

Proyecto Taller de Redes Inalámbricas

Curso 2024

Parte II - Laboratorio 2

Trabajo grupal

Para el resto del proyecto trabajarán en grupos de 2 o 3 personas, cada grupo contará con un **kit_IoT**.

Objetivo Global

El objetivo global del proyecto es introducir a los estudiantes en el mundo de IoT, "Internet de las cosas" o "Internet of things" y lo harán a través de un kit de desarrollo para tal fin.

Se persiguen **varios objetivos** con la realización de este proyecto donde los estudiantes:

- Tomen **contacto** con el **concepto** de **IoT**
- Estudio de dos **nuevas tecnología** de **acceso móvil** diseñada para IoT llamada Nb-IoT y Cat-M1
- Familiarizarse con la tecnología **Arduino** y escribir programas sencillos utilizando en sistema operativo en tiempo real **Free RTOS**
- Introducción al protocolo **MQTT** y comandos **AT**
- Conceptos básicos de lenguaje de programación **C**

Laboratorio 2 - Conexión Módulo BG96 a red móvil y envío de datos por MQTT (20 Puntos)

1.Introducción	2
2.Objetivos	2
2.1Objetivos particulares:	2
3.Materiales	3
4. Fundamentos	4
4.1 Comandos AT	4
4.3 MQTTX	5
4.4 LTE IoT 2 Click (BG96)	5
4.5 Terminal serie	5
4.6.1 Establecimiento conexión red de Antel	6
4.6.2 Conexión Servidor MQTT y envío de mensajes	7
5. Tareas y actividades a realizar	8
5.1 Intercambio de mensajes utilizando el pasadizo (passthrough)	8

1.Introducción

En este laboratorio se propone estudiar y poner en práctica los principales conceptos de **NB-IoT** y **LTE Cat-M1**¹ para realizar la conexión a la red celular (attach) y enviar datos vía MQTT. Se interactuará directamente con el módulo de comunicación utilizando una terminal serie (programando passthrough en el microcontrolador).

2.Objetivos

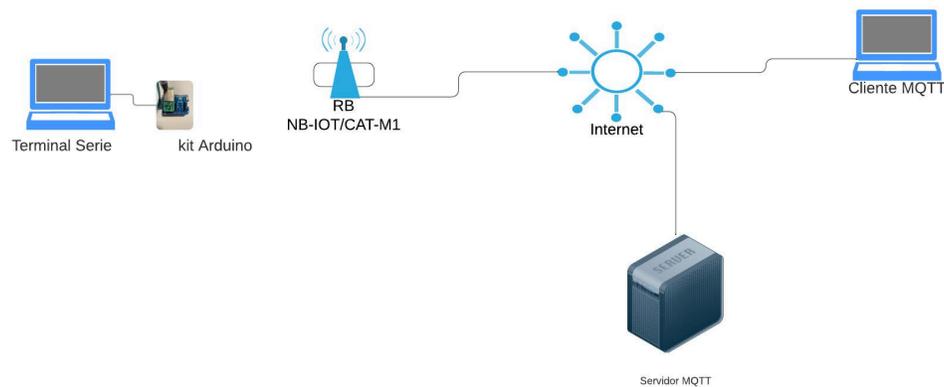
En este laboratorio del curso utilizaremos la placa de desarrollo Arduino Zero y el módulo BG96 de Quectel incorporado en la placa LTE IoT 2 click (MikroElektronika) conectados a través de la placa de adaptación.

Se utilizará Mosquitto que es un servidor/broker MQTT público basado en el software libre y de código abierto Eclipse Mosquitto™.

¹ Nb-IoT y LTE Cat-M1 son dos estándares de conectividad definidos por la 3GPP para redes LPWAN (Low Power Wide Area Network)

El objetivo es el de realizar la conexión a la red celular (NB-IoT o LTE Cat-M1) de Antel para enviar y recibir datos a través de un broker MQTT.

Diagrama de Red

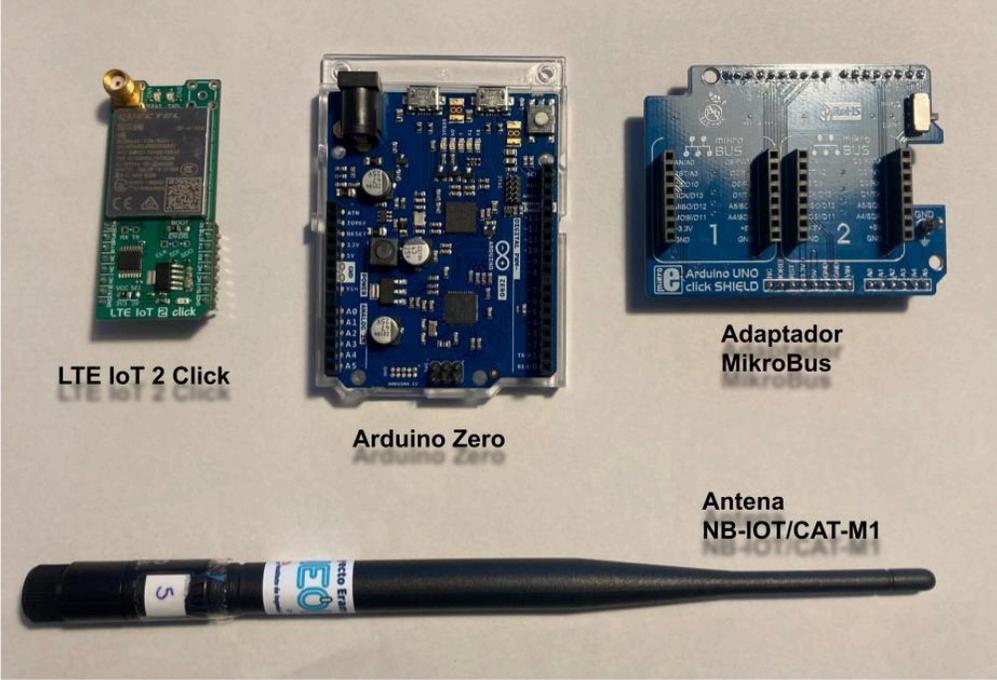


2.1 Objetivos particulares:

- Familiarizarse con los comandos AT en general y con los específicos del módulo BG96 y MQTT.
- Realizar el proceso de conexión (attach) a una red.
- Establecer una conexión a un broker MQTT.
- Enviar y recibir datos a través de MQTT.

3. Materiales

- Computadora con Arduino IDE instalado y configurado (ya utilizado en la parte A).
- Placa de desarrollo Arduino Zero.
- Placa de adaptación MikroBus para Arduino.
- Placa LTE IoT 2 Click (que incluye el módulo BG96).
- Antena NB-IoT/CAT-M1



LTE IoT 2 Click

Arduino Zero

Adaptador MikroBus

Antena NB-IOT/CAT-M1

4. Fundamentos

4.1 Comandos AT²

Los comandos **AT** a utilizar en este laboratorio tienen distintas sintaxis, pueden clasificarse, a grandes rasgos, en cinco tipos distintos:

1. Comandos de **sintaxis básica**: AT<x><n> donde <x> es el comando y <n> es el valor a establecer.
2. Comandos de **testeo**: AT+<x>=? Retorna la lista de parámetros y rangos de valores posibles que se le pueden pasar al comando de escritura.
3. Comandos de **lectura**: AT+<x>? Retorna el valor de uno o más parámetros.
4. Comandos de **escritura**: AT+<x>=<...> Establece un parámetro con valor definido por el usuario.
5. Comandos de **ejecución**: AT+<x> Realiza una lectura de un parámetro no variable (no modificable por el usuario) afectado por el comportamiento interno del módulo

4.2 Mosquitto³

El servidor (broker) **MQTT**⁴ a utilizar en este laboratorio es Mosquitto; se trata de un sistema público basado en el software libre y de código abierto Eclipse Mosquitto™. Tiene diversas ventajas y funcionalidades que lo hacen una opción muy práctica para probar nuestros desarrollos:

- Está basado en software libre y de código abierto.
- Soporta tanto IPv4 como IPv6.
- Soporta opcionalmente autenticación de usuario/contraseña.
- Soporta opcionalmente encriptación SSL.

4.3 MQTTX

MQTTX⁵ es un cliente MQTT libre y de código abierto multiplataforma, disponible para Linux, Windows y macOS. El código fuente puede encontrarse aquí.

4.4 LTE IoT 2 Click (BG96)

El módulo LTE IoT 2 Click de MikroElektronika cuenta con un módulo de comunicación BG96 y una interfaz MikroBus. El módulo BG96 utiliza comandos AT para su configuración y operación. En la página del curso se encuentran publicados documentos que detallan los comandos para algunas aplicaciones.

² También conocido como como set comandos Hayes, son muy usados en telecomunicaciones para la configuración de modems y equipamiento de red.

³ Servidor MQTT de código abierto, <https://mosquitto.org/>.

⁴ Message Queue Telemetry Transport. Protocolo liviano para mensajería IoT

⁵ <https://mqttx.app/>

4.5 Terminal serie

La terminal serie provista por el IDE de Arduino tiene la dificultad de que no facilita el envío del carácter Ctrl-Z (carácter 26 en decimal o 1A en hexadecimal) requerido por el comando AT+QMTPUB.

Hay varias formas de solventar esto:

1. Utilizar el software Minicom. Se trata de un software muy potente pero que no es tan sencillo de utilizar y además no se encuentra disponible para Windows. En Windows y en Linux se puede utilizar PuTTY o Coolterm.
2. Buscar un editor de texto que soporte este carácter (ya sea presionando Ctrl-Z o Alt-026), luego copiarlo y pegarlo en la terminal.
3. Evitar el uso de este carácter, optando por una implementación alternativa del comando AT+QMTPUB (por ejemplo especificando de antemano cuántos caracteres contendrá el mensaje).

En algunos terminales seriales, por ejemplo Minicom, puede resultar cómodo deshabilitar el eco del módulo, es decir, el hecho de que el módulo repita cada carácter que se le envía. Esto se realiza con el comando ATE0.

4.6 Pasos para envío y recepción de datos vía MQTT

A continuación se exponen a modo de ejemplo los comandos AT para enviar y recibir mensajes desde un broker MQTT.

Previo al envío de mensajes al broker MQTT es necesario establecer conexión a la red NB-IoT o Cat-M1 de Antel dependiendo de la SIM card incluida en el kit. Luego deberán conectarse al servidor y en última instancia enviar los mensajes.

Las líneas que comienzan con # corresponden a comentarios del comando, por más detalles se puede consultar la documentación publicada en la página del curso. Todos los comandos terminan con un carácter de retorno de carro (<CR> o \r) seguido de un carácter de retorno de línea (<LF> o \n).

4.6.1 Establecimiento conexión red de Antel

A continuación se listan los comandos AT básicos necesarios para conectarse a la red de Antel:

test de la comunicación (debe devolver OK):

AT

Opcional para deshabilitar eco del módulo:

ATE0

Consulta si la tarjeta SIM requiere PIN:

AT+CPIN?

Proporcionar el PIN de la tarjeta SIM (en caso de ser necesario):

```
AT+CPIN=1323
```

```
AT+CPIN?
```

Deshabilita la radio:

```
AT+CFUN=4
```

Parámetros de configuración para la selección de redes y servicios:

```
AT+QCFG="nwscanmode",3,1
```

```
AT+QCFG="iotopmode",1,1
```

```
AT+QCFG="servicedomain",1,1
```

Configuración de IP (IP -> IPv4; IPV6 -> IPv6), y APN (catm.iot para SIM CAT-M1 o internet.iot para SIM NB-IoT):

```
AT+CGDCONT=1,"IPV6","internet.iot","",0,0 #→ ingresar este comando si la SIM del kit es NB-IoT
```

```
AT+CGDCONT=1,"IPV6","catm.iot","",0,0 #→ ingresar este comando si la SIM del kit es CAT-M1
```

Restablece la radio:

```
AT+CFUN=1
```

Conecta con la red de Antel, seleccionar la opción que corresponda según la SIM del kit:

```
AT+COPS=1,2,"74801",9 #→ SIM NB-IoT
```

```
AT+COPS=1,2,"74801",8 #→ SIM CAT-M1
```

Consulta si se registró correctamente en la red

(debe devolver 0,1, en caso contrario esperar unos segundos y volver a consultar:

```
AT+CEREG?
```

Consulta la información de la red:

```
AT+QNWINFO
```

Consulta el IP asignado por la operadora (notar la rareza en la sintaxis del comando):

```
AT+CGPADDR=1
```

4.6.2 Conexión Servidor MQTT y envío de mensajes

test.mosquitto.org es un servidor MQTT que dispone el proyecto Mosquitto, es gratuito y posible usarlo para pruebas, existen otras opciones así como la posibilidad de instalar una versión de mosquitto en una máquina propia si así lo disponen.

Comandos para abrir conexión con el servidor, publicar y recibir mensajes:

Abre la conexión con el broker:

```
AT+QMTOPEN=0,"test.mosquitto.org",1883
```

```
# Se identifica ante el broker:
```

```
AT+QMTCONN=0,"nodo_tiot","",""
```

```
# Publica un mensaje en el tópico TloT/mensajes_de_prueba (el mensaje termina con Ctrl-Z).
```

```
AT+QMTPUB=0,0,0,0,"TloT/mensajes_de_prueba"
```

```
# Se suscribe a un tópico o un grupo de tópicos:
```

```
AT+QMTSUB=0,1,"TloT/#",2
```

5. Tareas y actividades a realizar

5.1 Intercambio de mensajes utilizando el pasadizo (passthrough)

Interactuar manualmente con la UART del módulo BG96 para enviar y recibir datos vía MQTT, verificando el correcto funcionamiento. Esto es:

- Programar la Arduino con el programa passthrough, conectar el módulo LTE IoT2 Click usando el adaptador y probar la comunicación con el comando de prueba AT.
- Abrir el cliente MQTTX, conectarse a test.mosquitto.org y suscribirse a tópico TIoT/#.
- Seguir la secuencia de comandos AT detallada más arriba para enviar un mensaje al broker en un sub-tópico TIoT/# y verlo en el cliente MQTTX. Sugerencia: Utilizar un tópico del tipo TIoT/<nombre> donde <nombre> es el nombre de pila del estudiante, por ejemplo TIoT/Felisberto.
- Continuar con la secuencia de comandos AT para suscribirse al tópico TIoT/<nombre>/para_mi.
- Enviar un mensaje desde el cliente MQTTX y verificar la recepción por el módulo BG96.
- Consultar la hora del dispositivo BG96.

5.2 Envío de temperatura

Utilizando el ejemplo **esqueleto_AT** provisto por los docentes del curso realizar un programa que implemente la conexión con el broker test.mosquitto.org, se suscriba al tópico TIoT/<nombre>/para_mi y envíe secuencialmente la temperatura del módulo (cada 5 segundos aproximadamente) utilizando la sintaxis JSON:

```
{"msg_id": <msg_id>, "temperature": <temperature>}
```

donde <msg_id> es un contador del número de mensaje (comienza en 0 y se incrementa), y <temperature> es la temperatura interna del microcontrolador.

Las funciones utilizadas en esqueleto_AT tratan de mostrar en la consola información relevante para depurar errores:

- Los comandos que se envían al módulo BG96 aparecen entre corchetes: {<comando>}.
- La respuesta que NO es analizada por el analizador (desechada por la directiva CLEAN_BUFFER), aparece entre paréntesis rectos: [<desecho>].
- Cuando el analizador se pierde al buscar la respuesta deseada, muestra el caracter numeral: #.
- Todo lo demás es analizado en busca de la respuesta que se espera recibir luego de

cada comando.

No es necesario tener en cuenta los errores, desconexiones o códigos no solicitados (URC).

Verificar el correcto funcionamiento de la solución. Esto es, que los mensajes que se envían desde la Arduino se visualicen correctamente en el cliente MQTT, y, de forma análoga, que los mensajes enviados desde el cliente MQTT se vean en el terminal serie de la Arduino (aunque no sean procesados por el programa realizado).

6. Entregable

Informe detallando las tareas realizadas en el laboratorio, incluyendo como mínimo:

- Breve descripción del estándar de conexión utilizado NB-IoT o CAT-M1 haciendo foco en las características de Capa 1 y 2.
- Descripción del protocolo MQTT
- Tareas desarrolladas en el punto 5)
 - Con descripción de comandos AT utilizados
 - Código desarrollado
- Archivos .ino programados

Referencias

- Quectel_BG96_AT_Commands_Manual_V2.3.pdf
- Quectel_BG96_MQTT_Application_Note_V1.2.pdf
- <https://docs.arduino.cc/hardware/zero>
- <https://www.mikroe.com/arduino-uno-click-shield>
- <https://www.mikroe.com/lte-iot-2-click>

Fecha de entrega del informe: x de noviembre.

Utilizar: fuente Time New Roman (o similar), tamaño 12 Incluir referencias!!!