# Jugando con Bluetooth


- Un poco de teoría Bluetooth
- La extensión en Makecode
- Programa en Makecode
- APP Serial Bluetooth Terminal
- Resultado con la APP Serial Bluetooth Terminal
- APP INVENTOR extensiones bluetooth LE y microbit
- APP INVENTOR tu programa


Copyright 2026

# Un poco de teoría Bluetooth

## ONDAS

Una onda es una señal que se propaga por un medio. Por ejemplo el sonido, que es una onda mecánica que viaja usando el aire o cualquier otro material. Pero en el caso de las señales eléctricas pueden ser enviadas por el cable o a través del vacío (no necesitan un medio para transmitirse).

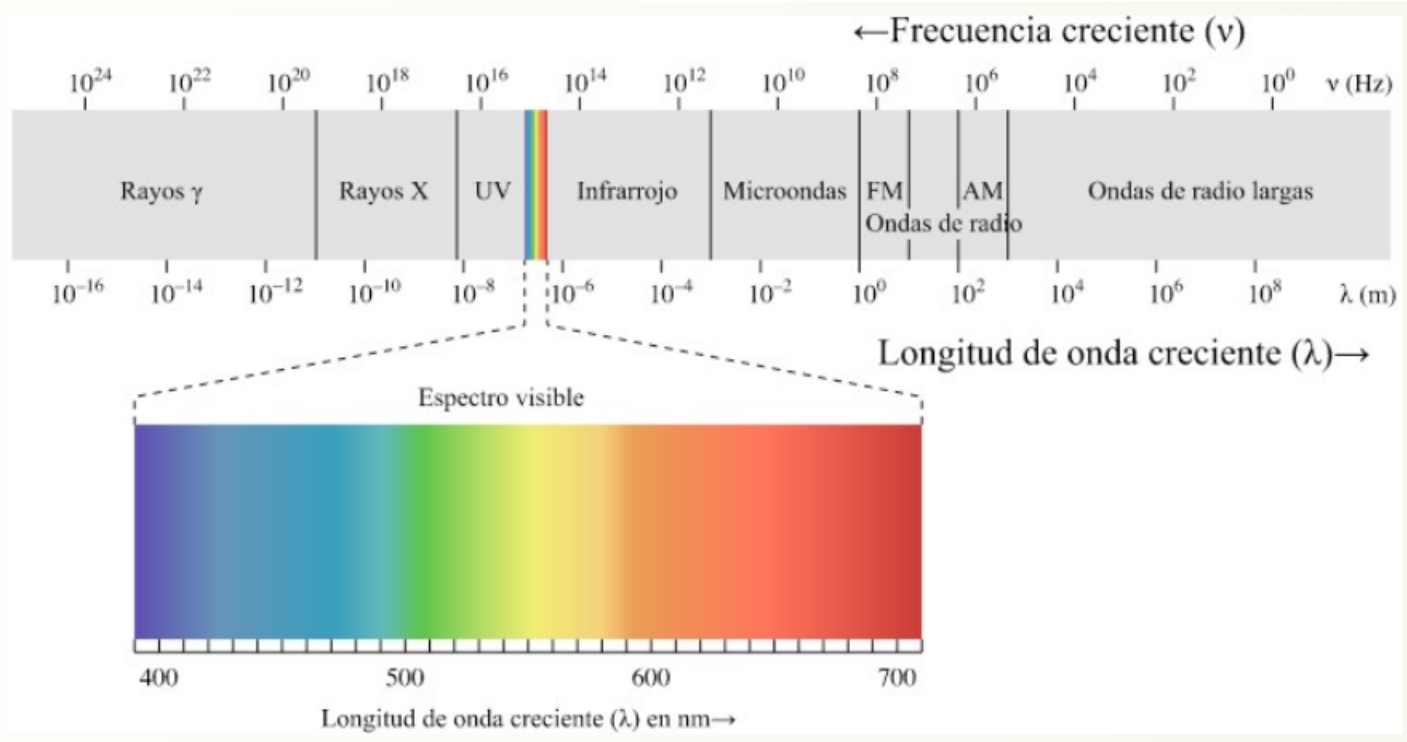
Dependen de 3 parámetros principalmente:

- **Amplitud:** altura máxima de la onda. Hablando de sonido representaría el volumen. Si nos referimos a una onda eléctrica estaríamos representando normalmente el voltaje.
- **Longitud de onda λ:** distancia entre el primer y último punto de un ciclo de la onda (que normalmente se repite en el tiempo).
- **Frecuencia f :** Número de veces que la onda repite su ciclo en 1 segundo (se mide en hertzios).
- **Periodo T** es simplemente es la inversa de la frecuencia. T=1/f

La relación entre ellas es muy fácil pues las ondas electromagnéticas viajan a la velocidad de la luz c y si velocidad es espacio/tiempo luego c = λ/T luego **c= λ*f**

Dentro del espectro electromagnético encontramos diferentes tipos de señales dependiendo de las características de su onda.

Copyright 2026




## TRANSMISIÓN INALÁMBRICA: BLUETOOTH.

- Hoy en día, este grupo está formado por miles de empresas y se utiliza no sólo para teléfonos sino para cientos de dispositivos.
- Bluetooth es una red inalámbrica de corto alcance pensada para conectar pares de dispositivos y crear una pequeña red punto a punto, (sólo 2 dispositivos).
- Utiliza una parte del espectro electromagnético llamado "**Banda ISM**", reservado para fines no comerciales de la industria, área científica y medicina. Dentro de esta banda también se encuentran todas las redes WIFI que usamos a diario. En concreto funcionan a 2,4GHz. (Un G son $10^9$) luego entre FM y Microondas.

> **¿Sabias que?**
> Su curioso nombre viene de un antiguo rey Noruego y Danés, y su símbolo, de las antiguas ruinas que representan ese mismo nombre.

Hay 3 clases de bluetooth que nos indican la máxima potencia a la que emiten y por tanto la distancia máxima que podrán alcanzar:

Copyright 2026

| CLASE | POTENCIA | DISTANCIA |
|---|---|---|
| Clase 1 | 100 mW | 100 m |
| Clase 2 | 2,5 mW | 10 m |
| Clase 3 | 1 mW | 1 m |

También es muy importante la velocidad a la que pueden enviarse los datos con este protocolo:

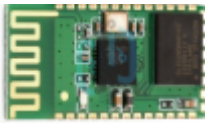
| Versión | Velocidad |
|---|---|
| 1.2 | 1 Mbps |
| 2 | 3 Mbps |
| 3 | 24 Mbps |
| 4 | 24 Mbps |

> Mbps : Mega Bits por segundo. MBps: Mega Bytes por segundo.
> kb = 1.024 b M = 1.024 k G = 1.024 M

## ¿Te atreves a calcularlo …. ?

¿Cuántos ciclos por segundo tendrán las ondas que están en la **Banda ISM?** ¿Cuál es la longitud de onda?

**Solución**
a) f = 2.4G
b) λ=c/f= 12.5cm o sea, las antenas tendrían que ser de esta longitud. Hay muchos trucos para reducirla, una de ellas es la forma de serpiente que puedes ver en el HC-06



## ¿Te atreves a calcularlo…?

¿A qué distancia y cuanto tiempo tardarían en enviarse los siguientes archivos por Bluetooth?

1. Un vídeo de 7Mb usando versión 2 clase 2
2. Una imagen de 2.5Mb usando versión 3 clase 1
3. Un archivo de texto de 240KB usando versión 1.2 clase 1

Copyright 2026

**Solución**

1) 7Mb / 3Mbs = 2.3 seg.
2) 2.5Mb / 24Mbs = 0.1 seg.
3) 240 kB 8b/B = 1.920 kb  1.920 kb  / 1.024 =  1.875 Mb  1.875Mb / 1Mbs  = 1.875 seg.

## ¿Bluetooth clásico o Bluetooth Low Energy = BLE?

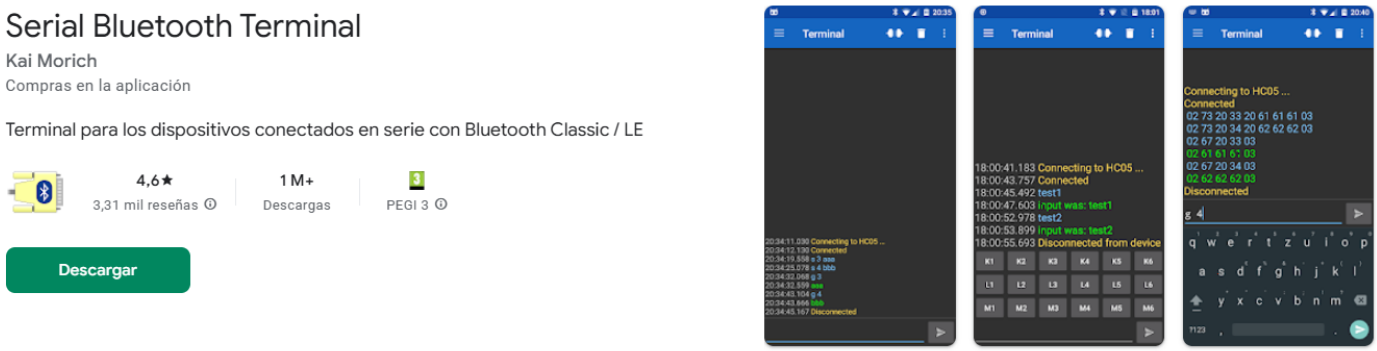
Es un protocolo similar al clásico Bluetooth pero diseñado a consumir menos potencia manteniendo funcionalidad. Su popularidad ha crecido en multitud de dispositivos

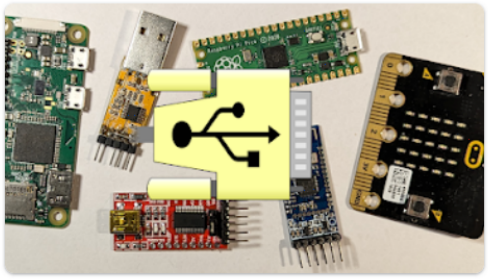
En robótica, el clásico device que utiliza BLE es la **Micro:bit**. Aunque la Micro:bit no tiene Wifi integrada, posee una radiofrecuencia que podemos configurar para Bluetooth (hay que elegir, o utilizar sus comandos de Radio o utilizar comandos de Bluetooth)

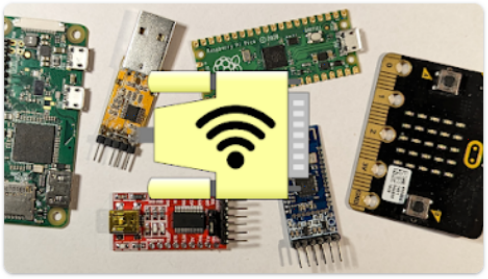
Por eso a la hora de elegir la APP tienes que tener en cuenta:

- Si acepta Bluetooth clásico o **BLE**
- Que la APP acepte **leer** datos desde el robot como **escribir** o enviar

Nosotros hemos elegido uno sencillo que cumple las dos condiciones (hay muchas APPs) Serial Bluetooth Terminal



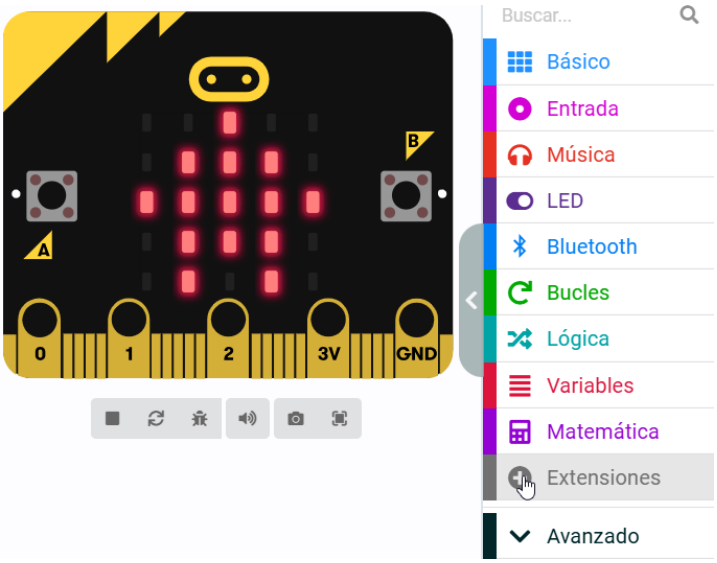


Copyright 2026

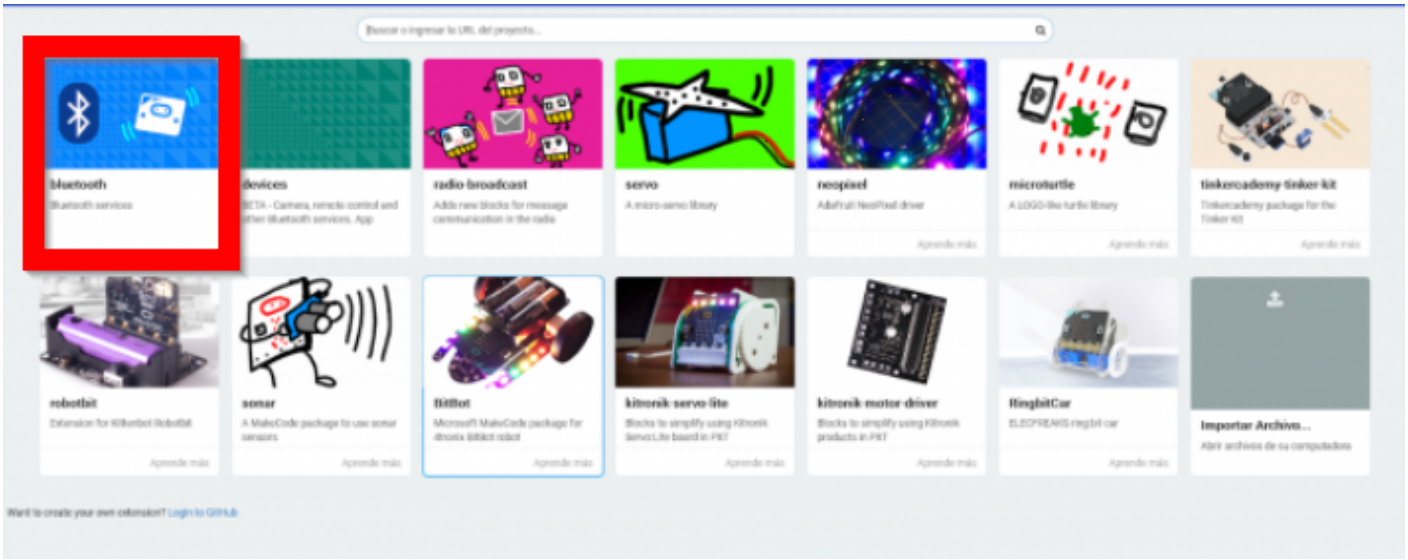
# La extensión en Makecode

En Makecode instalaremos la siguiente extensión

Entramos en **Extensiones**

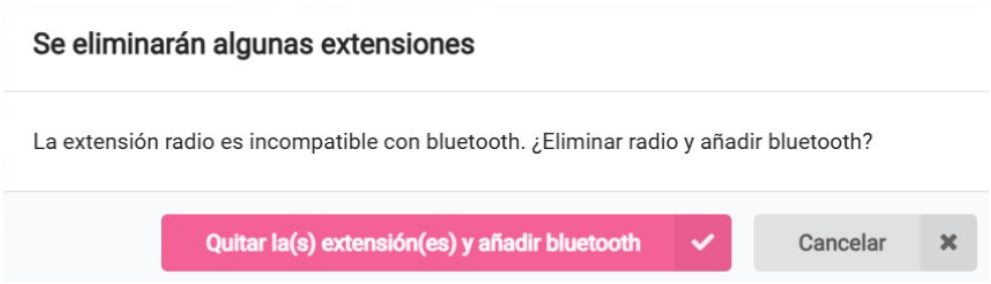



Buscamos **Bluetooth** y elegimos la esta :



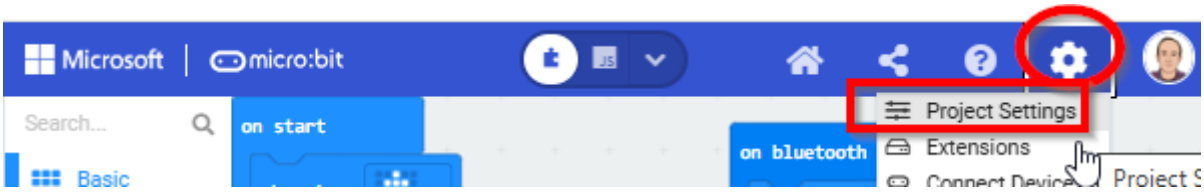

Nos dirá que es incompatible con la radio, y hay que eliminar la radio y poner Bluetotth, aceptamos :

Copyright 2026

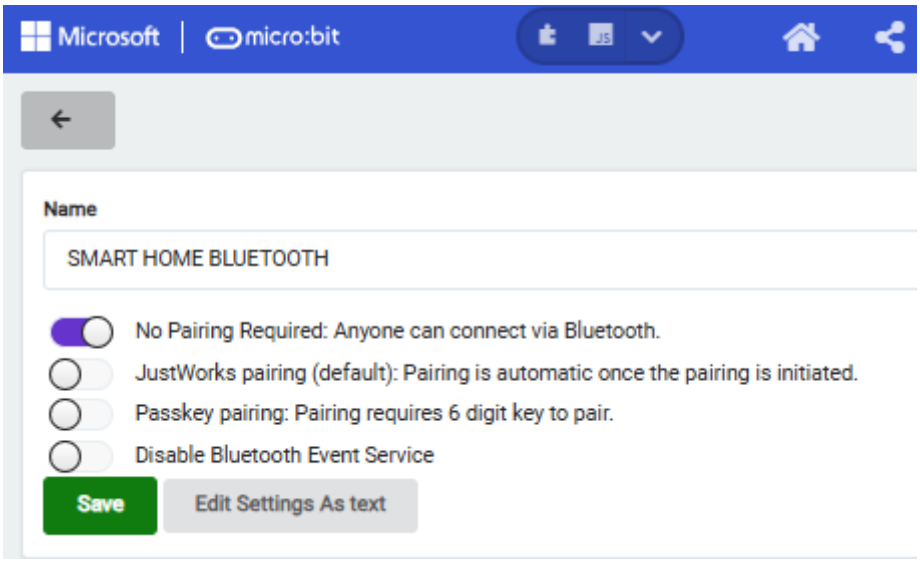



## Por si acaso...

En Makecode, si vamos a la rueda dentada - Project settings




Hay que tener que cualquiera se puede conectar via Bluetooth




Copyright 2026

# Programa en Makecode

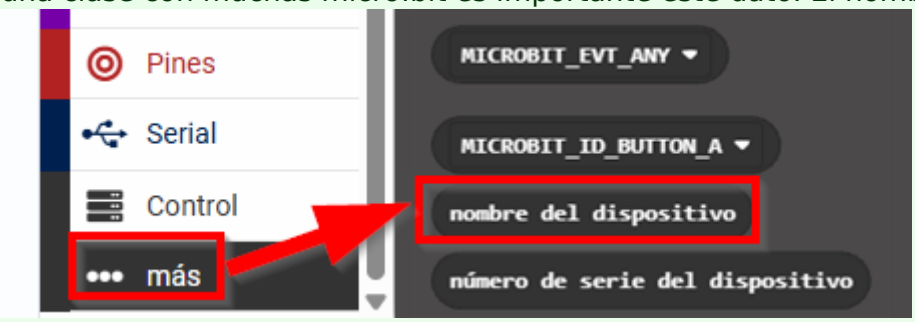
Parecido al que vimos con la micro:bit sola en https://libros.catedu.es/books/microbit-car/page/bluetooth-programa-en-makecode pero con la diferencia que ahora podemos añadir elementos de entrada y salida de la maqueta:

Realizamos un programa que :

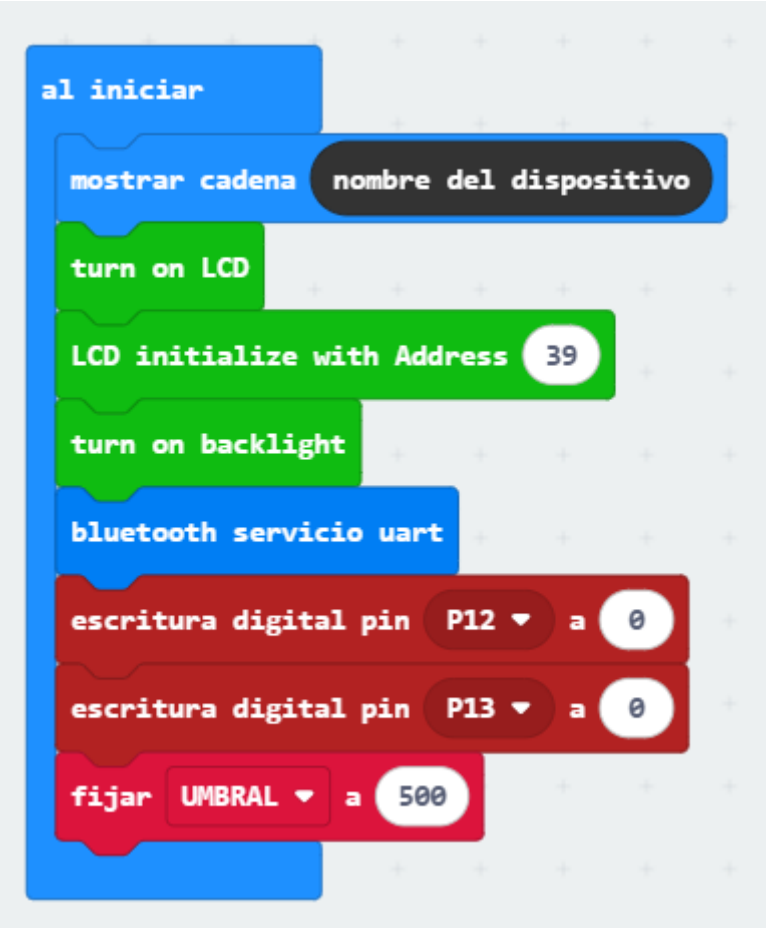
- **Extensiones**.
  - Instalamos la extensión LCD display
- En **inicio :**
  - que muestre el nombre del equipo (ver recuadro)
  - Inicializamos el LDC Display para que nos muestre qué ocurre
  - se active el servicio UART para el envío y recepción de mensajes,
  - apagamos el motor por si esta encendido
  - definimos el umbral de lluvia con 500
- **Al conectar Bluetooth**
  - que muestre un check
- **Al desconectar Bluetooth**
  - que muestre X
- **Al recibir datos**, hasta # (puede ser otro carácter)
  - que muestre la frase recibida
  - si la frase es un "ON" que encienda el motor
  - si la frase es un "OFF" que apague el motor
- **Cada segundo**
  - Que envíe la temperatura por bluetooth
  - si hay lluvia que envíe "ESTA LLOVIENDO"

Copyright 2026

**¿Para qué mostrar el nombre de la micro:bit?** Para saber a qué micro:bit conectarte. En una clase con muchas micro:bit es importante este dato. El nombre del equipo está en




Al inicio

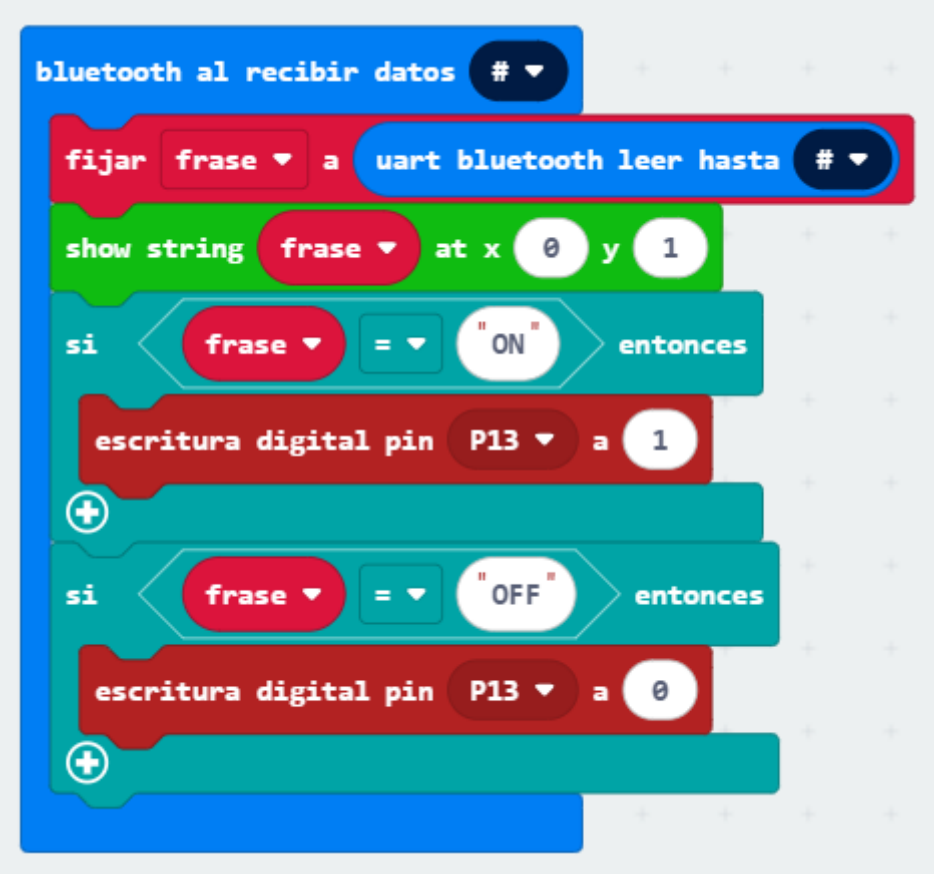



Al conectar Bluetooth o desconectar Bluetooth

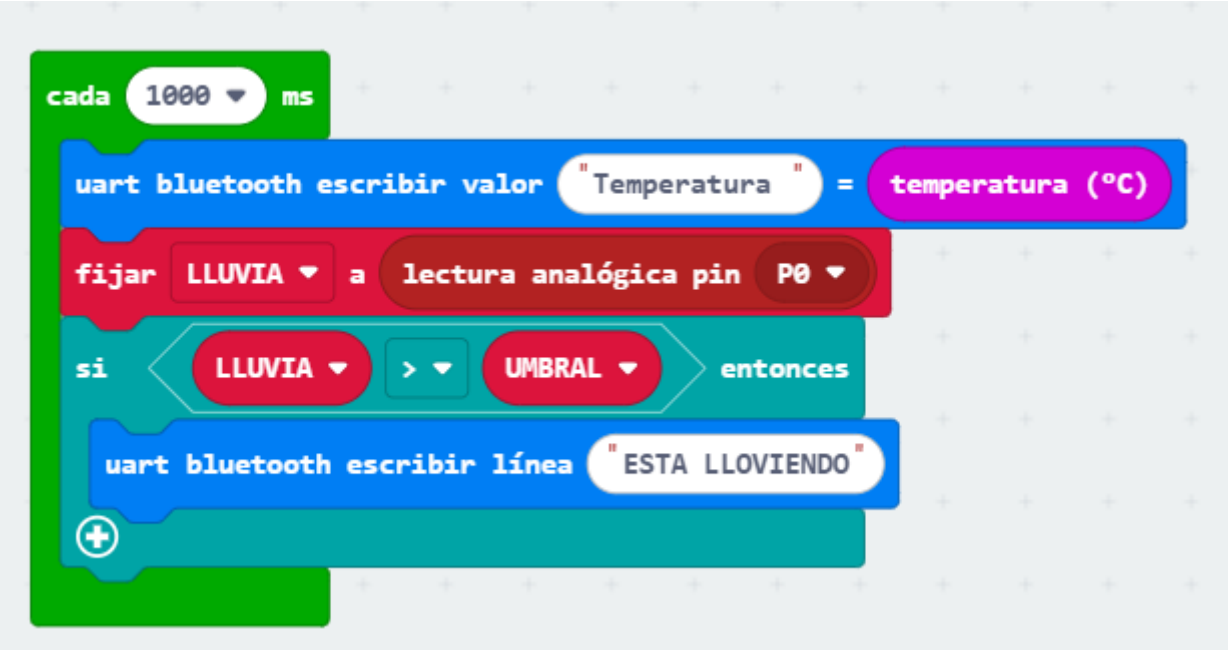
Copyright 2026




Cuando recibo datos #ON# o #OFF#




Y envío de la maqueta a la APP la temperatura, y si llueve que avise




Pregunta: ¿Esa temperatura es del sensor DHT11? Respuesta NO ES DEL SENSOR INTEGRADO DE LA PLACA MICROBIT

El programa entero

https://makecode.microbit.org/S07605-40747-09462-69361

Copyright 2026

https://makecode.microbit.org/#pub:S07605-40747-09462-69361

Copyright 2026

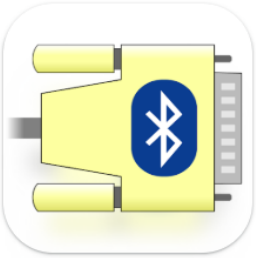
# APP Serial Bluetooth Terminal

## DESCARGA LA APP

Esta APP es muy sencilla y la puedes descargar aquí. Tiene las siguientes ventajas :

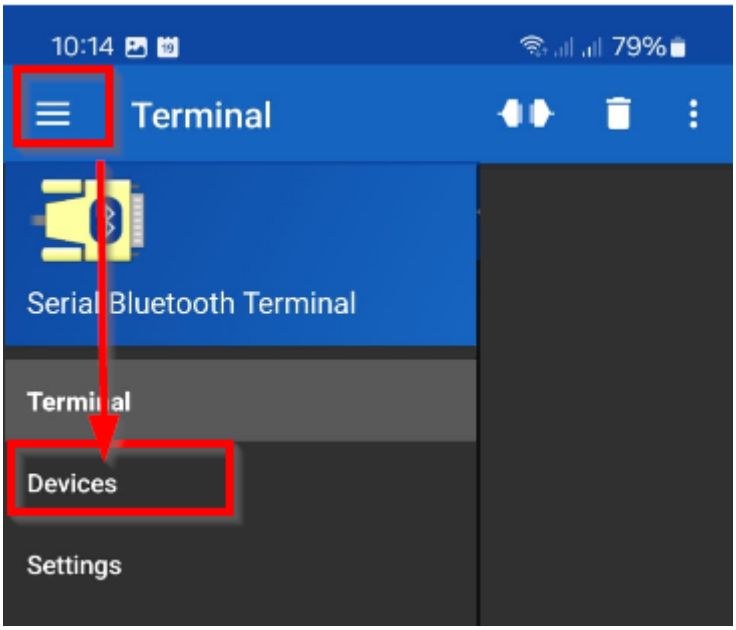
- **Enviar** / **Recibir** mensajes
- Permitir conexiones tanto
  - **BLUETOOTH CLÁSICO** por ejemplo HC06 de Arduino, Echidna, ESP32 ...
  - **BLUETOOTH LE** (Low emision)  por ejemplo para la MICRO:BIT



## EMPAREJAR DISPOSITIVOS

Si no esta emparejado con el móvil NO TE PUEDES CONECTAR, para ello entramos en Devices :

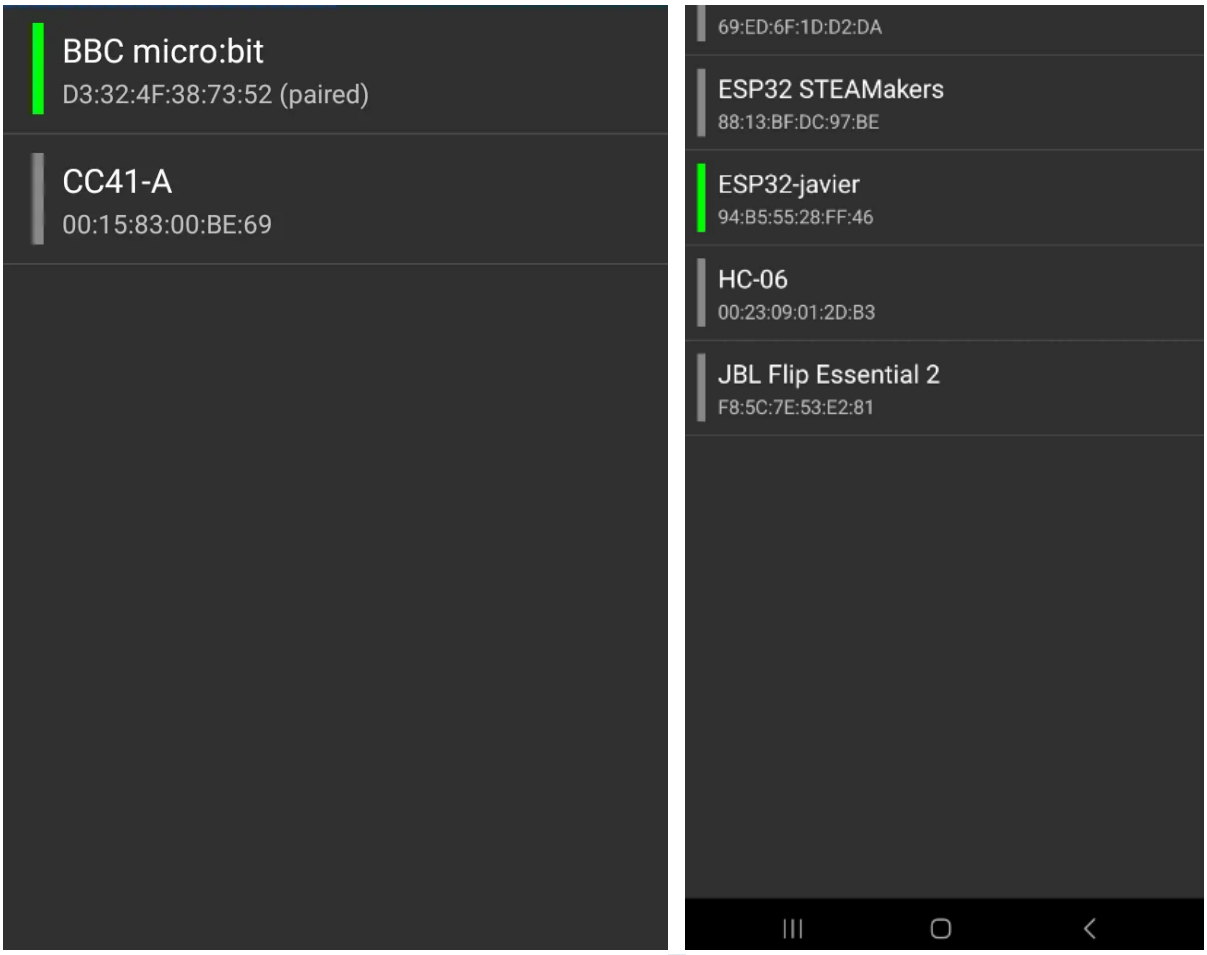



- **Microbit**: Entramos en **Devices** y en **Bluetooth LE** y nos conectamos a la Micro:bit
- **HC06** para Arduino Echidna **ESP32....** igual pero en **Bluetooth clásico**

Aquí puedes ver dos capturas de dispositivos en Bluetooth clásico y Bluetooth BLE

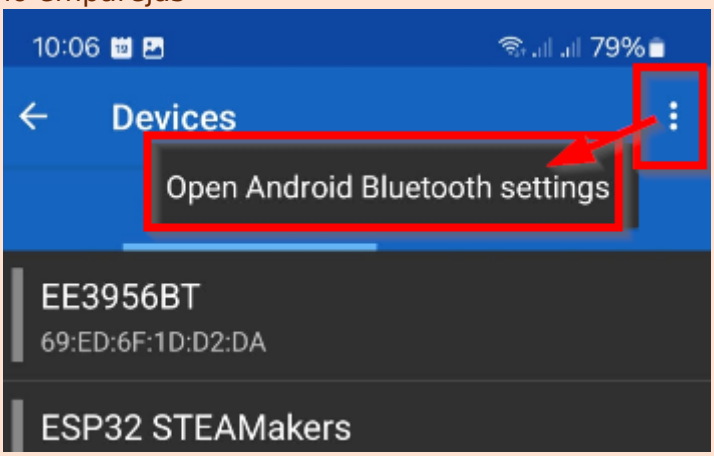
> LOS QUE ESTAN EN **VERDE** SON LOS QUE TIENES EMPAREJADOS Y PUEDES CONECTARTE

Copyright 2026




Copyright 2026

**¿Y si no aparece o no esta emparejado?** Entras en el diálogo de Android de Bluetooth y lo emparejas




## CONECTARTE

Una vez seleccionado el dispositivo emparejado ya puedes conectarte :

1. Menú
2. Entras en Terminal
3. Enchufe
4. Sale conectado, ya estas preparado para enviar y recibir



Copyright 2026

# Resultado con la APP Serial Bluetooth Terminal

## En Makecode

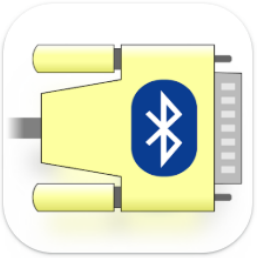
En la micro:bit hemos puesto este programa en Makecode :

https://makecode.microbit.org/#pub:S48588-28779-49040-07772

## En la APP

Copyright 2026



Podemos hacer:

- **Enviar** un mensaje, que como hemos definido anteriormente en Makecode tiene que ir entre #
  - En nuestro caso enviaremos #ON# o #OFF# para encender / apagar el motor
- **Recibir** un mensaje, se visualizará lo que nos envíe la micro:bit que en Makecode hemos definido que será
  - la temperatura
  - si llueve, muestra la advertencia

## Resultado

https://www.youtube.com/embed/Nysl6CQhAOw

Agradecimientos al tutorial de https://lopegonzalez.es/eso-y-bachillerato/proyectos-con-microbit/control-remoto-por-bluetooth/

Copyright 2026

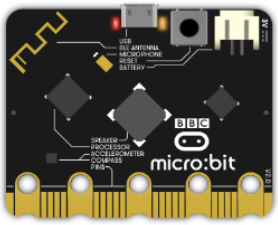
# APP INVENTOR extensiones bluetooth LE y microbit

## Descargas e instalación

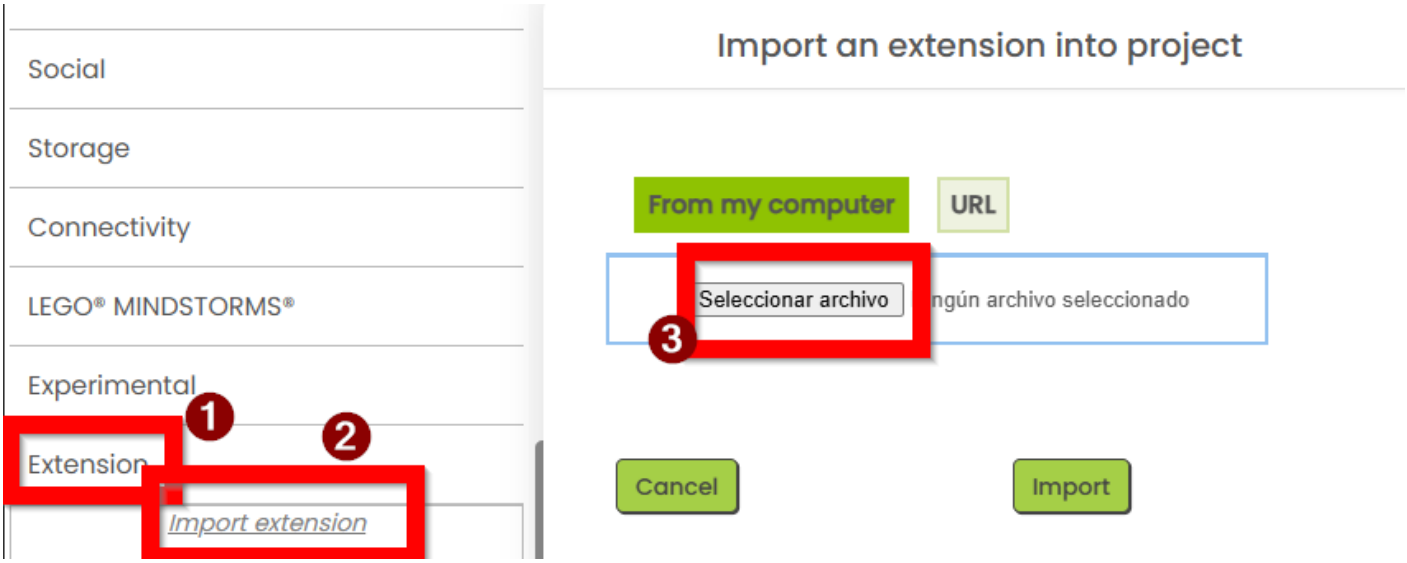
Necesitamos estas extensiones para poder crear una APP que pueda enviar y recibir con nuestra micro:bit

Página de descargas https://iot.appinventor.mit.edu/#/

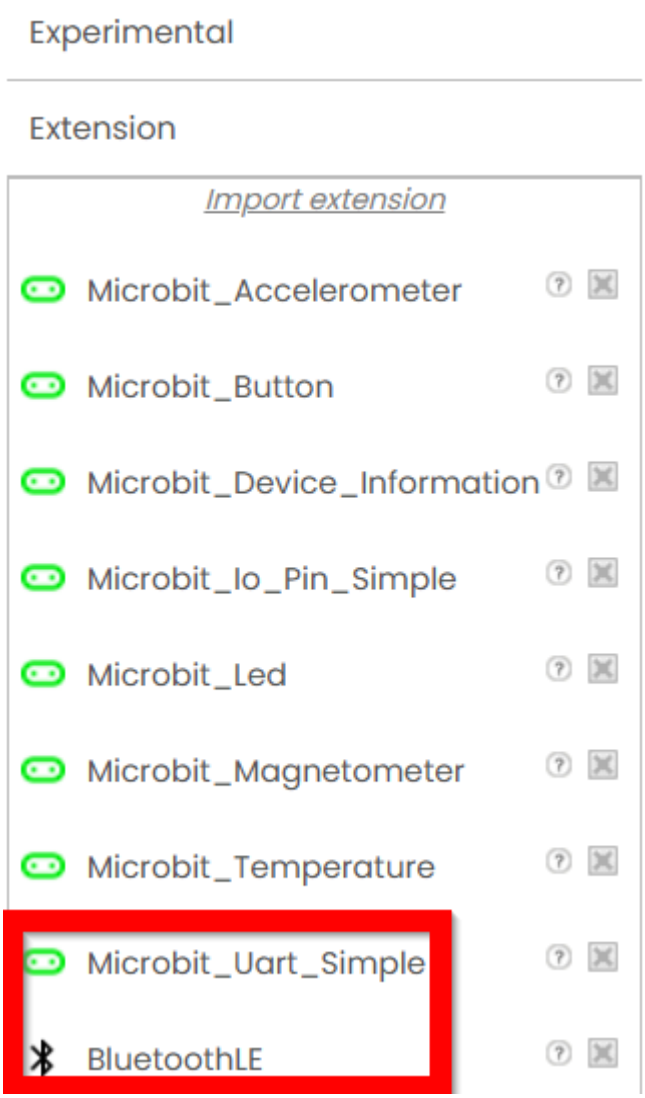
**Supported Extensions**

| Device | Description | Extension |
|---|---|---|
| Bluetooth® | **Bluetooth low energy**<br>Bluetooth Low Energy, also referred to as Bluetooth LE or simply BLE, is a new protocol similar to classic Bluetooth except that it is designed to consume less power while maintaining comparable functionality. | Download |
|  | **BBC micro:bit**<br>The micro:bit is a computing platform from the BBC. It is an open platform for developing all manner of projects and is programmable by many different editors, including a blocks editor provided by Microsoft. Learn more about the micro:bit at the Micro:bit Educational Foundation's website. | Download |

Una vez descargadas, vamos al APP INVENTOR https://ai2.appinventor.mit.edu y las instalamos en extensiones :




Una vez instaladas, se visualizan como extensiones abajo del menú. Las dos últimas son las que utilizaremos:




La extensión Bluetooth tiene diversas funciones que tienes su descripción aquí (English). Para poder instalarla, vamos al APP INVENTOR https://ai2.appinventor.mit.edu

# APP INVENTOR tu programa

Parecido al programa que realizamos con la Micro:bit sola en

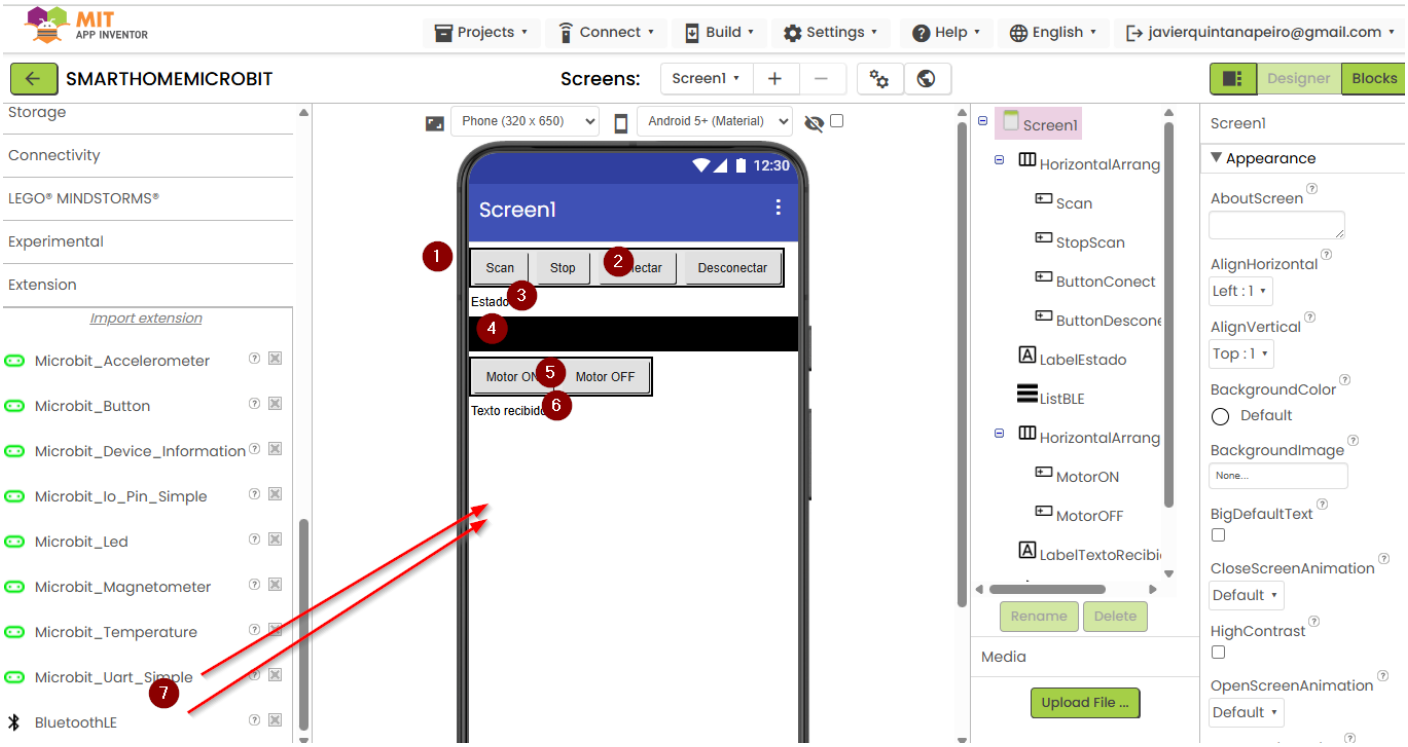
https://libros.catedu.es/books/microbit-car/page/bluetooth-app-inventor-programa

## En DESIGNER

incorporamos:

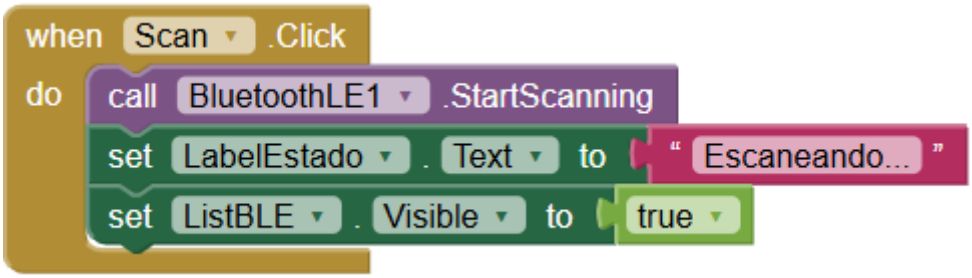
1. **HorizontalArrangment** para que los botones queden alineados horizontalmente
2. **Botones**
    1. Scan
    2. Stop
    3. Conectar
    4. Desconectar
3. **Label** que dirá el estado de la conexión. Lo llamaremos **LabelEstado**
4. **ListView** que lo llamaremos **ListBLE** donde mostrará los diferentes dispositivos Bluetooth LE que detecta
5. **Botones** para encender o apagar el motor de la maqueta
6. Un **Label** que lo llamaremos **LabelTextoRecibido** que mostrará el mensaje desde micro:bit
7. Añadimos los elementos de las extensiones que hemos instalado anteriormente
    1. Microbit_UART_Simple
    2. BluetoothLE

Copyright 2026

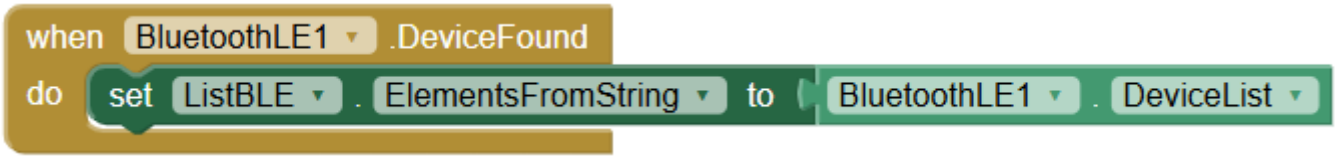



## En Blocks

Cuando escaneemos, que el elemento empiece el escaneado y la lista se vuelva visible, además de que LabelEstado diga que esta escaneando




Si ha encontrado un dispositivo, que lo vaya añadiendo a la lista ListBLE

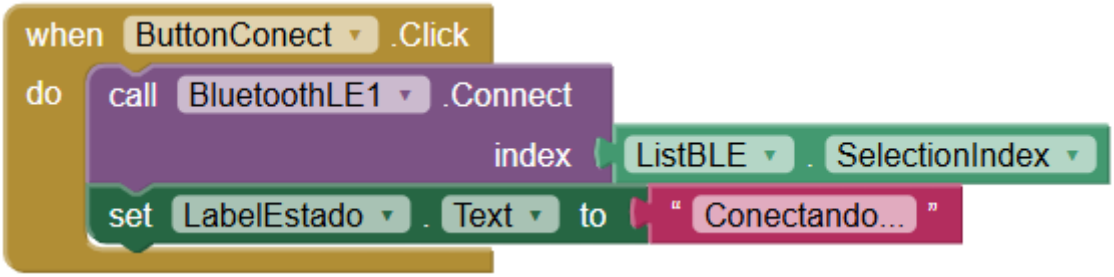



Cuando le digamos que pare, simplemente se lo mandamos al dispositivo y LabelEstado lo informa

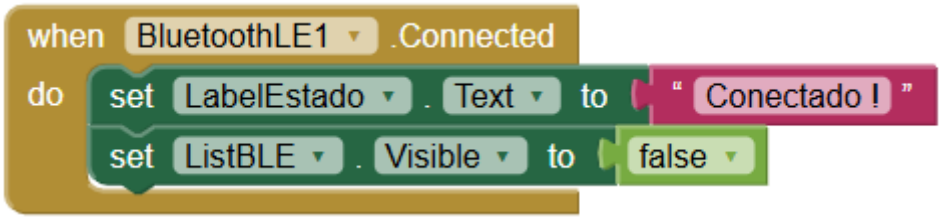
Copyright 2026




Cuando le demos a conectar, pues conecta con el seleccionado en ListBLE y LabelEstado informa



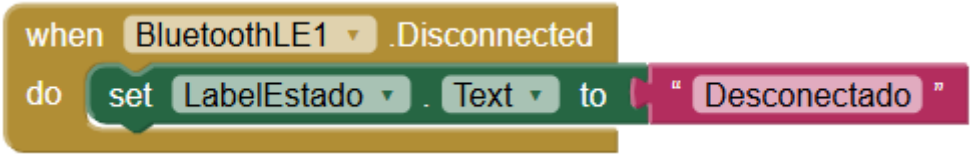

Si conecta, pues LabelEstado informa y ListBLE no es necesaria por lo tanto se oculta, pues entorpece la visión



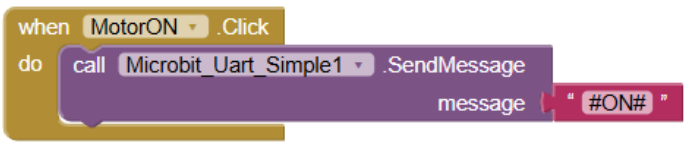

Si queremos desconectar, pues le decimos al elemento BluetoothLE que desconecte

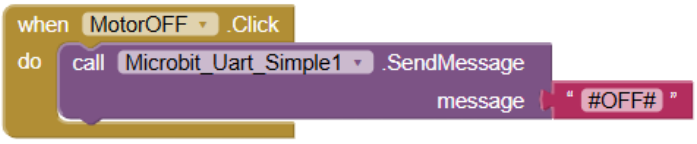



Si se ha desconectado (voluntariamente al dar al botón anterior, o involuntariamente pues el dispositivo se ha desconectado, o esta muy lejos... etc) que informe

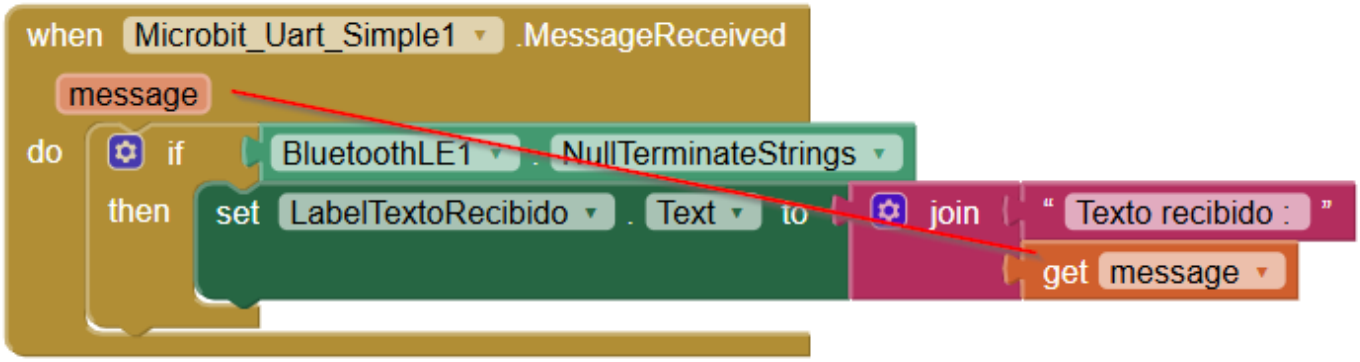


Si apretamos el botón ON o OFF, le enviamos el texto ON o OFF entre "#" pues así lo hemos definido en el programa Makecode: (si quieres saber más sobre estas instrucciones UART aquí)




Si se ha recibido un mensaje, pues que lo visualice, pero primero comprueba que el mensaje no este vacío
NOTA el mensaje "**message**" lo arrastras desde la instrucción "**when**.." tal y como señala la línea roja




**SMARTHOMEMICROBIT.aia**

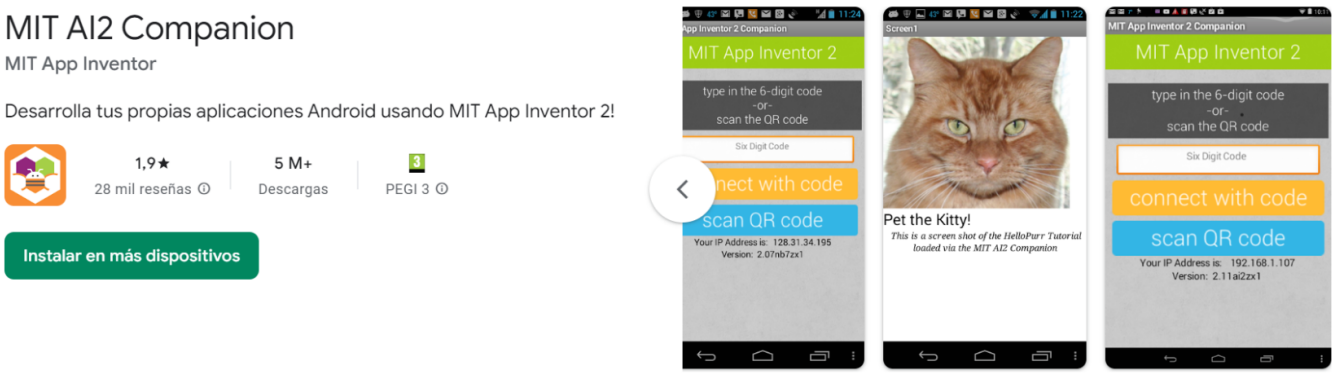
## La APP a tu móvil

Tienes dos opciones

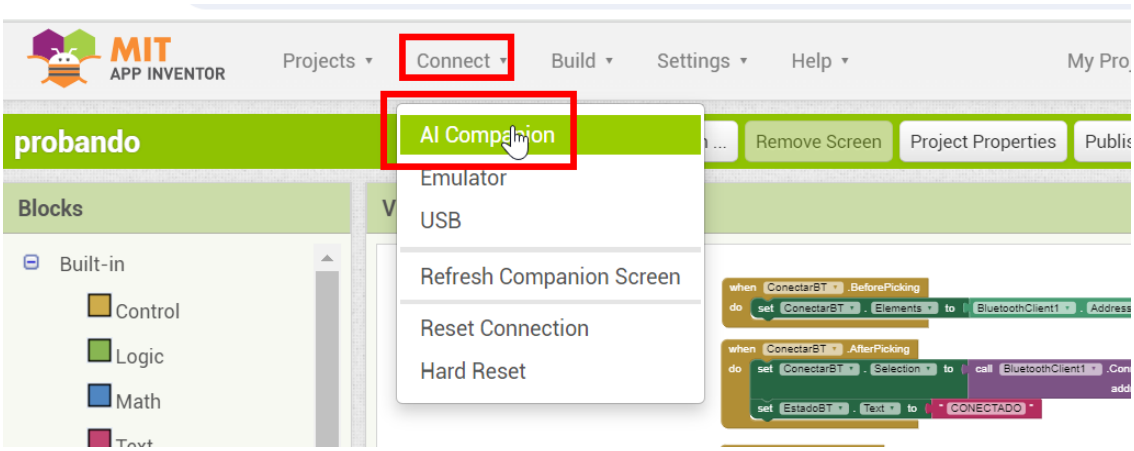
- **EN VIVO CONNECT - AI COMPANION** esta opción es la más rápida, y realmente lo simula a través de la APP INVENTOR.
  - Tienes que tener instalada la APP MIT AI2 COMPANION
  - Se le pasa el código de tu APP a la APP
- **OTRAS OPCIONES**
  - Ver https://appinventor.mit.edu/explore/ai2/setup

# OPCIÓN EN VIVO AI COMPANION

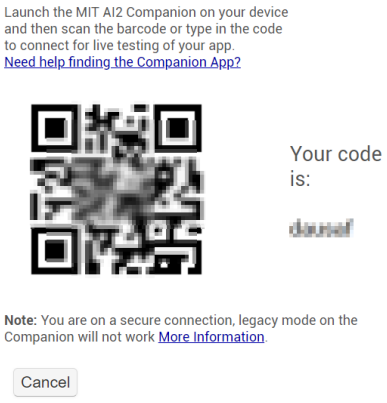
Instalas la APP MIT AI2 COMPANION



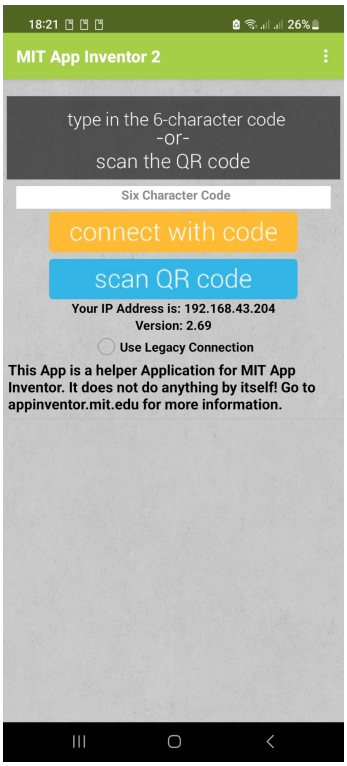

En APP INVENTOR




Y sale un código y un QR asociado al código




Abrimos la APP MIT AI2 COMPANION y metemos el código anterior (o lo escaneamos con el QR)




En APP INVENTOR verás que sale una barra de progreso enviando tu APP a tu móvil. Cuando termina automáticamente lo ejecuta.

## A jugar...

https://www.youtube.com/embed/ObcnN212PcQ