

# Laboratorio 1: Introducción a las herramientas de desarrollo

El objetivo de este laboratorio es introducir al estudiante en el manejo de las herramientas de desarrollo así como el hardware a utilizar durante los laboratorios del curso. Este laboratorio se deberá realizar completamente en forma individual.

## Introducción

Durante los laboratorios del curso utilizaremos la placa de desarrollo Arduino Zero, con una placa de adaptación que permite conectarle distintos módulos, en particular en este curso utilizaremos dos módulos de comunicación de Mikro-Electronica (o mikroe):

- LR 2 click (tecnología LoRa)
- LTE IoT 2 click (tecnología NB-IoT).

La placa Arduino Zero es muy sencilla de programar y depurar porque además de utilizar un lenguaje basado en C, incorpora una *depurador (debugger)* sin necesidad de ningún equipo extra.

## Objetivos

### Objetivos generales

Familiarizar al estudiante con las herramientas de desarrollo y el hardware a utilizar durante los laboratorios del curso.

### Objetivos particulares

- Escribir programas sencillos, basados en Arduino y utilizando el sistema operativo en tiempo real FreeRTOS.
- Programar y depurar software en la placa Arduino Zero.
- Tener un primer acercamiento a los comandos utilizados por los módulos de comunicación (comandos AT del módulo BG96 de Quectel de la placa LTE IoT 2 click y comandos propietarios del módulo RN2903 de la placa LR 2 click.

## Materiales

- Computadora con Arduino IDE instalado y configurado como indica la guía *Instalación del entorno de desarrollo*.
- Placa de desarrollo Arduino Zero<sup>1</sup>.
- Placa de adaptación MikroBus para Arduino<sup>2</sup>.

---

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

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

- Módulo LTE IoT 2 Click (BG96)<sup>3</sup> o módulo LR 2 Click (RN2903)<sup>4</sup>.

## Fundamentos

### Generalidades

Durante los laboratorios utilizaremos los 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**.

### 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. Durante el laboratorio utilizaremos la versión **FreeRTOS\_SAMD21**, adaptada para usar en procesadores SAMD21 sobre el framework de Arduino.

### 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. En este laboratorio nos centraremos en el uso del GPS.

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

---

<sup>3</sup><https://www.mikroe.com/lte-iot-2-click>

<sup>4</sup><https://www.mikroe.com/lr-2-click>

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 red.

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.

---

## LR 2 Click (RN2903)

El módulo *LR 2 Click* de MikroElectronica cuenta con un módulo de comunicación RN2903 de LoRaWAN y una interfaz MikroBus. El módulo RN2903 utiliza comandos de serie (propietario) para su funcionamiento.

En la página del curso se encuentran publicados documentos que detallan los comandos para algunas aplicaciones. En este laboratorio nos limitaremos a explorar comandos generales de configuración.

El encendido y apagado del módulo RN2903 se realiza de forma análoga al módulo BG96 mediante un pulso de 1 segundo en el pin PKY.

## Tareas y actividades a realizar

### 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:
  1. Probar el programa dejándolo correr (`play/pause`) y observar lo que sucede con los LEDs.
  2. Pausar el programa (`play/pause`). En caso que se haya pausado dentro de una función, volver a la pestaña del programa `blink_random.ino`.
  3. 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.

4. 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**.

### 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.

### 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: guardar los 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).

### 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 **pk** 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?

Se sugiere consultar la hoja de datos `Quectel_BG96_GNSS_Application_Note` publicada en EVA.

### Prueba del módulo RN2903

1. Conectar el módulo RN2903 a la placa Arduino.
2. Abrir el programa `passthrough` y analizar el código.
3. Configurar el pin `pk` 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. Abrir el puerto serie, configurar baudrate apropiado y el final de línea `Both NL & CR`.
5. Probar el programa con los siguientes comandos:

```
sys reset # Debería contestar con la versión del firmware.
```

```
sys get ver
```

```
mac get ch freq 0
```

Puede consultar otros comandos en la guía `RN2903 LoRa Technology Module Command Reference User Guide` publicada en EVA.

### Entregables y Defensa

Cada estudiante debe asistir a la defensa con el laboratorio realizado en su completitud, y habiendo contestado de manera individual -y en tiempo y forma- la encuesta correspondiente publicada en la plataforma EVA.

El día de la defensa deberán poder explicar cada una de las partes de este laboratorio.

### Referencias

- `Quectel_BG96_AT_Commands_Manual_V2.3.pdf`
- `Quectel_BG96_GNSS_Application_Note_V1.2.pdf`
- `Quectel_BG96_LPWA_Specification_V1.9.pdf`
- <https://docs.arduino.cc/hardware/zero>
- <https://www.mikroe.com/arduino-uno-click-shield>
- <https://www.mikroe.com/lte-iot-2-click>
- <https://www.mikroe.com/lr-2-click>
- Guía *Instalación del entorno de desarrollo* publicada en la web del curso.