

Obligatorio 9 - Control Pad.

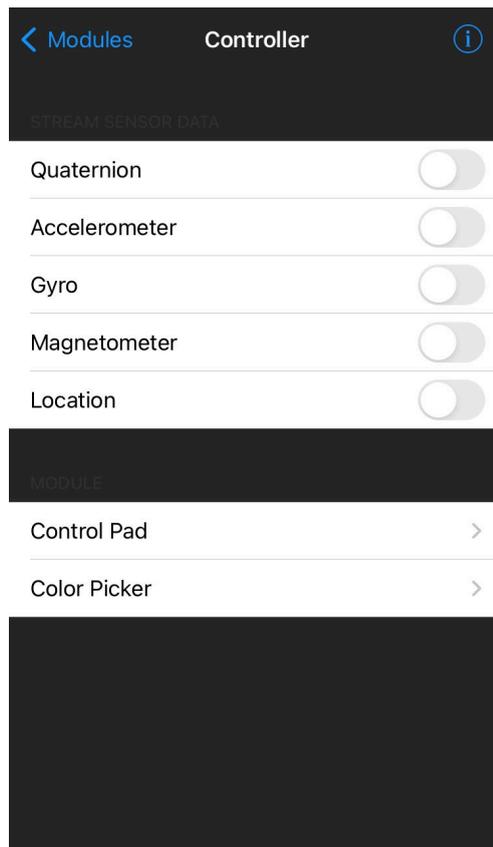
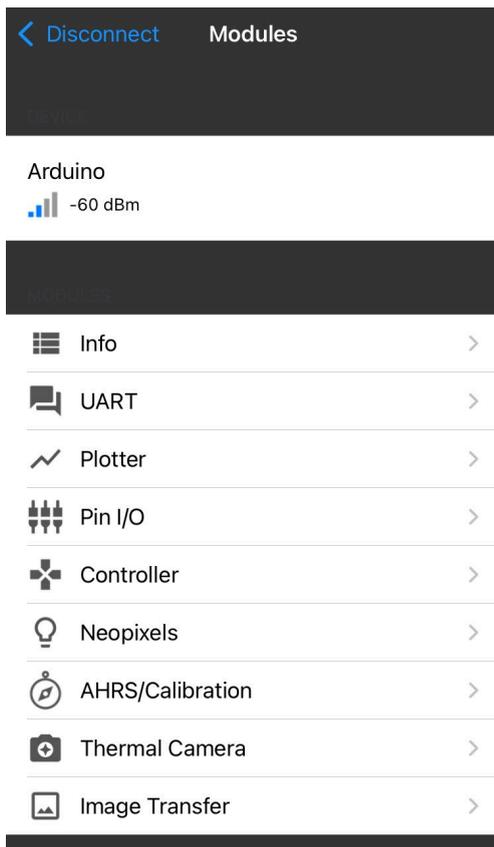
IIE - Facultad de Ingeniería - Universidad de la República

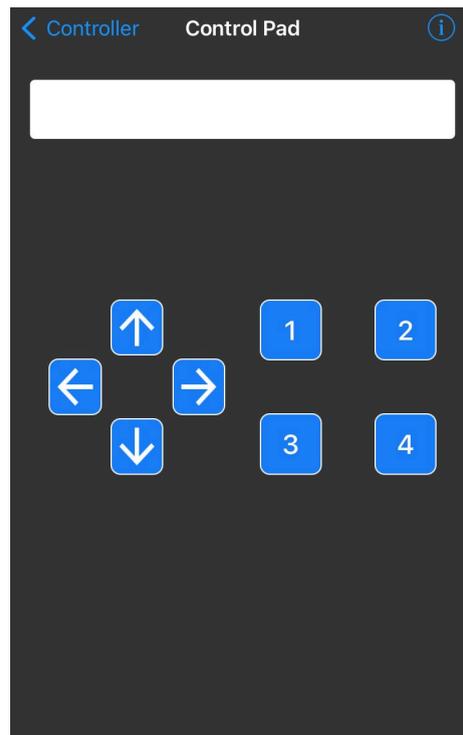
Tallerine Biónico 2025

El objetivo de este obligatorio es sustituir la comunicación serial mediante Bluetooth por un Control Pad también mediante bluetooth utilizando la aplicación Bluefruit.

1. Aplicación Bluefruit

Esta aplicación además de enviar texto mediante bluetooth, permite enviar comandos apretando botones de un joystick. Para eso entramos a la sección "Controller" y luego "Control Pad" como se muestra en la siguiente imagen.





Este joystick posee 8 botones, 4 para indicar movimiento y 4 para agregar otras funcionalidades.

2. Librerías

Para recibir qué botón se apretó o se soltó utilizamos funciones que interpretan la información enviada de la aplicación. Descargar los 3 archivos del eva: "prueba_ble_controller_pad.ino", "leerControllerPad" y "packetParser.cpp" y colocarlos en una misma carpeta llamada prueba_ble_controller_pad.

El archivo "prueba_ble_controller_pad" tiene un ejemplo de aplicación. El archivo "leerControllerPad" se encarga de leer un comando enviado por el celular y transformarlo a los siguientes números:

- * Boton1 --> 1
- * Boton2 --> 2
- * Boton3 --> 3
- * Boton4 --> 4
- * FlechaArriba --> 5
- * FlechaAbajo --> 6
- * Flechalzquierda --> 7
- * FlechaDerecha --> 8
- * Al soltar cualquier botón devuelve -2.
- * En caso de no haber datos, timeout o error en checksum devuelve -1.

El archivo "packetParser.cpp" tiene la lógica de cómo recibir un comando desde la aplicación.

Abrimos el ejemplo "prueba_ble_controller_pad.ino" ir a Sketch →Add File y agregar los archivos "leerControllerPad" y "packetParser.cpp". Una vez agregado compilar y verificar su funcionamiento. Notar que la aplicación es "prueba_ble_controller_pad.ino", los demás archivos no se deberán modificar.

3. Ejercicios

- 1) Incluir los archivos "leerControllerPad" y "packetParser.cpp" en el programa principal como se explicó anteriormente. Modificar el programa basándose en el ejemplo "prueba_ble_controller_pad" de forma que ahora el bicho se mueva mediante el Control Pad.

Aclaración: si utilizamos el Control Pad ya no podremos utilizar BLE UART para enviar comandos.

- 2) El Control Pad solo dispone de 8 botones por lo que no es suficiente para asignar un botón a cada movimiento. Para aumentar las posibilidades del Control Pad a 14 comandos, se propone realizar un programa que simule algo similar a como funciona una calculadora con el botón "shift". Por ejemplo, si se considera al Botón 1 como "shift", este botón activa y desactiva el "modo shift", siendo el comando devuelto el que corresponde a la siguiente tabla:

Modo actual	Botón	Comando	Próximo Modo
Normal	Botón 1	(Ninguno)	Shift
Normal	Botón 2	2	Normal
Normal	Botón 3	3	Normal
Normal	Botón 4	4	Normal
Normal	Flecha Arriba	5	Normal
Normal	Flecha Abajo	6	Normal
Normal	Flecha Izquierda	7	Normal
Normal	Flecha Derecha	8	Normal
Shift	Botón 1	(Ninguno)	Normal
Shift	Botón 2	9	Normal
Shift	Botón 3	10	Normal
Shift	Botón 4	11	Normal
Shift	Flecha Arriba	12	Normal
Shift	Flecha Abajo	13	Normal
Shift	Flecha Izquierda	14	Normal
Shift	Flecha Derecha	15	Normal

Se sugiere utilizar una memoria global que indique el Modo (Normal o Shift, o bien, 0 o 1).

Observar que en caso de que se oprima un botón de comando estando en modo shift, basta sumar 7 al valor que entrega el Control Pad. Y aún más simple si se define Normal = 0 y Shift = 1, se reduce a la siguiente ecuación: $\text{comando} = \text{Modo} * 7 + \text{ComandoControlPad}$.

Encender un led en el Bicho siempre que se esté en Modo shift.