
[ 5]  0.00-0.26    sec 13.7 KBytes  430 Kbits/sec  0.000 ms    0/10 (0%)  sender
[ 5]  0.00-0.26    sec 13.7 KBytes  430 Kbits/sec  473.755 ms  0/10 (0%)  receiver
iperf Done.
```

sender: en el modo directo, el emisor de datos es el cliente iperf. Los resultados muestran la velocidad de subida del cliente iperf al servidor iperf.

receiver: en el modo directo, el receptor de datos es el servidor iperf. Los resultados muestran la velocidad de bajada en el servidor iperf.

Idealmente, ambas medidas deberían ser las mismas.

sender: en el modo inverso, el emisor de datos es el servidor iperf. Los resultados muestran la velocidad de subida del servidor al cliente.

receiver: en el modo inverso, el receptor de datos es el cliente iperf. Los resultados muestran la velocidad de bajada en el cliente.

También en este caso, ambas medidas deberían ser iguales o muy parecidas.

Nota: Dado que el kit Arduino de NB-IoT no posee un software de control específico para utilizarlo como modem de internet, es necesario establecer una conexión PPP para realizar las pruebas. Esta tarea escapa de los objetivos del curso.

3.3 Preguntas de Repaso

- a) ¿Por qué es aconsejable medir el desempeño de la red y en diferentes momentos del día?
- b) ¿Por qué utilizamos Iperf3 con tramas UDP para las pruebas de NB-IoT?
- c) ¿Por qué motivo es preferible utilizar un servidor Iperf3 sobre la red interna del operador de NB-IoT?
- d) ¿Es comparable la velocidad obtenida en las pruebas de NB-IoT con la que se enuncia en el estándar?

4 Laboratorio 2: Prueba de latencia

Dado que NB-IoT está orientado a comunicaciones esporádicas, la latencia es mucho mayor que la existente en otras tecnologías, como por ejemplo LTE Cat-M1. Esta prueba consiste en el envío de paquetes de datos de manera de poder establecer la latencia y jitter existentes en una conexión NB-IoT. Se harán diferentes pruebas tanto con tramas UDP de longitud variable (Iperf) como con tramas ICMP (ping) de manera de obtener los valores de jitter y latencia en diferentes condiciones de transmisión.

4.1 Introducción

La latencia se refiere al tiempo de ida y vuelta que le toma a un paquete de datos salir del emisor, llegar al destino y volver al emisor. Es lo que se conoce como 'round-trip time'. Se diferencia del retardo de la red en que, este último, se refiere al tiempo que le toma a un paquete de datos ir de un extremo de la red a otro, es decir, en una sola dirección. Tanto la latencia como el retardo dependen principalmente de la cantidad de tráfico en un instante determinado, del tamaño del dato y de la cantidad de nodos que deba atravesar el dato.

Usualmente, los paquetes enviados a través de la red tendrán diferentes retardos. Esto se debe a que cada paquete se enruta individualmente y puede atravesar por diferentes caminos hasta alcanzar el destino.

Por último, también debemos mencionar la existencia del jitter, el cual es la variación del retardo y que depende del grado de congestión a lo largo de la red.

4.2 Procedimiento

Esta prueba consiste en establecer una conexión cliente – servidor a través de la herramienta Iperf3, como se hizo en la prueba de velocidad, utilizando las computadoras "PC1" y "PC2". El esquema de conexiones es el que se muestra en la Figura 4.1.

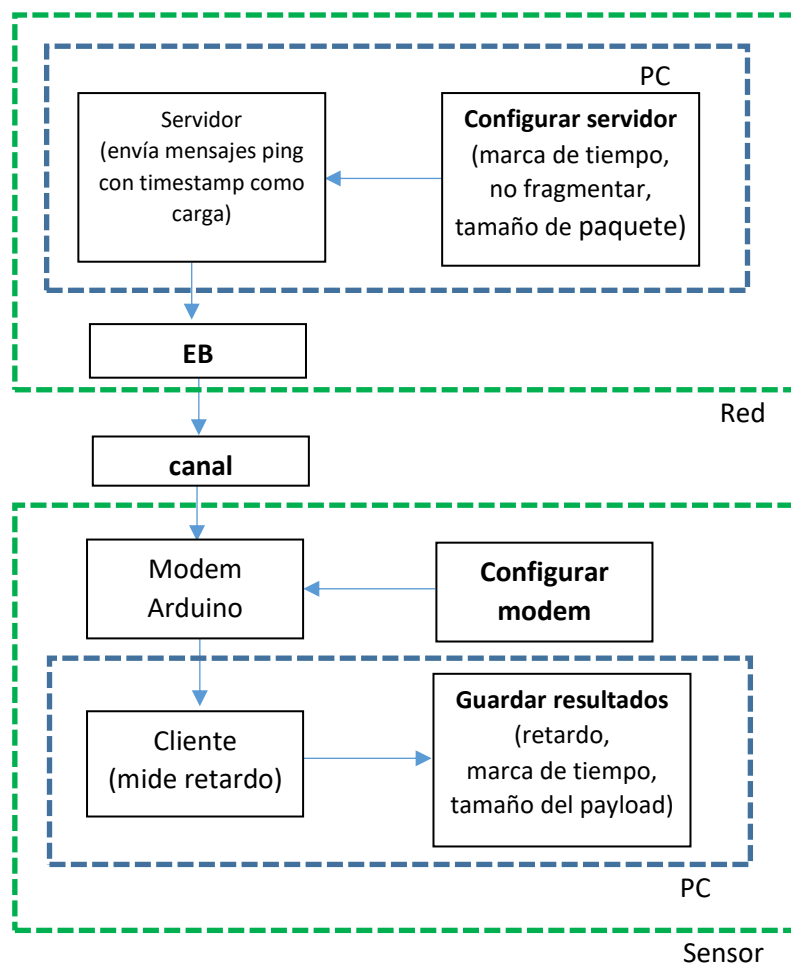


Figura 4.1. Esquema de conexiones para la prueba de medición de latencia.

Pruebas de jitter utilizando la herramienta Iperf3:

- Verificar que el kit está siendo detectado por la computadora "PC2". A través de comandos AT, determinar la dirección IP que le fue asignada al modem y la tecnología de radio empleada. Verificar que la conexión NB se encuentre activa. Esto puede realizarse desde el IDE Arduino o a través de una consola ejecutando:

Modem Arduino MKR NB1500	Modem Quectel BG96
AT	AT
AT+URAT?	AT+QNWINFO
AT+CREG?	
AT+CGDCONT?	

- Iniciar el cliente Iperf3 en la computadora "PC2" a la que se conecta el kit NB. Para la configuración del cliente Iperf3, se debe tener en cuenta el tipo de enlace a analizar. También en este caso se utilizarán tramas UDP.
- Utilizar las configuraciones dadas a continuación:

Comando Iperf3
iperf3 -c 181.9.64.1 -b 10m -u
iperf3 -c 181.9.64.1 -b 10m -u -R

Ejemplo 1: modo directo

```
pi@nviot:~$ iperf3 -c lotiperfuniversidad.telecom.com.ar -b 0 -k 8 -u
Connecting to host lotiperfuniversidad.telecom.com.ar, port 5201
[ 5] local 181.9.64.137 port 34621 connected to 181.94.5.30 port 5201
[ ID] Interval      Transfer    Bitrate      Total Datagrams
[ 5] 0.00-0.21 sec  13.4 KBytes 522 Kbits/sec 10
-----
[ ID] Interval      Transfer    Bitrate      Jitter        Lost/Total Datagrams
[ 5] 0.00-0.21 sec  13.4 KBytes 522 Kbits/sec 0.000 ms      0/8 (0%) sender
[ 5] 0.00-13.00 sec 13.4 KBytes 8.42 Kbits/sec 536.703 ms    0/8 (0%) receiver

iperf Done.
```

Ejemplo 2: modo inverso

```
pi@nviot:~$ iperf3 -c lotiperfuniversidad.telecom.com.ar -b 0 -k 8 -u -R
Connecting to host lotiperfuniversidad.telecom.com.ar, port 5201
Reverse mode, remote host lotiperfuniversidad.telecom.com.ar is sending
[ 5] local 181.9.64.137 port 53129 connected to 181.94.5.30 port 5201
[ ID] Interval      Transfer    Bitrate      Jitter        Lost/Total Datagrams
[ 5] 0.00-28.02 sec 1.34 KBytes 391 bits/sec 1773.956 ms    0/1 (0%)
^C[ 5] 28.02-79.61 sec 0.00 Bytes 0.00 bits/sec 1129.136 ms    0/0 (0%)
-----
[ ID] Interval      Transfer    Bitrate      Jitter        Lost/Total Datagrams
[ 5] 0.00-79.61 sec 0.00 Bytes 0.00 bits/sec 0.000 ms      0/0 (0%) sender
[ 5] 0.00-79.61 sec 10.7 KBytes 1.10 Kbits/sec 1129.136 ms    0/8 (0%) receiver
```

Pruebas de tiempo de ida y vuelta (RTT) utilizando la herramienta Ping:

- Verificar que la conexión NB se encuentre activa.
- Configurar las pruebas de Ping a diferentes direcciones IP teniendo en cuenta los parámetros de control.

Ejemplo 3: Ping a Google (DNS)

```
pi@nviot:~$ ping -c 10 -s 10 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 10(38) bytes of data:
18 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=371 ms
18 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=636 ms
18 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=389 ms
18 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=572 ms
18 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=198 ms
18 bytes from 8.8.8.8: icmp_seq=6 ttl=115 time=493 ms
18 bytes from 8.8.8.8: icmp_seq=7 ttl=115 time=760 ms
18 bytes from 8.8.8.8: icmp_seq=8 ttl=115 time=515 ms
18 bytes from 8.8.8.8: icmp_seq=9 ttl=115 time=697 ms
```

```
18 bytes from 8.8.8.8: icmp_seq=10 ttl=115 time=338 ms

--- 8.8.8.8 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 16ms
rtt min/avg/max/mdev = 198.287/497.004/760.097/166.375 ms
```

Ejemplo 4: Ping a Google (DNS)

```
pi@nviot:~ $ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=1168 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=323 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=363 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=657 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=283 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=115 time=688 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=115 time=829 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=115 time=442 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=115 time=1994 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=115 time=1118 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=115 time=306 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=115 time=711 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=115 time=337 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=115 time=592 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=115 time=222 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=115 time=483 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=115 time=747 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=115 time=396 ms

--- 8.8.8.8 ping statistics ---
19 packets transmitted, 18 received, 5.26316% packet loss, time 140ms
rtt min/avg/max/mdev = 221.989/647.629/1994.038/421.844 ms
```

Ejemplo 5: Ping al servidor del operador

```
pi@nviot:~ $ ping -c 8 -s 50 181.96.126.37
PING 181.96.126.37 (181.96.126.37) 50(78) bytes of data.
58 bytes from 181.96.126.37: icmp_seq=1 ttl=253 time=369 ms
58 bytes from 181.96.126.37: icmp_seq=2 ttl=253 time=637 ms
58 bytes from 181.96.126.37: icmp_seq=3 ttl=253 time=271 ms
58 bytes from 181.96.126.37: icmp_seq=4 ttl=253 time=550 ms
58 bytes from 181.96.126.37: icmp_seq=5 ttl=253 time=830 ms
58 bytes from 181.96.126.37: icmp_seq=6 ttl=253 time=469 ms
58 bytes from 181.96.126.37: icmp_seq=7 ttl=253 time=750 ms
58 bytes from 181.96.126.37: icmp_seq=8 ttl=253 time=387 ms

--- 181.96.126.37 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 14ms
rtt min/avg/max/mdev = 271.499/532.805/829.557/182.724 ms
```

Nota 1: los paquetes ICMP pueden ser procesados con baja prioridad, lo que puede afectar la precisión con que se mide de la latencia (tiempo de ida y vuelta).

Nota 2: el tamaño del paquete ICMP utilizado por Ping está compuesto por la carga útil (definida por `--s tamaño_de_paquete`) y el encabezado (8 bytes).

Nota 3: tener en cuenta que Ping tiene diferentes nombres de parámetros según el Sistema Operativo. Existen diferencias entre Windows y Linux.

Ejemplo 6: Traceroute al servidor lperf

```
pi@nviot:~$ traceroute lotiperfuniversidad.telecom.com.ar
traceroute to lotiperfuniversidad.telecom.com.ar (181.94.5.30), 30 hops max, 60 byte packets
1 * * *
2 172.25.218.161 (172.25.218.161) 1024.426 ms 1218.227 ms 1378.512 ms
3 host37.181-96-126.telecom.net.ar (181.96.126.37) 1441.007 ms 1648.829 ms 1680.893 ms
4 * * *
5 host244.181-88-80.telecom.net.ar (181.88.80.244) 2415.275 ms 2608.043 ms 2639.987 ms
6 host205.181-15-47.telecom.net.ar (181.15.47.205) 2876.839 ms 2063.609 ms 1901.804 ms
7 * * *
8 host1.181-105-192.telecom.net.ar (181.105.192.1) 3278.017 ms 2606.231 ms 2685.566 ms
9 host2.181-105-192.telecom.net.ar (181.105.192.2) 2861.243 ms host10.181-105-
192.telecom.net.ar (181.105.192.10) 2737.290 ms host2.181-105-192.telecom.net.ar
(181.105.192.2) 2653.502 ms
10 host30.181-94-5.telecom.net.ar (181.94.5.30) 2894.233 ms 1261.818 ms 397.759 ms
11 host30.181-94-5.telecom.net.ar (181.94.5.30) 426.532 ms 606.960 ms 193.720 ms
```

Traceroute al servidor lperf conectando a través del modem.

```
Microsoft Windows [Versión 10.0.19042.1288]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Guillermo>tracert lotiperfuniversidad.telecom.com.ar

Traza a la dirección lotiperfuniversidad.telecom.com.ar [181.94.5.30]
sobre un máximo de 30 saltos:

 1 <1 ms <1 ms <1 ms 192.168.36.10
 2 * * * Tiempo de espera agotado para esta solicitud.
 3 <1 ms <1 ms <1 ms swri-gate00.uns.edu.ar [200.49.237.129]
 4 <1 ms <1 ms <1 ms fiig1.uns.edu.ar [200.49.239.129]
 5 <1 ms <1 ms <1 ms griu.uns.edu.ar [170.210.184.1]
 6 <1 ms <1 ms <1 ms tasa-uns-500M.BAHIA-BLANCA.riu.edu.ar [170.210.4.113]
 7 25 ms 24 ms 25 ms host29.advance.com.ar [200.70.52.29]
 8 25 ms 25 ms 25 ms 200-26-92-34.advance.com.ar [200.26.92.34]
 9 * * * Tiempo de espera agotado para esta solicitud.
10 * * * Tiempo de espera agotado para esta solicitud.
11 * * * Tiempo de espera agotado para esta solicitud.
12 * * * Tiempo de espera agotado para esta solicitud.
13 28 ms 27 ms 29 ms riu-tasa-500m.buenos-aires.riu.edu.ar [170.210.4.2]
14 * 24 ms 23 ms 200.0.17.131
15 * * * Tiempo de espera agotado para esta solicitud.
16 * * * Tiempo de espera agotado para esta solicitud.
17 24 ms 29 ms 29 ms host6.181-88-111.telecom.net.ar [181.88.111.6]
18 23 ms 23 ms 23 ms host205.181-15-47.telecom.net.ar [181.15.47.205]
19 * * * Tiempo de espera agotado para esta solicitud.
20 24 ms 23 ms 23 ms host1.181-105-192.telecom.net.ar [181.105.192.1]
21 24 ms 24 ms 24 ms host2.181-105-192.telecom.net.ar [181.105.192.2]
22 24 ms 24 ms 25 ms host30.181-94-5.telecom.net.ar [181.94.5.30]
23 24 ms 24 ms 24 ms host30.181-94-5.telecom.net.ar [181.94.5.30]
```

Traceroute al servidor lperf conectando a través de LAN.

4.3 Preguntas de Repaso

- a) ¿Es comparable el jitter obtenido en NB-IoT con el que se enuncia en el estándar?
- b) ¿Es comparable el RTT obtenido con el que se enuncia en el estándar?
- c) ¿Se pierden paquetes desde el modem al servidor del operador? ¿Y desde el modem al servidor de Google?
- d) ¿Cómo son las latencias de la ruta? Compare los resultados del comando “traceroute”.

5 Laboratorio 3: Prueba de alcance

En este ensayo, se busca determinar la distancia máxima de recepción que puede existir entre el modem y la EB. Este ensayo está íntimamente ligado con la prueba de consumo, dado que, a mayor distancia entre EB y modem mayor será la potencia y número de repeticiones que deberá utilizar este último para lograr una transmisión exitosa. Asimismo, a través de la medición de los niveles potencia recibida, tanto RSRP (Reference Signal Received Power) como RSSI (Received Signal Strength Indicator) se podrán determinar otros parámetros de capa física como lo son el RSRQ (Reference Signal Received Quality) y el SINR (Signal-to-Interference and Noise Ratio).

5.1 Introducción

Desde su concepción, la tecnología NB-IoT se desarrolló para tener un mejor nivel de cobertura que su antecesora GPRS, permitiendo establecer una comunicación confiable en ambiente urbano, suburbano o rural. Si bien esto guarda una relación directa con la distancia máxima entre el UE y la EB, lo usual es determinar el grado cobertura a través del MCL (Maximum Coupling Loss). Podemos definir al MCL como la pérdida máxima total del enlace de comunicación entre la EB y el UE para la cual la comunicación se mantiene. Dicho de otra forma, el MCL asegura una completa independencia con la frecuencia de trabajo y el propio entorno de operación. Concretamente, con NB-IoT es posible obtener 164 dB de MCL, un valor mucho mayor a los 144 dB que ofrece GPRS. Este nivel de cobertura extendida y la posibilidad de controlar los modos de ahorro de energía son los que hacen a NB-IoT cada vez más popular en aplicaciones IoT.

5.2 Procedimiento

El ensayo de nivel de cobertura se realiza de manera indirecta. Esto significa que, para determinar la pérdida de acoplamiento, primero es necesario determinar otros parámetros del enlace a partir de los cuales podremos calcularla.

Las tramas de prueba utilizadas son las mismas en todos los casos. Con esto nos aseguramos de no variar la información en cada prueba de alcance.

El esquema empleado para las mediciones es el que se muestra en la Figura 5.1.

A través del uso de un atenuador variable se puede controlar la potencia recibida en el modem, con lo cual se emula la separación entre dicho modem y la EB. Esto permite realizar las pruebas sin desplazar el modem.

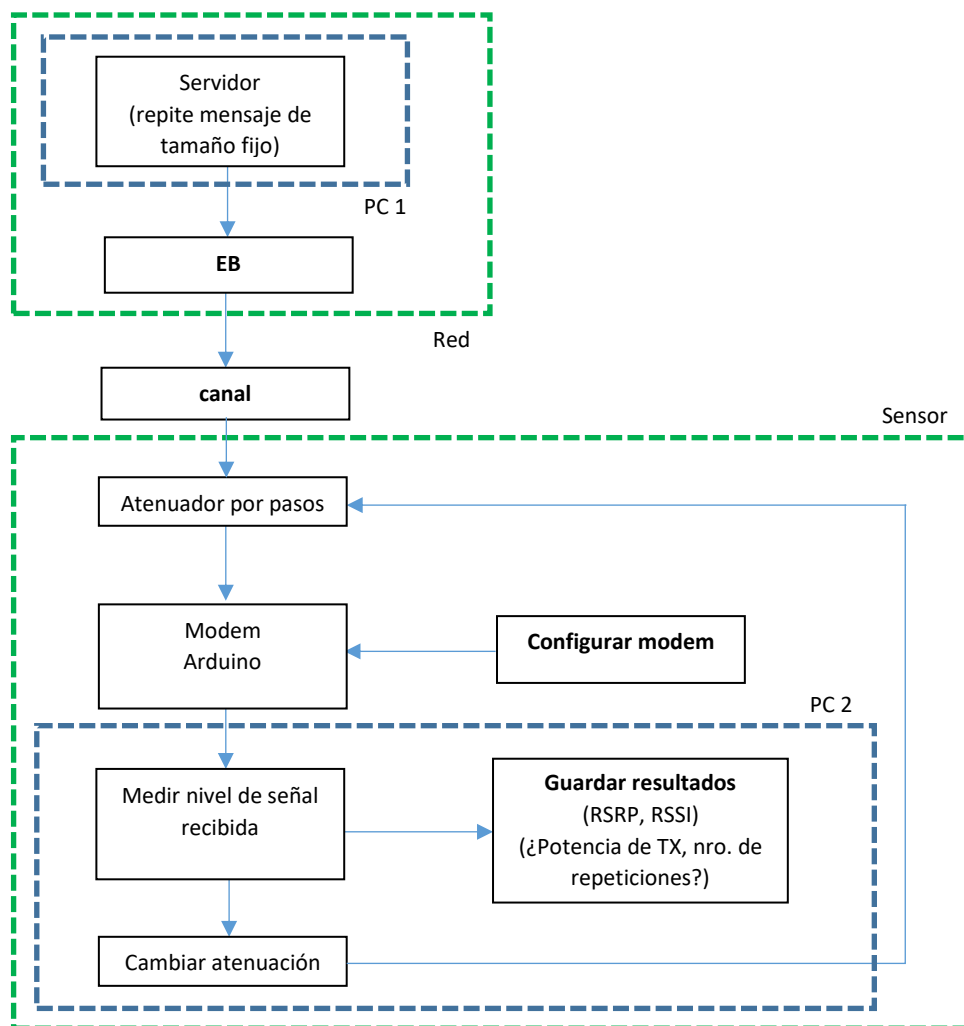


Figura 5.1. Esquema de conexiones para la prueba de alcance.

Las pruebas se realizaron empleando el modem BG96 y fue necesario utilizar comandos AT para obtener los parámetros de calidad de señal. Se recomienda referirse al manual de comandos AT del modem para mayores detalles.

Comando: [AT+QCSQ \(Query and Report Signal Strength\)](#)

Este comando permite obtener los valores de RSSI (received signal strength Indication), RSRP (reference signal received power), SINR (signal to interference plus noise ratio) y RSRQ (reference signal received quality).

<sysmdoe>	<value1>	<value2>	<value3>	<value4>
"NOSERVICE"				
"GSM"	<gsm_rssi>			
"CAT-M1"	<lte_rssi>	<lte_rsrp>	<lte_sinr>	<lte_rsrq>
"CAT-NB1"	<lte_rssi>	<lte_rsrp>	<lte_sinr>	<lte_rsrq>

<gsm_rssi>,<lte_rssi>	An integer indicating the received signal strength. These parameters are available for GSM and LTE mode respectively.
<lte_rsrp>	An integer indicating the reference signal received power (RSRP). This parameter is available for LTE mode.
<lte_sinr>	An integer indicating the signal to interference plus noise ratio (SINR). Logarithmic value of SINR. Values are in 1/5th of a dB. The range is 0-250 which translates to -20dB - +30dB.
<lte_rsrq>	An integer indicating the reference signal received quality (RSRQ) in dB.

Ejemplo:

```
AT+QCSQ
+QCSQ: "CAT-NB1",-74,-82,196,-9
OK
```

RSSI	RSRP	SINR	RSRQ
-74 dBm	-82 dBm	196	-9 dB
		Límites de la lectura de SINR: $0 \equiv -20 \text{ dB}$, $250 \equiv +30 \text{ dB}$	
		$\text{SINR (dB)} = \text{SINR (veces)} / 5 + (-20 \text{ dB})$	
		$\text{SINR} = 196 / 5 + (-20 \text{ dB}) = 19.2 \text{ dB}$	

El RSRP es un número que indica el nivel de potencia recibida de la señal de referencia. Por lo tanto, si se mantienen condiciones similares en las mediciones (posición de antena, ambiente exterior o interior) mientras más bajo sea su valor, mayor será la distancia entre EB y UE.

Ciertos modems, solo pueden indicar el valor de RSSI y no el de RSRP. Si bien esto es una desventaja ya que el RSSI representa la potencia total capturada en una cierta banda, incluyendo la potencia de la señal deseada, ruido e interferencias, puede utilizarse como una aproximación razonable cuando quiere determinarse el nivel de cobertura aproximado.

Comando: [AT+CSQ \(Signal Quality Report\)](#)

Test Command AT+CSQ=?	Response The Test Command returns values supported by the TA. +CSQ: (list of supported <rssi>s),(list of supported <ber>s) OK
Execution Command AT+CSQ	Response The Execution Command returns received signal strength indication <rssi> and channel bit error rate <ber> from the ME. +CSQ: <rssi>,<ber> OK If there is an error related to ME functionality, response: +CME ERROR: <err>
Maximum Response Time	300ms

Parameter

<rssi>	0	-113dBm or less
	1	-111dBm
	2...30	-109... -53dBm
	31	-51dBm or greater
	99	Not known or not detectable
<ber>	Channel bit error rate (in percent)	
	0...7	As RXQUAL values in the table in 3GPP TS 45.008 subclause 8.2.4
	99	Not known or not detectable

Límites de la lectura de RSSI: 0 \equiv -113 dBm, 31 \equiv -51 dBm

$\text{RSSI (dBm)} = \text{RSSI (veces)} \times 2 - 113 \text{ dBm}$

Ejemplo:

```
AT+CSQ
+CSQ: 21,99
OK
```

Lectura RSSI (veces) = 21

$\text{RSSI} = 21 \times 2 - 113 \text{ dBm} = -71 \text{ dBm}$

Comparar con el valor obtenido al ejecutar el comando QCSQ:

```
AT+QCSQ
+QCSQ: "CAT-NB1",-71,-82,205,-13
OK
```

RSSI = -71 dBm

Ejemplo: Estimar los valores de la pérdida de acoplamiento y pérdida de acoplamiento máxima.

El valor del MCL se puede estimar a partir de los cálculos indicados en TR 36.888 del 3GPP, adecuados a NB-IoT. La medición real del MCL es un proceso complejo que requiere conocer con exactitud el valor de todos los parámetros intervinientes.

Ref.	Parámetro	Valor en DL
(1)	Tx Power	+46 dBm
(2)	Rx Thermal Noise Density	-174 dBm/Hz
(3)	Rx Noise Figure	9 dB
(4)	Rx Interference Margin	0 dB
(5)	Rx Occupied Channel Bandwidth	360 (KHz)
(6)	Rx Effective Noise Power (2) + (3) + (4) + 10 log(5)	-109.4 dBm
(7)	Rx Required SINR	-4 (dB)
(8)	Receiver Sensitivity (6) + (7)	-113.4 (dBm)
(9)	Rx Processing Gain	0 (dB)
(10)	MCL (1) - (8)	159.4 (dB)

El MCL obtenido depende, principalmente, de la cifra de ruido del receptor, del ancho de banda utilizado y de la ganancia de procesamiento. En este caso, se utilizaron valores conservativos por lo que el MCL real podría acercarse e incluso superar a los 164 dB declarados para NB-IoT.

Para el caso de las mediciones realizadas anteriormente, la pérdida de acoplamiento (CL) obtenida es:

$$CL \approx P_{TX,RS} - RSRP = 27.2 \text{ dBm} - (-82 \text{ dBm}) = 109.2 \text{ dBm}$$

donde $P_{TX,RS}$ es la potencia de transmisión declarada por la compañía que presta el servicio y RSRP es el valor de señal recibido.

Como se observa, todavía existe un gran margen entre este valor y el MCL, lo que permitiría incrementar la distancia entre el UE y la EB en condiciones de uso exterior o llevar al UE a zonas de difícil acceso de señal en condiciones de uso interior.

5.3 Preguntas de Repaso

- ¿Qué diferencia existe entre el valor indicado por el RSSI y el RSRP?
- ¿De qué depende la ganancia de procesamiento?
- ¿Cómo podrían incrementarse en una prueba de laboratorio los valores de CL hasta llegar al valor de MCL real?

6 Laboratorio 4: Prueba de consumo

Este ensayo tiene como fin determinar la energía demandada por el modem en sus diferentes modos de funcionamiento. A partir de esta prueba, se podrá inferir la autonomía esperable en el caso de una aplicación alimentada por baterías como también el intervalo de transmisión mínimo en el caso de una aplicación alimentada por cosecha de energía. Se analizará la influencia del modo de funcionamiento tanto en la velocidad de transmisión como en la latencia. Por último, con la ayuda de un osciloscopio, se dispondrá de un esquema de medición que permita visualizar la gráfica del consumo en función del tiempo tratando de identificar las fases involucradas en la comunicación.

6.1 Introducción

En NB-IoT la mejora en la cobertura se debe al incremento del MCL, lo que permite alcanzar dispositivos en zonas lejanas o con poco nivel de señal. Una de las principales técnicas adoptadas para alcanzar este objetivo, consiste en incrementar el número de repeticiones con el fin de combinarlas y lograr un incremento en la SNR. El número de retransmisiones puede llegar a 128 en el caso del uplink y hasta 2048 para el downlink.

Sin embargo, la mejora en la cobertura a expensas de utilizar repeticiones introduce una reducción en la tasa de datos efectiva a la vez que incrementa el retraso en el envío de información y ocasiona un incremento en el consumo de energía.

Una técnica que ayudaría a reducir el consumo de energía sería apagar completamente el módulo de radio cuando no está siendo utilizado. Sin embargo, esto requeriría que el UE se registre en la red cada vez que se vuelve a encender, lo que conlleva un gasto extra de energía en el procedimiento de reconexión.

En el caso de los dispositivos NB-IoT, además del modo de recepción discontinua extendida (eDRX), existe el modo de ahorro de energía (PSM), el cual es controlado por dos valores de temporizado, T3324 (control del tiempo activo) y T3412 (control del TAU periódico). Mientras el UE está en modo PSM, no puede ser contactado por la red, sin embargo, todavía se encuentra registrado en ella, por lo que al salir del estado de PSM no necesitará rehacer el procedimiento de conexión.

6.2 Procedimiento

El procedimiento para obtener el consumo de corriente instantáneo del modem puede realizarse de manera directa, utilizando un instrumento específico para tal fin o de manera indirecta, con la ayuda de un osciloscopio en modo adquisidor o una placa con capacidad de muestrear señales analógicas como Arduino UNO. Las pruebas realizadas en esta guía utilizaron los dos últimos métodos.

El esquema utilizado para las mediciones se muestra en la Figura 6.1, mientras que el conexionado del instrumental se presenta en la Figura 6.2. Notar que el método indirecto requiere medir caída de tensión sobre una resistencia de conocida para luego proceder a calcular el valor de la corriente. Este método está lejos de ser exacto, pero tiene la ventaja que nos permite conocer la relación de magnitudes que alcanza la corriente en las diferentes etapas de intercambio de información entre el modem y la red. Los ensayos se realizaron sobre el kit Arduino MKR NB 1500.

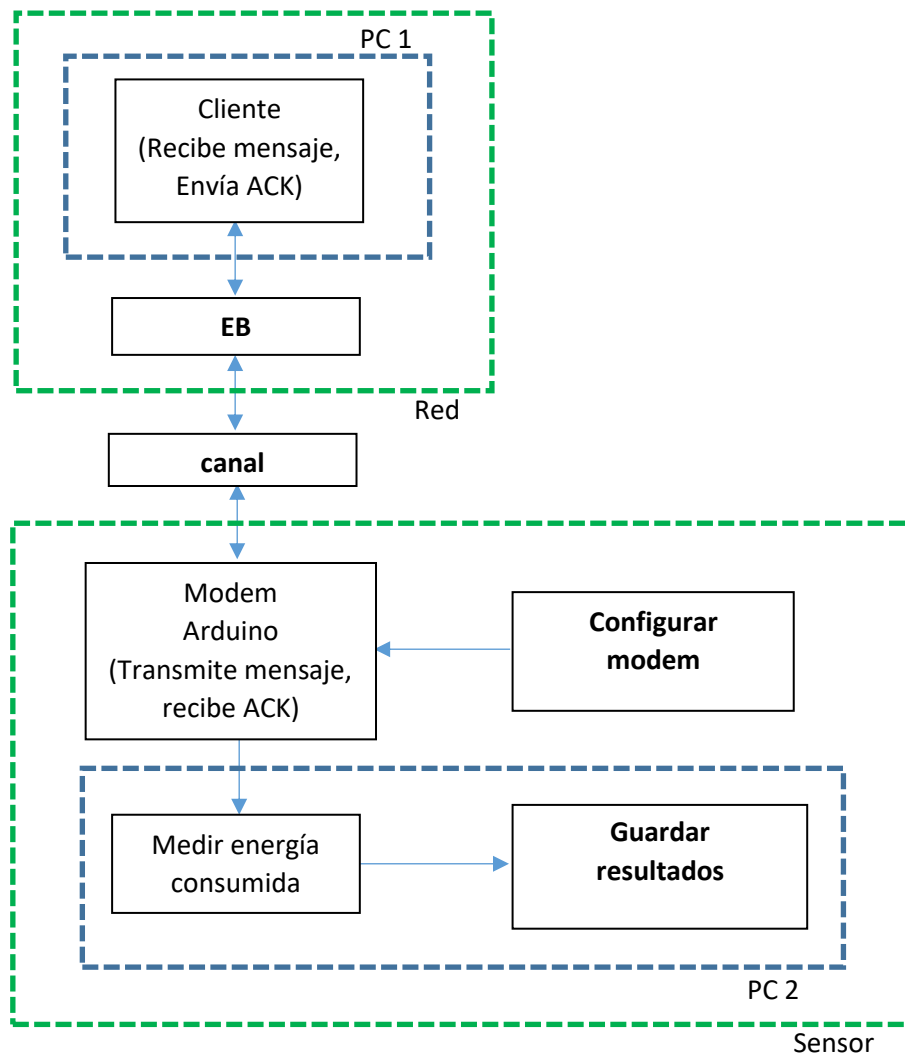


Figura 6.1. Esquema de conexiones para la prueba de consumo.

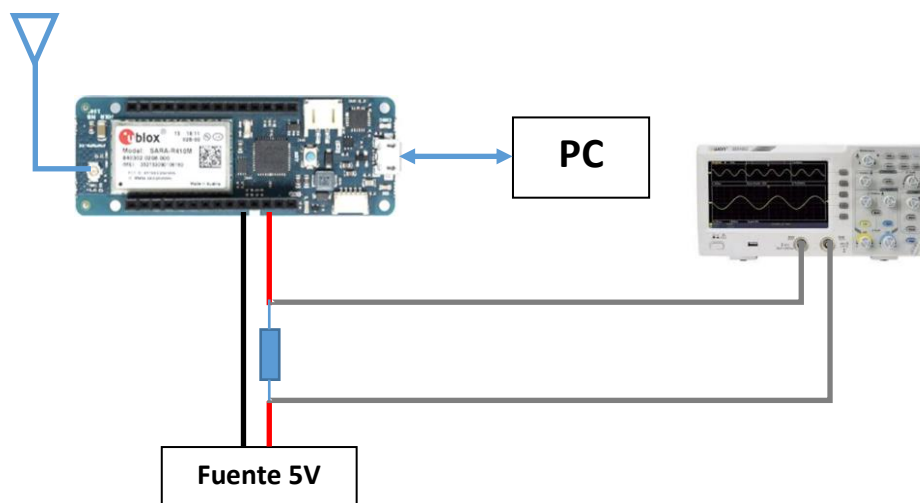


Figura 6.2. Conexionado del instrumental utilizado en las pruebas.

6.2.1 Conexión a la red

Partiendo del estado de reset del modem, comienza a registrarse la corriente demandada por el mismo con la ayuda de un osciloscopio (Figura 3).

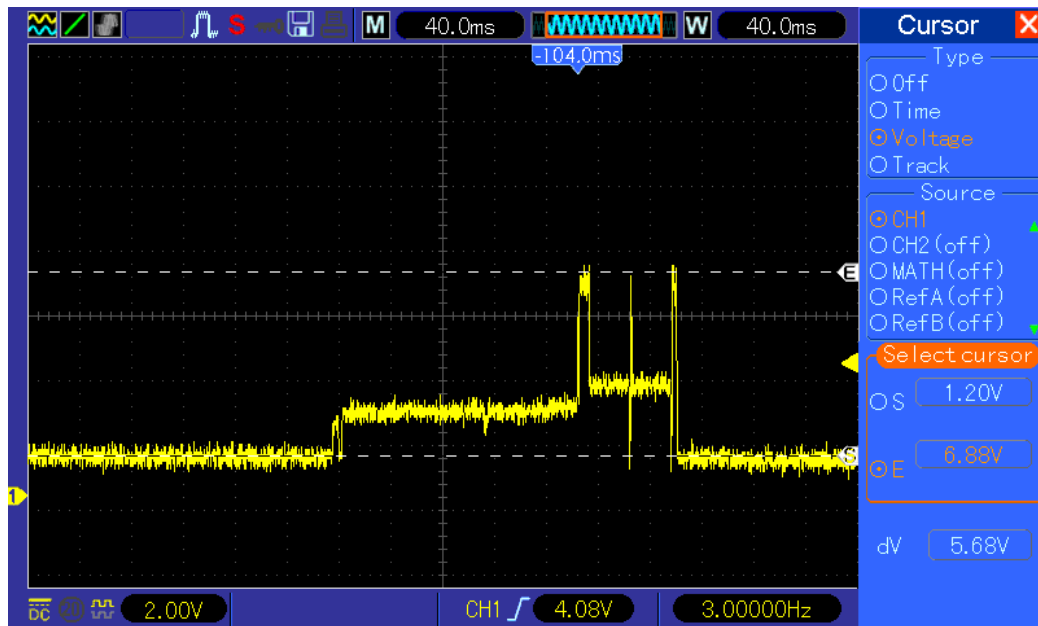


Figura 3. Inicio de la conexión a la red por parte del modem.

El mismo procedimiento fue repetido utilizando una placa Arduino UNO capturando un intervalo de tiempo mayor (Figura 4).

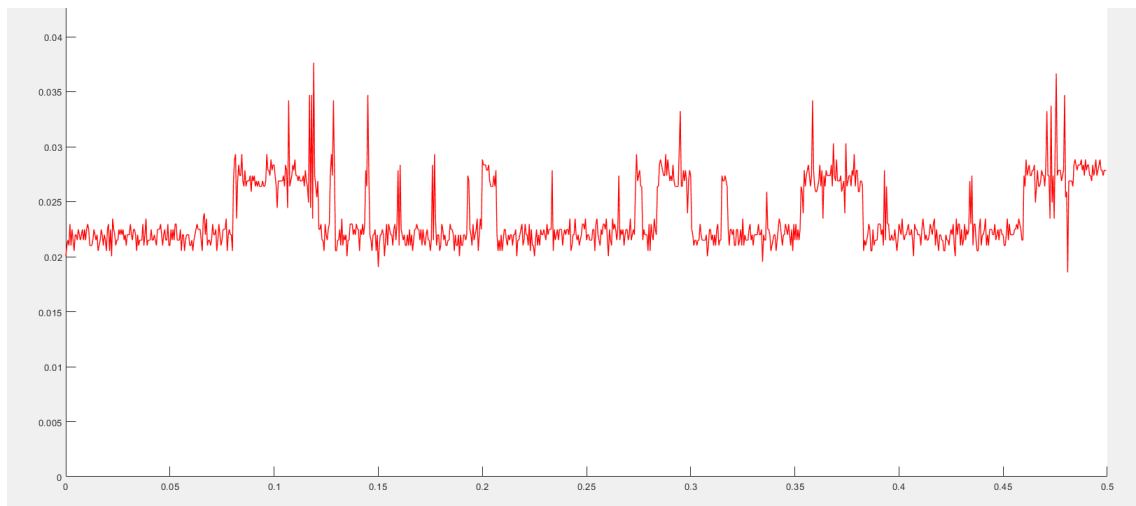


Figura 4. Inicio de la conexión a la red por parte del modem.

La gráfica obtenida en las figuras anteriores, corresponde a la negociación de conexión entre el modem y la red.

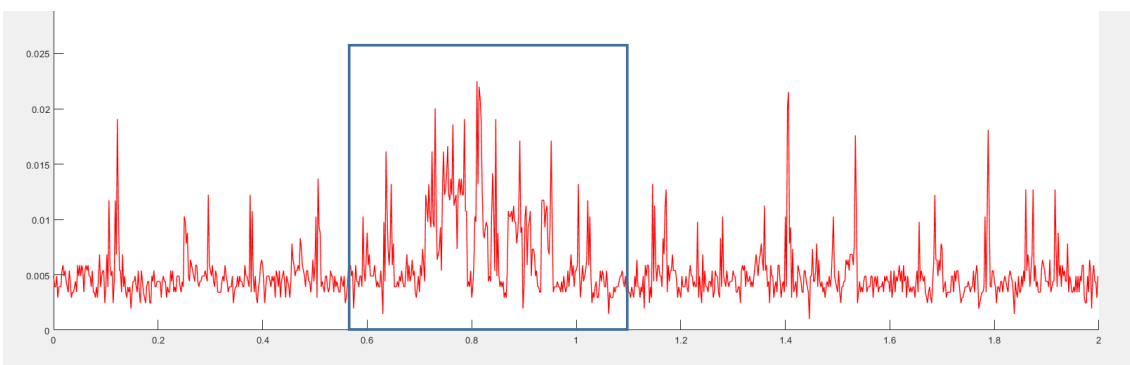
6.2.2 Envío de datos

Para las pruebas de consumo con envío de datos, se utilizó el envío de paquetes Ping con tramas de diferentes tamaños. Los resultados obtenidos se muestran las Figuras siguientes.

Ejemplo 1: 8 paquetes de datos de 8 bytes cada uno

```
pi@enviot:~$ sudo hping3 -l -d 8 -c 8 8.8.8.8
HPING 8.8.8.8 (ppp0 8.8.8.8): icmp mode set, 28 headers + 8 data bytes
len=36 ip=8.8.8.8 ttl=116 id=0 icmp_seq=0 rtt=559.3 ms
len=36 ip=8.8.8.8 ttl=116 id=0 icmp_seq=1 rtt=839.0 ms
len=36 ip=8.8.8.8 ttl=116 id=0 icmp_seq=2 rtt=478.8 ms
len=36 ip=8.8.8.8 ttl=116 id=0 icmp_seq=3 rtt=758.5 ms
len=36 ip=8.8.8.8 ttl=116 id=0 icmp_seq=4 rtt=398.2 ms
len=36 ip=8.8.8.8 ttl=116 id=0 icmp_seq=5 rtt=677.9 ms
len=36 ip=8.8.8.8 ttl=116 id=0 icmp_seq=6 rtt=337.6 ms
len=36 ip=8.8.8.8 ttl=116 id=0 icmp_seq=7 rtt=617.4 ms

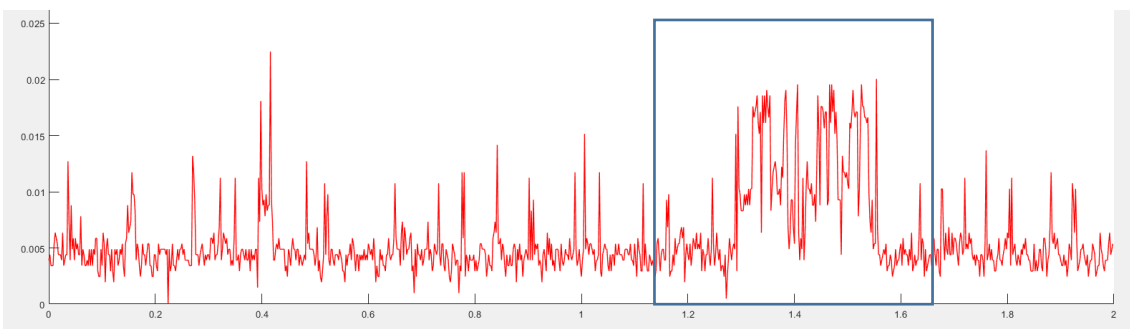
--- 8.8.8.8 hping statistic ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 337.6/583.4/839.0 ms
```



Ejemplo 2: 8 paquetes de datos de 800 bytes cada uno

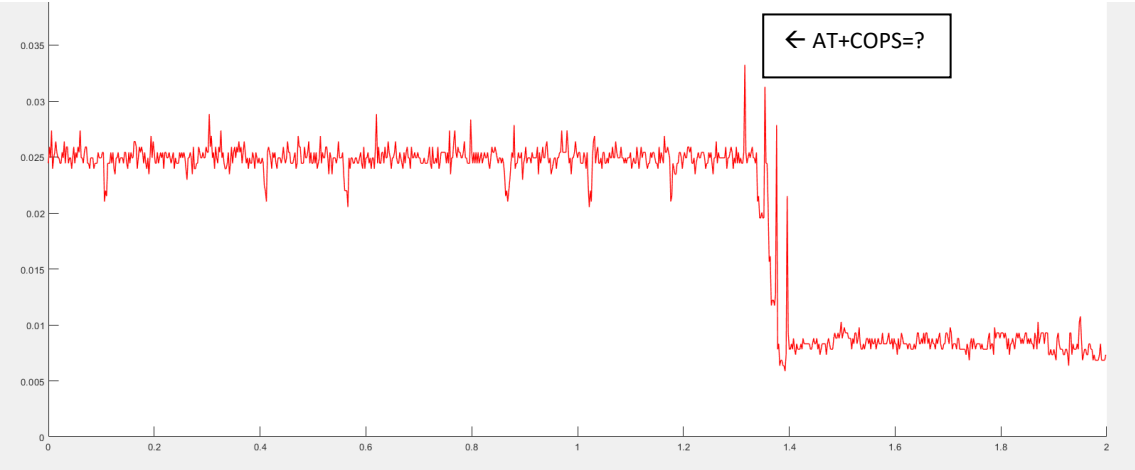
```
pi@enviot:~$ sudo hping3 -l -d 800 -c 8 8.8.8.8
HPING 8.8.8.8 (ppp0 8.8.8.8): icmp mode set, 28 headers + 800 data bytes
len=96 ip=8.8.8.8 ttl=116 id=0 icmp_seq=0 rtt=1418.8 ms
len=96 ip=8.8.8.8 ttl=116 id=0 icmp_seq=1 rtt=1048.5 ms
len=96 ip=8.8.8.8 ttl=116 id=0 icmp_seq=2 rtt=1348.2 ms
len=96 ip=8.8.8.8 ttl=116 id=0 icmp_seq=3 rtt=967.9 ms
len=96 ip=8.8.8.8 ttl=116 id=0 icmp_seq=4 rtt=1327.5 ms
len=96 ip=8.8.8.8 ttl=116 id=0 icmp_seq=5 rtt=947.2 ms
len=96 ip=8.8.8.8 ttl=116 id=0 icmp_seq=6 rtt=1186.9 ms
len=96 ip=8.8.8.8 ttl=116 id=0 icmp_seq=7 rtt=806.6 ms

--- 8.8.8.8 hping statistic ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 806.6/1131.5/1418.8 ms
```

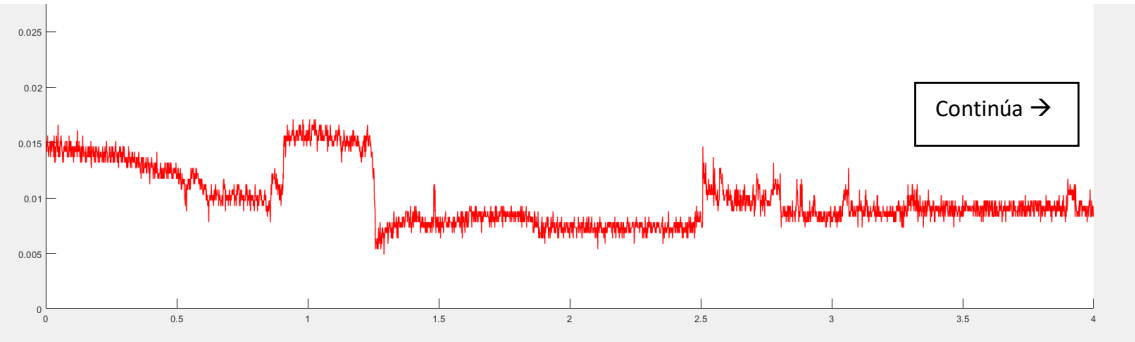
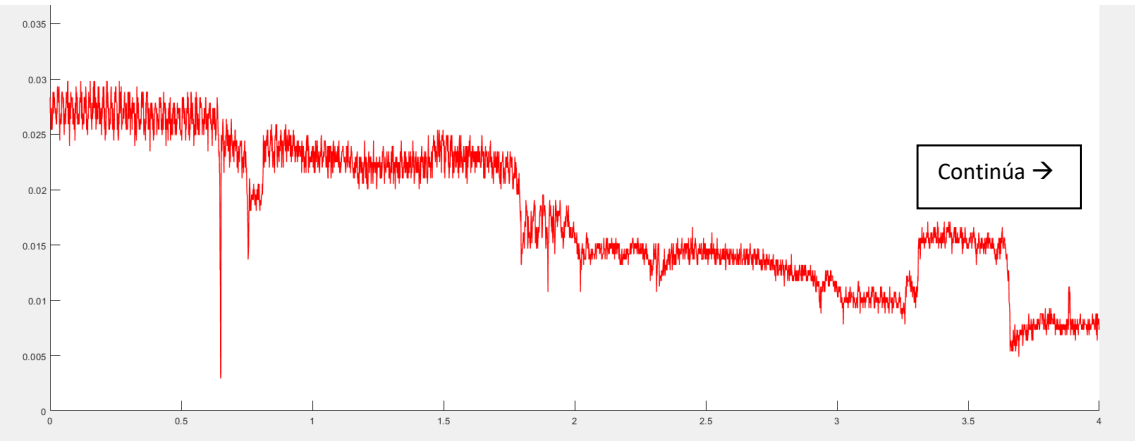


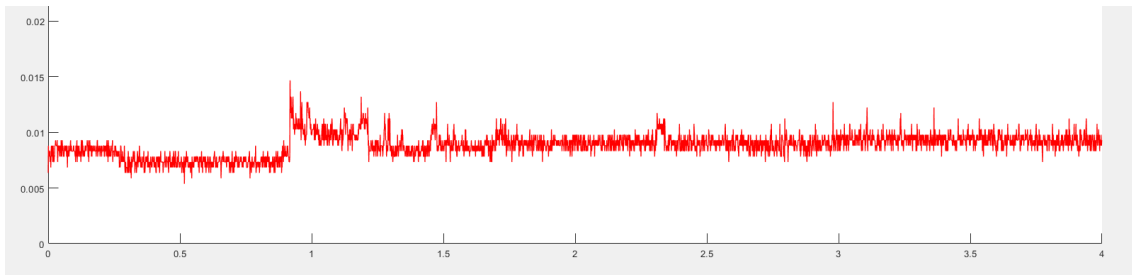
Ejemplo 3: escaneo de redes (AT+COPS=?)

```
OK
AT+COPS=?
+COPS: (3,"722 310","722 310","722310",9),(1,"722 07","722 07","72207",9),,(0,1,2,3,4),(0,1,2)
OK
```

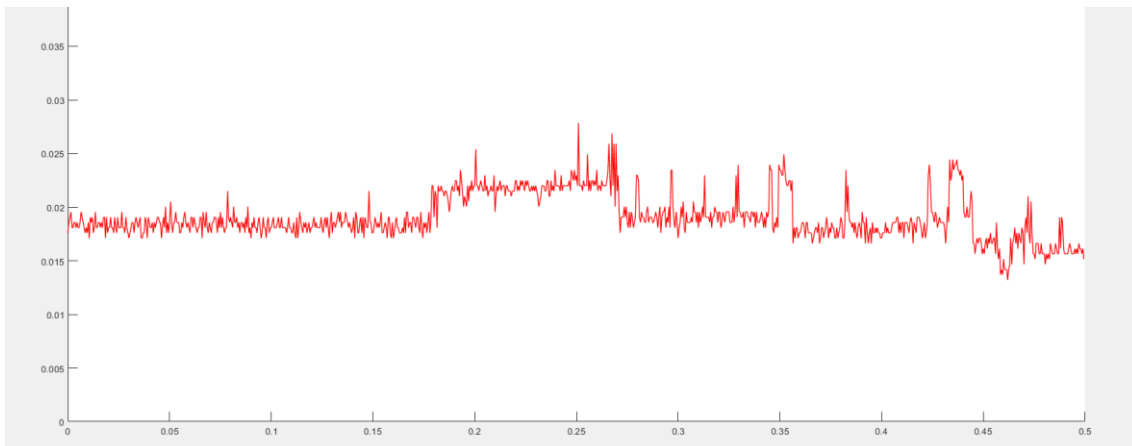


Ejemplo 4: dispositivo en mínimo consumo sin PSM activo (AT+CPSMS=0, AT+CFUN=0)





Ejemplo 5: ejecución del comando AT+CREG=2



El ajuste del modo PSM requiere del envío de comandos específicos para negociar con la red los valores de temporizado que utilizará el modem. En el Apéndice A3, se listan los comandos AT que controlan este modo de operación para el modem BG96.

6.3 Preguntas de Repaso

- ¿Podrían Identificarse los estados de conexión de la Figura 3?
- Repase los ejemplos 1 y 2. En ambos casos no hubo pérdida de paquetes. Si hubiera existido una gran cantidad de pérdida de paquetes ¿Hubiera cambiado el pico de consumo de corriente?
- ¿Qué hace el dispositivo en el modo de mínimo consumo AT+CFUN=0?
- ¿Qué hace el dispositivo al ejecutar AT+CREG=2?

Apéndice

A.1 Instalación y uso de Iperf3

A continuación, se dan los pasos a seguir para efectuar las pruebas utilizando PCs con sistema operativo Windows. El procedimiento para ejecutar Iperf3 sobre Linux es similar.

- 1) Descomprimir “iperf-3.1.3-win64.zip” o (“iperf-3.1.3-win32.zip”) en una carpeta fácilmente accesible (para mayor comodidad). Por ejemplo, D:\iperf-3. Dentro de esta carpeta “iperf” debe estar todo el contenido del “zip”.
- 2) Abrir la consola de Windows: botón derecho sobre logo Windows → Ejecutar → cmd
- 3) Ir a la carpeta **D:\iperf-3**

Usar “cd\” para llegar a la raíz.

Cambiar de unidad con “D:”. (Esto se adecua según la estructura del disco).

Después, se debe cambiar al directorio donde está el ejecutable iperf: “cd iperf-3” (o el nombre que se haya puesto a la carpeta).

Ahora se debería estar en: “D:\iperf-3>”

- 4) Para establecer el **servidor iperf3**, (en la computadora “A”) escribir lo siguiente:

Iperf3.exe -s

Si todo está correcto, se habrá iniciado el servidor iperf y se mostrará la leyenda:

“Server listening on 5201”

El puerto 5201 es el que utiliza por defecto Iperf3.

- 5) A continuación, se debe ejecutar el **cliente iperf3**. Esto puede hacerse en la misma PC que actúa como servidor, en otra PC dentro de la misma LAN o en una PC fuera de la LAN (internet). Para este último caso, puede ser necesario hacer una redirección de puertos (port forwarding) en el router principal.
- 6) **Cliente 1:** la misma PC que actúa como servidor.

- a) Se debe abrir otra consola (cmd) y llegar a:

“D:\iperf-3>”

- b) Escribir lo siguiente:

Iperf3.exe -c 127.0.0.1

- c) Si todo está correcto, el cliente se conectará al servidor y arrojará los resultados de la conexión:

“Connecting to host 127.0.0.1, port 5201”

“local 127.0.0.1 port 57110 connected to 127.0.0.1 port 5201”

```

D:\No sincronizado\iperf-3>iperf3.exe -c 127.0.0.1
Connecting to host 127.0.0.1, port 5201
[ 4] local 127.0.0.1 port 57752 connected to 127.0.0.1 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-1.00      sec    263 MBytes  2.21 Gbits/sec
[ 4] 1.00-2.00      sec    241 MBytes  2.02 Gbits/sec
[ 4] 2.00-3.00      sec    264 MBytes  2.21 Gbits/sec
[ 4] 3.00-4.00      sec    213 MBytes  1.78 Gbits/sec
[ 4] 4.00-5.00      sec    267 MBytes  2.24 Gbits/sec
[ 4] 5.00-6.00      sec    263 MBytes  2.20 Gbits/sec
[ 4] 6.00-7.00      sec    248 MBytes  2.09 Gbits/sec
[ 4] 7.00-8.00      sec    217 MBytes  1.82 Gbits/sec
[ 4] 8.00-9.00      sec    268 MBytes  2.25 Gbits/sec
[ 4] 9.00-10.00     sec    264 MBytes  2.22 Gbits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-10.00     sec    2.45 GBytes  2.10 Gbits/sec
[ 4] 0.00-10.00     sec    2.45 GBytes  2.10 Gbits/sec
iperf Done.
D:\No sincronizado\iperf-3>

```

Figura A.1.1. Resultados Iperf3 del cliente local host.

7) **Ciente 2:** otra PC conectada en la misma LAN.

- a) Se debe abrir la consola (cmd) y llegar a:

"D:\iperf-3>"

- b) Escribir lo siguiente, teniendo en cuenta la dirección IP local del servidor:

iperf3.exe -c 192.168.0.102

- c) Si todo está correcto, el cliente se conectará al servidor y arrojará los resultados de la conexión:

"Connecting to host 192.168.0.102, port 5201"

"local 192.168.0.105 port 49904 connected to 192.168.0.102 port 5201"

```

D:\No sincronizado\iperf-3>iperf3.exe -c 192.168.0.102
Connecting to host 192.168.0.102, port 5201
[ 4] local 192.168.0.104 port 54851 connected to 192.168.0.102 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-1.00      sec    768 KBytes  6.28 Mbits/sec
[ 4] 1.00-2.01      sec    128 KBytes  1.04 Mbits/sec
[ 4] 2.01-3.02      sec    640 KBytes  5.20 Mbits/sec
[ 4] 3.02-4.01      sec    256 KBytes  2.12 Mbits/sec
[ 4] 4.01-5.01      sec    384 KBytes  3.14 Mbits/sec
[ 4] 5.01-6.01      sec    512 KBytes  4.21 Mbits/sec
[ 4] 6.01-7.01      sec    256 KBytes  2.10 Mbits/sec
[ 4] 7.01-8.00      sec    512 KBytes  4.22 Mbits/sec
[ 4] 8.00-9.01      sec    384 KBytes  3.13 Mbits/sec
[ 4] 9.01-10.01     sec    384 KBytes  3.12 Mbits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-10.01     sec    4.12 MBytes  3.46 Mbits/sec
[ 4] 0.00-10.01     sec    3.88 MBytes  3.25 Mbits/sec
iperf Done.

```

Figura A.1.2. Resultados Iperf3 del cliente LAN.

8) **Ciente 3:** otra PC conectada fuera de la LAN (internet).

Nota: Para este ensayo, el servidor deberá situarse en una red fuera del ámbito de la UNS ya que al estar filtrado el puerto de Iperf3 la prueba no funcionará.

- a) Se debe abrir la consola (cmd) y llegar a:

"D:\iperf-3>"

- b) Escribir lo siguiente, teniendo en cuenta la dirección IP pública del servidor, habiendo abierto y redireccionando previamente los puertos del router principal:

lperf3.exe -c 181.9.0.130

- c) Si todo está correcto, el cliente se conectará al servidor y arrojará los resultados de la conexión:

“Accepted connection from 181.9.148.214, port 4297”

“local 181.9.0.130 port 5201 connected to 181.9.148.214 port 4298”.

Lo que se está haciendo es ejecutar el cliente lperf3, apuntado a la dirección pública 181.9.0.130.

```
-----
Server listening on 5201
-----
Accepted connection from 181.9.148.214, port 4297 [ 5] local 181.9.0.130 port
5201 connected to 181.9.148.214 port 4298
[ ID] Interval      Transfer  Bitrate
[ 5] 0.00-1.00 sec  5.46 KBytes  44.7 Kbits/sec
[ 5] 1.00-2.00 sec  9.56 KBytes  78.3 Kbits/sec
[ 5] 2.00-3.00 sec  10.9 KBytes  89.5 Kbits/sec
[ 5] 3.00-4.00 sec  10.9 KBytes  89.5 Kbits/sec
[ 5] 4.00-5.00 sec  12.3 KBytes  101 Kbits/sec
[ 5] 5.00-6.00 sec  9.56 KBytes  78.3 Kbits/sec
[ 5] 6.00-7.00 sec  10.9 KBytes  89.5 Kbits/sec
[ 5] 7.00-8.00 sec  9.56 KBytes  78.3 Kbits/sec
[ 5] 7.00-8.00 sec  9.56 KBytes  78.3 Kbits/sec
-----
[ ID] Interval      Transfer  Bitrate
[ 5] 0.00-8.00 sec  86.0 KBytes  88.1 Kbits/sec      receiver
iperf3: the client has terminated
```

Figura A.1.3. Resultados lperf3 del cliente NB-IoT (lado servidor).

La Tabla A.1.1 resume una lista con los comandos de lperf3.

Server or Client:	
-p, --port #	server port to listen on/connect to
-f, --format [kmgKMG]	format to report: Kbits, Mbits, KBytes, MBytes
-i, --interval #	seconds between periodic bandwidth reports
-F, --file name	xmit/recv the specified file
-B, --bind <host>	bind to a specific interface
-V, --verbose	more detailed output
-J, --json	output in JSON format
--logfile f	send output to a log file
-d, --debug	emit debugging output
-v, --version	show version information and quit
-h, --help	show this message and quit
Server specific:	
-s, --server	run in server mode
-D, --daemon	run the server as a daemon
-l, --pidfile file	write PID file
-1, --one-off	handle one client connection then exit
Client specific:	
-c, --client <host>	run in client mode, connecting to <host>
-u, --udp	use UDP rather than TCP
-b, --bandwidth #[KMG][/#]	target bandwidth in bits/sec (0 for unlimited) (default 1 Mbit/sec for UDP, unlimited for TCP) (optional slash and packet count for burst mode)
-t, --time #	time in seconds to transmit for (default 10 secs)
-n, --bytes #[KMG]	number of bytes to transmit (instead of -t)
-k, --blockcount #[KMG]	number of blocks (packets) to transmit (instead of -t or -n)
-l, --len #[KMG]	length of buffer to read or write (default 128 KB for TCP, 8 KB for UDP)
--cport <port>	bind to a specific client port (TCP and UDP, default: ephemeral port)
-P, --parallel #	number of parallel client streams to run
-R, --reverse	run in reverse mode (server sends, client receives)
-w, --window #[KMG]	set window size / socket buffer size
-M, --set-mss #	set TCP/SCTP maximum segment size (MTU - 40 bytes)
-N, --no-delay	set TCP/SCTP no delay, disabling Nagle's Algorithm
-4, --version4	only use IPv4
-6, --version6	only use IPv6
-S, --tos N	set the IP 'type of service'
-Z, --zerocopy	use a 'zero copy' method of sending data
-O, --omit N	omit the first n seconds
-T, --title str	prefix every output line with this string
--get-server-output	get results from server
--udp-counters-64bit	use 64-bit counters in UDP test packets

Tabla A.1.1. Resumen de comandos Iperf3.

A.2 Parámetros de configuración de Ping

La herramienta Ping está disponible por defecto tanto en Windows como en Linux. Dependiendo del S.O. pueden existir pequeñas variaciones en los parámetros de configuración.

Windows	
ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS] [-r count] [-s count] [[-j host-list] [-k host-list]] [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p] [-4] [-6] nombre_destino	
Opciones:	
-t	Hacer ping al host especificado hasta que se detenga. Para ver estadísticas y continuar, presione Ctrl-Interrumpir; para detener, presione Ctrl+C.
-a	Resolver direcciones en nombres de host.
-n count	Número de solicitudes de eco para enviar.
-l size	Enviar tamaño de búfer.
-f	Establecer marca No fragmentar en paquetes (solo IPv4).
-i TTL	Período de vida.
-v TOS	Tipo de servicio (solo IPv4. Esta opción está desusada y no tiene ningún efecto sobre el campo de tipo de servicio del encabezado IP).
-r count	Registrar la ruta de saltos de cuenta (solo IPv4).
-s count	Marca de tiempo de saltos de cuenta (solo IPv4).
-j host-list	Ruta de origen no estricta para lista-host (solo IPv4).
-k host-list	Ruta de origen estricta para lista-host (solo IPv4).
-w timeout	Tiempo de espera en milisegundos para cada respuesta.
-R	Usar encabezado de enrutamiento para probar también la ruta inversa (solo IPv6). Por RFC 5095 el uso de este encabezado de enrutamiento ha quedado en desuso. Es posible que algunos sistemas anulen solicitudes de eco si usa este encabezado.
-S srcaddr	Dirección de origen que se desea usar.
-c compartment	Enrutamiento del identificador del compartimiento.
-p	Hacer ping a la dirección del proveedor de Virtualización de red de Hyper-V.
-4	Forzar el uso de IPv4.
-6	Forzar el uso de IPv6.
Linux	
Usage: ping [-aAbBdDfhLnOqrRUvV64] [-c count] [-i interval] [-I interface] [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos] [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline] [-W timeout] [hop1 ...] destination	
Opciones:	
Option Description	
a	Use this option for a beep sound when the peer is reachable
b	Use this option to allow pinging a broadcast address
B	Use this option if you do not want to allow the ping to change the source address of the probe
c (count)	Use this option to set the number of times to send the ping request

d	Use this option to set the SO-DEBUG option on the socket being used
f	Use this option to flood the network by sending hundred or more packets per second
i (interval)	Use this option to specify an interval between successive packet transmissions. The default value of interval is 1 second
I (interface address)	Use this option to set source address to the specified interface address. This option is required when pinging IPv6 link local address. Its argument can be an IP address or name of the device.
I (preload)	Use this option to set the number of packets to send without waiting for a reply. For selecting a value more than 3, you need to be a super user.
n	Use this option to display network addresses as numbers rather than as hostnames
q	Use this option to display a quiet output. It means that only the summary is displayed at startup and finish time
T (ttl)	Use this option to set the Time To Live
v	Use this option for verbose output
V	Use this option to display the version and exit
w (deadline)	Use this option to specify a timeout, in seconds, before ping exits, regardless of how many packets have been sent or received.
W (timeout)	Use this option to set the time(seconds) to wait for a response
-s	Packet size (20 to 9096 bytes).

A.3 Configuración del Modo de Ahorro de Energía en modem BG96

Comando: [AT+CPSMS \(Power Saving Mode Setting\)](#)

AT+CPSMS Power Saving Mode Setting	
Test Command AT+CPSMS=?	Response +CPSMS: (list of supported <mode>s),(list of supported
	<Requested_Periodic-RAU>s),(list of supported <Requested_GPRS-READY-timer>s),(list of supported <Requested_Periodic-TAU>s),(list of supported <Requested_Active-Time>s)
	OK
Write Command AT+CPSMS=[<mode>,<Requested_Periodic-RAU>,<Requested_GPRS-READY-timer>,<Requested_Periodic-TAU>,<Requested_Active-Time>]]]]	Response OK If there is any error, response: ERROR
Read Command AT+CPSMS?	Response +CPSMS: <mode>,<Requested_Periodic-RAU>,<Requested_GPRS-READY-timer>,<Requested_Periodic-TAU>,<Requested_Active-Time>
	OK
Maximum Response Time	300ms
Reference 3GPP TS 27.007	

Parameter

<mode>	Integer type. Disable or enable the use of PSM in the UE. 0 Disable the use of PSM 1 Enable the use of PSM
<Requested_Periodic-RAU>	String type. One byte in an 8 bit format. Requested extended periodic RAU value (T3312) to be allocated to the UE in GERAN. (e.g. "01000111" equals to 70 hours) Bits 5 to 1 represent the binary coded timer value Bits 6 to 8 define the timer value unit as follows: Bits 8 7 6 0 0 0 value is incremented in multiples of 10 minutes 0 0 1 value is incremented in multiples of 1 hour 0 1 0 value is incremented in multiples of 10 hours 0 1 1 value is incremented in multiples of 2 seconds 1 0 0 value is incremented in multiples of 30 seconds 1 0 1 value is incremented in multiples of 1 minute
<Requested_GPRS-READY-timer>	String type. One byte in an 8 bit format. Requested GPRS READY timer value (T3314) to be allocated to the UE in GERAN. (e.g. "01001010" equals to 1 hours) Bits 5 to 1 represent the binary coded timer value

	<p>Bits 6 to 8 define the timer value unit as follows:</p> <p>Bits</p> <p>8 7 6</p> <p>0 0 0 value is incremented in multiples of 2 seconds</p> <p>0 0 1 value is incremented in multiples of 1 minute</p> <p>0 1 0 value is incremented in multiples of decihours</p> <p>1 1 1 value indicates that the timer is deactivated.</p>
<Requested_Periodic-TAU>	<p>String type. One byte in an 8 bit format. Requested extended periodic TAU value (T3412) to be allocated to the UE in E-UTRAN.</p> <p>(e.g. "00001010" equals to 100 minutes)</p> <p>Bits 5 to 1 represent the binary coded timer value.</p> <p>Bits 6 to 8 define the timer value unit as follows:</p> <p>Bits</p> <p>8 7 6</p> <p>0 0 0 value is incremented in multiples of 10 minutes</p> <p>0 0 1 value is incremented in multiples of 1 hour</p> <p>0 1 0 value is incremented in multiples of 10 hours</p> <p>0 1 1 value is incremented in multiples of 2 seconds</p> <p>1 0 0 value is incremented in multiples of 30 seconds</p> <p>1 0 1 value is incremented in multiples of 1 minute</p>
<Requested_Active-Time>	<p>String type. One byte in an 8 bit format. Requested Active Time value (T3324) to be allocated to the UE.</p> <p>(e.g. "00001111" equals to 30 seconds)</p> <p>Bits 5 to 1 represent the binary coded timer value.</p> <p>Bits 6 to 8 define the timer value unit as follows:</p> <p>Bits</p> <p>8 7 6</p> <p>0 0 0 value is incremented in multiples of 2 seconds</p> <p>0 0 1 value is incremented in multiples of 1 minute</p> <p>0 1 0 value is incremented in multiples of decihours</p> <p>1 1 1 value indicates that the timer is deactivated.</p>

Example

```
AT+CPSMS=1,,"00000100","00001111" //Set the requested T3412 value to 40 minutes, and set the
requested T3324 value to 30 seconds.
OK
```

Comando: [AT+QPSMS \(Extended Power Saving Mode Setting\)](#)

AT+QPSMS Extended Power Saving Mode Setting	
Test Command AT+QPSMS=?	Response +QPSMS: (list of supported <mode>s),(list of supported <Requested_Periodic-RAU>s),(list of supported <Requested_GPRS-READY-timer>s),(list of supported <Requested_Periodic-TAU>s),(list of supported <Requested_Active-Time>s) OK
Write Command AT+QPSMS=[<mode>,<Requested_Periodic-RAU>,<Requested_GPRS-READY-timer>,<Requested_Periodic-TAU>,<Requested_Active-Time>]]]]	Response OK If there is any error, response: ERROR
Read Command AT+QPSMS?	Response +QPSMS: <mode>,<Network_Periodic-RAU>,<Network_GPRS-READY-timer>,<Network_Periodic-TAU>,<Network_Active-Time> OK
Maximum Response Time	300ms

Parameter

<mode>	Integer type. Disable or enable the use of PSM in the UE. 0 Disable the use of PSM 1 Enable the use of PSM
<Requested_Periodic-RAU>	String type. One byte in an 8 bit format. Requested extended periodic RAU value (T3312) to be allocated to the UE in GERAN. (e.g. "01000111" equals to 70 hours) Bits 5 to 1 represent the binary coded timer value. Bits 6 to 8 define the timer value unit as follows: Bits 8 7 6 0 0 0 value is incremented in multiples of 10 minutes 0 0 1 value is incremented in multiples of 1 hour 0 1 0 value is incremented in multiples of 10 hours 0 1 1 value is incremented in multiples of 2 seconds 1 0 0 value is incremented in multiples of 30 seconds 1 0 1 value is incremented in multiples of 1 minute
<Requested_GPRS-READY-timer>	String type. One byte in an 8 bit format. Requested GPRS

	<p>READY timer value (T3314) to be allocated to the UE in GERAN. (e.g. "01001010" equals to 1 hours)</p> <p>Bits 5 to 1 represent the binary coded timer value.</p> <p>Bits 6 to 8 define the timer value unit as follows:</p> <p>Bits 8 7 6</p> <p>0 0 0 value is incremented in multiples of 2 seconds</p> <p>0 0 1 value is incremented in multiples of 1 minute</p> <p>0 1 0 value is incremented in multiples of decihours</p> <p>1 1 1 value indicates that the timer is deactivated.</p>
<Requested_Periodic-TAU>	<p>String type. One byte in an 8 bit format. Requested extended periodic TAU value (T3412) to be allocated to the UE in E-UTRAN. (e.g. "00001010" equals to 100 minutes)</p> <p>Bits 5 to 1 represent the binary coded timer value.</p> <p>Bits 6 to 8 define the timer value unit as follows:</p> <p>Bits 8 7 6</p> <p>0 0 0 value is incremented in multiples of 10 minutes</p> <p>0 0 1 value is incremented in multiples of 1 hour</p> <p>0 1 0 value is incremented in multiples of 10 hours</p> <p>0 1 1 value is incremented in multiples of 2 seconds</p> <p>1 0 0 value is incremented in multiples of 30 seconds</p> <p>1 0 1 value is incremented in multiples of 1 minute</p>
<Requested_Active-Time>	<p>String type. One byte in an 8 bit format. Requested Active Time value (T3324) to be allocated to the UE. (e.g. "00001111" equals to 1 minute)</p> <p>Bits 5 to 1 represent the binary coded timer value.</p> <p>Bits 6 to 8 define the timer value unit as follows:</p> <p>Bits 8 7 6</p> <p>0 0 0 value is incremented in multiples of 2 seconds</p> <p>0 0 1 value is incremented in multiples of 1 minute</p> <p>0 1 0 value is incremented in multiples of decihours</p> <p>1 1 1 value indicates that the timer is deactivated.</p>
<Network_Periodic-RAU>	<p>Integer type. Extended periodic RAU value (T3312) to be allocated to the UE in GERAN, and the value is specified by network.</p>
<Network_GPRS-READY-timer>	<p>Integer type. GPRS READY timer value (T3314) to be allocated to the UE in GERAN, and the value is specified by network.</p>
<Network_Periodic-TAU>	<p>Integer type. Extended periodic TAU value (T3412) to be allocated to the UE in E-UTRAN, and the value is specified by network.</p>
<Network_Active-Time>	<p>Integer type. Active timer value (T3324) to be allocated to the UE in E-UTRAN, and the value is specified by network.</p>

Example

```

AT+QPSMS=1,,,"00000100","00001111" //Set the requested T3412 value to 40 minutes, and set the
                                     requested T3324 value to 30 seconds.
OK
AT+QPSMS?                               //Query the PSM mode and the timer from network.
+QPSMS: 1,,,"86400","2"
OK

```

Comando: [AT+QPSMCFG](#) (Extended Power Saving Mode Setting)

AT+QPSMCFG PSM Feature and Minimum Threshold Value Setting	
Test Command AT+QPSMCFG=?	Response +QPSMCFG: (list of supported <threshold>s),(list of supported <psm_version>s) OK
Write Command AT+QPSMCFG=[<threshold>,<psm_version>]	Response OK If there is any error, response: ERROR
Read Command AT+QPSMCFG?	Response +QPSMCFG: <threshold>,<psm_version> OK

Parameter

<threshold>	Minimum threshold value to enter PSM. Range: <u>20</u> -4294967295. Unit: second.
<psm_version>	Bitmask to indicate PSM modes (1 – Enable/0 – Disable). Each bit is configured independently. Range: 0- <u>4</u> -15. Bit 0 – PSM without network coordination Bit 1 – Rel 12 PSM without context retention Bit 2 – Rel 12 PSM with context retention Bit 3 – PSM in between eDRX cycles

Example

```

AT+QPSMCFG=100           //Set the threshold to 100 seconds.
OK
AT+QPSMCFG?              //Query the threshold value and PSM mode.
+QPSMCFG: 100,5
OK

```

Referencias

- [1] Aduino MKR NB 1500. Low Power Narrowband Board. [STORE.ARDUINO.CC/ARDUINO-MKR-NB-1500](https://store.arduino.cc/arduino-mkr-nb-1500).
- [2] Aduino MKR NB 1500 Starter Guide.
- [3] Arduino MKR NB 1500 User Guide.
- [4] SARA-R4 series, Size-optimized LTE Cat M1 / NB1 / GPRS modules, AT commands manual, R19 - 09-Dec-2020.
- [5] SARA-R4 series, LTE Cat M1 / NB1 and EGPRS modules, Data sheet, R20 - 28-Aug-2020.
- [6] SARA-R4 series, System integration manual, R14 - 23-Dec-2019.
- [7] SARA-R4 series, LwM2M objects and commands, Application note, R02 - 11-Jun-2020.
- [8] Mariusz Ciesielski, U-BLOX IOT MODULES, Microdis, 2018.