

Obligatorio 5 - Bluetooth y más movimientos.

IIE - Facultad de Ingeniería - Universidad de la República

Tallerine Biónico 2025

El objetivo es familiarizarse con el módulo bluetooth de la placa Microbit y la conexión con una aplicación de celular para realizar proyectos inalámbricos.

Se continúa trabajando con el bicho, afinando movimientos ya vistos e incorporando nuevos.

Además, ya comenzar a elaborar el preinforme.

● Bluetooth en el celular

Hasta ahora hemos recibido comandos desde el Monitor Serial. Se busca recibir estos comandos desde un celular por medio de Bluetooth. Para esto se debe instalar en el celular una aplicación que permite enviar texto a través de Bluetooth.

La microbit cuenta con una radio bluetooth de baja energía (BLE, Bluetooth Low Energy) incorporada que se puede comunicar de forma sencilla con la aplicación **Adafruit Bluefruit Connect** la cual se encuentra disponible para Android e iOS (*Play Store* y *App Store* respectivamente).

Se puede acceder por más información de la aplicación al siguiente enlace: <https://learn.adafruit.com/bluefruit-le-connect>

Una vez instalada la aplicación, se abre y se elige el dispositivo a conectarse (figura 1) . Luego se debe seleccionar el módulo "UART" (figura 2).

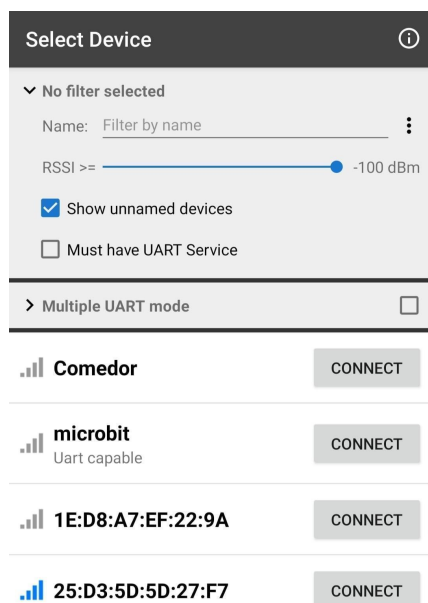


Figura 1. Detección de dispositivos

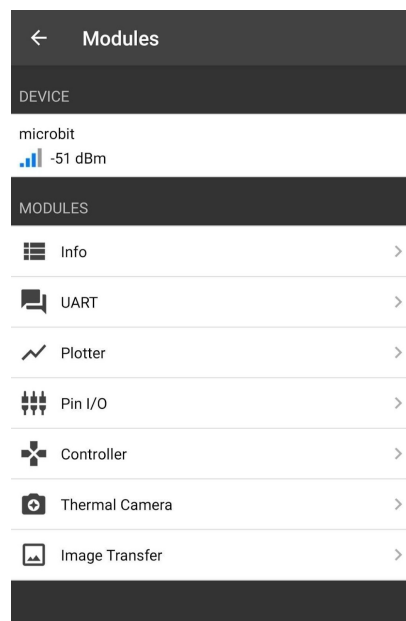


Figura 2. Módulos

● Bluetooth en la Microbit

Para utilizar la comunicación Bluetooth de la placa no es necesario conocer sus detalles ya que se dispone de una biblioteca que lo resuelve: **Adafruit Microbit**.

La biblioteca **Adafruit Microbit** tiene varias funcionalidades, brinda una interfaz para el uso de la matriz LED, el sensor de temperatura, el acelerómetro y en este obligatorio se trabajará con ella pues brinda también una interfaz para usar con la aplicación **Adafruit Bluefruit Connect** a través de la conexión Bluetooth.

Para que todo funcione se deberán instalar las siguientes bibliotecas:

- Adafruit Microbit: <https://www.arduino.cc/reference/en/libraries/adafruit-microbit-library/>
- BLEPeripheral: <https://www.arduino.cc/reference/en/libraries/bleperipheral/>
- Adafruit GFX : <https://www.arduino.cc/reference/en/libraries/adafruit-gfx-library/>
- Adafruit_BusIO: <https://www.arduino.cc/reference/en/libraries/adafruit-busio/>

Para descargar las bibliotecas, seleccionar “Releases” y luego la versión “latest”.

Para instalar cada librería en Arduino IDE seleccionar *Programa > Incluir Librería > Añadir biblioteca.ZIP...* Una vez realizado esto, reiniciar Arduino IDE y verificar si se encuentra en el listado de librerías, *Programa > Incluir Librería*.

● Ejemplo de comunicación Bluetooth

El siguiente código **ble_uart_microbit_v2.ino** (disponible en EVA) muestra cómo se comunica la placa Microbit a través de un puerto serie a la PC y un dispositivo inalámbrico (el celular en nuestro caso):

```
#include <Adafruit_Microbit.h>
```

```
Adafruit_Microbit microbit; //Define la variable microbit para utilizar las funciones.
```

```
void setup()
```

```
{
  Serial.begin(9600);
  Serial.println("Microbit ready!");
  microbit.BTLESerial.setLocalName("Microbit");
                                     //Microbit es el nombre de la conexión
  microbit.BTLESerial.begin();
}
```

```
void loop()
```

```
{
  microbit.BTLESerial.poll(); //Permite escribir en el celular
  if (microbit.BTLESerial.available())
  {
    Serial.write(microbit.BTLESerial.read());
  }
}
```

```

    }
    if (Serial.available())
    {
        microbit.BTLESerial.write(Serial.read());
    }
}

```

Aclaración:

La interfaz que se utiliza en la microbit se llama UART (Universal Asynchronous Receiver-Transmitter) que consta de dos pines: Rx y Tx. El pin Rx se utiliza para recibir datos y el pin Tx para transmitirlos. La recepción dispone de memoria interna, llamada búfer (en inglés *buffer*), en la cual se van encolando los datos que llegan por Rx hasta que sean procesados. Este búfer interno de la microbit tiene 64 bytes. Una vez lleno, no puede recibir datos. Para asegurarse de leer los datos antes de que el búfer esté lleno, y que los datos no se pierdan, se recomienda enviar hasta un máximo de 10 caracteres desde la terminal serial de la PC.

● Ejercicios

1. Comencemos con la comunicación inalámbrica entre celular y microbit.
 - a. Instalar la aplicación Adafruit Bluefruit Connect en un celular.
 - b. Realizar la conexión Bluetooth entre el celular y la placa.
 - c. Estudiar el código *ble_uart_microbit_v2.ino*. Se debe comprender que se espera que suceda.
 - d. Utilizar la aplicación del celular en Modo Terminal UART. Investigar el funcionamiento para enviar y recibir caracteres ("a", "b", etc)
 - e. Probar el código en la placa.
 - f. Modificar el código de forma tal que si el último dato enviado desde el celular es la letra "a" representar con un símbolo a elección en la matriz LED (y dejar encendido). En el caso de que el último comando recibido NO sea la letra "a" se puede apagar la matriz LED o representar con otro símbolo que el dato enviado no es "a".
2. Continuar perfeccionando los movimientos ya conocidos y desarrollar otros que permitan rotar sobre sí mismo en sentido horario y antihorario. Considerar, dentro de lo posible, el ejercicio 7 del obligatorio 4 para la realización de este ejercicio. Se pide:
 - a. A partir de la función **Avanzar()** ya desarrollada, se deben fijar los retardos y ángulos óptimos para que el avance sea lo más rápido posible. Tener presente que algunos retardos debieron ser suprimidos para realizar movimientos simultáneos.
 - b. Repetir lo anterior para las funciones **Retroceder()**, y **Saludar()**.
 - c. Partiendo de las funciones *Avanzar()* y *Retroceder()*, desarrollar la función **GiroHorario()** de forma que el Bicho realice 2 pasos girando en sentido horario. El movimiento debe ser óptimo (ver parte a.)
 - d. Desarrollar la función opuesta **GiroAntiHorario()** en forma óptima.
3. Tomando las funciones de movimientos ya realizadas, implementar un programa que utilice una conexión bluetooth tal que el bicho realice en forma continuada el movimiento correspondiente al último comando recibido por la microbit de acuerdo a la tabla 1.

Comando recibido	Movimiento
'a'	Avanzar
'r'	Retroceder
'g'	Girarhorario
'j'	Girarantihorario
's'	Saludar
'i'	Init patas
'p'	Pausa (quieto donde estaba).

Tabla 1. Correspondencia Comando - Movimiento

Si se recibe un comando no incluido en la tabla 1 se debe hacer caso omiso.

Inicialmente se considera que fue oprimida la letra “i”.

4. Preinforme

- a. Lea atentamente la guía “como hacer un informe.pdf” disponible en EVA.
- b. Comenzar a elaborar el preinforme, por los siguientes puntos:
 - Carátula
 - Marco teórico
 - Desarrollo, con introducción y objetivos.
 - Conclusiones

No se pide una estética final, pero debe ser prolijo para que se entienda. Puede tener faltante de contenido. El objetivo de esta etapa es tener un avance del preinforme para ir madurando su elaboración.

Se debe presentar el preinforme preliminar en formato impreso para la clase 6 conteniendo lo indicado arriba.

- c. Se debe presentar el preinforme completo en formato pdf en la plataforma EVA antes de la fecha indicada en el EVA (fecha aproximada 11/5).

IMPORTANTE: No se permite realizar un copiar-pegar de Internet (incluido ChapGPT). Esta práctica es considerada una falta grave y, por lo tanto, descalifica el informe completo.

Para tener en cuenta

Al abrir varias ventanas del IDE de Arduino puede darse el caso que no se seleccione el campo Softdevice: S110. Si sucede esto y se carga el programa, se borrará parte del código que tiene el micro:bit por defecto para manejar el bluetooth. Notarán que le cargan el programa y no responde. Para solucionar esto deben ir a <https://makecode.microbit.org/> generar un proyecto nuevo, descargar un programa vacío, este termina en .hex y subirlo a la micro:bit. Esto volverá a cargar la parte borrada. Luego de esto volver a programar en arduino.