

Proyecto Taller de Redes Inalámbricas

Curso 2025

Parte II - Laboratorio 1

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, el objetivo se completará a través de dos laboratorios.

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

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

Laboratorio 1 - Configuración del entorno (10 Puntos)

1. Introducción	2
2. Objetivos	2
3. Materiales	2
4. Fundamentos	3
4.1 Generalidades	3
4.2 FreeRTOS	3
4.3 LTE IoT 2 Click (BG96)	3
5. Tareas y actividades a realizar	4
5.1 Parpadeo simple	4
5.2 Parpadeo con FreeRTOS	5
5.3 Cola de mensajes con FreeRTOS	5
5.4 Prueba del módulo BG96 (GPS) → hay que soldar los conectores SMA a la placa	5
6. Entregable	6

1. Introducción

Durante el presente laboratorio utilizaremos la placa de desarrollo **Arduino Zero**, con una placa de adaptación que permite conectar distintos módulos, en particular en este curso utilizaremos un módulo de comunicación MikroElektronika (o mikroe):

- **LTE IoT 2 click** (tecnología NB-IoT).

Hacemos uso de la placa Arduino Zero por su relativa sencillez para programar y depurar, que además de utilizar un lenguaje basado en C, incorpora un depurador (debugger) sin necesidad de ningún equipo extra.

2. Objetivos

El objetivo de este laboratorio es introducir al estudiante en el manejo de la herramienta de desarrollo así como el hardware a utilizar durante los dos laboratorios del curso.

Los estudiantes tomarán contacto con la tecnología Arduino donde deberán descargar y configurar el entorno de trabajo y realizar una serie de ejercicios sencillos para familiarizarse con la herramienta.

3. Materiales

- Computadora con Arduino IDE instalado y configurado como indica la guía Instalación del entorno de desarrollo que se encuentra en EVA.
- Placa Arduino Zero¹
- Placa de adaptación mikroBus para Arduino²
- Módulo LTE IoT 2 Click (BG96)³

4. Fundamentos

4.1 Generalidades

Durante los laboratorios utilizaremos módulos de comunicación desarrollados por **MikroElectronica**, que hacen uso del conector **MikroBus**. Para ello contamos con placas de adaptación **Arduino-MikroBus** que permiten la conexión de hasta dos módulos MikroBus simultáneamente. Estos conectores comparten la **UART** y el bus **I2C**, pero difieren en otros pines como el de RESET usado para el encendido y reinicio de los módulos.

En general puede utilizarse un conector u otro pero debe prestarse atención a los cambios requeridos en el código de encendido/apagado/reinicio de los módulos MikroBus.

La placa de adaptación tiene un interruptor para conectar o desconectar la UART (pines RX/TX) de los módulos conectados al MikroBus. Esto sirve fundamentalmente para viejas placas Arduino en que la UART disponible para los periféricos era la misma que la utilizada para programar el firmware, entonces la forma de que los periféricos no interfieran durante la programación de la placa era colocar el interruptor en **PROG**.

Hasta ahora no hemos detectado problemas para programar las placas Arduino ZERO indistintamente de la posición del interruptor. Para utilizar la UART disponible en los conectores MikroBus es necesario que el interruptor se encuentre en la posición UART.

4.2 FreeRTOS

FreeRTOS es un **sistema operativo** para **tiempo real (RTOS)** diseñado para utilizar en sistemas embebidos. Es de código abierto y se encuentra portado a diversas plataformas. Utilizaremos la **versión FreeRTOS_SAMD21**, adaptada para usar en **procesadores SAMD21** sobre el framework de Arduino.

¹ <https://docs.arduino.cc/hardware/zero>

² <https://www.mikroe.com/arduino-uno-click-shield>

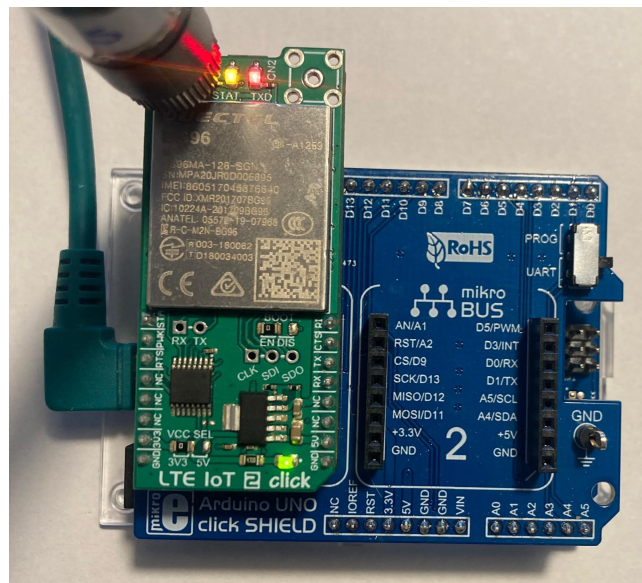
4.3 LTE IoT 2 Click (BG96)

El módulo LTE IoT 2 Click de MikroElectronica cuenta con un **módulo** de comunicación **BG96** y una **interfaz MikroBus**. El módulo BG96 integra **comunicación NB-IoT** (entre otras) y varias tecnologías de posicionamiento global (**GPS**, **GLONASS** y otras), utilizando **comandos AT** para su funcionamiento.

En la página del curso se encuentran publicados documentos que detallan los comandos para algunas aplicaciones.

El **encendido** del módulo BG96 se realiza mediante el **pin PKY** conectado al **pin 17** o **16** de la Arduino dependiendo de si se utilizan los **conectores 1** o **2** respectivamente. Tanto para el encendido como para el apagado, el pin debe llevarse a 1 (VDD) durante 1 segundo.

El **LED verde** simplemente indica que el **módulo** está **energizado**. El LED **amarillo** indica que el módulo está **encendido**. El LED **rojo** indica el estado de **conexión** a la.



Este módulo acepta dos antenas; una antena para la conexión celular (larga y rígida) que va colocada en el conector CN1, y una antena secundaria para la geolocalización (cuadrada y con cable) que va colocada en el conector CN2.

Advertencia: Siempre que se utilice la conexión celular, el módulo debe tener la antena correspondiente enchufada en todo momento. Encender el módulo sin antena celular puede dañarlo permanentemente.

5. Tareas y actividades a realizar

5.1 Parpadeo simple

1. Abrir el proyecto **blink_random** en el IDE de Arduino (disponible junto a esta página).
2. Analizar el código y deducir lo que hace.
3. Asegurarse que la opción **Optimize for Debugging** está activada, compilar el código y programarlo en la placa Arduino Zero utilizando el debugger.
4. **Trabajando con el debugger:**
 - a. Probar el programa dejándolo correr (play/pause) y observar lo que sucede con los LEDs.
 - b. Pausar el programa (play/pause). En caso que se haya pausado dentro de una función volver a la pestaña del programa **blink_random.ino**.
 - c. Agregar un **breakpoint** en la línea que dice **digitalWrite(PIN_LED, randomNumber)**; haciendo click a la izquierda del número de línea, lo que se indicará con un círculo rojo.
 - d. Dejar correr el programa nuevamente (play/pause) y cuando llegue al breakpoint averiguar cuánto vale randomNumber. Hay muchas formas para hacer esto, una opción es pararse con el ratón arriba de la variable; otra opción es buscar la variable entre las variables locales dentro del desplegable **VARIABLES** del menú de la izquierda.
5. Buscando la variable randomNumber en el menú desplegable VARIABLES de la izquierda, y haciendo doble click sobre el valor, cambiar el valor (si está en 1 pasarla a 0 y viceversa). Verificar este cambio al volver a correr el programa con el botón play/pause.

5.2 Parpadeo con FreeRTOS

1. Abrir el proyecto **blink_freertos** en el IDE de Arduino.
2. Analizar el código y deducir lo que hace.
3. Programarlo directamente en la placa (sin debugger) y verificar su funcionamiento.

5.3 Cola de mensajes con FreeRTOS

1. Abrir el proyecto **queue_freertos**.
2. Analizar el código y deducir lo que hace.
3. Programarlo en la placa y probarlo.
4. Modificar el código para que las tareas manden solamente valores enteros: una tarea la **temperatura interna** del **microcontrolador** y la otra **valores aleatorios** entre **20** y **25**.
5. Para visualizar los datos con Serial Plotter, modificar el código de la tarea 'SerialTask' para imprimir los valores de temperatura y valores aleatorios separados por coma.

Sugerencia: guardarlos datos en variables auxiliares y cada vez que llega un nuevo dato se actualiza el correspondiente y luego se imprimen ambos (uno actualizado y el otro último)

valor).

5.4 Prueba del módulo BG96 (GPS)

1. Conectar el módulo BG96 a la placa Arduino.
2. Abrir el programa *passthrough* y analizar el código.
3. Configurar el pin pky según el lugar en que se haya conectado el módulo, asegurarse que el **#define** del comienzo está en la opción adecuada y programar el código en la placa Arduino.
4. Esperar que el módulo se encienda (LED amarillo). En caso que no se encienda reiniciar la placa con el botón que hay montado en ella.
5. Abrir el puerto serie, configurar baudrate apropiado y el final de línea Both NL & CR.
6. Probar el programa con los siguientes comandos:

AT # Debería responder OK, en caso que no funcione revisar la configuración.

AT+QGPS=1 # Debería responder OK

AT+QGPSLOC?

Para más información sobre comandos GPS se sugiere consultar la hoja de datos Quectel_BG96_GNSS_Application_Note publicada en EVA.

6. Entregable

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

- Procedimiento para instalación del entorno de trabajo
- Conceptos adquiridos sobre HW y SW :
 - Arduino
 - FreeRTOS
 - Skecth
 - otros
- Tareas desarrolladas en el punto 5) . Incluir descripción de comandos AT utilizados

Fecha de entrega del informe: 12 de noviembre.

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