

# CyberPi

## Guía de usuario

José Manuel Ruiz Gutiérrez



makeblock  
education

**ROBOTIX**  
Hands-on Learning

## Datos del Autor

Profesor. **José Manuel Ruiz Gutiérrez**

Ingeniero de Telecomunicaciones

Catedrático de Tecnología (Educación Secundaria)

Experto Universitario en Informática Educativa (UNED)

ExProfesor Asociado, Dto de Sistemas en la Escuela Superior de Informática UCL  
(Ciudad Real)

ExProfesor en la Escuela de Artes de Tomelloso (Ciudad Real). Dto. Diseño de  
Productos

Blogs:

- [Educación Secundaria y Universitaria: Robótica Control Simulación](#)
- [IA IoT En la Educacion](#)
- [M5 Famyli STEM Design for Education](#)

## Índice

1. **CyberPi** Visión General
2. Shield Batería/Conexiones
3. Guía de Funcionamiento de **CyberPi**
  - 3.1. Antes de usar **CyberPi**
    - 3.1.1. Conectar **CyberPi**
  - 3.2. Conozca su **CyberPi**
    - 3.2.1. Funciones de Hardware
      - 3.2.1.1. Funciones predeterminadas del Joystick y los botones
  - 3.3. La página de inicio de **CyberOS**
    - 3.3.1. Configuración del Idioma del Sistema
    - 3.3.2. Cambiar de Programa
4. Programación
  - 4.1. Descarga e instalación del software requerido
  - 4.2. Programación basada en bloques
    - 4.2.1. Agregar y conectar **CyberPi**
    - 4.2.2. Programas de ejemplo
    - 4.2.3. Ayuda de los bloques
    - 4.2.4. Lenguajes de programación
    - 4.2.5. Ejemplo: Programa “Marquesina”
  - 4.3. Programación Python
5. Preguntas
6. Más Información
7. Librerías específicas para **CyberPi**
  - 7.1. Librería: Audio
    - 7.1.1. Reproduce sonido y espera hasta que acaba.
    - 7.1.2. Reproduce sonido especificado
    - 7.1.3. Empieza a grabar sonidos
    - 7.1.4. Detiene la grabación
    - 7.1.5. Reproduce los sonidos grabados hasta que acaba
    - 7.1.6. Reproduce los sonidos grabados
    - 7.1.7. Reproduce el sonido especificado para los ritmos elegidos.
    - 7.1.8. Cambia la velocidad de reproducción de **CyberPi**
    - 7.1.9. Establece la velocidad de reproducción de **CyberPi**
    - 7.1.10. Devuelve la velocidad de reproducción de **CyberPi**
    - 7.1.11. Cambia el volumen de reproducción de **CyberPi**
    - 7.1.12. Establece el volumen de reproducción de **CyberPi**
    - 7.1.13. Devuelve el volumen de reproducción de **CyberPi**
    - 7.1.14. Reproduce un sonido con frecuencia y duración específica
    - 7.1.15. Reproduce un sonido a la frecuencia especificada
    - 7.1.16. Detiene la reproducción de todos los sonidos
  - 7.2. Librería: LED
    - 7.2.1. Muestra animación LED especificada hasta que finaliza
    - 7.2.2. Ilumina los LEDs de **CyberPi** con los colores especificados
    - 7.2.3. El color de los LEDs se desplaza de izquierda a derecha en el número especificado de posiciones
    - 7.2.4. Ilumina los LEDs especificados en el color especificado durante el período especificado

- 7.2.5. Ilumina los LEDs especificados en el color especificado
- 7.2.6. Ilumina los LED en el color especificado que es la combinación de la intensidad de rojo, verde y azul durante el período establecido.
- 7.2.7. Ilumina los LEDs en el color especificado que es la combinación de la intensidad de rojo, verde y azul
- 7.2.8. Cambia el brillo de los LED de **CyberPi**
- 7.2.9. Establece el brillo de los LED de **CyberPi**
- 7.2.10. Informa el brillo actual de los LED de **CyberPi**
- 7.2.11. Apaga los LED especificados en **CyberPi**
- 7.3. Librería: PANTALLA
  - 7.3.1. Muestra textos en la pantalla de **CyberPi** con saltos de línea forzados
  - 7.3.2. Muestra textos en la pantalla de **CyberPi** con saltos de línea automáticos
  - 7.3.3. Muestra el texto especificado en el tamaño especificado en la pantalla de **CyberPi**
  - 7.3.4. Muestra el texto en la posición especificada de la pantalla de **CyberPi** con el tamaño de fuente especificado
  - 7.3.5. Muestra el texto en el tamaño de fuente especificado en la pantalla de **CyberPi** con las coordenadas especificadas como punto de inicio.
  - 7.3.6. Agrega un dato y muestra un gráfico de líneas en la pantalla de **CyberPi**
  - 7.3.7. Establece el espacio entre los puntos de datos en un gráfico de líneas
  - 7.3.8. Agrega un dato y muestra un gráfico de barras en la pantalla de **CyberPi**
  - 7.3.9. Agrega un dato y muestra una tabla en la pantalla de **CyberPi**
  - 7.3.10. Establece el color del pincel
  - 7.3.11. Establece el pincel en el color que es la combinación especificada de rojo, verde y azul.
  - 7.3.12. Establece la pantalla de **CyberPi** en la dirección de visualización especificada
  - 7.3.13. Borra la pantalla de **CyberPi**
- 7.4. Librería: SENSORES DE MOVIMIENTO
  - 7.4.1. Hace que el sprite especificado siga a **CyberPi** con la sensibilidad especificada
  - 7.4.2. Determina si **CyberPi** se mueve en la posición indicada.
  - 7.4.3. Determina si **CyberPi** se coloca en la postura especificada
  - 7.4.4. Informa con qué fuerza se agita **CyberPi**
  - 7.4.5. Informa la dirección en la que se mueve **CyberPi**
  - 7.4.6. Informa la velocidad a la que se mueve **CyberPi**
  - 7.4.7. Informa el ángulo de la postura especificada
  - 7.4.8. Informa la aceleración de **CyberPi** en el eje especificado
  - 7.4.9. Informa la velocidad angular a la que **CyberPi** gira alrededor del eje especificado
  - 7.4.10. Informa que el ángulo **CyberPi** se gira alrededor del eje especificado
  - 7.4.11. Restablece el ángulo que **CyberPi** gira alrededor del eje especificado (ejes) a cero
  - 7.4.12. Restablece el ángulo de guiñada de **CyberPi** a cero
- 7.5. Librería: SENSORES
  - 7.5.1. Determina si el joystick se mueve en la dirección especificada
  - 7.5.2. Informa el número de veces que se mueve el joystick en la dirección especificada
  - 7.5.3. Restablece el número de veces que se mueve el joystick en la dirección especificada a cero

- 7.5.4. Determina si se presiona el botón especificado
- 7.5.5. Informa el número de veces que se presiona el botón especificado
- 7.5.6. Restablece el número de veces que se presiona el botón especificado a cero
- 7.5.7. Informa el volumen ambiental detectado por el micrófono en **CyberPi**
- 7.5.8. Informa la intensidad de la luz ambiental detectada por el sensor de luz en **CyberPi**
- 7.5.9. Reporta el valor de conteo del temporizador
- 7.5.10. Restablece el temporizador a cero
- 7.5.11. Informa el nombre de host de **CyberPi**
- 7.5.12. Informa el nivel de batería
- 7.6. Librería: LAN
  - 7.6.1. Transmite el mensaje especificado en la LAN
  - 7.6.2. Difunde el mensaje especificado con el valor especificado en la LAN
  - 7.6.3. Ejecuta los bloques siguientes cuando se recibe el mensaje especificado
  - 7.6.4. Informa el valor recibido por **CyberPi** a través del mensaje especificado
  - 7.6.5.
- 7.7. Librería: IA
  - 7.7.1. Conecta **CyberPi** a la red Wi-Fi especificada
  - 7.7.2. Determina si **CyberPi** se ha conectado a Wi-Fi
  - 7.7.3. Reconoce el texto especificado y lo lee en voz alta en el idioma reconocido
  - 7.7.4. Reconoce el idioma especificado durante el período especificado
  - 7.7.5. Informa las palabras reconocidas en un discurso.
  - 7.7.6. Reporta la traducción del texto especificado
- 7.8. Librería: IoT
  - 7.8.1. Conecta **CyberPi** a la red Wi-Fi especificada
  - 7.8.2. Determina si **CyberPi** se ha conectado a Wi-Fi
  - 7.8.3. Envía el mensaje de nube de usuario especificado
  - 7.8.4. Envía mensaje a la nube y el valor especificado
  - 7.8.5. Ejecuta los bloques cuando se recibe el mensaje de nube
  - 7.8.6. Devuelve el valor recibido de la nube
  - 7.8.7. Devuelve las condiciones meteorológicas seleccionadas del distrito especificado
  - 7.8.8. Devuelve la calidad del aire del lugar especificado
  - 7.8.9. Devuelve la hora de salida o puesta del sol del distrito especificado
- 8. Ejercicios realizados con **CyberPi**
  - 8.1. Básicas
    - 8.1.1. Temporizador
    - 8.1.2. Arco Iris
    - 8.1.3. Contador de pasos
    - 8.1.4. Detector de Movimiento
    - 8.1.5. Detector de Sonido
    - 8.1.6. Funciones Trigonométricas
    - 8.1.7. Grabación y Reproducción de Sonidos
    - 8.1.8. Simulación de Sirena
    - 8.1.9. Toca Canción
    - 8.1.10. Traductor de Bolsillo
    - 8.1.11. Control de Motor con la Voz
  - 8.2. Proyectos IoT

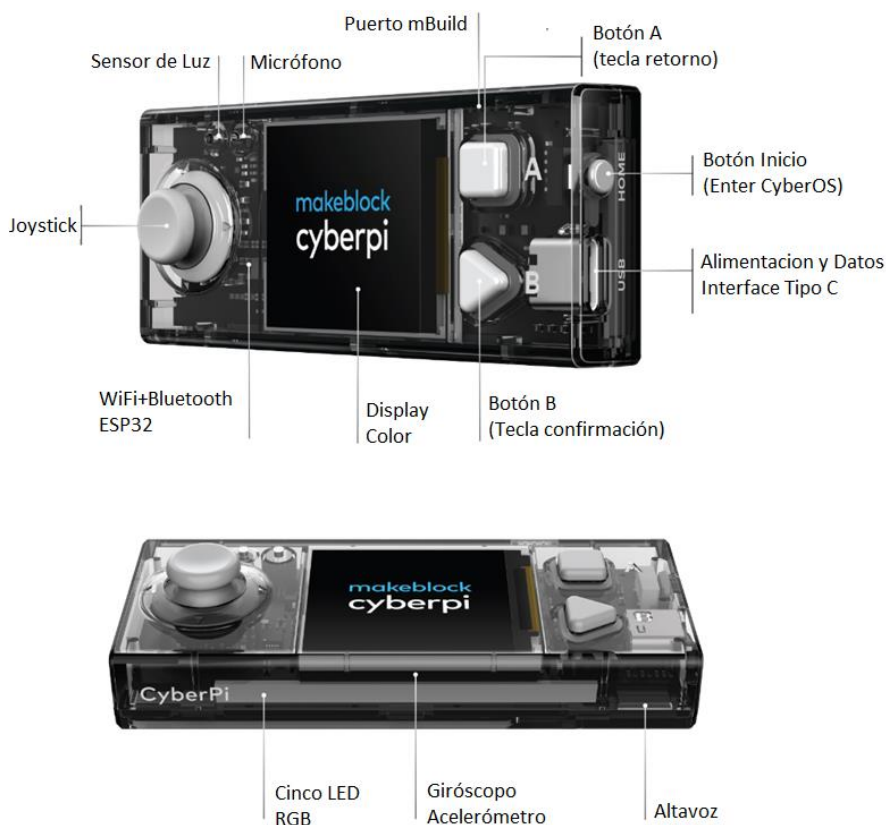
- 8.2.1. Estación Meteorológica Remota
- 8.2.2. Encendido Farolas
- 8.2.3. Telemedida y Telemando

## 1. CyberPi Visión General



**CyberPi** es una tarjeta electrónica de control encapsulada en una caja que ha sido desarrollada de forma independiente por **Makeblock**. Con la estructura compacta y los puertos integrados, se puede ampliar fácilmente. Es compatible con **mBlock5** y [mBlock-Python Editor](#), es aplicable a múltiples escenarios educativos, incluida la enseñanza a gran escala, la enseñanza comunitaria y la educación y formación en línea / fuera de línea; cubre múltiples campos de enseñanza, incluidos la programación, la creación y los robots; y, por lo tanto, puede satisfacer las necesidades educativas diversificadas, como **IA**, **IoT**, **ciencia de datos** y diseño de **interfaz de usuario**.

En la siguiente imagen vemos sus principales partes y funciones.



## Especificaciones técnicas

Los criterios con los que esta tarjeta se pone en circulación son los que se acogen a las siguientes características técnicas.

Nombre		CyberPi
<b>Chip</b>		ESP32-WROVER-B
<b>Procesador</b>	<b>Procesador Principal</b>	Xtensa® 32-bit LX6 dual-core
	<b>Clock frequency</b>	240 MHz
<b>Memoria en tarjeta</b>	<b>ROM</b>	448 KB
	<b>SRAM</b>	520 KB
<b>Memoria Extendida</b>	<b>SPI Flash</b>	8 MB
	<b>PSRAM</b>	8 MB
<b>Sistema Operativo</b>		<b>CyberOS</b> , desarrollado de forma independiente por Makeblock
<b>Comunicación Inalámbrica</b>		Wi-Fi Modo-Dual-modo Bluetooth
<b>Puertos físicos</b>		Micro USB puerto (Tipo-C) Puerto para conectar placas de extensión Puerto para conectar módulos electrónicos (comunicación en serie)
<b>Hardware versión</b>		V1.0
<b>Dimensiones</b>		84 mm × 35 mm × 13 mm (altura × anchura × profundidad)
<b>Peso</b>		36 g

## Características Funcionales

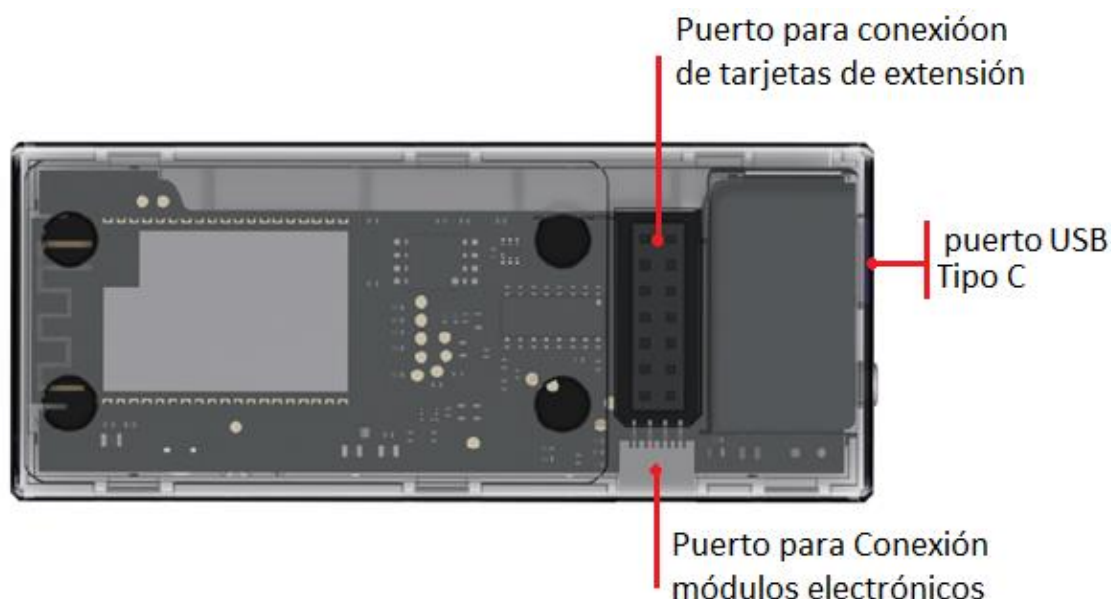
- Pantalla a todo color, que proporciona interfaces de usuario fáciles de usar para la interacción hombre-máquina
- Sistema **CyberOS**, que le permite ejecutar los programas predefinidos, configurar el idioma del sistema y actualizar el sistema a través del joystick y los botones integrados
- Un puerto micro USB (tipo C) para conectarse a PC para suministro de energía y comunicación
- Un puerto de módulo electrónico para conectar módulos electrónicos
- Un puerto de placa de extensión para conectar a placas de extensión
- Múltiples sensores integrados, como el sensor de luz y el giroscopio, que proporcionan múltiples tipos de salida de datos.
- Cinco LED, que le permiten presentar abundantes efectos de luz
- Módulo integrado de Bluetooth y Wi-Fi, que permite la comunicación inalámbrica
- Admite la programación **mBlock5**, que está destinada a usuarios de todas las edades, incluidos aquellos sin experiencia en programación
- Admite la programación de Python, para la que se proporciona la biblioteca **CyberPi**

## Dimensiones



## Descripción de puerto

**CyberPi** está equipado con un puerto USB tipo C, un puerto de módulo electrónico y un puerto de placa de extensión, que le permiten conectarse fácil y rápidamente a varios tipos de módulos electrónicos y placas de extensión.



### Micro USB puerto (Tipo-C)

El puerto Micro USB permite que **CyberPi** se conecte a varios tipos de dispositivos informáticos para la alimentación y la comunicación.

### Puerto para conectar tarjetas de extensión

Puede conectar fácilmente **CyberPi** a una placa de extensión a través del puerto de la placa de extensión. Actualmente, la placa de extensión Pocket Shield está disponible para **CyberPi**.

Pocket Shield está equipado con una batería recargable incorporada que puede suministrar energía para **CyberPi** y proporciona interfaces de 2 y 3 pines que se pueden para conectar servos, tiras de LED y motores, lo que mejora significativamente la extensibilidad de **CyberPi**. Para más información, ver "[Pocket Shield](#)."

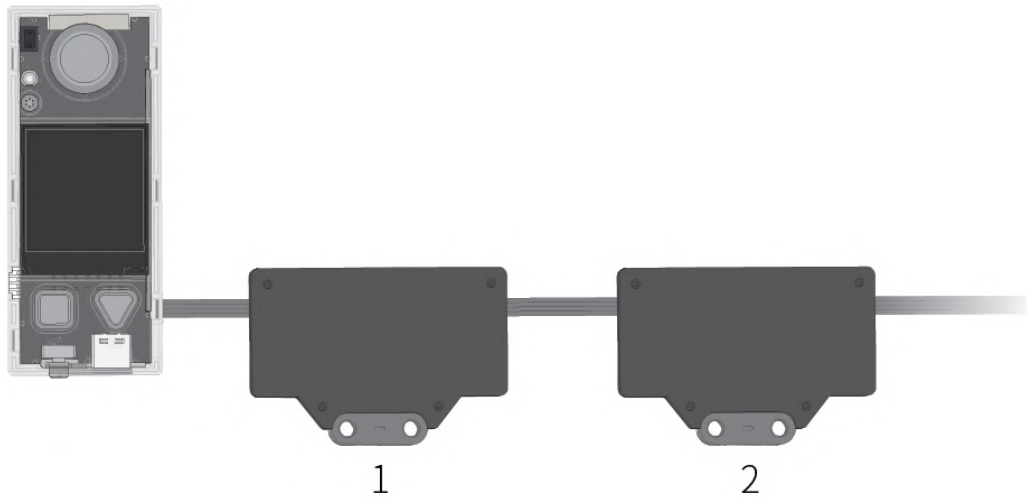
### Puerto para conectar módulos electrónicos

Puede conectar **CyberPi** a varios módulos electrónicos en serie a través del puerto del módulo electrónico.

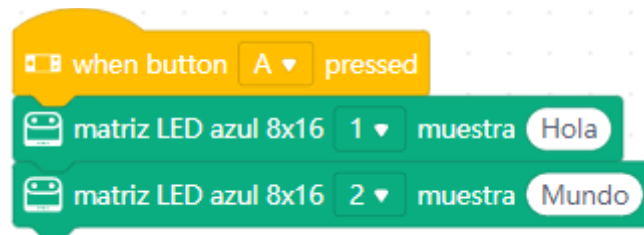
**CyberPi** puede identificar de forma inteligente la dirección de los módulos, lo que simplifica su programación. No es necesario que establezca la información sobre la dirección de los módulos cuando agrega o quita un módulo.

## Ejemplo:

Conecte **CyberPi** a múltiples matrices LED

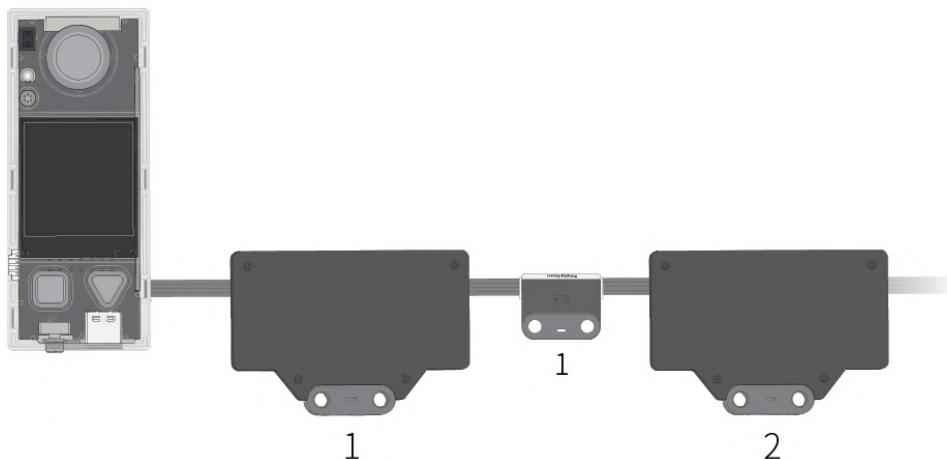


Después de conectar **CyberPi** a múltiples matrices de LED, solo necesita especificar el lugar de una matriz de LED entre las conectadas en lugar de especificar el puerto al que está conectada la matriz de LED al compilar un programa. Como se muestra en la figura anterior, la primera matriz de LED conectada a **CyberPi** tiene el número 1, la segunda tiene el número 2, y así sucesivamente.



Cuando presiona el botón A en **CyberPi**, la primera matriz de LED muestra "Hola" y la segunda muestra "Mundo".

Cambie las posiciones de los módulos, como se muestra en la siguiente figura.



El programa anterior aún funciona después de agregar un sensor de rango. Cuando presiona el botón A en **CyberPi**, la primera matriz de LED muestra "Hola" y la segunda muestra "Mundo".

## Programación

Puede utilizar **mBlock5** para programar **CyberPi**. **mBlock5** proporciona dos editores, a saber, el editor gráfico basado en bloques (el editor predeterminado, denominado **mBlock5**) y el editor de Python (denominado **mBlock-Python Editor**). Para obtener detalles sobre la programación, consulte "[Software de Programación](#)."

## Más Información

[CyberPi Operation Guide](#)

[Pocket Shield Operation Guide](#)

[CyberPi Series User Manual](#)

[Python API Documentation for CyberPi](#)

[mBlock5 Online Help](#)

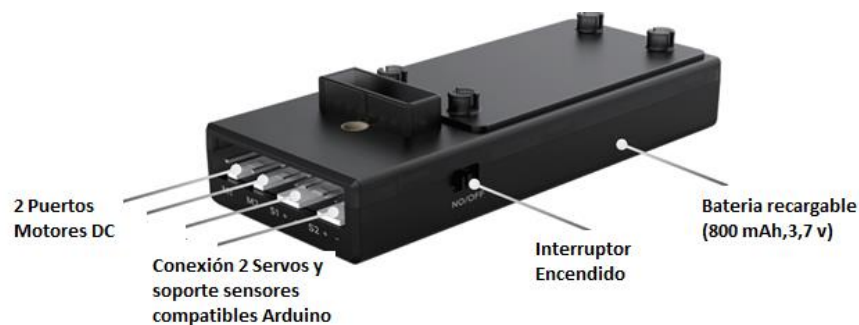
[mBlock-Python Editor Online Help](#)

## 2. Shield Batería-Conexiones



### Vista general

**Pocket Shield** (así lo denomina el fabricante) está equipado con una batería recargable incorporada que puede suministrar energía para **CyberPi** y proporciona interfaces de 2 y 3 pines que se pueden usar para conectar servos, tiras de LED y motores, lo que mejora significativamente la extensibilidad de **CyberPi**.



### Especificaciones

Producto parámetros	Descripción
Micro-procesador	GD32
Batería	3.7V 800mAh
Entrada voltaje/intensidad	5V 700mA
Salida voltaje/intensidad	5V 2A
Duración de Batería	4 horas (Escenarios de uso general, solo como referencia)

Tiempo de Carga	1~2 horas
Modos Comunicación	Comunicación Serie: Tarjeta de control principal a tarjeta de extensión Señal PWM: Puerto de servomotor digital Señal Digital: Puerto de motor DC
Versión Hardware	V1.1
Dimensiones	84×35×19mm
Peso	48g

## Características

- Batería de iones de litio recargable incorporada, que se utiliza para suministrar energía a **CyberPi**
- Dos puertos de motor de CC, utilizados para conectar y accionar motores de CC
- Dos puertos servo digitales, que se utilizan para conectar y controlar servos o tiras de LED
- Un puerto de placa de control principal, lo que le permite conectar fácilmente Pocket Shield a **CyberPi**
- Compatible con la programación **mBlock5**, que está destinada a usuarios de todas las edades, incluidos aquellos sin experiencia en programación.
- Compatible con la programación de Python, para la que se proporciona la biblioteca **CyberPi**

## Actualizar Firmware

Después de que se lanza una nueva versión de firmware, puede conectar **Pocket Shield** a **mBlock5** a través de **CyberPi** para ver y actualizar su firmware. Para obtener detalles sobre cómo actualizar el firmware, consulte "[Como actualizar el firmware?](#)"

## Programación

Puede utilizar **mBlock5** para programar **Pocket Shield**. **mBlock5** proporciona dos editores, a saber, el editor gráfico basado en bloques (el editor predeterminado, denominado **mBlock5**) y el editor de Python (denominado **mBlock-Python Editor**). Para obtener detalles sobre la programación, consulte "[Software Programación](#)."

## Mas Información

[CyberPi Operation Guide](#)  
[Pocket Shield Operation Guide](#)  
[CyberPi Series User Manual](#)  
[Python API Documentation for CyberPi](#)  
[mBlock5 Online Help](#)  
[mBlock-Python Editor Online Help](#)

## 3. Guía de funcionamiento de CyberPi

### 3.1. Antes de usar CyberPi

#### 3.1.1. Conectar CyberPi

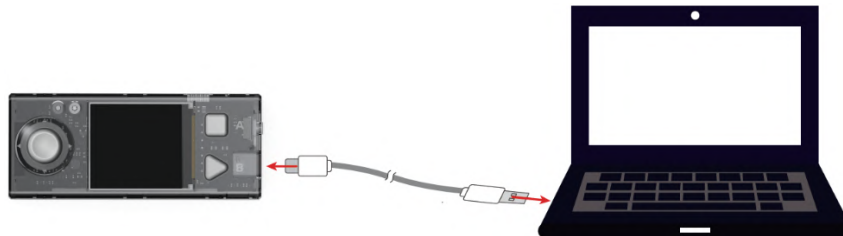
Utilice un cable Micro USB (tipo C) para conectar **CyberPi** a su PC. El PC puede suministrar energía a **CyberPi**.

**Nota:**



- El cable Micro USB no está incluido en el paquete. Necesita comprar uno por separado.
- También puede utilizar una unidad de energía para suministrar energía a **CyberPi**.

Al cargar **CyberPi**, asegúrese de que el voltaje sea inferior a 6 V. El hardware de **CyberPi** puede dañarse si el voltaje es igual o superior a 6 V.



Alternativamente, puede comprar [Pocket Shield](#).

**Pocket Shield** está equipado con una batería recargable incorporada que puede suministrar energía para **CyberPi** y proporciona interfaces de 2 y 3 pines que se pueden usar para conectar servos, tiras de LED y motores, lo que mejora significativamente la extensibilidad de **CyberPi**.

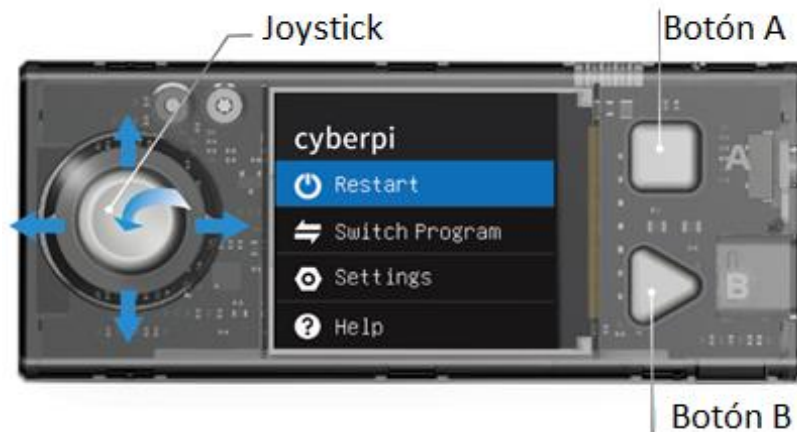
## 3.2. Conozca su CyberPi

### 3.2.1. Funciones de hardware

#### 3.2.1.1. Funciones predeterminadas del joystick y los botones

**CyberPi** está equipado con un joystick y dos botones.

Las funciones predeterminadas del joystick y los botones se describen a continuación.



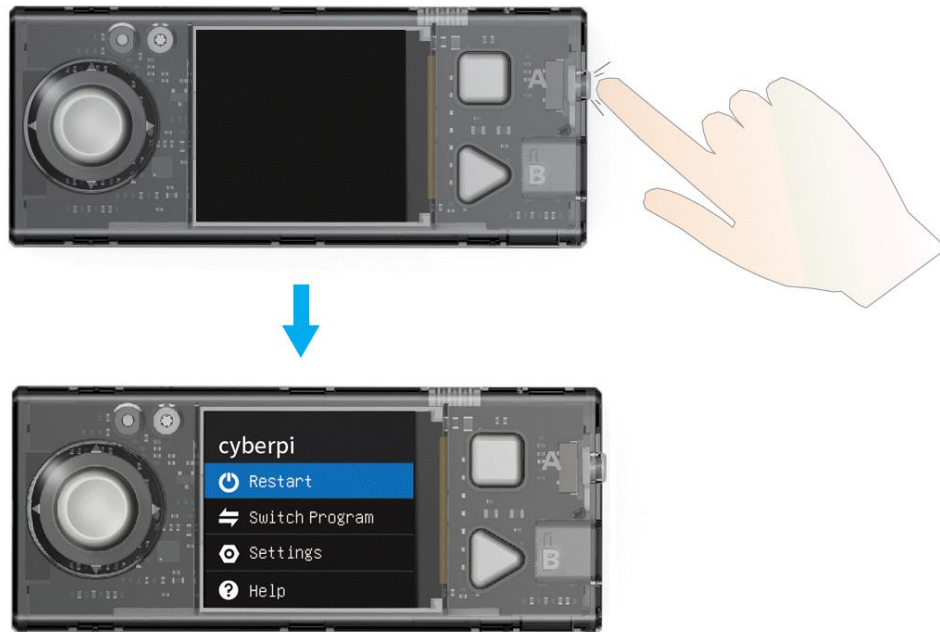
- **Mueva el joystick hacia arriba:** para seleccionar el elemento anterior
- **Mueva el joystick hacia abajo:** para seleccionar el siguiente elemento
- **Mueva el joystick hacia la izquierda:** para seleccionar el elemento de la izquierda
- **Mueva el joystick hacia la derecha:** para seleccionar el elemento correcto
- **Presione el joystick en el medio:** para confirmar la selección e ingresar al menú de nivel inferior, si lo hubiera
- **Botón A:** para volver al menú de nivel superior
- **Botón B:** para confirmar la selección e ingresar al menú de nivel inferior, si lo hubiera

## 3.3. La página de inicio de CyberOS

**CyberPi** ejecuta **CyberOS**.

Después de encenderlo, **CyberPi** muestra la página de inicio del sistema.

Si no muestra la página de inicio, puede presionar el botón Inicio en el lado derecho de **CyberPi** para ingresar a la página de inicio.



Después de iniciar **CyberOS**, puede usar los programas predefinidos, configurar el idioma del sistema y actualizar el sistema a través de Internet.

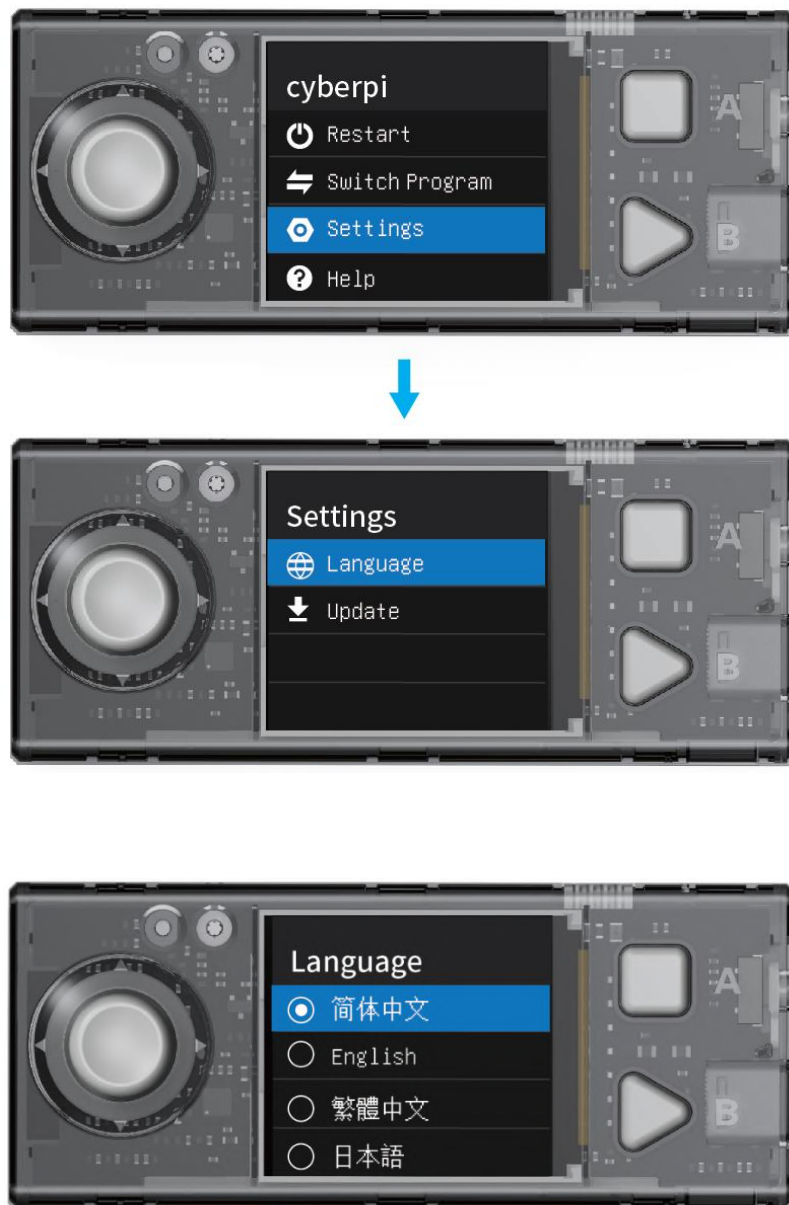
**Puede salir de la página de inicio seleccionando "Reiniciar programa".**

**Nota:** Cuando **CyberPi** está conectado a **mBlock5** y se está programando en el modo **En vivo**, el botón **INICIO** no tiene efecto y no puede ingresar a la página de inicio presionándolo.

### 3.3.1. Configurar el idioma del sistema

Si el idioma predeterminado del sistema no es su idioma nativo, puede cambiarlo. A continuación, se muestran los pasos para cambiar el idioma de inglés a chino como ejemplo:

- (1) Ingrese a la página de inicio, seleccione Configuración moviendo el joystick hacia arriba o hacia abajo, y presione B para ingresar a la página de Configuración "**Settings**".
- (2) En la página Configuración, seleccione Idioma moviendo el joystick hacia arriba o hacia abajo y presione el botón B para ingresar a la página Idioma.



Después de configurar un idioma, el sistema cambia al idioma y regresa a la página de inicio.

### 3.3.2. Cambiar de programa

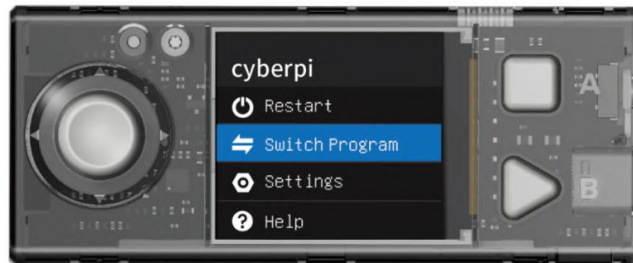
**CyberPi** puede almacenar varios programas y puede cambiarlos. Además, **CyberPi** se entrega con varios programas de ejemplo predeterminados para ayudarlo a comprender sus funciones principales.

**Nota:**

- Se requiere acceso a Internet para los programas de **IA** e **IoT** y, por lo tanto, no están incluidos en los programas de ejemplo predeterminados y se proporcionan en los programas de ejemplo en **mBlock5**. Encuéntrelos seleccionando *Tutoriales > Programas de ejemplo* en **mBlock5**.

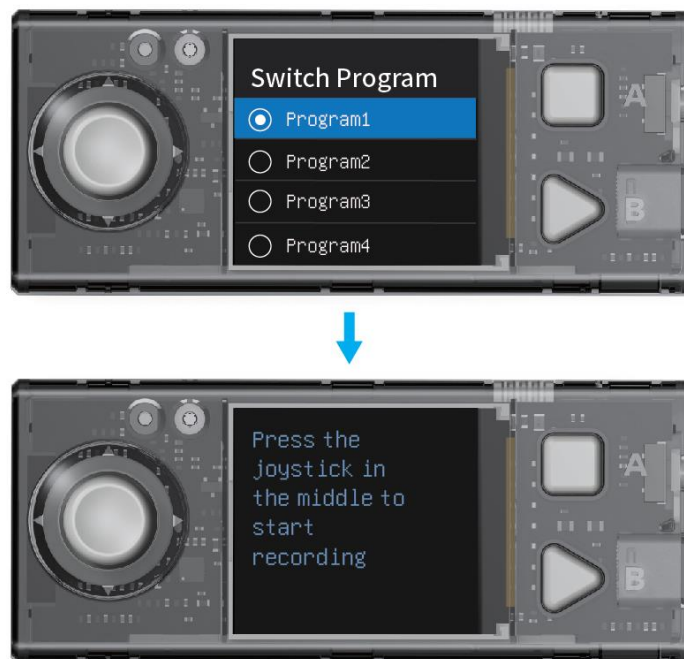
- El programa que cargue en **CyberPi** desde **mBlock5** reemplaza al que utilizó la última vez. Se proporcionan los programas de ejemplo predeterminados y puede restaurarlos cargándolos en su **CyberPi**.

(1) Ingrese a la página de inicio, seleccione Cambiar programa moviendo el joystick hacia arriba o hacia abajo y presione B para ingresar a la página Cambiar programa.



(2) En la página Cambiar programa, seleccione un programa, por ejemplo, Programa1, moviendo el joystick hacia arriba o hacia abajo, y luego presione el botón B.

**CyberPi** se reinicia y ejecuta el Programa 1.



**CyberPi** muestra primero el nombre del programa y luego le solicita que realice los pasos necesarios para ejecutar el programa.

## 4. Programación

Esta sección describe cómo programar **CyberPi** en **mBlock5**. Puede usar las poderosas funciones de **CyberPi** a través de la programación, de las que se realizara una detallada descripción más adelante.

### 4.1. Descarga e instalación del los software requerido

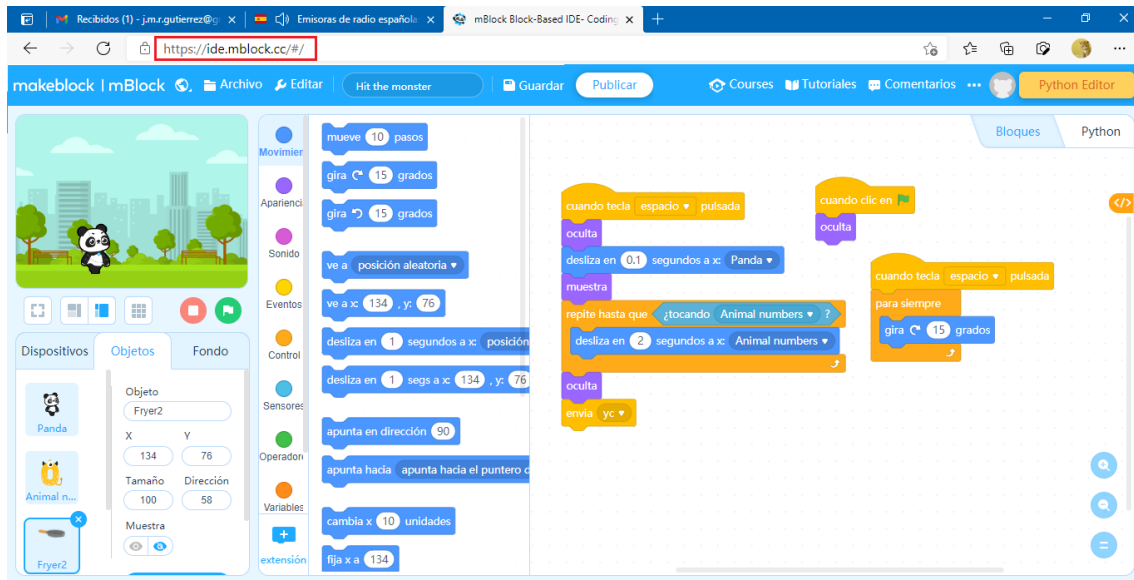
Actualmente, **CyberPi** admite la programación gráfica basada en bloques y la programación Python. Asegúrese de haber descargado e instalado el software necesario.

Lenguaje	Editor	mBlock5 versión	Software requerido	Dirección URL
Scratch, MicroPython	mBlock Block-based Editor	Version PC on Windows/Mac 	<a href="#">mBlock5 for Windows</a> <a href="#">mLink2 for Windows</a>  <a href="#">mBlock5 for Mac</a> <a href="#">mLink2 for Mac</a>	N/A
		Versión Web Windows/Mac 	<a href="#">Google Chrome</a> <a href="#">mLink2 for Windows</a> <a href="#">mLink2 for Mac</a>	<a href="https://ide.mBlock.cc">https://ide.mBlock.cc</a>
		Versión Web version en Chromebook 	<a href="#">mLink for Chromebook</a>  Para detalles de Cómo utilizar <b>mBlock5</b> para programar <b>CyberPi</b> on Chromebook, ver " <a href="#">Programming CyberPi on Chromebook</a> ."	<a href="https://ide.mBlock.cc">https://ide.mBlock.cc</a>
Python, MicroPython	mBlock-Python Editor	Versión Web en Windows/Mac 	<a href="#">Google Chrome</a> <a href="#">mLink2 for Windows</a> <a href="#">mLink2 for Mac</a>	<a href="https://python.makeblock.com/">https://python.makeblock.com/</a>

## 4.2. Programación basada en bloques

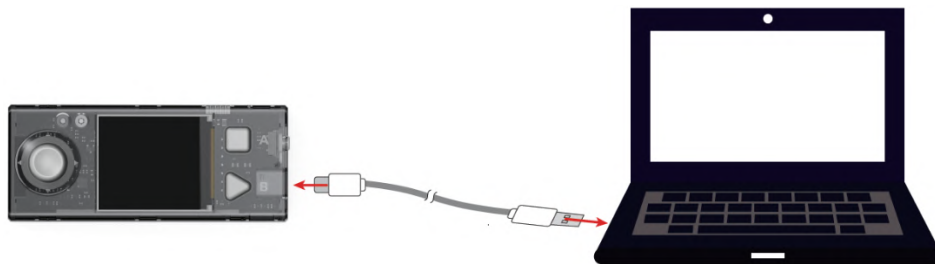
### 4.2.1. Agregar y conectar CyberPi

(1) Abra la [versión web de mBlock5](https://ide.mblock.cc/) en su PC.



**Nota:** Para utilizar algunas funciones del servicio en la nube, debe iniciar sesión en su cuenta **mBlock5**. Registre una cuenta si no la tiene.

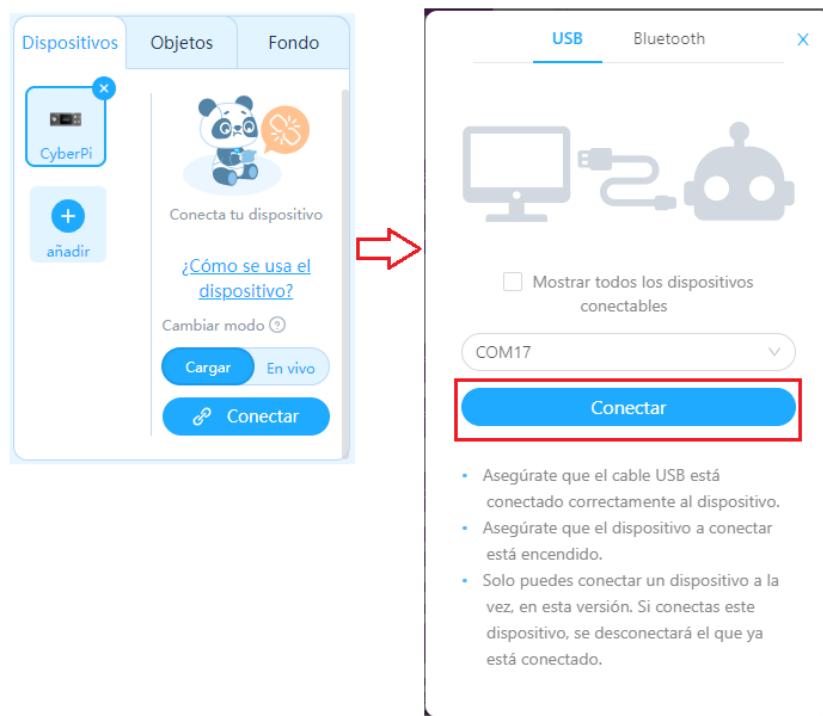
(2) Utilice un cable Micro USB (tipo C) para conectar **CyberPi** a su PC.



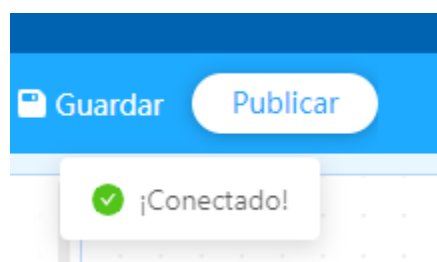
(3) Haga clic en + agregar en la pestaña Dispositivos, seleccione **CyberPi** en la biblioteca de dispositivos y haga clic en Aceptar.



(4) Haga clic en **Conectar** para conectar **CyberPi** a **mBlock5**.



Se muestra un mensaje después de que **CyberPi** está conectado, lo que indica que la conexión se realizó correctamente.



(5) Configure el modo de programación.

**mBlock5** proporciona dos modos de programación, a saber, “**En vivo**” y “**Cargar**”. Puede hacer clic para cambiar los modos.



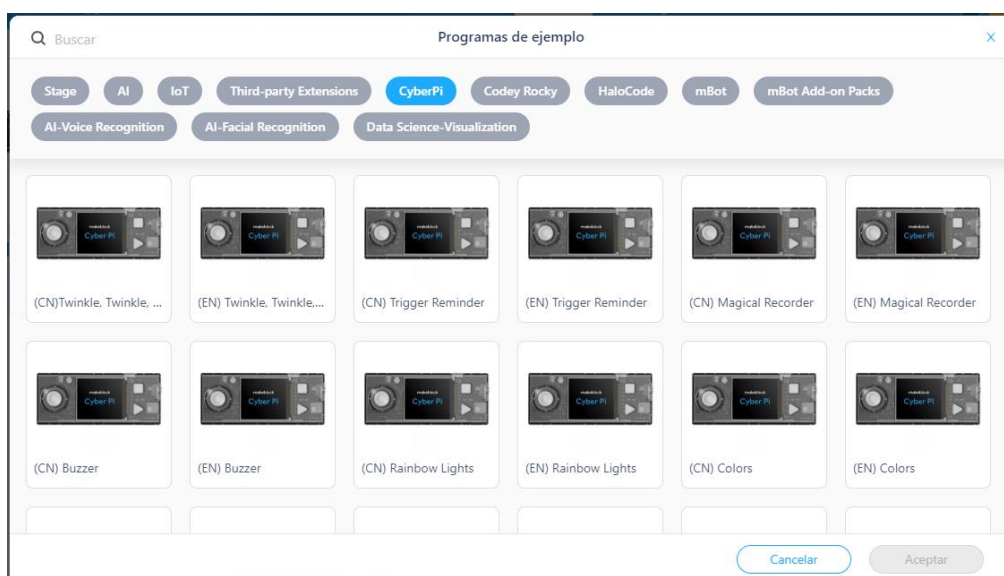
**En vivo:** En este modo, puede ver el efecto de ejecución del programa en tiempo real, lo que facilita la depuración del programa. En este modo, debe mantener **CyberPi** conectado a **mBlock5**. Si están desconectados, el programa no se puede ejecutar.

**Cargar:** en este modo, debe cargar el programa compilado en **CyberPi**. Después de cargarse correctamente, el programa aún puede ejecutarse correctamente en **CyberPi** cuando se desconecta de **mBlock5**.

## 4.2.2 Programas de ejemplo

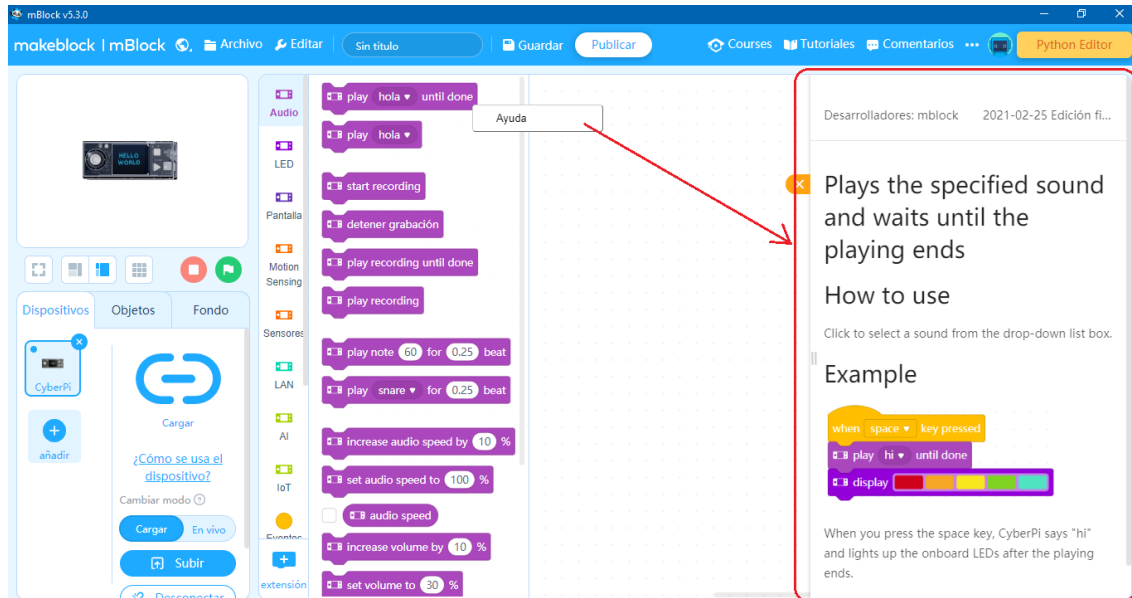
**mBlock5** proporciona una gran cantidad de programas de ejemplo.

Elija Tutoriales> Programas de ejemplo y haga clic en **CyberPi** para ver los programas de ejemplo proporcionados para **CyberPi**.



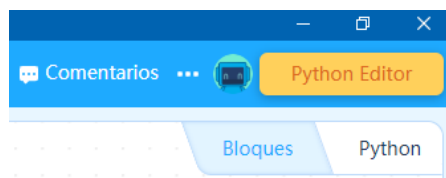
### 4.2.3 Ayuda de los bloques

**mBlock5** proporciona varios tipos de bloques para **CyberPi**. Si no entiende un bloque al usarlo, puede hacer clic con el botón derecho y hacer clic en Ayuda que aparece.



### 4.2.4 Lenguajes de programación

**mBlock5** proporciona dos lenguajes de programación para **CyberPi**, a saber, programación basada en bloques y Python. En el modo de carga, puede hacer clic en los botones de la derecha para cambiar los lenguajes de programación.



Además, al programar **CyberPi** en el lenguaje basado en bloques, puede hacer clic en el botón de cambio a la derecha para ver las declaraciones de Python correspondientes (obtenidas al convertir los bloques).



**Nota:** Puede ver la [documentación de la API de Python para CyberPi](#) para conocer más funciones de **CyberPi**.

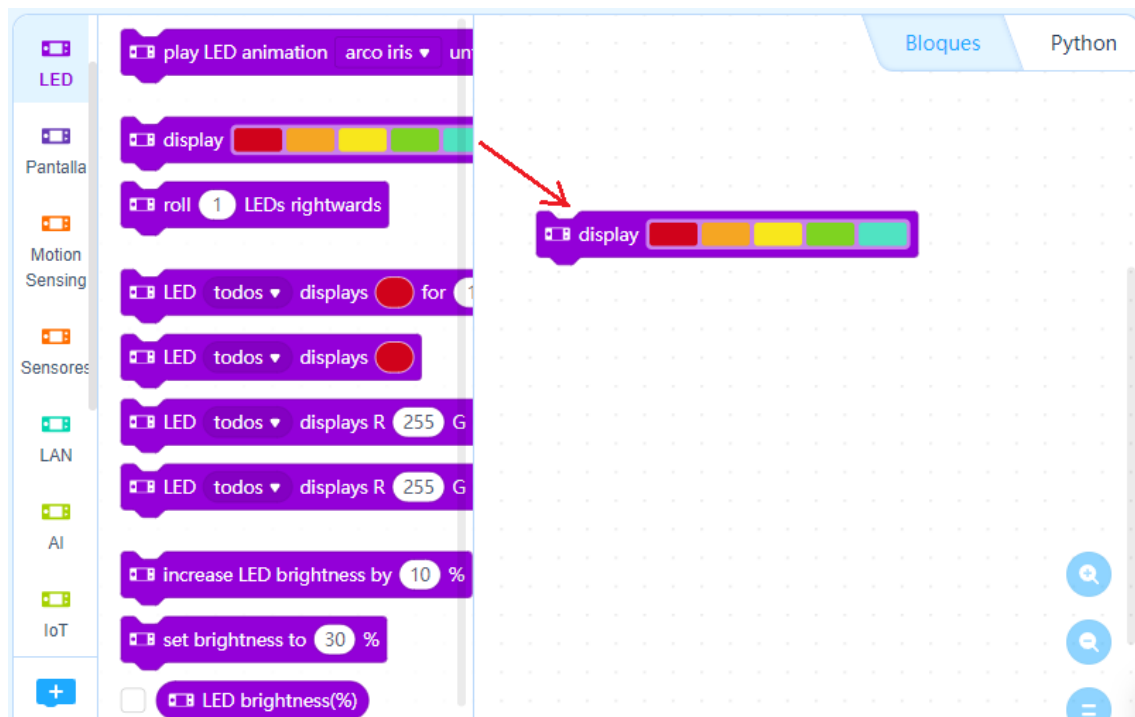
#### 4.2.5 Ejemplo: Programa "Marquesina"

Realice los siguientes pasos para compilar un programa usando bloques en modo “**En vivo**” para implementar el efecto de marquesina de los LED en **CyberPi**.

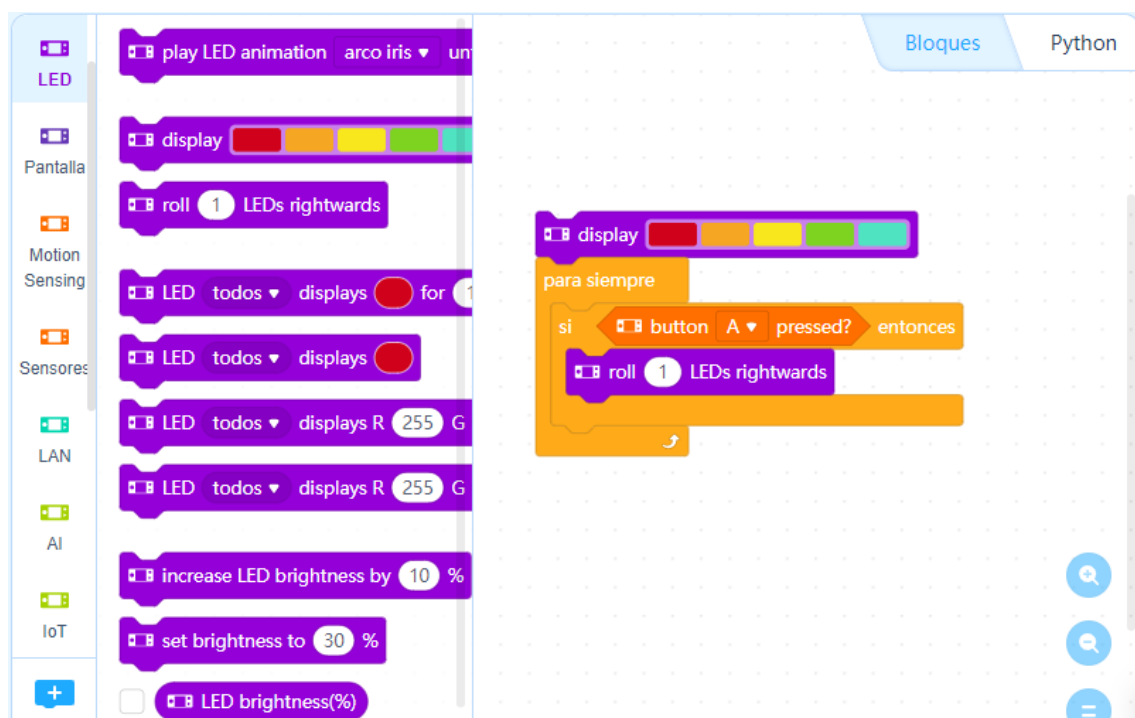
(1) Establezca el modo de programación “**En vivo**”.



(2) Configure los colores iniciales de los LED.

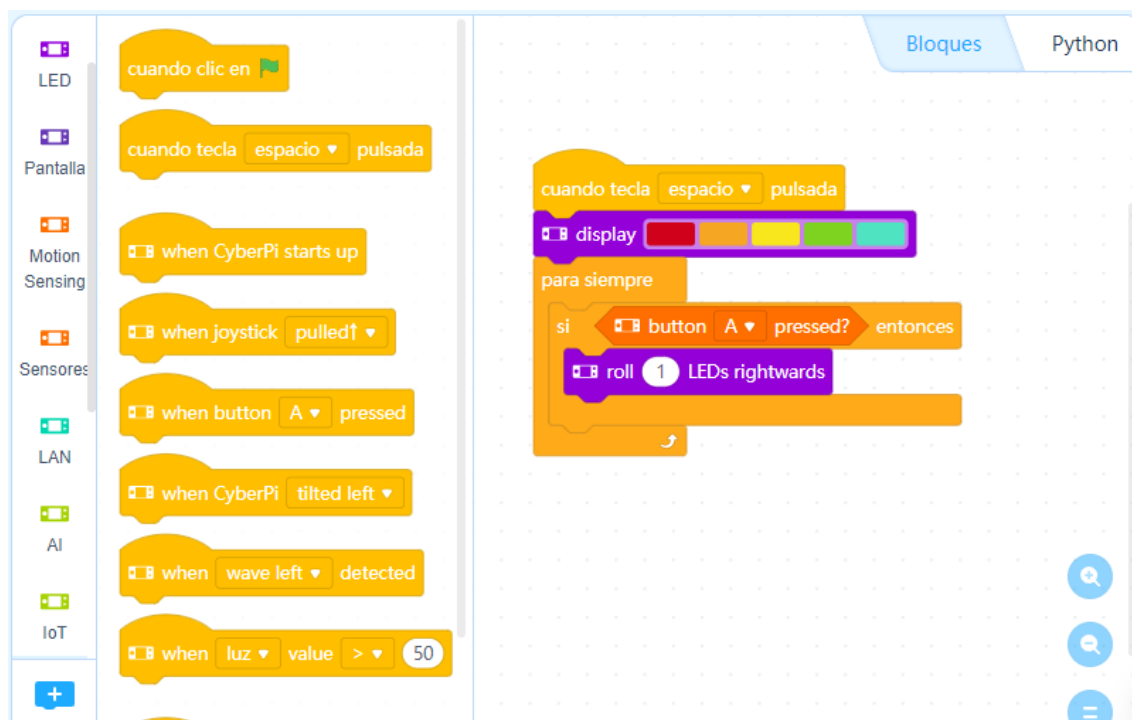


Establezca los colores iniciales por defecto: rojo, naranja, amarillo, verde y cian.



Cada vez que se presiona el **botón A** en **CyberPi**, los colores se mueven de izquierda a derecha en una posición.

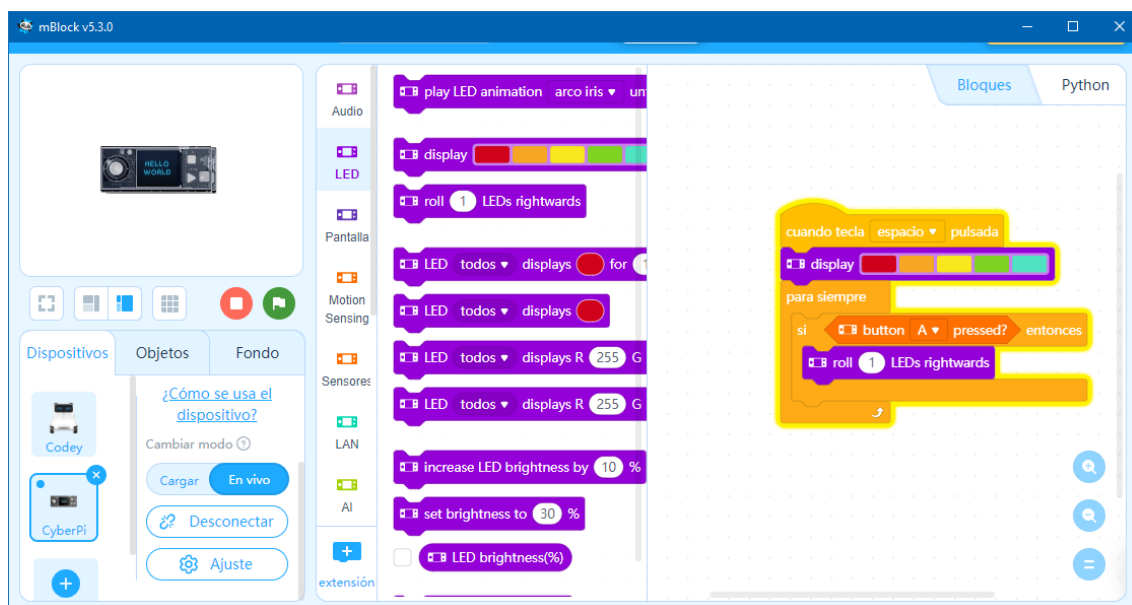
(4) Agregue un evento que active el programa.



El programa se ejecutará cuando presione la tecla espaciadora.

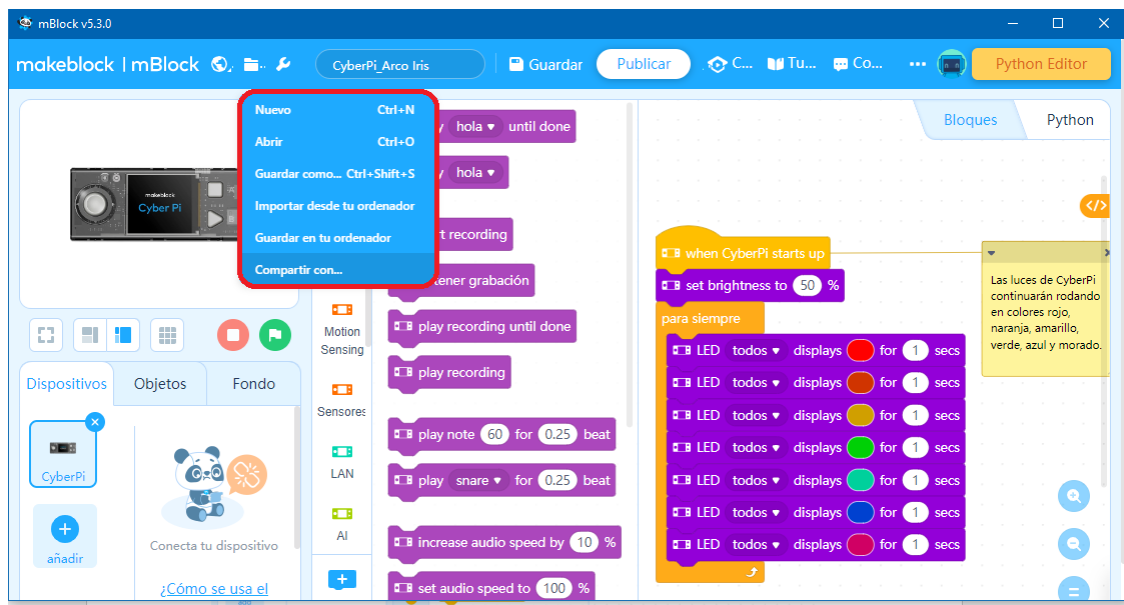
El programa Marquesina está completo.

Presione la tecla de **espacio**, y luego siga presionando el **botón A** para ver el resultado de la ejecución del programa.



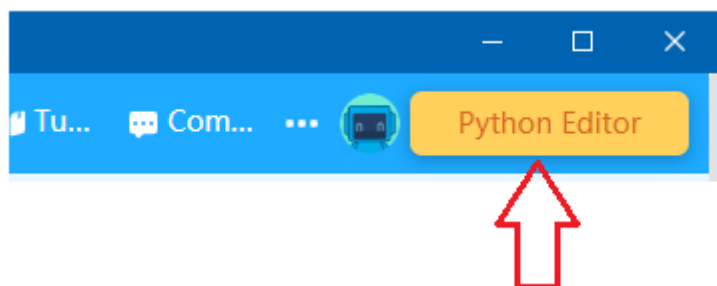


**Nota:** Para guardar proyectos, primero debe iniciar sesión en **mBlock5**. También puede guardar su proyecto en el disco local.



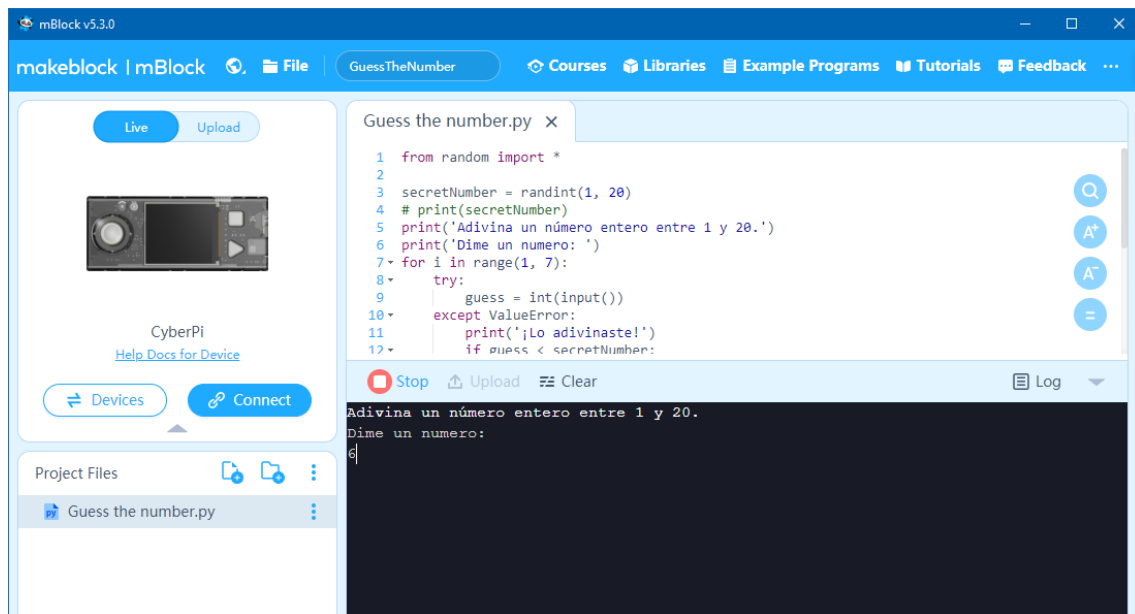
### 4.3 Programación Python

Abierto **mBlock5** se selecciona el botón de la imagen



Y aparece el editor de Python mediante el cual podremos escribir nuestro programa.

En otro lugar describiremos el manejo de este editor.



Esta es una imagen del Editor de Python

Para trabajar en modo online podemos entrar en la dirección siguiente [mBlock-Python Editor](#) en Google Chrome.

Para obtener detalles sobre cómo programar **CyberPi** en **mBlock-Python Editor**, consulte [mBlock-Python Editor Online Help](#).

## 5. Preguntas

### ¿Cómo puedo diseñar un juego con **CyberPi**?

Para garantizar la experiencia del usuario, solo se proporciona un permiso de control limitado en la pantalla de **CyberPi** para los usuarios actualmente. Más funciones están en camino.

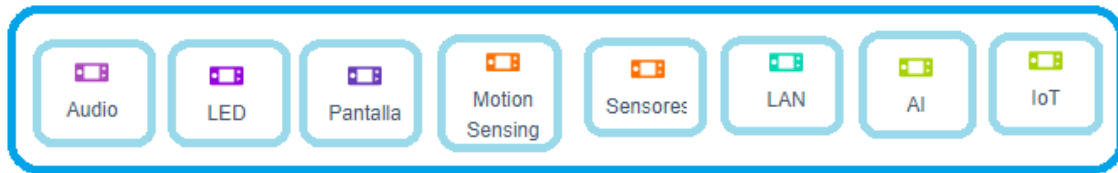
Actualmente, puede usar **mBlock-Python Editor** en combinación con algunas bibliotecas de **Python** como **pygame** para implementar el control sobre un juego de Python, usando **CyberPi** como control remoto. Puede encontrar ejemplos en el [\*\*mBlock-Python Editor Online Help\*\*](#).

Alternativamente, puede usar las extensiones de **CyberPi Lab** para usar las últimas funciones de **CyberPi**, para lo cual es posible que deba actualizar el firmware de **CyberPi**.

## 6. Más Información

[\*\*CyberPi Operation Guide\*\*](#)  
[\*\*Pocket Shield Operation Guide\*\*](#)  
[\*\*CyberPi Series User Manual\*\*](#)  
[\*\*Python API Documentation for CyberPi\*\*](#)  
[\*\*mBlock5 Online Help\*\*](#)  
[\*\*mBlock-Python Editor Online Help\*\*](#)

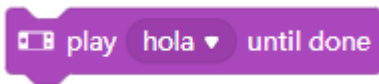
## 7. Librerías específicas para CyberPi.



### 7.1. Librería: Audio



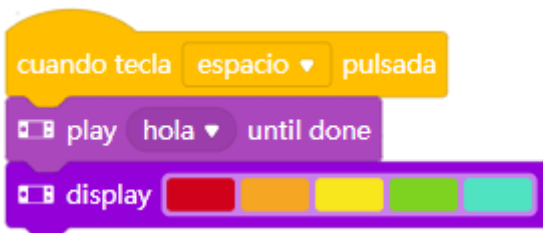
#### 7.1.1. Reproduce el sonido especificado y espera hasta que acaba.



#### Cómo usar

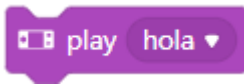
Haga clic para seleccionar un sonido del cuadro de lista desplegable.

#### Ejemplo



Cuando presiona la tecla de espacio, **CyberPi** dice "*hola*" y enciende los LED integrados después de que termina la reproducción.

### 7.1.2. Reproduce el sonido especificado



#### Cómo usar

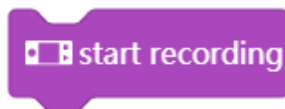
Haga clic para seleccionar un sonido del cuadro de lista desplegable.

#### Ejemplo



Cuando presiona la tecla de espacio, **CyberPi** dice "hola" y enciende los LED integrados al mismo tiempo.

### 7.1.3. Empieza a grabar sonidos



#### Cómo usar

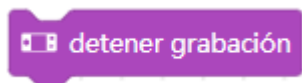
Cuando se ejecuta este bloque, **CyberPi** comienza a grabar sonidos hasta que se ejecuta el bloque de detener la grabación o la grabación supera los 10 segundos. El archivo grabado se almacena en la RAM, por lo que el archivo grabado se pierde si **CyberPi** está apagado.

#### Ejemplo

En el modo **En vivo**, presione la tecla espaciadora y luego presione el botón A en **CyberPi** para grabar sonidos; presione el botón B para detener la grabación; y mueva el joystick hacia abajo para reproducir los sonidos grabados.



#### 7.1.4. Detiene la grabación



##### Cómo usar

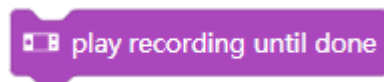
Este bloque debe usarse en combinación con el bloque de inicio de grabación. Cuando se ejecuta, **CyberPi** detiene la grabación.

## Ejemplo



En el modo **En vivo**, presione la tecla espaciadora y luego presione el botón A en **CyberPi** para grabar sonidos; presione el botón B para detener la grabación; y mueva el joystick hacia abajo para reproducir los sonidos grabados.

### 7.1.5. Reproduce los sonidos grabados hasta que acaba.



#### Cómo usar

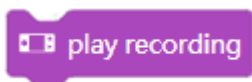
Cuando se ejecuta este bloque, los bloques siguientes se ejecutan después de que finaliza la reproducción.

## Ejemplo



En el modo **En vivo**, presione la tecla espaciadora y luego presione el botón A en **CyberPi** para grabar sonidos; presione el botón B para detener la grabación; y mueva el joystick hacia abajo para reproducir los sonidos grabados. Una vez finalizada la reproducción, se reproduce el sonido de aviso “timbre”.

### 7.1.6. Reproduce los sonidos grabados



#### Cómo usar

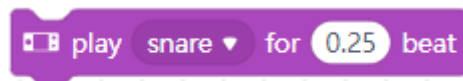
Cuando se ejecuta este bloque, **CyberPi** reproduce los sonidos grabados.

#### Ejemplo



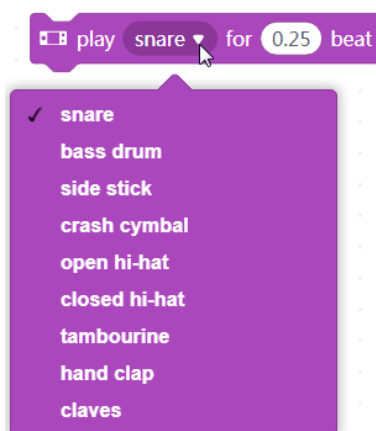
En el modo **En vivo**, presione la tecla espaciadora y luego presione el botón A en **CyberPi** para grabar sonidos; presione el botón B para detener la grabación; y mueva el joystick hacia abajo para reproducir los sonidos grabados.

### 7.1.7. Reproduce el sonido especificado para los ritmos elegidos.



#### Cómo usar

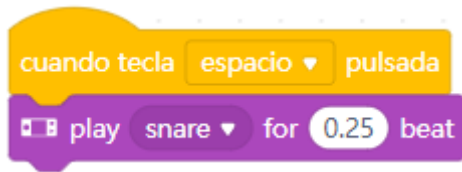
Haga clic para seleccionar un sonido.



**ritmo:** duración de la reproducción de un sonido  
Rango de ajuste: latido > 0

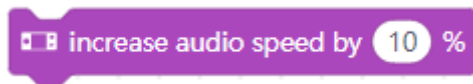
A la velocidad de reproducción general, un tiempo equivale a un segundo.

### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** reproduce el sonido de **snare**.

### 7.1.8. Cambia la velocidad de reproducción de **CyberPi**.



#### Cómo usar

Haga clic para establecer el porcentaje por el cual se cambiará la velocidad de reproducción, donde el porcentaje se refiere al porcentaje de la velocidad de reproducción normal.

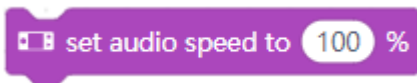
El valor que establezca debe ser un número entero. Un valor negativo indica disminución y uno positivo indica aumento.

### Ejemplo

En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** toca la nota del piano dos veces y la velocidad de reproducción por segunda vez es mayor.



### 7.1.9. Establece la velocidad de reproducción de CyberPi.



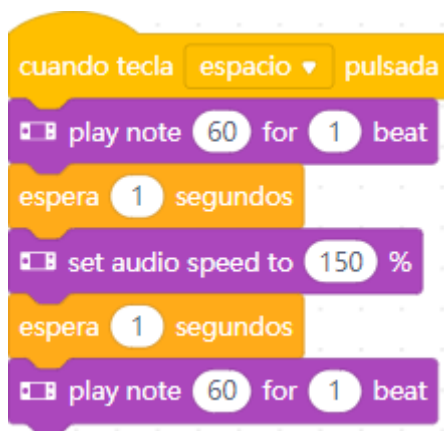
#### Cómo usar

Haga clic para establecer la velocidad de reproducción del altavoz.

El valor que establezca indica el porcentaje de la velocidad de reproducción normal.

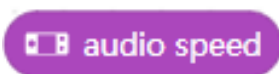
Intervalo de configuración: 25–400 (%)

#### Ejemplo



En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** toca la nota del piano dos veces y la velocidad de reproducción por segunda vez es mayor.

### 7.1.10. Devuelve la velocidad de reproducción de CyberPi



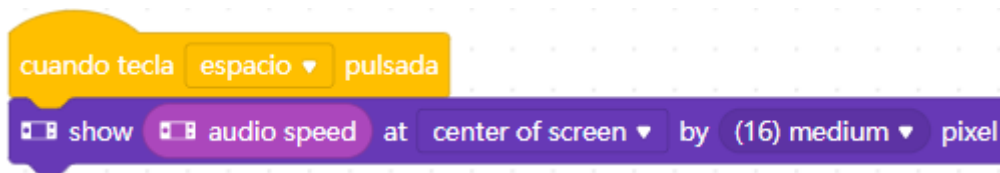
#### Cómo usar

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver la velocidad de reproducción en el escenario.

El valor 100 (%) indica la velocidad de reproducción normal.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

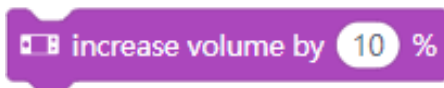
#### Ejemplo



Utilice este bloque en combinación con un bloque de **visualización**.

En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** muestra la velocidad de reproducción en el centro de su pantalla.

### 7.1.11. Cambia el volumen de reproducción de CyberPi.



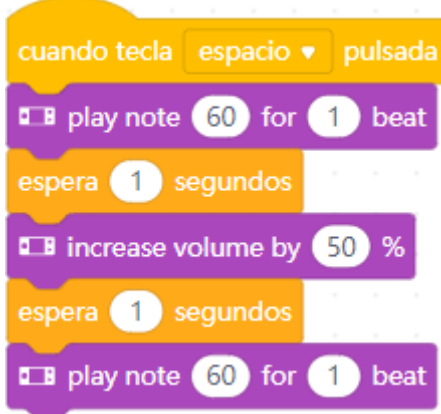
#### Cómo usar

Haga clic para establecer el porcentaje por el cual se cambiará el volumen.

El valor que establezca debe ser un número entero. Un valor negativo indica disminución y uno positivo indica aumento.

Rango de ajuste: -100– + 100 (%)

#### Ejemplo



En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** toca la nota del piano dos veces y el volumen de reproducción por segunda vez es más alto.

### 7.1.12. Establece el volumen de reproducción de CyberPi.



#### Cómo usar

Haga clic para configurar el volumen de reproducción del altavoz.  
Rango de ajuste: 0-100 (%)

### Ejemplo



En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** toca la nota del piano dos veces y el volumen de reproducción por segunda vez es más alto.

### 7.1.13. Devuelve el volumen de reproducción de CyberPi

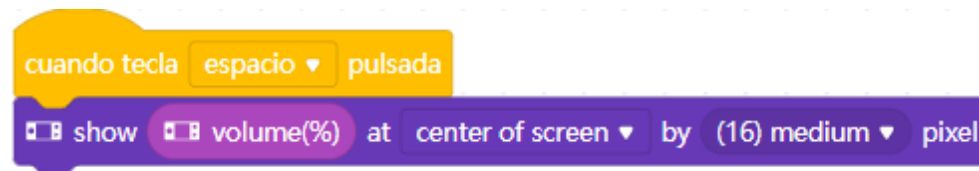


#### Cómo usar

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el volumen de reproducción en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

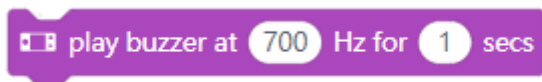
### Ejemplo



Utilice este bloque en combinación con un bloque de **visualización**.

En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** muestra el volumen de reproducción en el centro de su pantalla.

### 7.1.14. Reproduce un sonido con frecuencia y duración especificada



#### Cómo usar

Haga clic para establecer la frecuencia y la hora.

Rango de ajuste de la frecuencia: 20-5000 Hz

Para proteger sus oídos, no se emite ningún sonido si la frecuencia está fuera del rango. Es posible que le duelan los oídos si escucha sonidos de alta frecuencia durante un período prolongado.

También puede cambiar la música configurando la frecuencia, consulte las notas y sus frecuencias correspondientes, como se muestra en la siguiente figura.

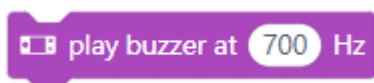
Note	Frequency	Note	Frequency	Note	Frequency
C2	65	D4	294	E6	1319
D2	73	E4	330	F6	1397
E2	82	F4	349	G6	1568
F2	87	G4	392	A6	1760
G2	98	A4	440	B6	1976
A2	110	B4	494	C7	2093
B2	123	C5	523	D7	2349
C3	131	D5	587	E7	2637
D3	147	E5	659	F7	2794
E3	165	F5	698	G7	3136
F3	175	G5	784	A7	3520
G3	196	A5	880	B7	3951
A3	220	B5	988	C8	4186
B3	247	C6	1047	D8	4699
C4	262	D6	1175		

#### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** reproduce un sonido durante un segundo y luego enciende los LED integrados.

### 7.1.15. Reproduce un sonido a la frecuencia especificada



#### Cómo usar

Haga clic para establecer la frecuencia.

Rango de ajuste de la frecuencia: 20-5000 Hz

Para proteger sus oídos, no se emite ningún sonido si la frecuencia está fuera del rango. Es posible que le duelan los oídos si escucha sonidos de alta frecuencia durante un período prolongado.

#### Ejemplo




En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** reproduce un sonido y enciende los LED integrados al mismo tiempo, y la reproducción del sonido se detiene solo después de presionar el botón A en **CyberPi**.

También puede cambiar la música configurando la frecuencia, consulte las notas y sus frecuencias correspondientes, como se muestra en la siguiente figura.

Note	Frequency	Note	Frequency	Note	Frequency
C2	65	D4	294	E6	1319
D2	73	E4	330	F6	1397
E2	82	F4	349	G6	1568
F2	87	G4	392	A6	1760
G2	98	A4	440	B6	1976
A2	110	B4	494	C7	2093
B2	123	C5	523	D7	2349
C3	131	D5	587	E7	2637
D3	147	E5	659	F7	2794
E3	165	F5	698	G7	3136
F3	175	G5	784	A7	3520
G3	196	A5	880	B7	3951
A3	220	B5	988	C8	4186
B3	247	C6	1047	D8	4699
C4	262	D6	1175		

### 7.1.16. Detiene la reproducción de todos los sonidos.

 detener todos los sonidos

#### Cómo usar

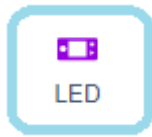
Este bloque se usa para detener todos los sonidos reproducidos por el altavoz de **CyberPi**.

#### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** sigue emitiendo un sonido; y cuando presiona la tecla de flecha ↑, **CyberPi** detiene todos los sonidos.

## 7.2. Librería: LED



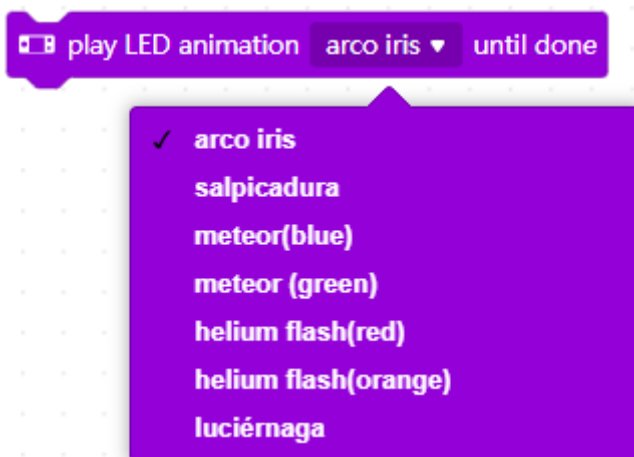
### 7.2.1. Muestra animación LED especificada hasta que finaliza.



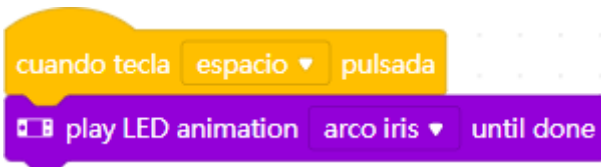
#### Cómo usar

Cuando se ejecuta este bloque, los bloques siguientes se ejecutan después de que finaliza la pantalla.

Haga clic para seleccionar una animación LED del cuadro de lista desplegable.



#### Ejemplo



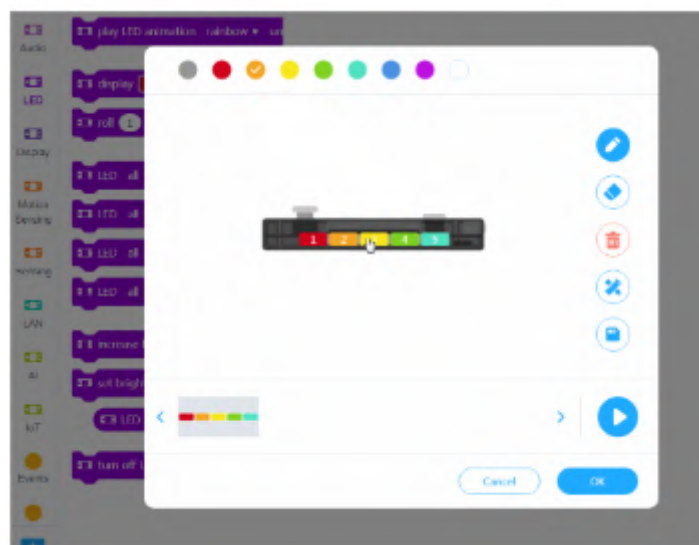
En el modo **En vivo**, cuando presiona la tecla espaciadora, los LED de **CyberPi** muestran toda la animación del arco iris.

### 7.2.2. Ilumina los LEDs de CyberPi con los colores especificados



#### Cómo usar

Haga clic para configurar los colores de los LED.  
 Seleccione un color y luego haga clic en un LED.  
 El LED se establece en el color seleccionado.



#### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, los LED de **CyberPi** se iluminan en naranja.

### 7.2.3. El color de los LEDs se desplaza de izquierda a derecha en el número especificado de posiciones

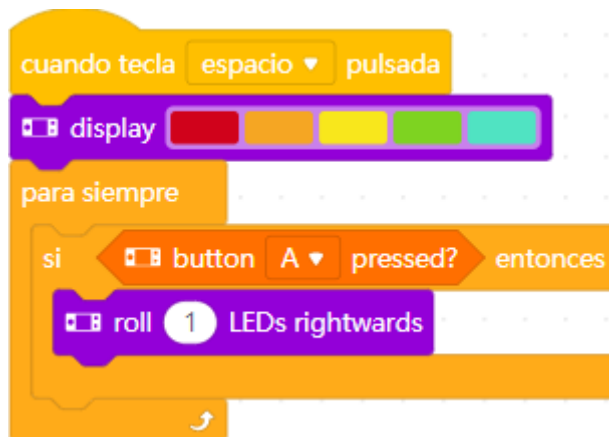


#### Cómo usar

**CyberPi** está equipado con cinco LED, y a cada uno se le asigna una posición.

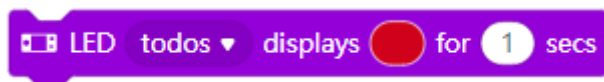
Rango de ajuste: -4 a +4

### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, los LED se iluminan con los colores que establezca; y luego, cuando presiona el botón A en **CyberPi**, los colores de los LED se mueven de izquierda a derecha en una posición.

#### 7.2.4. Ilumina los LED especificados en el color especificado durante el período especificado



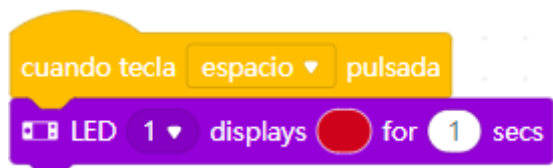
### Cómo usar

Haga clic para configurar los LED, el color y la hora.

Puede seleccionar todos los LED o un LED específico. Los números del 1 al 5 indican las posiciones de los LED, como se muestra en la siguiente figura.



### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, el primer LED de **CyberPi** se ilumina en rojo durante un segundo.

### 7.2.5. Ilumina los LEDs especificados en el color especificado



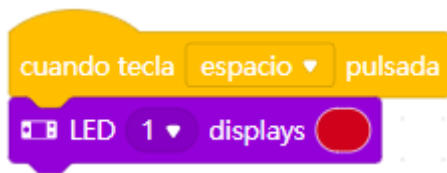
#### Cómo usar

Haga clic para configurar los LED y el color.

Puede seleccionar todos los LED o un LED específico. Los números del 1 al 5 indican las posiciones de los LED, como se muestra en la siguiente figura.



#### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, el primer LED de **CyberPi** se ilumina en rojo.

### 7.2.6. Ilumina los LED en el color especificado que es la combinación de la intensidad de rojo, verde y azul durante el período establecido.



#### Cómo usar

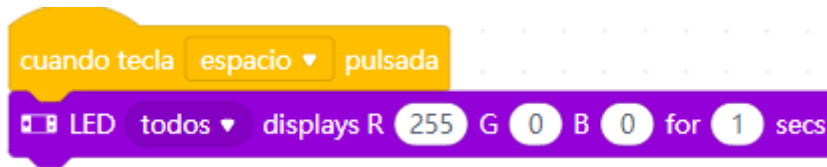
Haga clic para configurar los LED, el color y la hora.

Puede seleccionar todos los LED o un LED específico. Los números del 1 al 5 indican las posiciones de los LED, como se muestra en la siguiente figura.



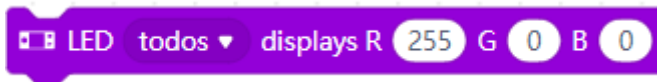
Rango de intensidad de rojo, verde y azul: 0-255

### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, todos los LED de **CyberPi** se iluminan en rojo durante un segundo.

### 7.2.7. Ilumina los LEDs en el color especificado que es la combinación de la intensidad de rojo, verde y azul



### Cómo usar

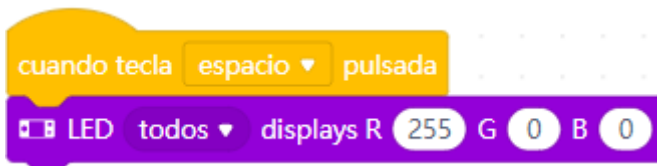
Haga clic para configurar los LED y el color.

Puede seleccionar todos los LED o un LED específico. Los números del 1 al 5 indican las posiciones de los LED, como se muestra en la siguiente figura.



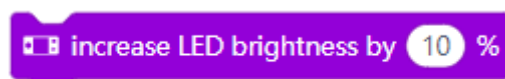
Rango de intensidad de rojo, verde y azul: 0-255

### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, todos los LED de **CyberPi** se iluminan en rojo.

### 7.2.8. Cambia el brillo de los LED de CyberPi



## Cómo usar

Haga clic para establecer el porcentaje por el cual se cambiará el brillo.

El valor que establezca debe ser un número entero. Un valor negativo indica disminución y uno positivo indica aumento.

Rango de ajuste: -100– + 100 (%)

## Ejemplo



En el modo **En vivo**, después de presionar la tecla espaciadora, los LED de **CyberPi** se iluminan en los colores especificados y el brillo aumenta en un segundo.

### 7.2.9. Establece el brillo de los LED de CyberPi

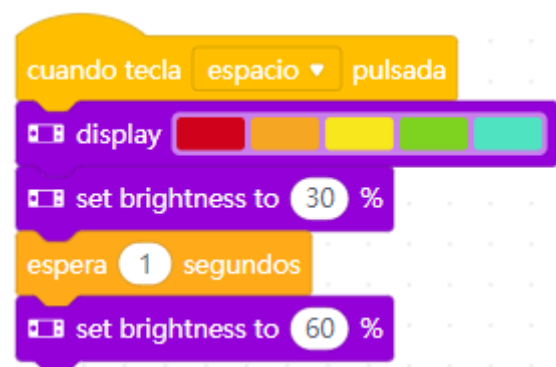


## Cómo usar

Haga clic para establecer el brillo de los LED.


Intervalo de configuración: 0-100 (%)

## Ejemplo



En el modo **En vivo**, después de presionar la tecla espaciadora, los LED de **CyberPi** se iluminan en los colores especificados y el brillo aumenta en un segundo.

### 7.2.10. Informa el brillo actual de los LED de CyberPi

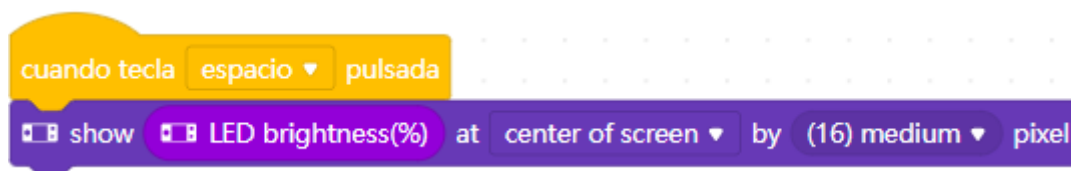
 LED brightness(%)

#### Cómo usar

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el brillo en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.


#### Ejemplo



Utilice este bloque en combinación con un bloque de **visualización**.

En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** muestra el brillo de los LED en el centro de su pantalla.

### 7.2.11. Apaga los LED especificados en CyberPi

 turn off LED todos ▼

#### Cómo usar

Este bloque se utiliza para apagar los LED especificados en **CyberPi**.

Puede seleccionar todos los LED o un LED específico. Los números del 1 al 5 indican las posiciones de los LED, como se muestra en la siguiente figura.



#### Ejemplo

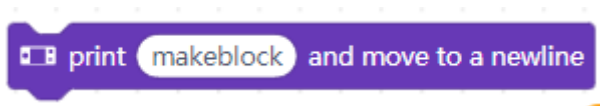


En el modo **En vivo**, cuando presiona la tecla espaciadora, los LED de **CyberPi** se iluminan en los colores especificados; y cuando presiona la tecla de flecha ↑, todos los LED se apagan.

## 7.3. Librería: Pantalla



### 7.3.1. Muestra textos en la pantalla de **CyberPi** con saltos de línea forzados

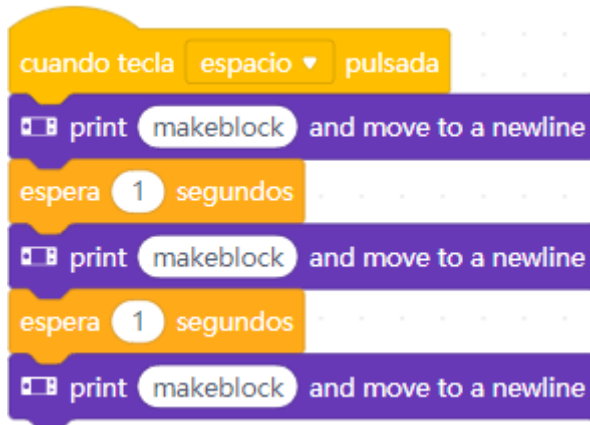


#### Cómo usar

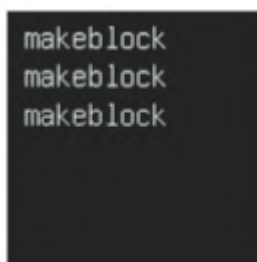
Haga clic para ingresar el texto que se mostrará.

Cuando se ejecuta este bloque, se inserta un salto de línea forzado después de que se muestra el texto actual, y el texto siguiente se muestra en una nueva línea.

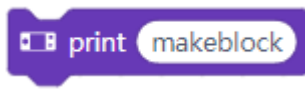
#### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, los textos se muestran en la pantalla de **CyberPi** con saltos de línea forzados, como se muestra en la siguiente figura.



### 7.3.2. Muestra textos en la pantalla de **CyberPi** con saltos de línea automáticos



#### Cómo usar

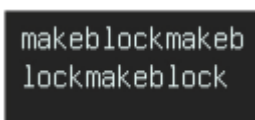
Haga clic para ingresar el texto que se mostrará.

Cuando se ejecuta este bloque, los textos se muestran en la misma línea, y cuando una línea está llena, los textos continúan en nuevas líneas.

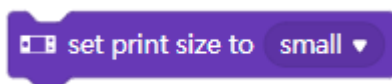
#### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, los textos se muestran en la pantalla de **CyberPi** con saltos de línea automáticos, como se muestra a continuación.



### 7.3.3. Muestra el texto especificado en el tamaño especificado en la pantalla de **CyberPi**



#### Cómo usar

Haga clic para seleccionar un tamaño de fuente en el cuadro de lista desplegable.

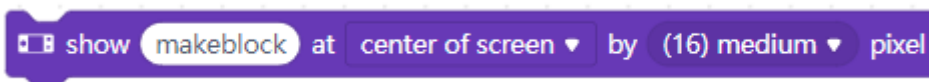
#### Ejemplo



Después de presionar la tecla de espacio, **CyberPi** muestra "makeblock", y un segundo después, **CyberPi** muestra "makeblock" en dos líneas y el tamaño de todos los textos se agranda.

**Nota:** **CyberPi** no puede mostrar textos en dos tamaños diferentes a la vez.

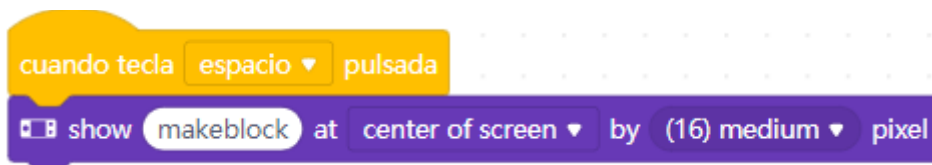
#### 7.3.4. Muestra el texto en la posición especificada de la pantalla de **CyberPi** con el tamaño de fuente especificado



##### Cómo usar

Haga clic para ingresar el texto que se mostrará y establecer la posición y el tamaño de fuente.

##### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** muestra "makeblock" en el centro de la pantalla.

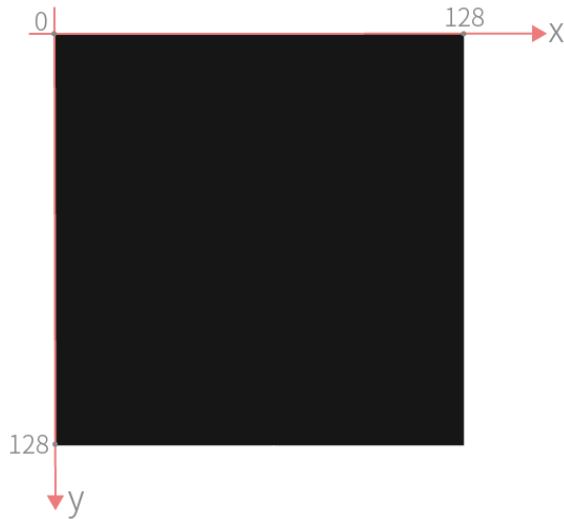
#### 7.3.5. Muestra el texto en el tamaño de fuente especificado en la pantalla de **CyberPi** con las coordenadas especificadas como punto de inicio.



## Cómo usar

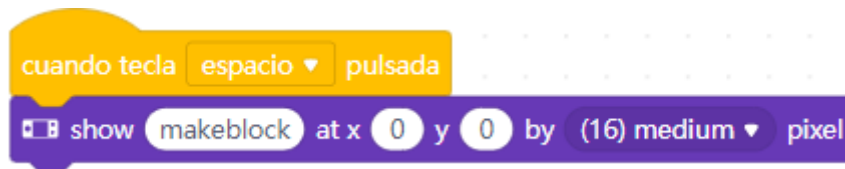
Haga clic para ingresar el texto que se mostrará y establecer las coordenadas y el tamaño de fuente.

La siguiente figura muestra las coordenadas de la pantalla de **CyberPi**.



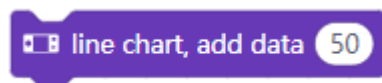
Rango de ajuste de xey: 0-128

## Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** muestra "makeblock" en la esquina superior izquierda de la pantalla.

### 7.3.6. Agrega un dato y muestra un gráfico de líneas en la pantalla de **CyberPi**



## Cómo usar

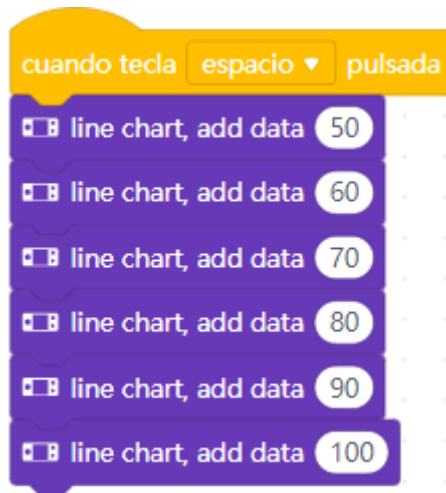
Rango de configuración de datos: 0-100

Si un valor a ingresar excede el rango de configuración, cambie la escala de todos los datos originales para que estén dentro del rango de configuración.

Para Ejemplo, puede cambiar la escala de los valores 200, 300 y 400 a 20, 30 y 40, respectivamente.

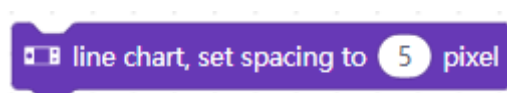
Este bloque se usa para agregar un dato a la vez. Puede usar este bloque varias veces para agregar varios datos.

### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, se muestra un gráfico de líneas en la pantalla de **CyberPi**.

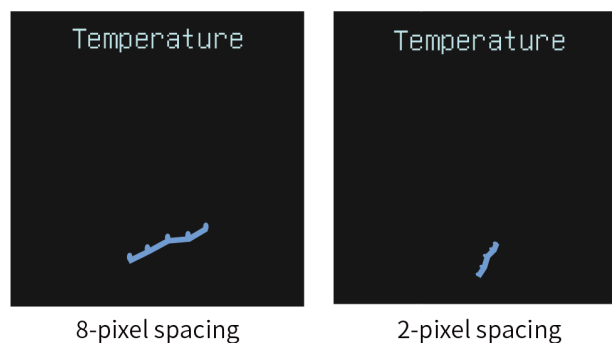
### 7.3.7. Establece el espacio entre los puntos de datos en un gráfico de líneas



#### Cómo usar

Rango de ajuste de espaciado: 0-128

Para Ejemplo, las siguientes figuras muestran los gráficos de líneas con los mismos datos pero diferentes espacios entre puntos de datos.



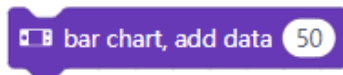
## Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, se muestra un gráfico de líneas en la pantalla de **CyberPi**.

Puede cambiar la forma del gráfico de líneas cambiando el espaciado.

### 7.3.8. Agrega un dato y muestra un gráfico de barras en la pantalla de **CyberPi**



#### Cómo usar

Rango de configuración de datos: 0-100

Si un valor a ingresar excede el rango de configuración, cambie la escala de todos los datos originales para que estén dentro del rango de configuración.

Para Ejemplo, puede cambiar la escala de los valores 200, 300 y 400 a 20, 30 y 40, respectivamente.

Este bloque se usa para agregar un dato a la vez. Puede usar este bloque varias veces para agregar varios datos.

**Nota:** Para agregar varios datos en un gráfico de barras, debe configurar varios colores de pincel para dibujar varias barras de datos.

## Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, se muestra un gráfico de barras con dos barras de datos en la pantalla de **CyberPi**.



### 7.3.9. Agrega un dato y muestra una tabla en la pantalla de CyberPi



#### Cómo usar

Se admite una tabla máxima de 4 (filas) × 3 (columnas).

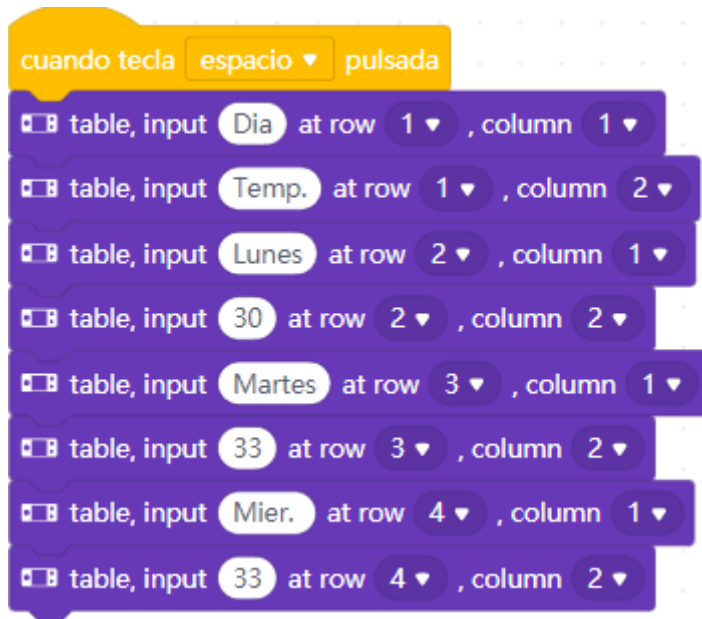
Haga clic para establecer la fila, la columna y el contenido que se introducirá.

Defina una celda estableciendo la fila y la columna.

Cuando el contenido que se muestra en una celda es demasiado largo, sigue desplazándose hacia la derecha en la celda.

Puede introducir textos y los iconos predefinidos en la tabla.

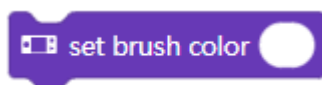
## Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, se muestra una tabla de  $4 \times 2$  en la pantalla de **CyberPi**.

Dia	Temp.
Lunes	30
Martes	33
Mier.	33

### 7.3.10. Establece el color del pincel

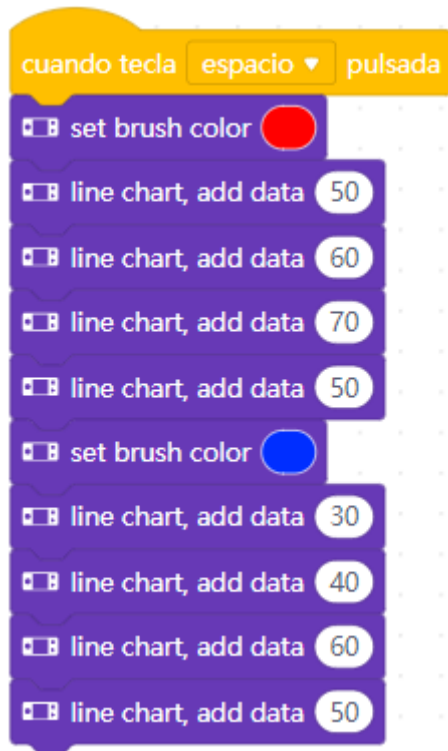


#### Cómo usar

El pincel aquí se usa para dibujar gráficos de líneas, gráficos de barras y tablas. Al establecer el color del pincel, puede dibujar un gráfico de líneas con diferentes colores de líneas, un gráfico de barras con diferentes colores de columnas y una tabla con el color de fuente especificado.

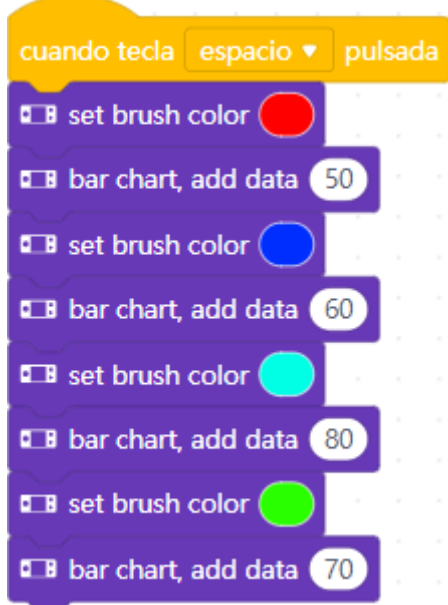
Haga clic para establecer el color del pincel.

## Ejemplo 1

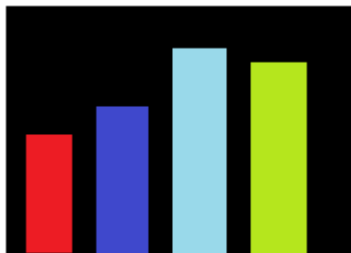


En el modo **En vivo**, cuando presiona la tecla espaciadora, se muestra un gráfico de líneas con dos colores diferentes de líneas en la pantalla de **CyberPi**.

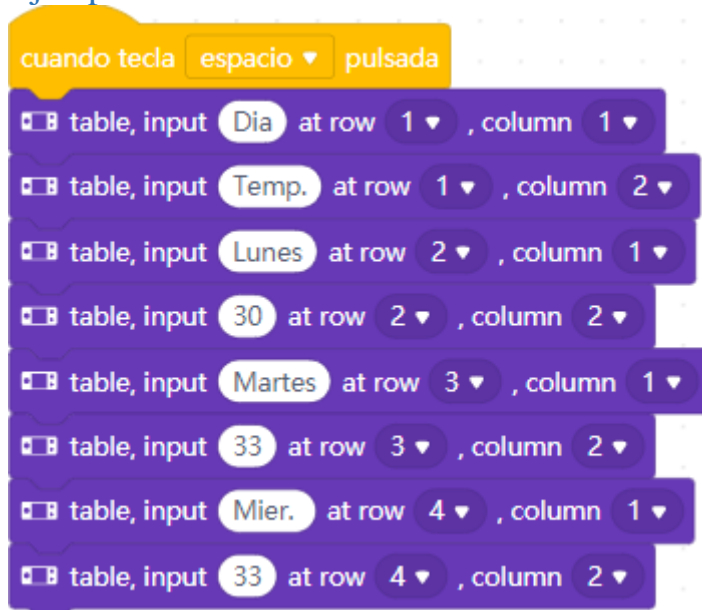
## Ejemplo 2



En el modo **En vivo**, cuando presiona la tecla espaciadora, se muestra un gráfico de barras con cuatro colores diferentes de barras en la pantalla de **CyberPi**.

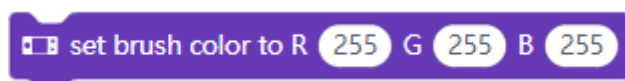


### Ejemplo 3



En el modo **En vivo**, cuando presiona la tecla espaciadora, se muestra una tabla con el color de fuente rojo en la pantalla de **CyberPi**.

#### 7.3.11. Establece el pincel en el color que es la combinación especificada de rojo, verde y azul.



#### Cómo usar

El pincel aquí se usa para dibujar gráficos de líneas, gráficos de barras y tablas. Al establecer el color del pincel, puede dibujar un gráfico de líneas con diferentes colores de líneas, un gráfico de barras con diferentes colores de columnas y una tabla con el color de fuente especificado.

Haga clic para establecer la intensidad de rojo, verde y azul.

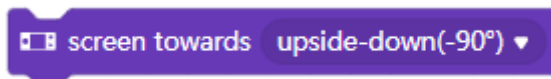
Rango de ajuste de intensidad: 0-255

### Ejemplo



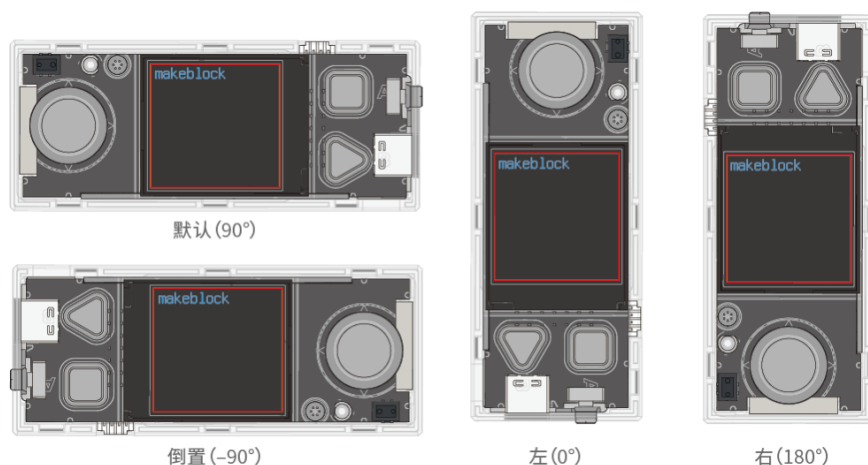
En el modo **En vivo**, cuando presiona la tecla espaciadora, se muestra un gráfico de barras con cuatro colores diferentes de barras en la pantalla de **CyberPi**.

### 7.3.12. Establece la pantalla de **CyberPi** en la dirección de visualización especificada

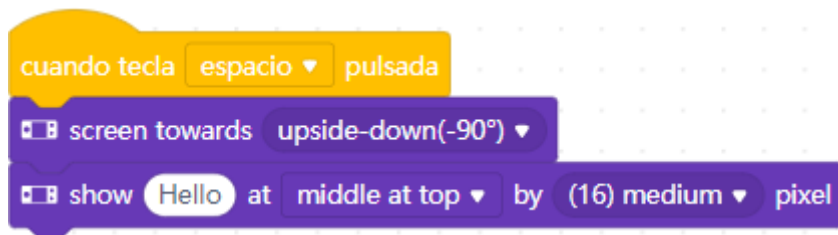


#### Cómo usar

Con este bloque, puede rotar la dirección de visualización de la pantalla de **CyberPi**. Se proporcionan cuatro opciones de dirección, como se muestra a continuación:



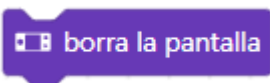
## Ejemplo



En el modo **En vivo**, cuando presiona la tecla de espacio, **CyberPi** muestra "Hola" en su pantalla, como se muestra en la siguiente figura.



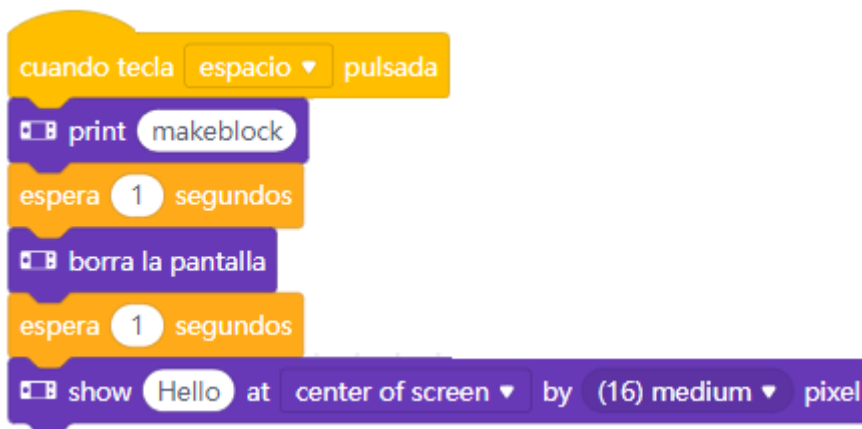
### 7.3.13. Borra la pantalla de CyberPi



## Cómo usar

Para mostrar información en la pantalla de **CyberPi**, primero debe usar este bloque para borrar la pantalla. De lo contrario, la información puede superponerse a la información mostrada anteriormente.

## Ejemplo



En el modo **En vivo**, cuando presiona la tecla de espacio, **CyberPi** muestra "makeblock" en su pantalla y luego borra la pantalla antes de mostrar "Hola".

## 7.4. Librería: Sensores de Movimiento



### 7.4.1. Hace que el sprite especificado siga a **CyberPi** con la sensibilidad especificada



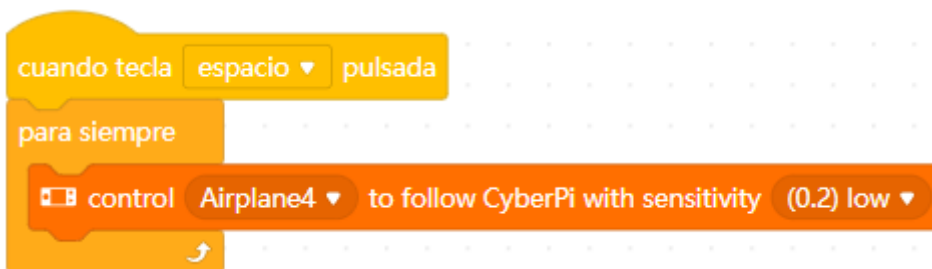
#### Como usar

Este bloque reconoce todos los sprites en el escenario. Haga clic para seleccionar un objeto del cuadro de lista desplegable. Para saber cómo agregar un objeto, consulte [Add a Sprite](#).

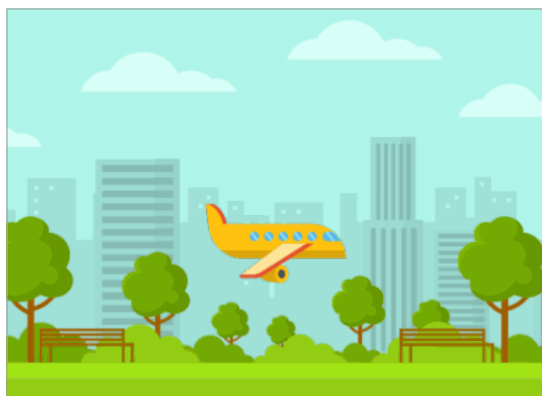
Haga clic para seleccionar una sensibilidad del cuadro de lista desplegable.

Una mayor sensibilidad indica que el objeto responde más rápidamente al movimiento de **CyberPi**.

#### Ejemplo



En el modo **En vivo**, cuando presiona la tecla espaciadora, el avión en el escenario se mueve a medida que mueve **CyberPi**.



### 7.4.2. Determina si **CyberPi** se mueve en la posición indicada.

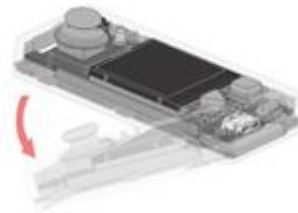


#### Cómo utilizar

Haga clic para seleccionar un estado de movimiento en el cuadro de lista desplegable. La siguiente figura muestra los estados de movimiento definidos para **CyberPi**.



Mover arriba



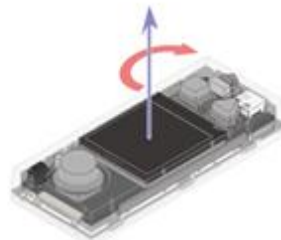
Mover abajo



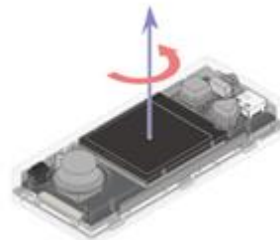
Mover Izquierda



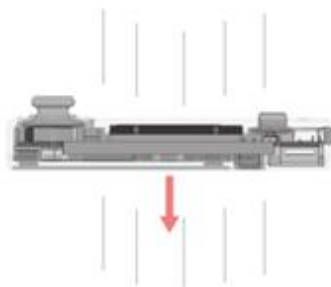
Mover Derecha



Rotar Derecha



Rotar Izquierda



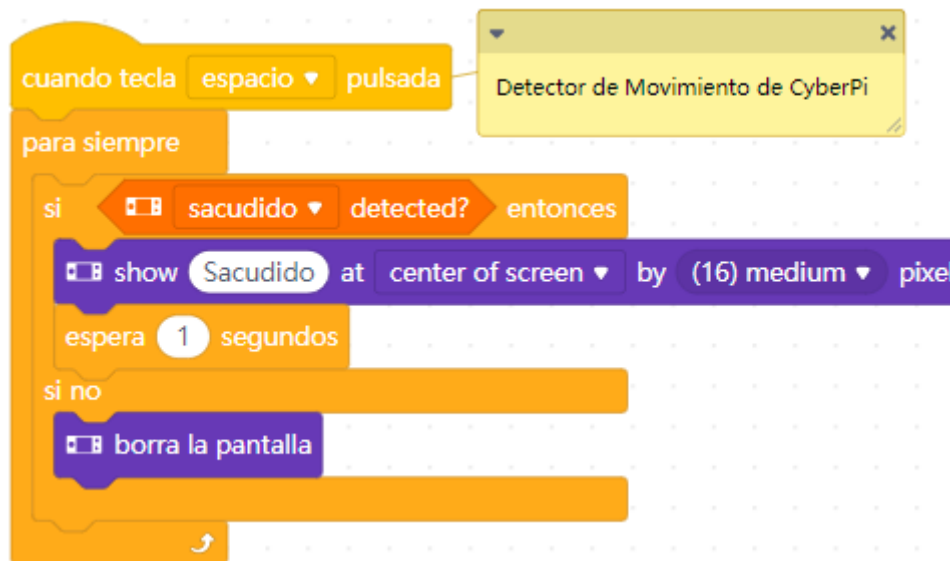
Cayendo



Agitado

Este es un bloque booleano que contiene una condición. Úselo en combinación con otro bloque que requiera una condición.

### Ejemplo



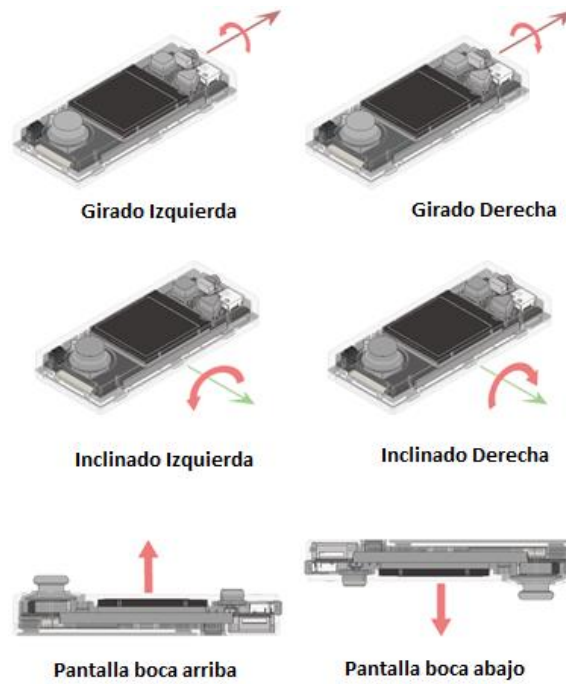
En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra Arriba en su pantalla cuando lo mueve hacia arriba y borra la pantalla cuando deja de hacerlo.

### 7.4.3. Determina si **CyberPi** se coloca en la postura especificada



#### Cómo utilizar

Haga clic para seleccionar una postura del cuadro de lista desplegable. La siguiente figura muestra las posturas definidas para **CyberPi**.



Este es un bloque booleano que contiene una condición. Úselo en combinación con otro bloque que requiera una condición.

### Ejemplo



En el modo **En vivo**, después de presionar la tecla espaciadora, los LED de **CyberPi** se encienden cuando inclina **CyberPi** hacia adelante y se apagan cuando lo coloca en otra postura.

### 7.4.4. Informa con qué fuerza se agita CyberPi

 fuerza de sacudida

#### Cómo utilizar

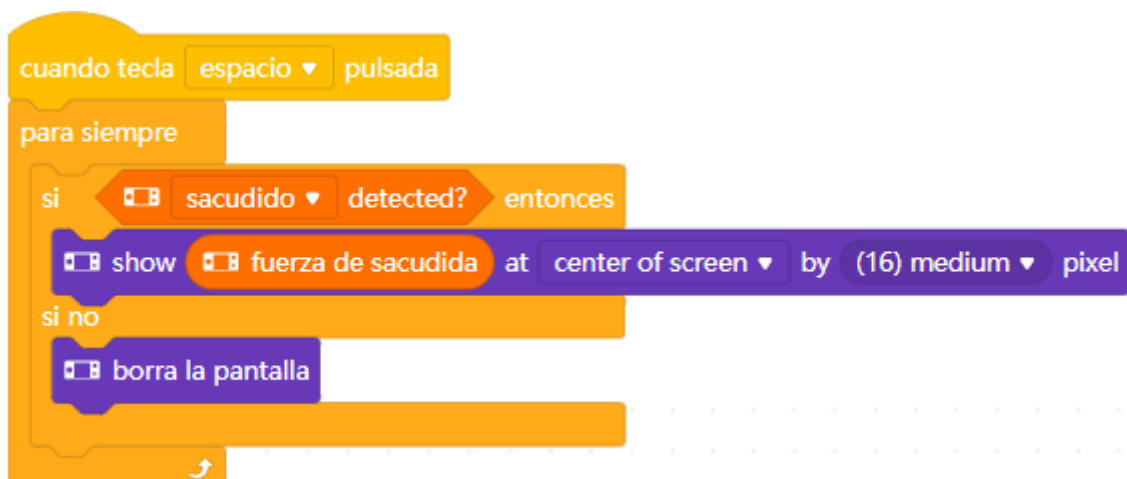
La fuerza de la agitación está relacionada positivamente con la frecuencia y amplitud de la agitación.

Intervalo de valores: 0–100

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver la fuerza del temblor en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

#### Ejemplo



Utilice este bloque en combinación con el bloque “show” de la librería “Pantalla”.

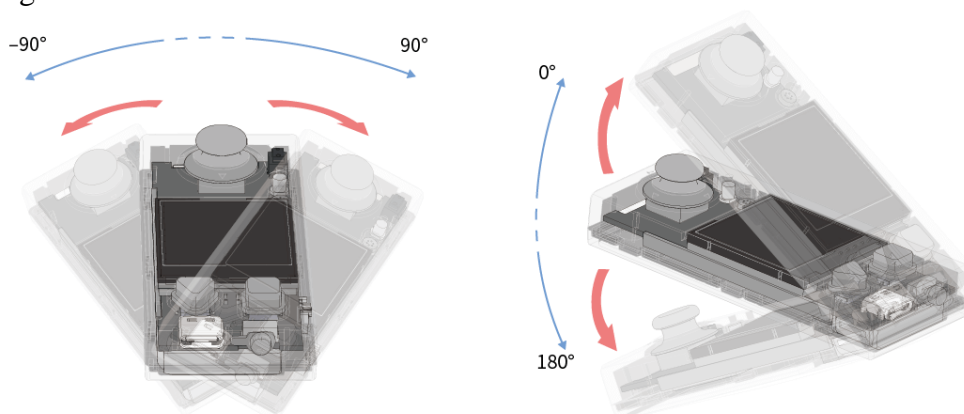
En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra la fuerza del temblor en el centro de su pantalla cuando la agita y limpia la pantalla cuando deja de agitarla.

### 7.4.5. Informa la dirección en la que se mueve CyberPi

 waving direction (°)

#### Cómo utilizar

La dirección de ondulación se define por el ángulo, como se muestra en la siguiente figura.

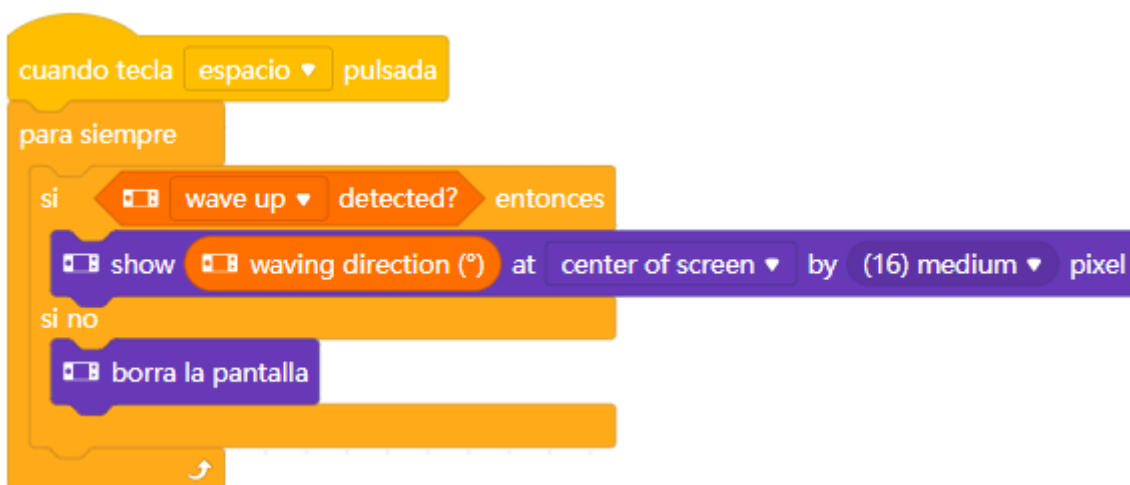


Rango de valores: -179 a +180, en grados (°)

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver la dirección del movimiento en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

#### Ejemplo



Utilice este bloque en combinación con el bloque “show” de la librería “Pantalla”.

En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra la dirección del movimiento en el centro de la pantalla cuando lo mueve hacia arriba y borra la pantalla cuando deja de hacerlo.

### 7.4.6. Informa la velocidad a la que se mueve **CyberPi**

 waving speed

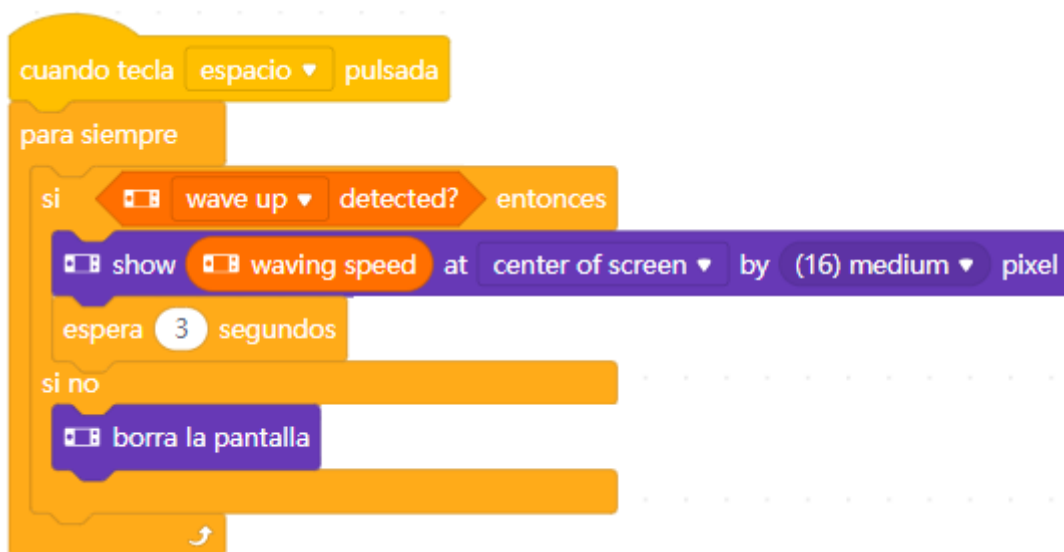
#### Cómo utilizar

Intervalo de valores: 0–100

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver la velocidad de ondulación en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

#### Ejemplo



Utilice este bloque en combinación con el bloque “**show**” de la librería “**Pantalla**”.

En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra la velocidad de agitación en el centro de su pantalla cuando la agita y borra la pantalla cuando deja de agitarla.

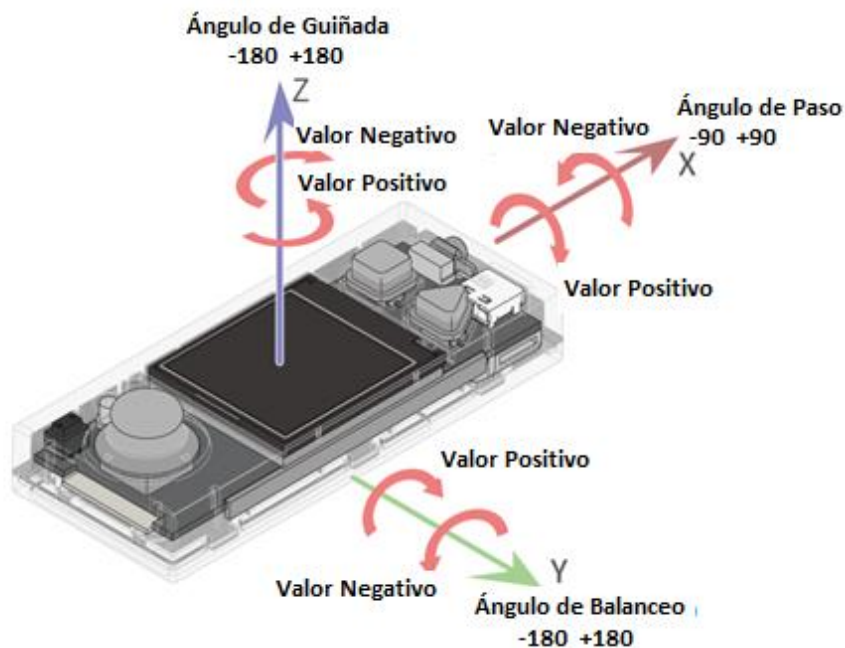
### 7.4.7. Informa el ángulo de la postura especificada

 tilted forward ▼ angle (°)

#### Cómo utilizar

Haga clic para seleccionar una postura del cuadro de lista desplegable.

Los ángulos de cabeceo, balanceo y guiñada se definen como sigue.



Rangos de valores:

Ángulo de paso: de  $-90$  a  $+90$ , en grados

Ángulo de balanceo:  $-180$  a  $+180$ , en grados

Ángulo de guiñada:  $-180$  a  $+180$ , en grados

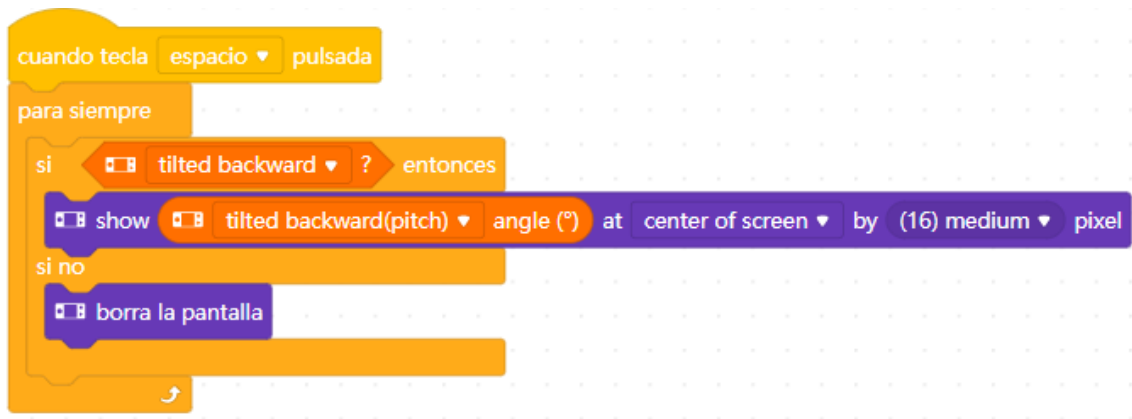
Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el ángulo de la postura en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

#### Ejemplo

Utilice este bloque en combinación con el bloque “show” de la librería “Pantalla”.

En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra el ángulo de postura en el centro de su pantalla cuando la inclina hacia atrás y limpia la pantalla cuando deja de inclinarla.



#### 7.4.8. Informa la aceleración de CyberPi en el eje especificado



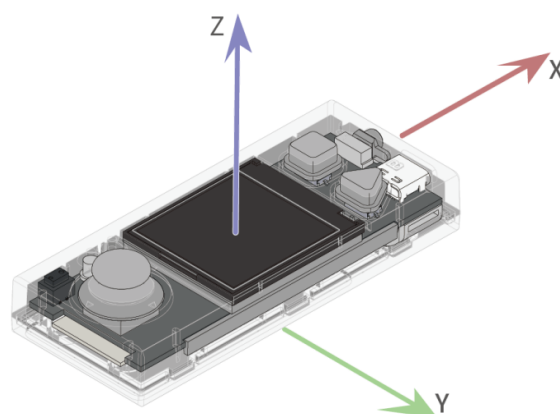
#### Cómo utilizar

Haga clic para seleccionar un eje del cuadro de lista desplegable.

**CyberPi** está equipado con un sensor de movimiento que puede detectar la aceleración de **CyberPi** en los ejes x, y y z.

Tenga en cuenta que el acelerómetro mide la gravedad de la Tierra. Cuando **CyberPi** se coloca sobre un escritorio, la salida del acelerómetro en el eje z es  $-9,8 \text{ m/s}^2$  ( $1g = 9,8 \text{ m/s}^2$ ).

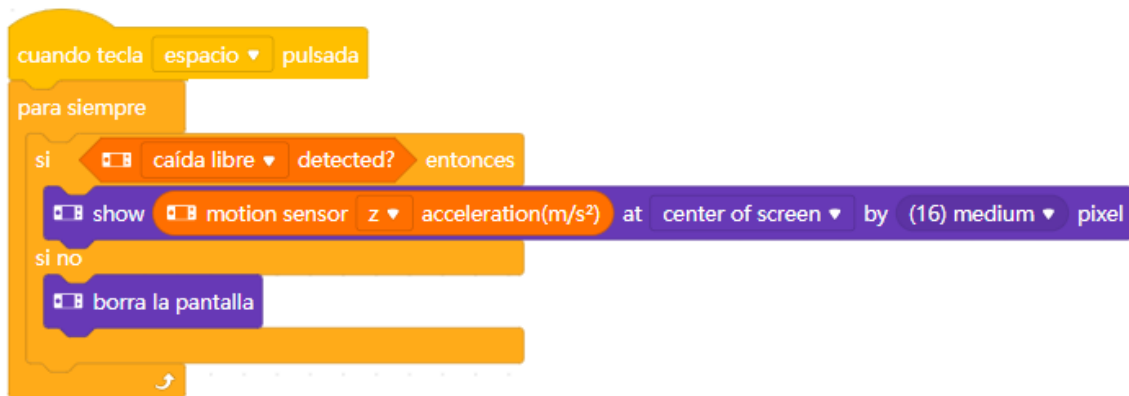
Los ejes se definen de la siguiente manera.



Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver la aceleración en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

## Ejemplo



Utilice este bloque en combinación con el bloque “show” de la librería “Pantalla”.

En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra la aceleración en el z-aixs en el centro de su pantalla cuando cae y borra la pantalla cuando lo mantiene presionado.

**Nota:** Cuando **CyberPi** se caiga, déjelo caer sobre una superficie blanda o cójalo con la mano para evitar dañar **CyberPi**.

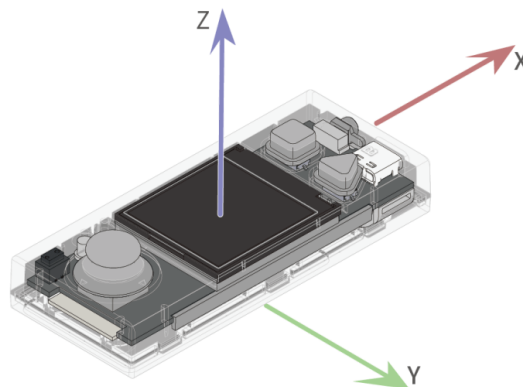
### 7.4.9. Informa la velocidad angular a la que CyberPi gira alrededor del eje especificado



#### Cómo utilizar

Haga clic para seleccionar un eje del cuadro de lista desplegable.

Los ejes se definen como sigue.



Rango de valores: -500 a +500, en grados / segundo (°)

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver la velocidad angular en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

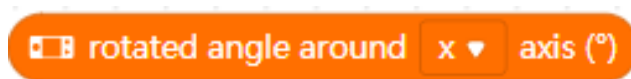
### Ejemplo



Utilice este bloque en combinación con un bloque de **visualización**.

En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra la velocidad angular de **CyberPi** alrededor del eje x en el centro de su pantalla cuando lo gira alrededor del eje x.

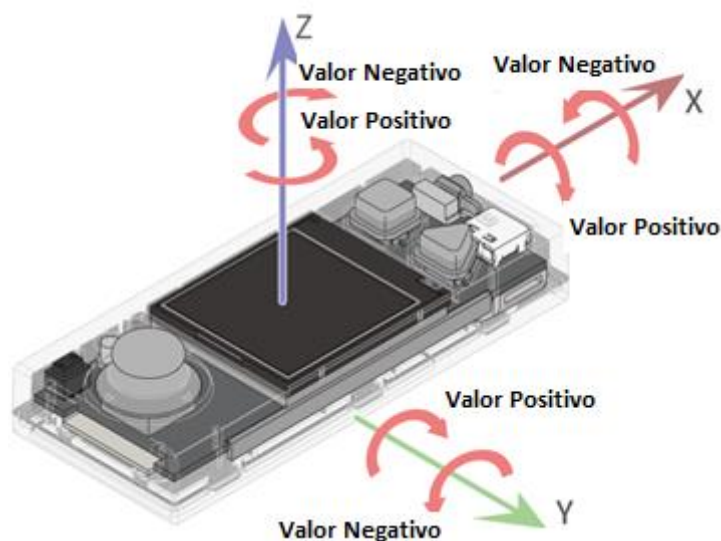
### 7.4.10. Informa que el ángulo **CyberPi** se gira alrededor del eje especificado



### Cómo utilizar

Haga clic para seleccionar un eje del cuadro de lista desplegable.

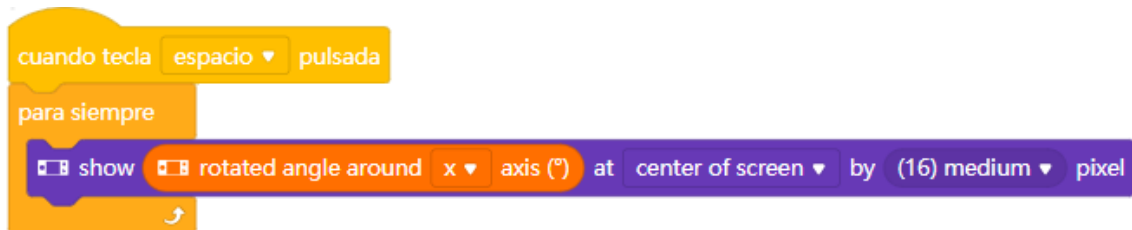
Los ejes y ángulos se definen como sigue.



Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el ángulo en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

### Ejemplo



Utilice este bloque en combinación con un bloque “show” de la librería “Pantalla”

En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra el ángulo en el centro de su pantalla cuando lo gira alrededor del eje x.

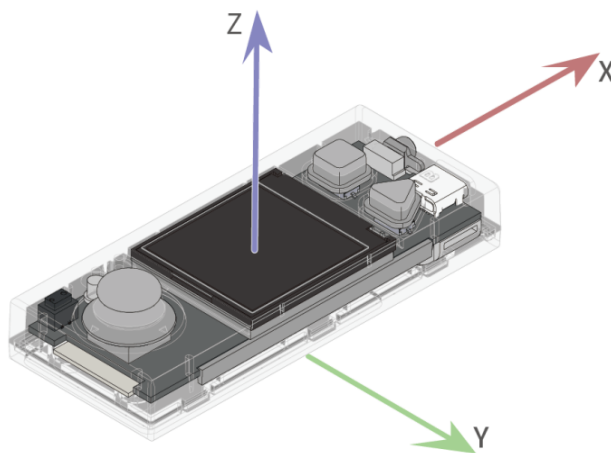
### 7.4.11. Restablece el ángulo que CyberPi gira alrededor del eje especificado (ejes) a cero



### Cómo utilizar

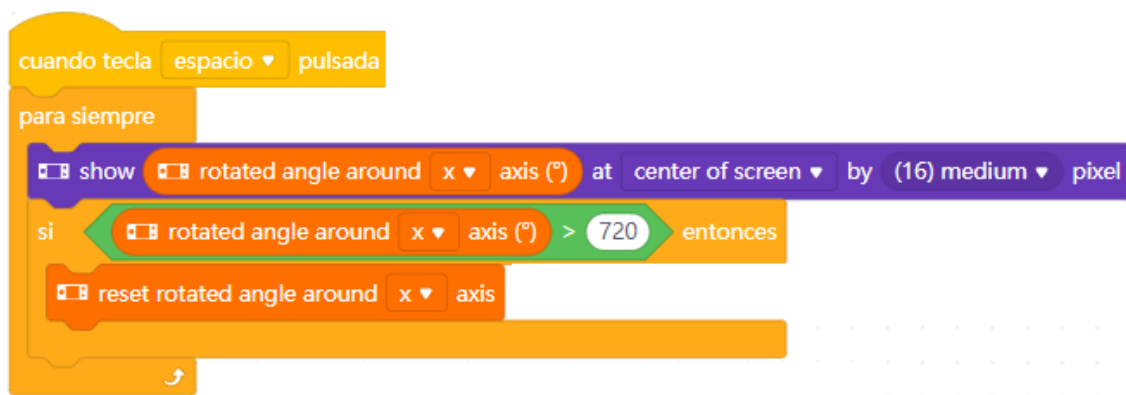
Haga clic para seleccionar todos los ejes o un eje del cuadro de lista desplegable.

Los ejes se definen como sigue.



Después de ejecutar este bloque, el ángulo **CyberPi** que gira alrededor del eje especificado se cuenta desde cero.

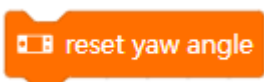
### Ejemplo



Utilice este bloque en combinación con un bloque “**show**” de la librería “**Pantalla**”.

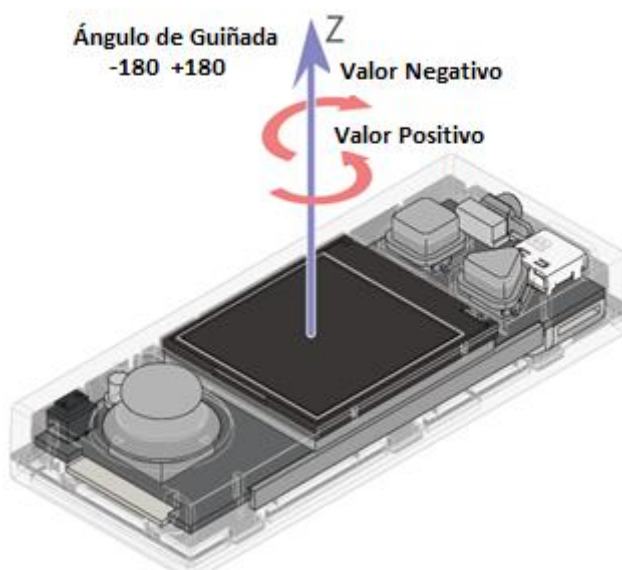
En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra el ángulo en el centro de su pantalla cuando lo gira alrededor del eje x. Cuando el ángulo es superior a 720, se pone a cero y se vuelve a contar.

### 7.4.12. Restablece el ángulo de guiñada de **CyberPi** a cero



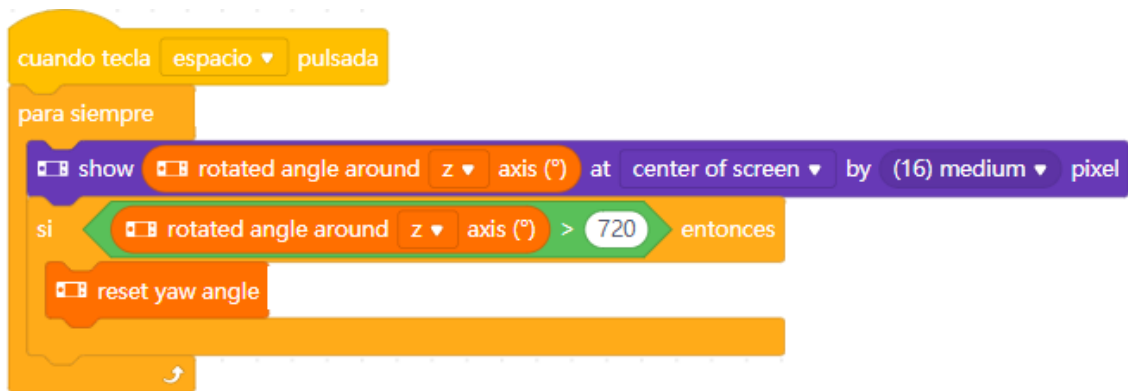
#### Cómo utilizar

El ángulo de guiñada se define como sigue.



Una vez ejecutado este bloque, el ángulo de guiñada de **CyberPi** se cuenta desde cero.

## Ejemplo



Utilice este bloque en combinación con un bloque de **visualización**.

En el modo **En vivo**, después de presionar la tecla espaciadora, **CyberPi** muestra el ángulo de guiñada en el centro de su pantalla cuando lo gira alrededor del eje z. Cuando el ángulo es superior a 179, se pone a cero y se vuelve a contar.

## 7.5. Librería: Sensores



Son las funciones de interacción que se ejercen mediante los botones y el Joystick de la unidad **CyberPi**, además de los valores obtenidos de los sensores de luz, sonido, estado batería, etc.

### 7.5.1. Determina si el joystick se mueve en la dirección especificada



#### Cómo utilizar

**CyberPi** está equipado con un joystick que se puede mover en cinco direcciones.



Haga clic para seleccionar una dirección del cuadro de lista desplegable.

Este es un bloque booleano que contiene una condición. Úselo en combinación con otro bloque que requiera una condición.

#### Ejemplo



En el modo **En vivo**, después de presionar la tecla de espacio, puede mover el joystick para mostrar "makeblock" en diferentes posiciones.

### 7.5.2. Informa el número de veces que se mueve el joystick en la dirección especificada

☒ joystick   middle pressed ▼   counts

#### Cómo utilizar

**CyberPi** está equipado con un joystick que se puede mover en cinco direcciones.

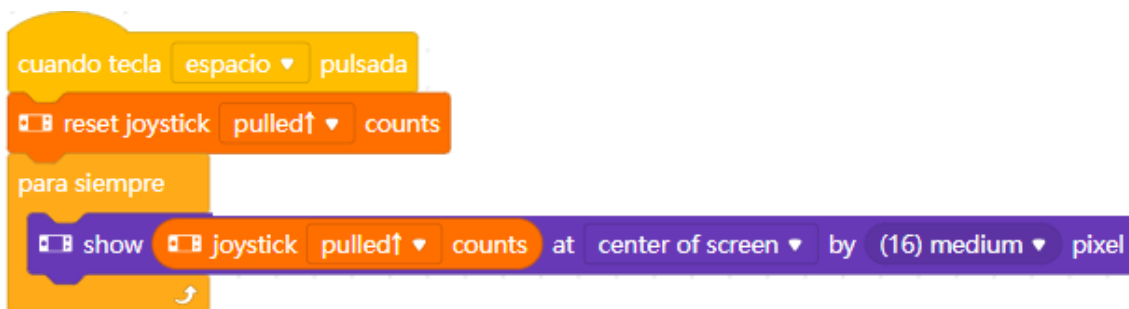


Haga clic para seleccionar una dirección del cuadro de lista desplegable.

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el número de veces en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

#### Ejemplo



Con este programa, **CyberPi** se puede utilizar como contador.

En el modo **En vivo**, después de presionar la tecla espaciadora, mueva repetidamente el joystick hacia arriba. **CyberPi** cuenta desde cero y muestra el número de veces en el centro de su pantalla.

### 7.5.3. Restablece el número de veces que se mueve el joystick en la dirección especificada a cero



#### Cómo utilizar

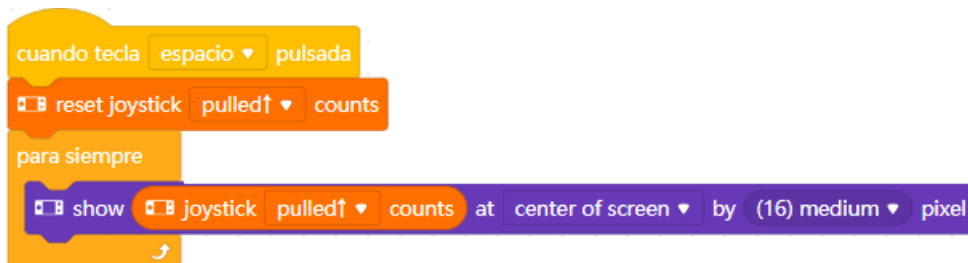
**CyberPi** está equipado con un joystick que se puede mover en cinco direcciones.



Haga clic para seleccionar una dirección del cuadro de lista desplegable.

Después de ejecutar este bloque, el número de veces que se mueve el joystick en la dirección especificada se cuenta desde cero..

#### Ejemplo



Con este programa, **CyberPi** puede funcionar como contador.

En el modo **En vivo**, después de presionar la tecla espaciadora, mueva repetidamente el joystick hacia arriba. **CyberPi** cuenta desde cero y muestra el número de veces en el centro de su pantalla.

### 7.5.4. Determina si se presiona el botón especificado



#### Cómo utilizar

**CyberPi** está equipado con dos botones, A y B.



Haga clic para seleccionar un botón del cuadro de lista desplegable.

Este es un bloque booleano que contiene una condición. Úselo en combinación con otro bloque que requiera una condición.

### Ejemplo



Con este programa, **CyberPi** puede funcionar como una grabadora.

En el modo **En vivo**, después de presionar la tecla espaciadora, presione A y B para iniciar y detener la grabación.

### 7.5.5. Informa el número de veces que se presiona el botón especificado

button A press counts

#### Cómo utilizar

**CyberPi** está equipado con dos botones, A y B.

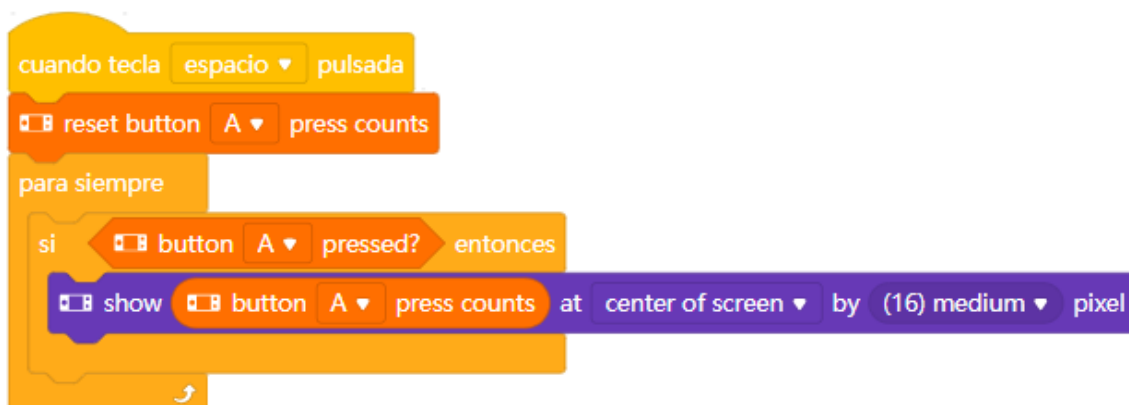


Haga clic para seleccionar un botón del cuadro de lista desplegable.

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el número de veces en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

### Ejemplo



Con este programa, **CyberPi** se puede utilizar como contador.

En el modo **En vivo**, después de presionar la tecla de espacio, presione repetidamente el botón A. **CyberPi** cuenta desde cero y muestra el número de veces en el centro de su pantalla.

### 7.5.6. Restablece el número de veces que se presiona el botón especificado a cero



### Cómo utilizar

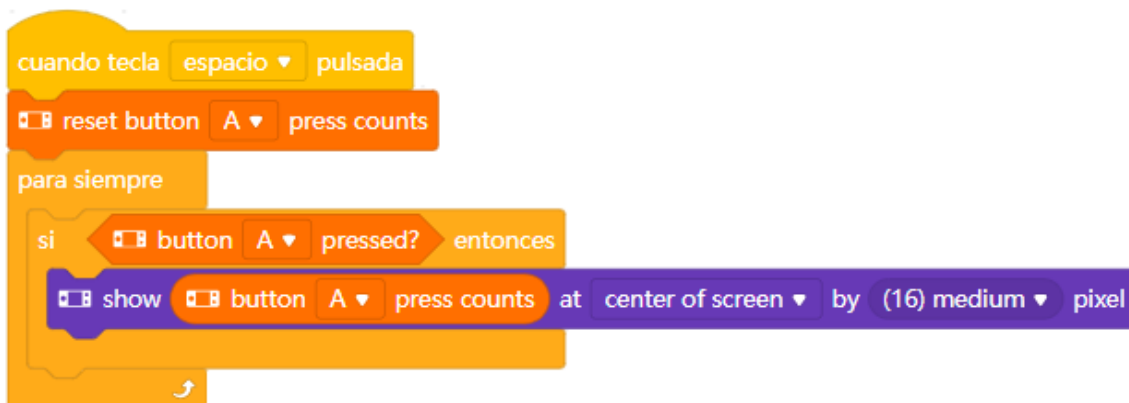
**CyberPi** está equipado con dos botones, A y B.



Haga clic para seleccionar un botón del cuadro de lista desplegable.

Después de ejecutar este bloque, el número de veces que se presiona el botón especificado se cuenta desde cero.

### Ejemplo



Con este programa, **CyberPi** se puede utilizar como contador.

En el modo **En vivo**, después de presionar la tecla de espacio, presione repetidamente el botón A. **CyberPi** cuenta desde cero y muestra el número de veces en el centro de su pantalla.

### 7.5.7. Informa el volumen ambiental detectado por el micrófono en CyberPi

 intensidad del sonido

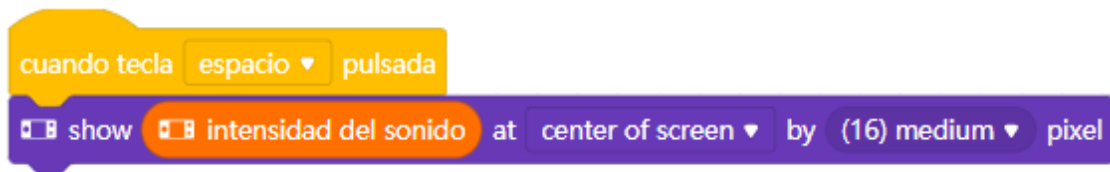
## Cómo utilizar

Intervalo de valores: 0–100

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el volumen del ambiente en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

## Ejemplo



Utilice este bloque en combinación con un bloque “**show**” de la librería “**Pantalla**”.

En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** muestra el volumen ambiental detectado por el micrófono en el centro de su pantalla.

## 7.5.8. Informa la intensidad de la luz ambiental detectada por el sensor de luz en CyberPi



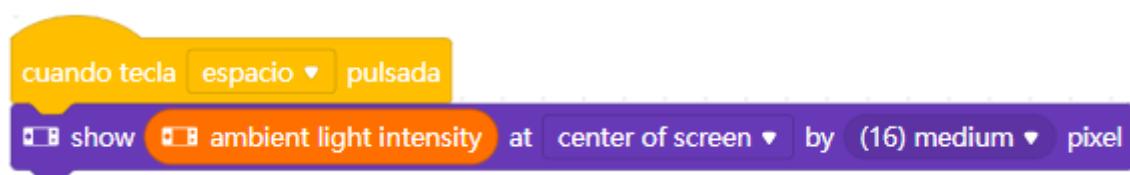
## Cómo utilizar

Intervalo de valores: 0–100

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver la intensidad de la luz ambiental en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

## Ejemplo



Utilice este bloque en combinación con un bloque “**show**” de la librería “**Pantalla**”.

En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** muestra el volumen ambiental detectado por el micrófono en el centro de su pantalla.

### 7.5.9. Reporta el valor de conteo del temporizador

 temporizador(es)

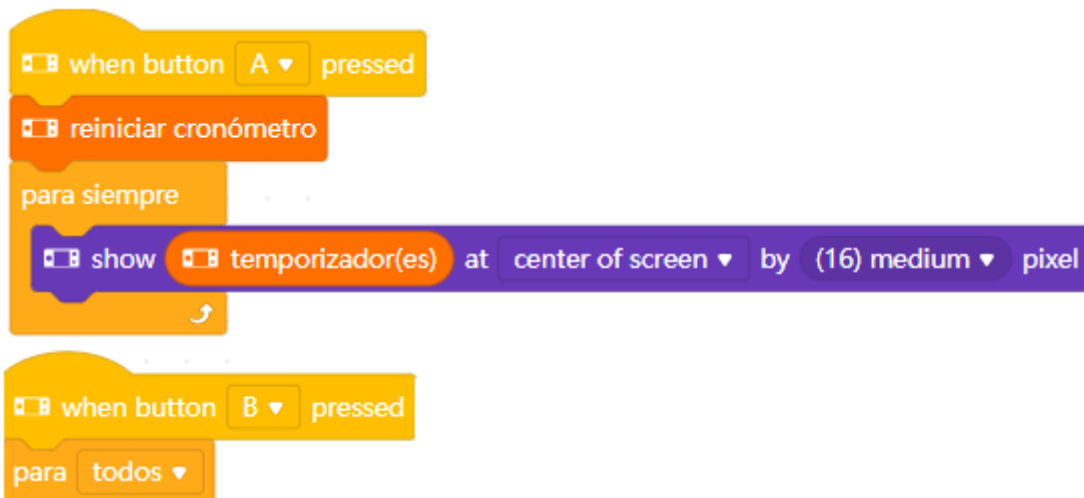
#### Cómo utilizar

El temporizador comienza a contar desde 0 cada vez que se enciende **CyberPi**. Este bloque informa la duración de **CyberPi** se ejecuta este tiempo.

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el valor de recuento del temporizador en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

#### Ejemplo



Con este programa, **CyberPi** puede funcionar como temporizador.

En el modo **Cargar**, cuando presiona el botón A en **CyberPi**, **CyberPi** comienza a contar desde 0 y muestra el tiempo en el centro de su pantalla. Cuando presiona el botón B, **CyberPi** deja de contar.

### 7.5.10. Restablece el temporizador a cero

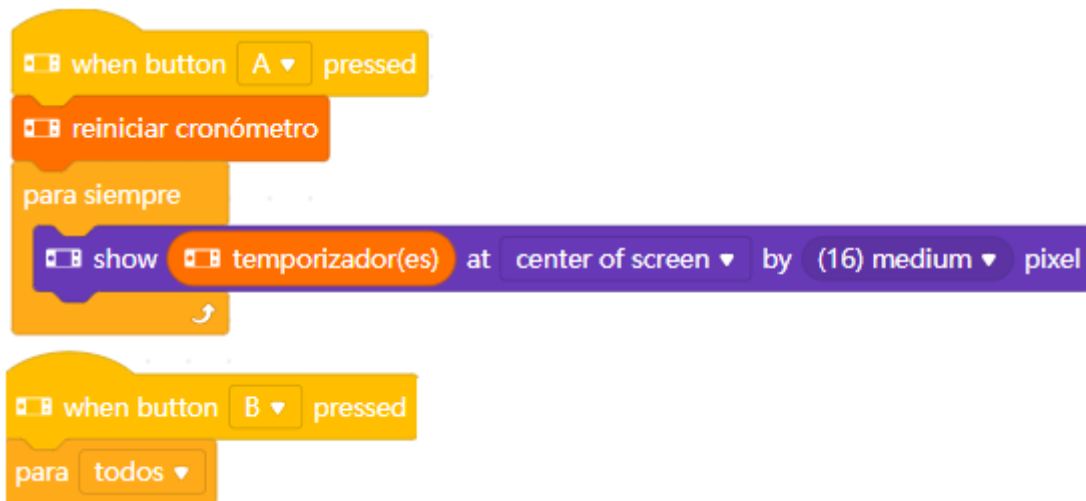
 reiniciar cronómetro

## Cómo utilizar

El temporizador de **CyberPi** comienza a contar desde cero cada vez que se enciende **CyberPi**.

Puede usar este bloque para reiniciar el temporizador, haciendo que comience a contar desde cero nuevamente.

## Ejemplo



Con este programa, **CyberPi** puede funcionar como temporizador.

En el modo **Cargar**, cuando presiona el botón A en **CyberPi**, **CyberPi** comienza a contar desde 0 y muestra el tiempo en el centro de su pantalla. Cuando presiona el botón B, **CyberPi** deja de contar.

### 7.5.11. Informa el nombre de host de **CyberPi**



## Cómo utilizar

El nombre de host predeterminado de **CyberPi** es **CyberPi**.

El nombre de host se muestra en la interfaz de usuario del sistema **CyberOS** que se ejecuta en su **CyberPi**. Al configurar los nombres de los dispositivos, los usuarios pueden encontrar sus **CyberPis** o identificarse entre sí en una LAN o IoT.

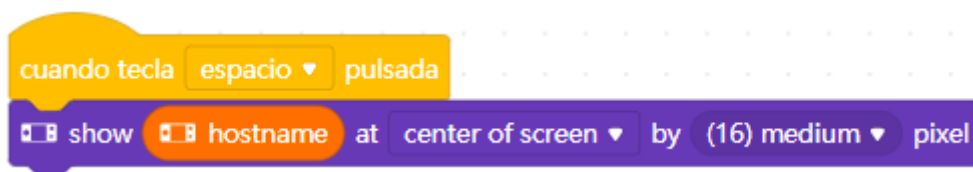
Puede configurar el nombre de host de la siguiente manera.



Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el nombre de host en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

### Ejemplo



Utilice este bloque en combinación con un bloque de **visualización**.

En el modo **En vivo**, cuando presiona la tecla espaciadora, **CyberPi** muestra el nombre de host que estableció en el centro de su pantalla.

### 7.5.12. Informa el nivel de batería



#### Cómo utilizar

El nivel de la batería se refiere al del protector de bolsillo conectado a **CyberPi**, en un rango de 0 a 100. No use esta opción si no hay ningún protector de bolsillo conectado.

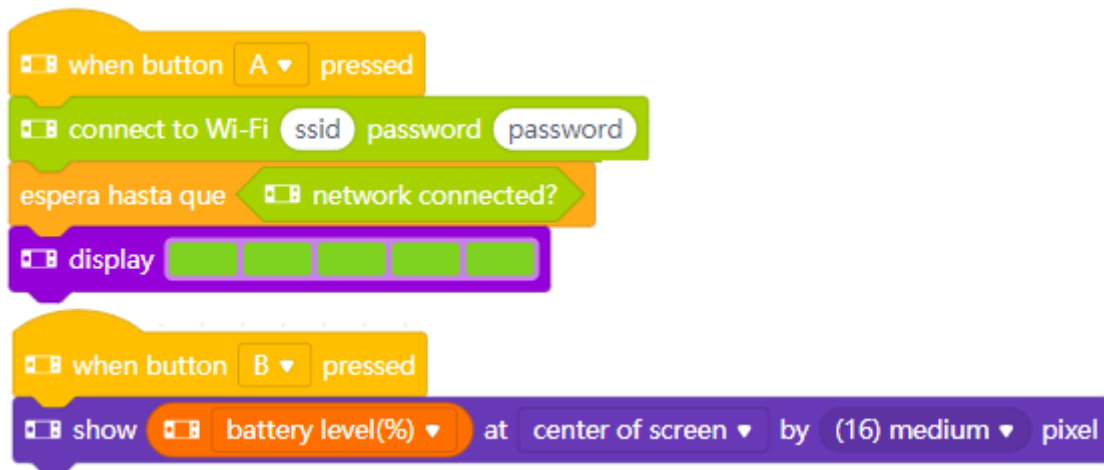
La dirección MAC se utiliza para conectarse a Wi-Fi. Tenga en cuenta que la dirección MAC que obtiene al usar este bloque puede ser incorrecta si **CyberPi** no está conectado a Wi-Fi. El valor de salida es una cadena de 12 bytes.

El nombre de Bluetooth se refiere a la transmisión por **CyberPi**, utilizada para la conexión Bluetooth.

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver la información correspondiente en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

### Ejemplo



**Nota:** Configure la cuenta de Wi-Fi y la contraseña a las disponibles.

Compile este programa en el modo de carga y cárguelo en **CyberPi**.

Presione el botón A. **CyberPi** comienza a conectarse a Wi-Fi.

Después de que los LED se enciendan en verde, presione el botón B. **CyberPi** muestra el nivel de la batería en la pantalla.

## 7.6. Librería: LAN



Envío y recepción de mensajes a través de una red.

### 7.6.1. Transmite el mensaje especificado en la LAN



#### Cómo utilizar

Haga clic para establecer el nombre del mensaje que se transmitirá.

El mensaje transmitido por **CyberPi** a través de este bloque puede ser recibido por todos los demás dispositivos en la misma LAN.

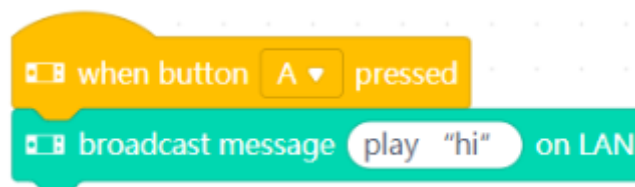
**Nota:** Todos los **CyberPis** en la misma LAN deben usar el mismo canal para la comunicación. Sin conexión de enrutador, **CyberPi** usa el mismo canal predeterminado y, por lo tanto, pueden comunicarse entre sí en la LAN; cuando **CyberPi** está conectado a un enrutador, su canal depende de la configuración del enrutador, que puede ser diferente del canal predeterminado y, por lo tanto, es posible que no se comunique con otro **CyberPi** que no esté conectado al enrutador.

Por lo tanto, para garantizar una comunicación adecuada, si usa un enrutador, conecte todos los **CyberPis** en la misma LAN al enrutador.

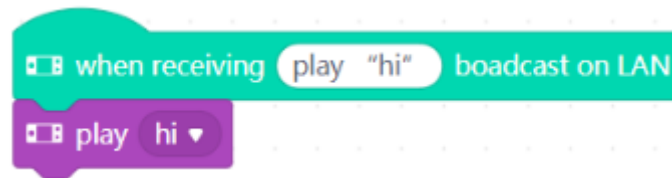
#### Ejemplo

Los siguientes programas permiten que dos **CyberPis** interactúen entre sí.

##### CyberPi 1



##### CyberPi 2

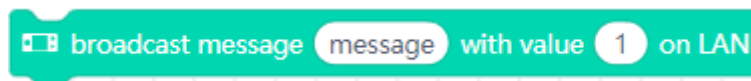


Compile los dos programas en el modo de **carga** y cárguelos en **CyberPis 1** y **2**, respectivamente.

Reinicie **CyberPis 1** y **2**.

Presione el botón A en **CyberPi 1**. **CyberPi 2** dice "hola".

### 7.6.2. Difunde el mensaje especificado con el valor especificado en la LAN



#### Cómo usar

Haga clic para establecer el nombre del mensaje y el valor que se transmitirá.

El mensaje transmitido por **CyberPi** a través de este bloque puede ser recibido por todos los demás dispositivos en la misma LAN.

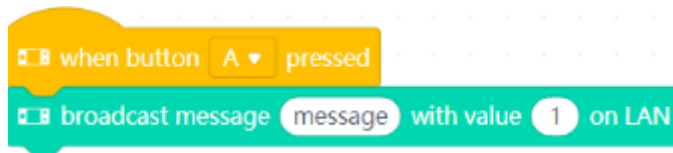
#### Nota:

Para garantizar una comunicación adecuada, si usa un enrutador, conecte todos los **CyberPi** en la misma LAN al enrutador.

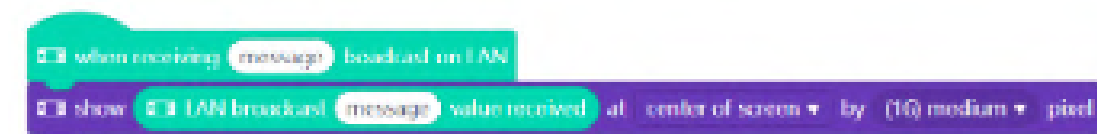
#### Ejemplo

Los siguientes programas permiten que dos **CyberPis** interactúen entre sí.

#### CyberPi 1



#### CyberPi 2



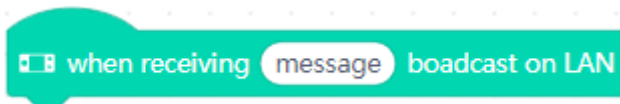
Compile los dos programas en el modo de **carga** y cárguelos en **CyberPis** 1 y 2, respectivamente.

Reinicie **CyberPis** 1 y 2.

Presione el botón A en **CyberPi** 1.

**CyberPi** 2 muestra el valor 1 en su pantalla.

### 7.6.3. Ejecuta los bloques siguientes cuando se recibe el mensaje especificado



#### Cómo usar

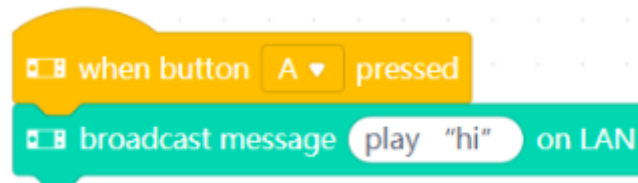
**CyberPi** puede recibir mensajes transmitidos por otros dispositivos en la misma LAN.

**Nota:** Para garantizar una comunicación adecuada, si usa un enrutador, conecte todos los **CyberPis** en la misma LAN al enrutador.

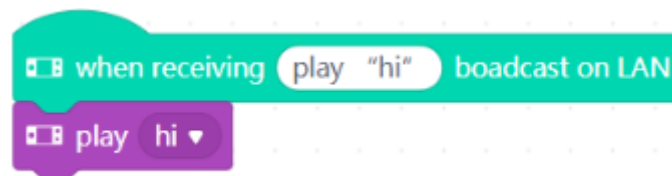
#### Ejemplo

Los siguientes programas permiten que dos **CyberPis** interactúen entre sí.

#### CyberPi 1



#### CyberPi 2

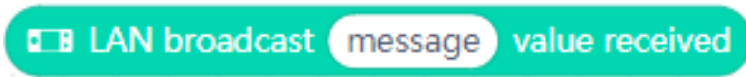


Compile los dos programas en el modo de **carga** y cárguelos en **CyberPis** 1 y 2, respectivamente.

Reinicie **CyberPis** 1 y 2.

Presione el botón A en **CyberPi** 1. **CyberPi** 2 dice "hola".

### 7.6.4. Informa el valor recibido por **CyberPi** a través del mensaje especificado



## Cómo usar

Después de estar conectado a Wi-Fi, **CyberPi** puede recibir mensajes y valores transmitidos por otros dispositivos conectados al mismo Wi-Fi.

Puede seleccionar la casilla de verificación a la izquierda de este bloque para ver el valor recibido en el escenario.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

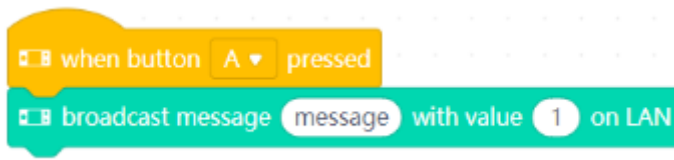
### Nota:

Para garantizar una comunicación adecuada, si usa un enrutador, conecte todos los **CyberPis** en la misma LAN al enrutador.

## Ejemplo

Los siguientes programas permiten que dos **CyberPis** interactúen entre sí.

### CyberPi 1



### CyberPi 2



Compile los dos programas en el modo de **carga** y cárguelos en **CyberPis** 1 y 2, respectivamente.

Reinicie **CyberPis** 1 y 2.

Presione el botón A en **CyberPi** 1.

**CyberPi** 2 muestra el valor 1 en su pantalla.

## 7.7. Librería: IA



### 7.7.1. Conecta **CyberPi** a la red Wi-Fi especificada



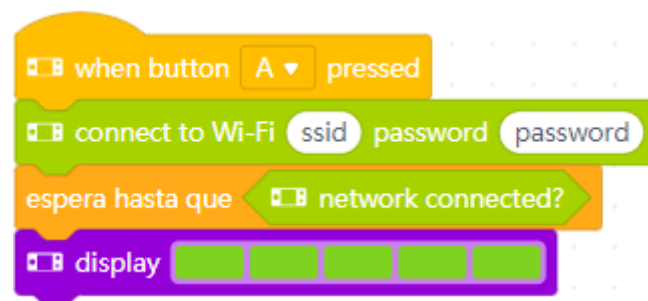
#### Cómo usar

**CyberPi** está equipado con un módulo que admite las siguientes funciones:

- Funciones de IA: permite a **CyberPi** reconocer discursos y traducir textos
- Funciones de IoT: facilita la interacción y el control entre redes y dispositivos

Haga clic para configurar la cuenta Wi-Fi y la contraseña que están disponibles.

#### Ejemplo



Compile este programa en el modo de **carga** y cárguelo en **CyberPi**. Presione el botón A. **CyberPi** comienza a conectarse al Wi-Fi especificado. Cuando los LED se iluminan en verde, la conexión se realiza correctamente.

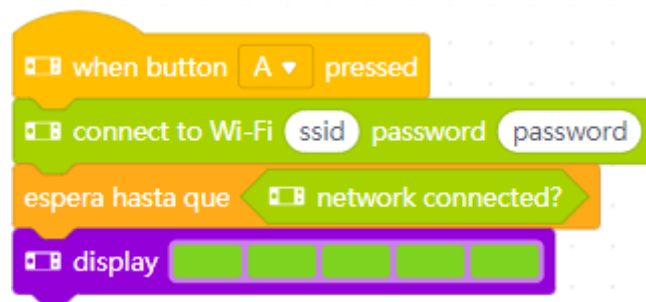
### 7.7.2. Determina si **CyberPi** se ha conectado a Wi-Fi



#### Cómo usar

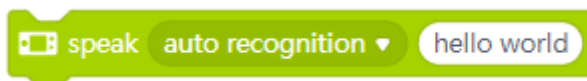
Este es un bloque booleano que contiene una condición. Úselo en combinación con otro bloque que requiera una condición.

#### Ejemplo



Compile este programa en el modo de **carga** y cárguelo en **CyberPi**. Presione el botón A. **CyberPi** comienza a conectarse al Wi-Fi especificado. Cuando los LED se iluminan en verde, la conexión se realiza correctamente.

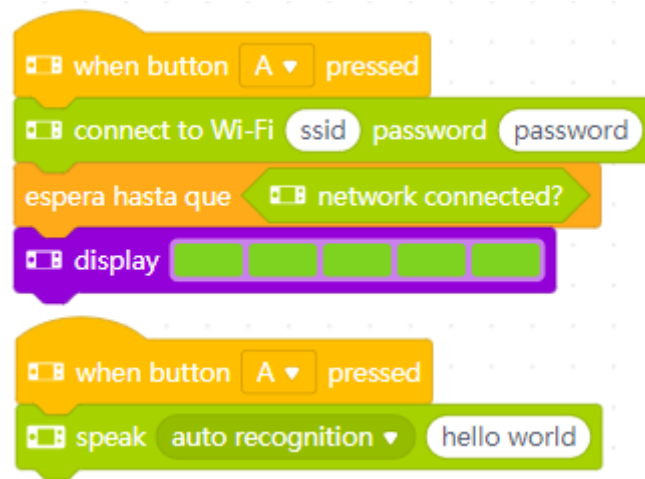
### 7.7.3. Reconoce el texto especificado y lo lee en voz alta en el idioma reconocido



#### Cómo usar

Haga clic para ingresar el texto a ser reconocido.

#### Ejemplo



**Nota:** Configure la cuenta Wi-Fi y la contraseña a las disponibles.

Compile este programa en el modo de **carga** y cárguelo en **CyberPi**.

Presione el botón A. **CyberPi** comienza a conectarse a Wi-Fi.

Después de que los LED se enciendan en verde, presione el botón B. **CyberPi** dice "hello world".

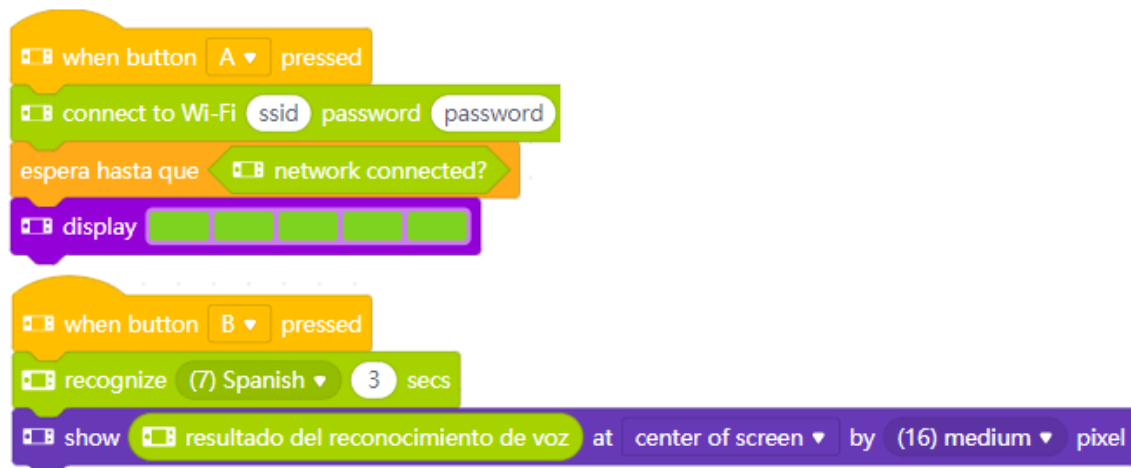
#### 7.7.4. Reconoce el idioma especificado durante el período especificado



##### Cómo usar

Haga clic para seleccionar un idioma del cuadro de lista desplegable y establecer la hora.


##### Ejemplo



**Nota:** Configure la cuenta Wi-Fi y la contraseña a las disponibles.

Compile este programa en el modo de **carga** y cárguelo en **CyberPi**. Presione el botón A. **CyberPi** comienza a conectarse a Wi-Fi. Después de que los LED se enciendan en verde, presione el botón B y hable en inglés. **CyberPi** muestra las palabras reconocidas en su pantalla.

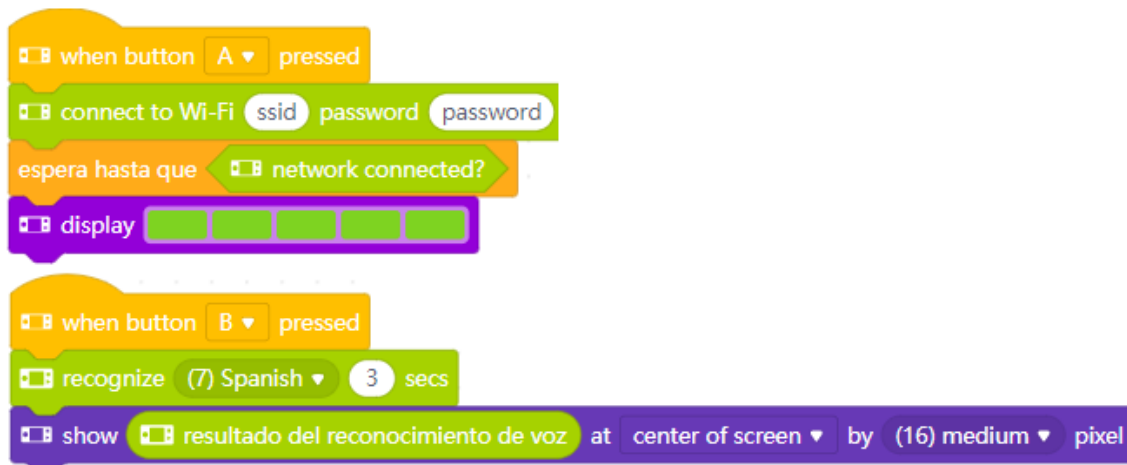
### 7.7.5. Informa las palabras reconocidas en un discurso.

 resultado del reconocimiento de voz

#### Cómo usar

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

#### Ejemplo



**Nota:** Configure la cuenta Wi-Fi y la contraseña a las disponibles. Compile este programa en el modo de **carga** y cárguelo en **CyberPi**. Presione el botón A. **CyberPi** comienza a conectarse a Wi-Fi. Después de que los LED se enciendan en verde, presione el botón B y hable en inglés. **CyberPi** muestra las palabras reconocidas en su pantalla.

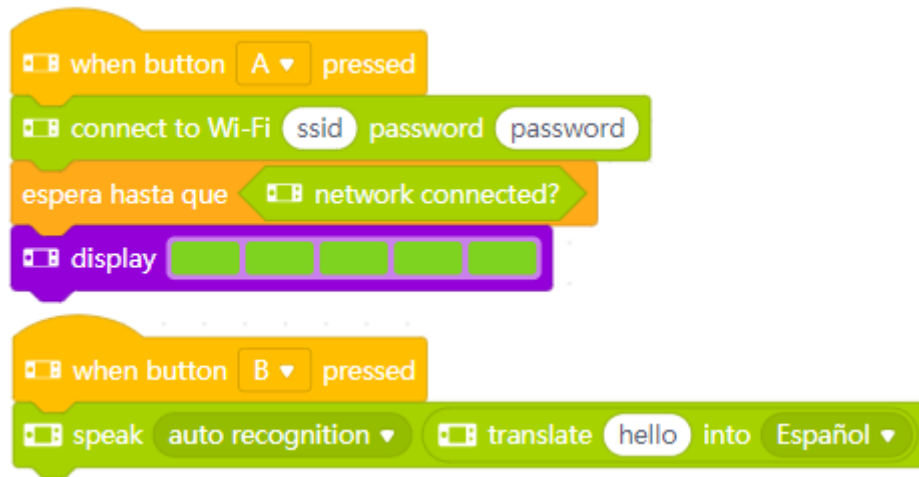
### 7.7.6. Reporta la traducción del texto especificado

 translate hello into Chino

#### Cómo usar

Haga clic para ingresar el texto a traducir y el idioma de destino. Actualmente, este bloque solo admite la traducción chino-inglés e inglés-chino.

## Ejemplo



**Nota:** Configure la cuenta Wi-Fi y la contraseña a las disponibles.  
Compile este programa en el modo de **carga** y cárguelo en **CyberPi**.  
Presione el botón A. **CyberPi** comienza a conectarse a Wi-Fi.  
Después de que los LED se iluminen en verde, presione el botón B. **CyberPi** dice "Hello" traducido al español

## 7.8. Librería: IoT



### 7.8.1. Conecta **CyberPi** a la red Wi-Fi especificada



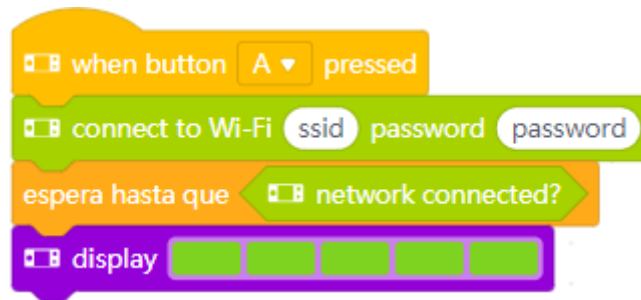
#### Cómo usar

**CyberPi** está equipado con un módulo que admite las siguientes funciones:

- Funciones de IA: permite a **CyberPi** reconocer discursos y traducir textos
- Funciones de IoT: facilita la interacción y el control entre redes y dispositivos

Haga clic para configurar la cuenta Wi-Fi y la contraseña que están disponibles.

## Ejemplo



Compile este programa en el modo de **carga** y cárguelo en **CyberPi**. Presione el botón A. **CyberPi** comienza a conectarse al Wi-Fi especificado. Cuando los LED se iluminan en verde, la conexión se realiza correctamente.

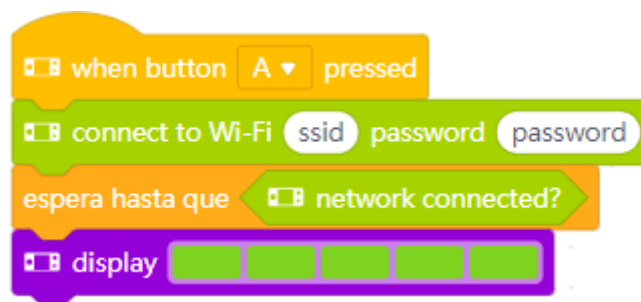
### 7.8.2. Determina si **CyberPi** se ha conectado a Wi-Fi



#### Cómo usar

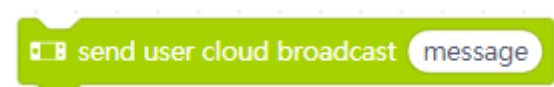
Este es un bloque booleano que contiene una condición. Úselo en combinación con otro bloque que requiera una condición.

## Ejemplo



Compile este programa en el modo de **carga** y cárguelo en **CyberPi**. Presione el botón A. **CyberPi** comienza a conectarse al Wi-Fi especificado. Cuando los LED se iluminan en verde, la conexión es exitosa.

### 7.8.3. Envía el mensaje de nube de usuario especificado



## Cómo usar

### ¿Qué es un mensaje de nube de usuario?

Con la misma cuenta **mBlock5**, los mensajes en la nube del usuario permiten la comunicación entre dispositivos y proyectos, es decir, puede permitir que **CyberPi** interactúe con otros **CyberPis** y sprites después de iniciar sesión con la misma cuenta **mBlock5**.

Antes de utilizar la función de mensaje en la nube del usuario, debe:

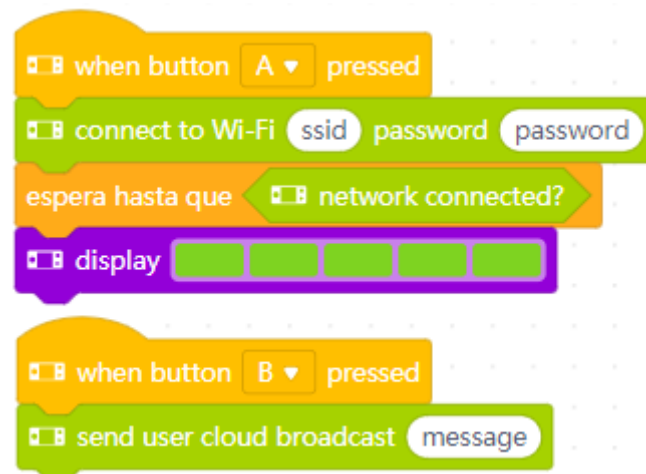
1. Configure **mBlock5** en el modo de **carga**
2. Inicie sesión en su cuenta **mBlock5**
3. Conecte **CyberPi** a Wi-Fi

Haga clic para establecer el nombre del mensaje de nube de usuario que se enviará.

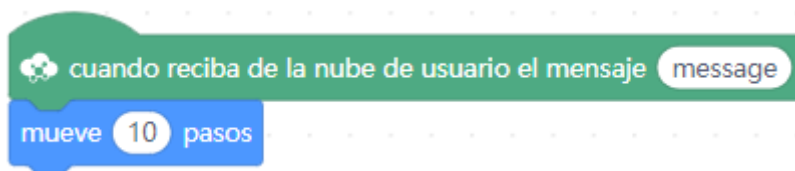
## Ejemplo

Los siguientes programas permiten a **CyberPi** interactuar con un objeto.

### CyberPi



### Script



**Nota:** Configure la cuenta Wi-Fi y la contraseña a las disponibles.

Compile los dos programas en el modo de **carga** y cargue el programa en **CyberPi**.

Presione el botón A. **CyberPi** comienza a conectarse a Wi-Fi.

Después de que los LED de **CyberPi** se enciendan en verde, presione el botón B.

**CyberPi** envía el mensaje de nube del usuario llamado **mensaje**.

Después de recibir el mensaje de la nube del usuario, el sprite se mueve 10 pasos.

### 7.8.4. Envía mensaje a la nube y el valor especificado



#### Cómo usar

Haga clic para establecer el nombre del mensaje de nube del usuario y el valor que se enviará.

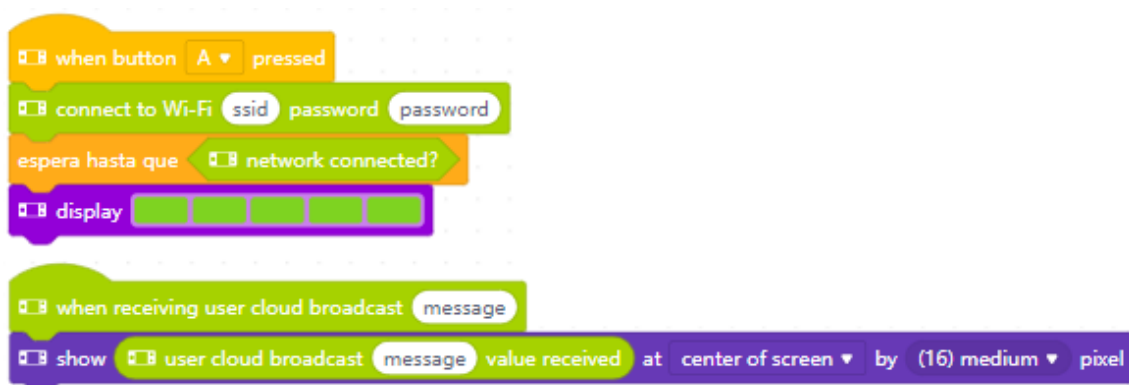
#### Ejemplo

Los siguientes programas permiten que **CyberPi** interactúe con otro **CyberPi**.

#### CyberPi 1



#### CyberPi 2



**Nota:** Configure la cuenta Wi-Fi y la contraseña a las disponibles.

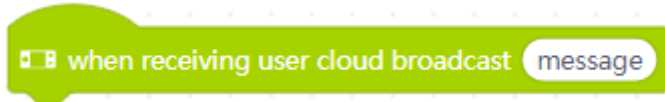
Compile los dos programas en el modo de **carga** y cárguelos en **CyberPis** 1 y 2, respectivamente.

Presione el botón A en **CyberPis** 1 y 2. Comienzan a conectarse a Wi-Fi.

Después de que los LED de **CyberPi** se enciendan en verde, presione el botón B en **CyberPi** 1. **CyberPi** 1 envía el mensaje de nube del usuario llamado **mensaje** y el valor 1.

Después de recibir el mensaje de la nube del usuario, **CyberPi** 2 muestra el valor 1 en su pantalla.

### 7.8.5. Ejecuta los bloques cuando se recibe el mensaje de nube



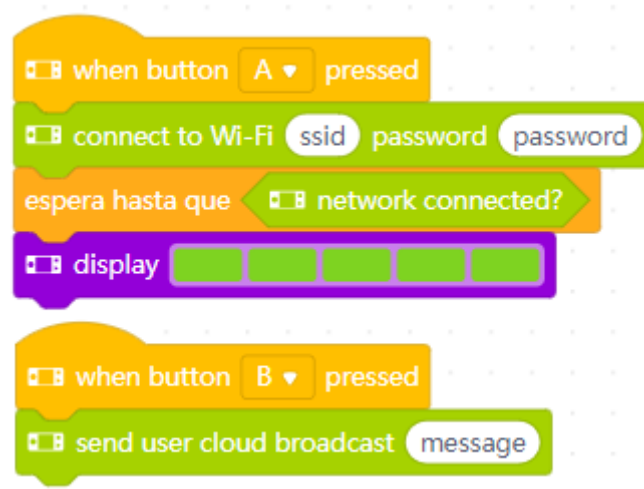
#### Cómo usar

Después de estar conectado a Wi-Fi, **CyberPi** puede interactuar con un sprite u otro **CyberPi** que esté programado con la misma cuenta **mBlock5**.

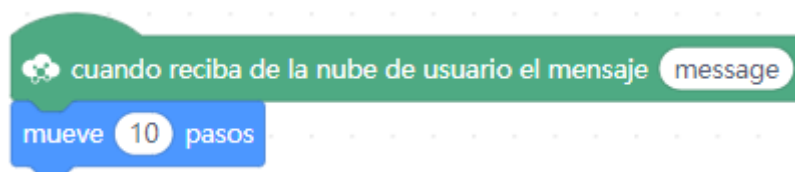
#### Ejemplo

Los siguientes programas permiten a **CyberPi** interactuar con un objeto.

#### CyberPi



#### Script



**Nota:** Configure la cuenta Wi-Fi y la contraseña a las disponibles.

Compile los dos programas en el modo de **carga** y cargue el programa en **CyberPi**.

Presione el botón A. **CyberPi** comienza a conectarse a Wi-Fi.

Después de que los LED de **CyberPi** se enciendan en verde, presione el botón B. **CyberPi** envía el mensaje de nube del usuario llamado **mensaje**. Después de recibir el mensaje de la nube del usuario, el objeto se mueve 10 pasos.

### 7.8.6. Devuelve el valor recibido de la nube

 user cloud broadcast **message** value received

#### Cómo usar.

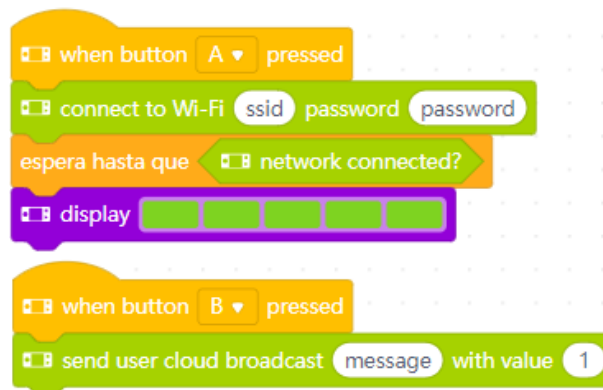
Después de estar conectado a Wi-Fi, **CyberPi** puede interactuar con un sprite u otro **CyberPi** que esté programado con la misma cuenta **mBlock5**.

Este es un bloque informador que debe usarse en combinación con otro bloque que requiera datos.

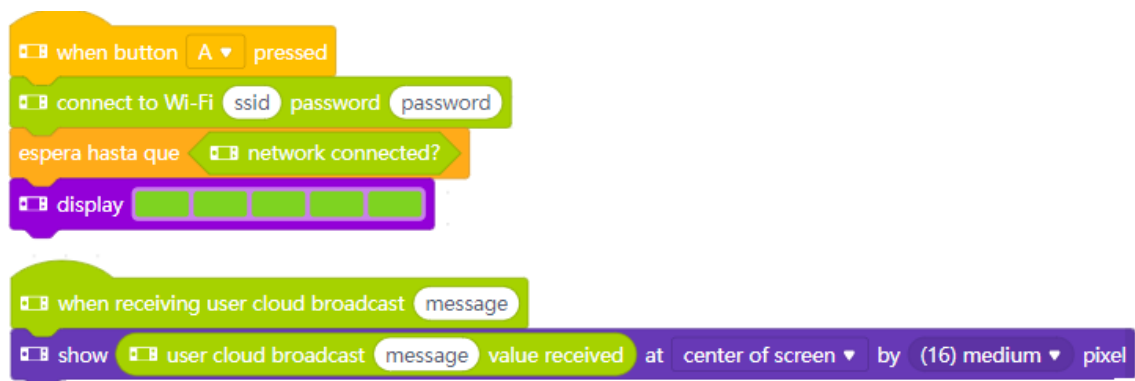
#### Ejemplo

Los siguientes programas permiten que **CyberPi** interactúe con otro **CyberPi**.

#### CyberPi 1



#### CyberPi 2



**Nota:** Configure la cuenta Wi-Fi y la contraseña a las disponibles.

Compile los dos programas en el modo de carga y cárguelos en **CyberPi** 1 y 2, respectivamente.

Presione el botón A en **CyberPi** 1 y 2. Comienzan a conectarse a Wi-Fi.

Después de que los LED de **CyberPi** se enciendan en verde, presione el botón B en **CyberPi** 1. **CyberPi** 1 envía el mensaje de nube del usuario llamado mensaje y el valor 1.

Después de recibir el mensaje de la nube del usuario, **CyberPi** 2 muestra el valor 1 en su pantalla.

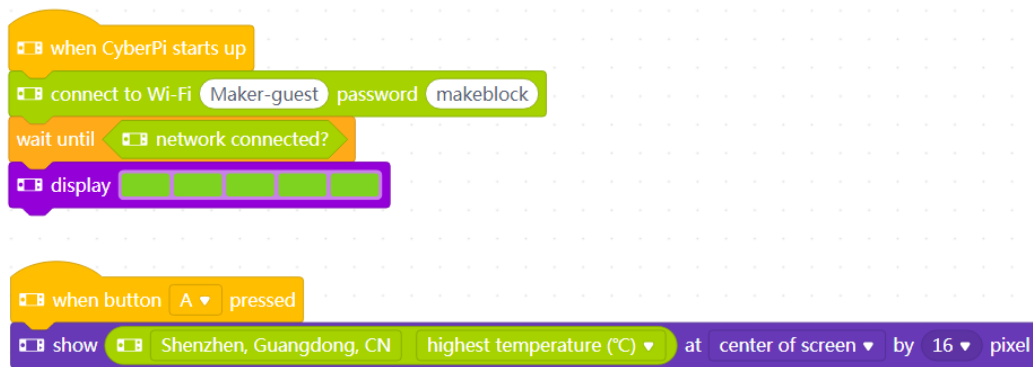
### 7.8.7. Devuelve las condiciones meteorológicas seleccionadas del distrito especificado



#### Cómo usar

Haga clic para establecer el distrito y las condiciones meteorológicas que desea conocer.

#### Ejemplo



**Nota:** Configure la cuenta de Wi-Fi y la contraseña a las disponibles. Compile el programa en modo **Cargar** y cárguelo en **CyberPi**. Después de que los LED de **CyberPi** se enciendan en verde, presione el botón A. **CyberPi** muestra la temperatura más alta de Shenzhen en su pantalla.

### 7.8.8. Devuelve la calidad del aire del lugar especificado



#### Cómo usar

Click to set the district and the weather quality condition you want to know about.

## Ejemplo



**Nota:** Configure la cuenta de Wi-Fi y la contraseña a las disponibles. Compile el programa en modo Cargar y cárguelo en **CyberPi**. Después de que los LED de **CyberPi** se enciendan en verde, presione el botón A. **CyberPi** muestra el índice de calidad del aire (AQI) de hoy en Shenzhen en su pantalla.

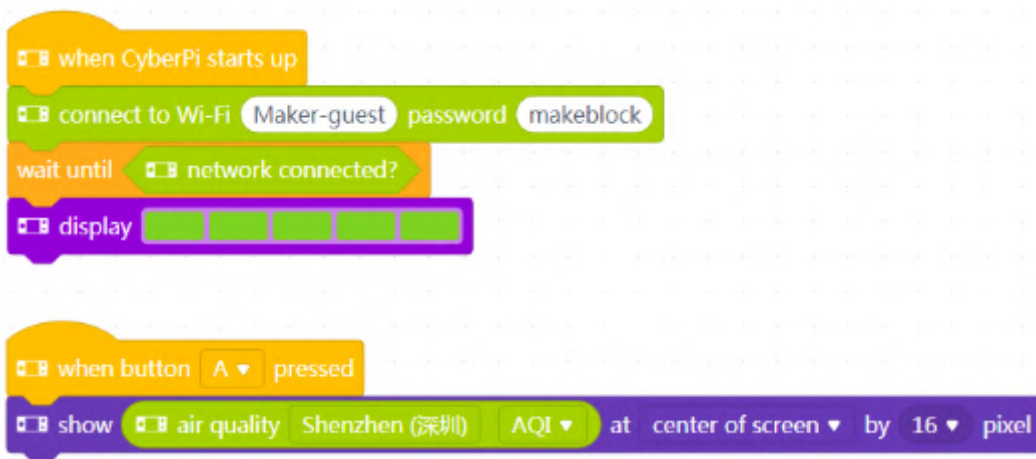
### 7.8.9. Devuelve la hora de salida o puesta del sol del distrito especificado

posición salida del sol hora

## Cómo usar

Haga clic para establecer el distrito y la hora de salida o puesta del sol que desea conocer.

## Ejemplo



**Nota:** Configure la cuenta de Wi-Fi y la contraseña a las disponibles.

Compile el programa en modo **Cargar** y cárguelo en **CyberPi**.

Después de que los LED de **CyberPi** se enciendan en verde, presione el botón A. **CyberPi** muestra la hora del amanecer hoy en Shenzhen en su pantalla.

## 8. Ejercicios realizados con CyberPi

A continuación vamos a plantear y resolver algunos ejemplos de programación usando la tarjeta **CyberPi**, algunos de los cuales vienen como ejemplo en el software **mBlock**.

### 8.1. CyberPi Prácticas Básicas

#### 8.1.1. Temporizador

##### Funcionamiento

Con este ejemplo vamos a experimentar con el temporizador interno que incorpora **CyberPi**. Este temporizador entrega su valor en el bloque “**temporizador**” de la librería “**Sensores**” propia del hardware **CyberPi**. Realmente lo que vamos a implementar es un sencillo cronómetro.

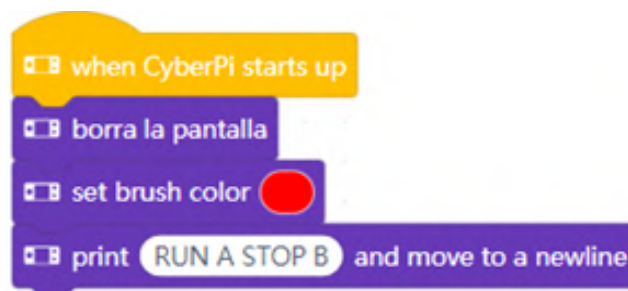
Nuestra aplicación será montada para trabajar en modo “cargar”, es decir para que funcione en modo off-line (con la unidad Cyberpi desconectada del PC)

Nuestro programa funcionará de la siguiente manera: Cuando pulsemos el **botón A** de la unidad se reinicia el tiempo mostrándose el valor en el display. Cuando se pulse el **botón B** se detendrá la cuenta del temporizador.

##### Diseño del script asociado al dispositivo

Para empezar debemos colocar el bloque de la librería “**Eventos**” llamado “**when CyberPi starts up**” es decir, cuando se inicia la unidad **CyberPi**.

Pondremos el color del texto escrito en “**set brush color**” en Rojo e imprimiremos en pantalla un texto, “**print..... and move to a newline**” que nos indicara que al pulsar el botón A se inicia la temporización y al pulsar el botón B se detiene.



Seguidamente vamos a implementar el script asociado a la pulsación sobre el botón B, este se activará con el bloque “**when button A pressed**” de la librería de “**Eventos**”

Lo que se debe hacer es, en primer lugar poner a cero el temporizador, con el bloque “**reiniciar cronómetro**” seguidamente cambiamos el color de la letra de pantalla a “Azul” y a continuación entraremos en un bucle del tipo “**repite hasta que**” de la librería “**Control**” cuya misión será estar ejecutándose hasta que se pulse el botón B de **CyberPi**.



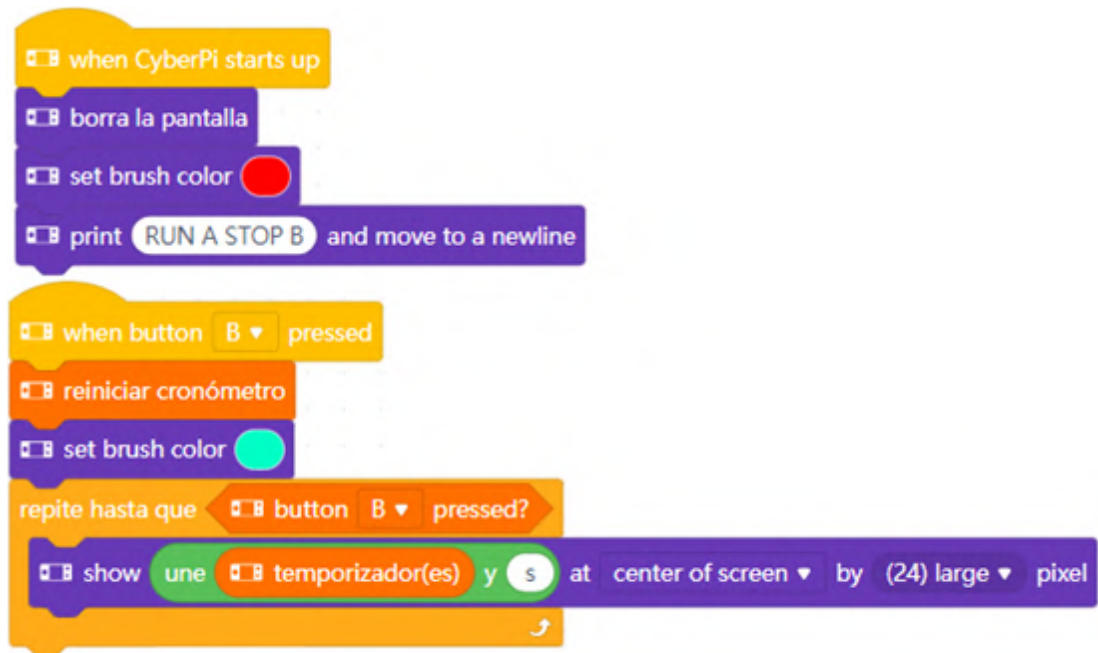
### Prueba.

El aspecto de la pantalla de **CyberPi** sería el siguiente:

- Para iniciar: Pulsando A.
- Para detener: Pulsando B



A continuación se muestra el script completo,



## 8.1.2. Arco Iris

**CyberPi** posee en uno de sus laterales una cadena de 5 LEDs RGB con la que se pueden realizar distintas simulaciones de efecto luminoso encendiendo y apagando estos LEDs.

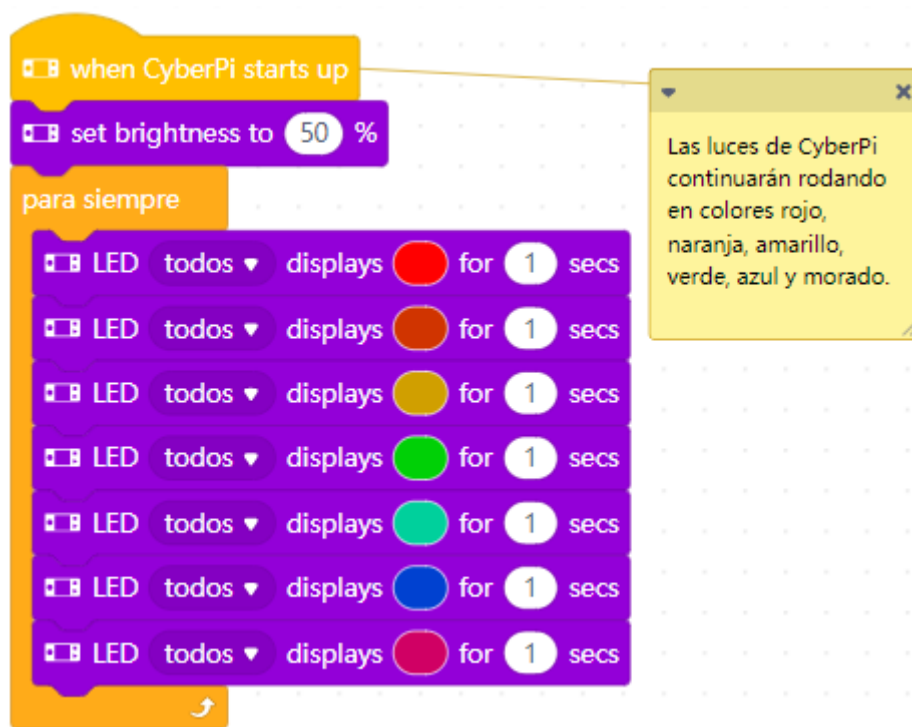
### Funcionamiento.

En esta aplicación vamos a hacer una simulación que nos recordará el espectro de colores del Arco Iris. Se trata de ir encendiendo los 5 leds con los colores básicos de dicho fenómeno meteorológico.

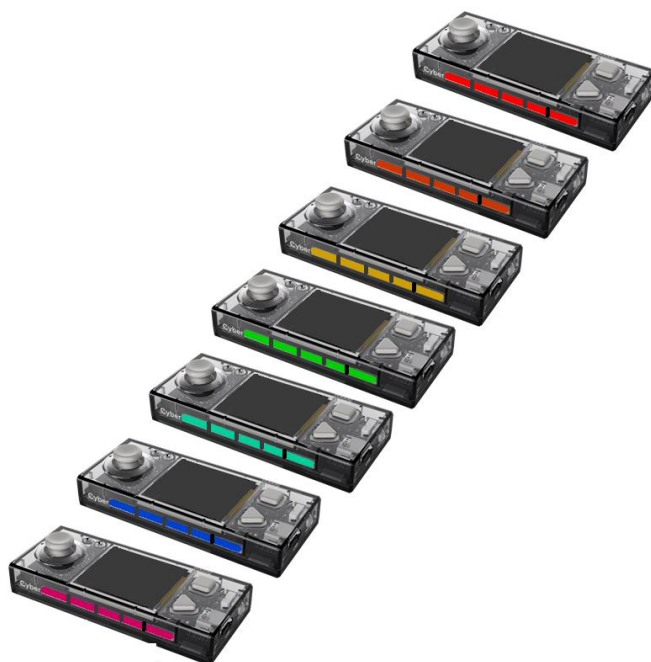
La aplicación, como la anterior, funcionará en modo “carga” por lo tanto off-line.

### Construcción del script del dispositivo

En este caso el programa empieza con el bloque **“when CyberPi starts up”** a partir del cual colocamos el brillo de los leds al 50% mediante la función **“set brightness to..”** seguidamente entraremos en un bucle “por siempre” dentro del cual colocaremos las funciones **“LED <todos> displays <color> for <tiempo> secs”** con un tiempo de encendido de 1seg, y cada uno de los colores que integran el espectro del Arco Iris



## Prueba de funcionamiento



### 8.1.3. Contador de Pasos

Sabemos que **CyberPi** incorpora un acelerómetro y por ello vamos a utilizarlo en este ejemplo para facilitar a un deportista que corra la posibilidad de contar los pasos. Esta aplicación ya está implementada en un sinnúmero de dispositivos deportivos pero nosotros vamos a reinventarla usando **CyberPi**.

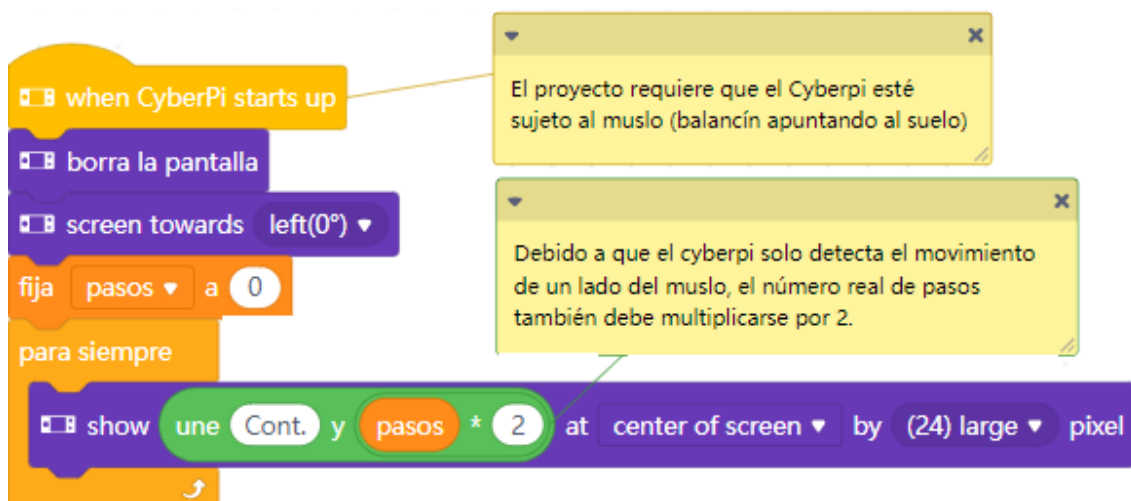
#### Funcionamiento

Se trata de fijar a la pierna el dispositivo **CyberPi** de tal manera que cuando esta flexione el dispositivo sea capaz de reconocer el cambio de posición angular del acelerómetro. Se creará una variable a la que podemos nombrar con el nombre de “pasos” que se verá incrementada cada vez que se produzca una alteración del ángulo de inclinación hacia la izquierda (en nuestro caso) entre los valores de 30° y 70° que es el ángulo de flexión estimado. Dispondremos del botón B para reiniciar el contador de pasos.

#### Creación del Script para el dispositivo **CyberPi**

Vamos a establecer dos hilos o ejecuciones en la acción de reinicio del dispositivo, lo cual permite nuestro lenguaje de programación **mBlock**.

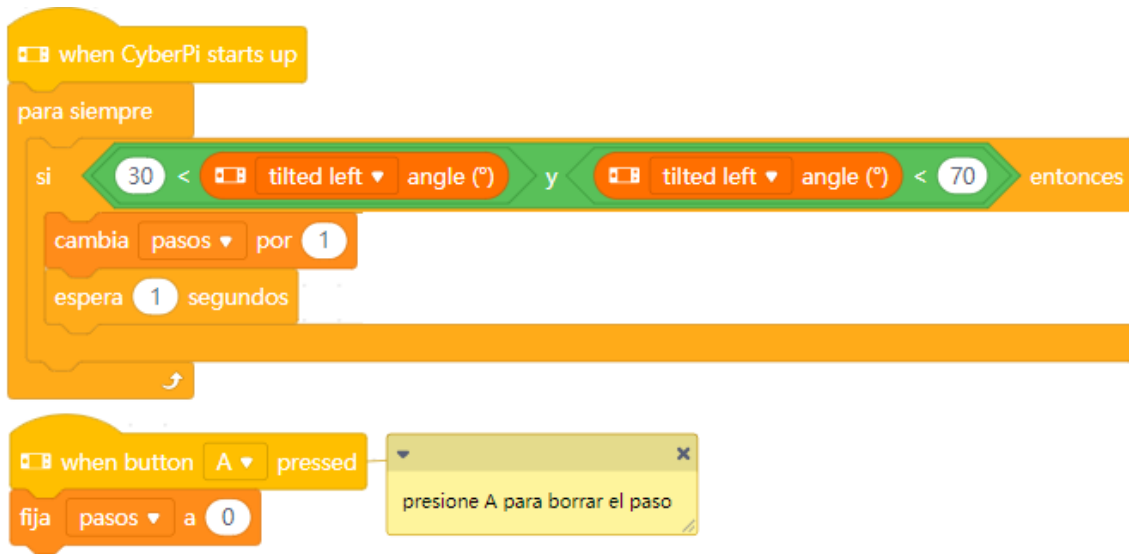
Lo que hacemos es, para empezar, usar el bloque de función “**when CyberPi start up**”. Borramos pantalla y forzamos su posición (vertical) con el bloque “**screen towards left(0°)**” y ponemos a cero la variable “*pasos*” “**fija pasos a 0**”



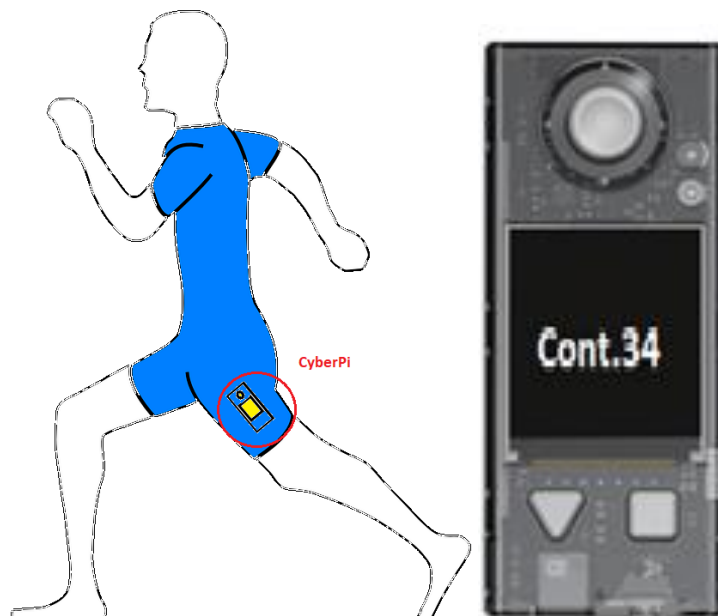
Seguidamente se procede a colocar un bloque “para siempre” que permanentemente estará escribiendo en la pantalla mediante el bloque “show” el texto que se forma al unir la palabra “**Cont.**” Y “*pasos*”\*2”

Por otra parte lo que hacemos, también al iniciar el dispositivo **CyberPi** se entra en un “**para siempre**” (bucle de repetición) en cuyo interior se lee la condición de un condicional “**si**” que establece el valor del ángulo entre 30° y 70° a la vez que dentro del condicional lo que hacemos es incrementar la variable “pasos”. En realidad se trata de detectar y contar los pasos. Se coloca un retardo, en nuestro caso de 1 seg “**espera 1 segundos**”, con el fin de que no se produzcan rebotes o dobles lecturas en el bucle.

Finalmente colocamos la función de puesta a cero del contador mediante el pulsador A que se realiza usando los bloques “**when button A pressed**” y el bloque “**fija pasos a 0**”



Seguidamente se muestra una imagen con la aplicación ya realizada e instalada.

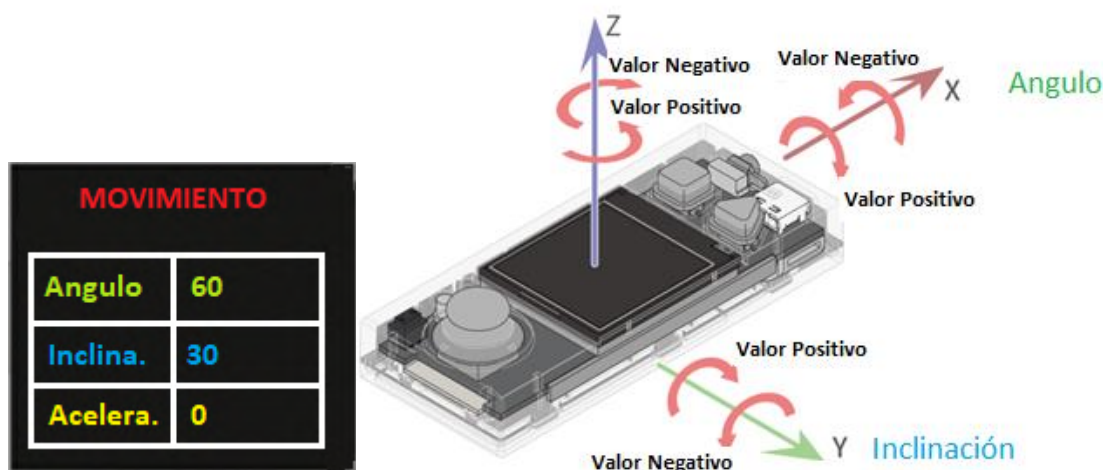


### 8.1.4. Detector de Movimiento

En esta aplicación vamos a seguir estudiando las posibilidades del acelerómetro que incluye **CyberPi**.

#### Funcionamiento

Queremos realizar la presentación de los datos en una tabla en la que podremos leer el “ángulo”, la “inclinación” y la “aceleración” o “fuerza de sacudida”

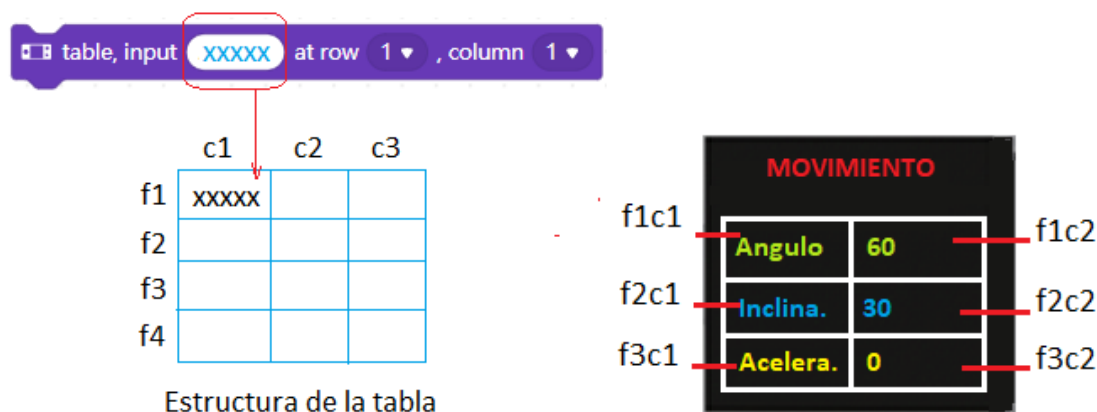


Se muestra en la figura el aspecto de la pantalla de **CyberPi** con los datos de las medidas que recogemos.

#### Diseño del Script del dispositivo

Para realizar la aplicación lo haremos en modo de programación Off-line, es decir en modo “**cargar**” en relación al dispositivo.

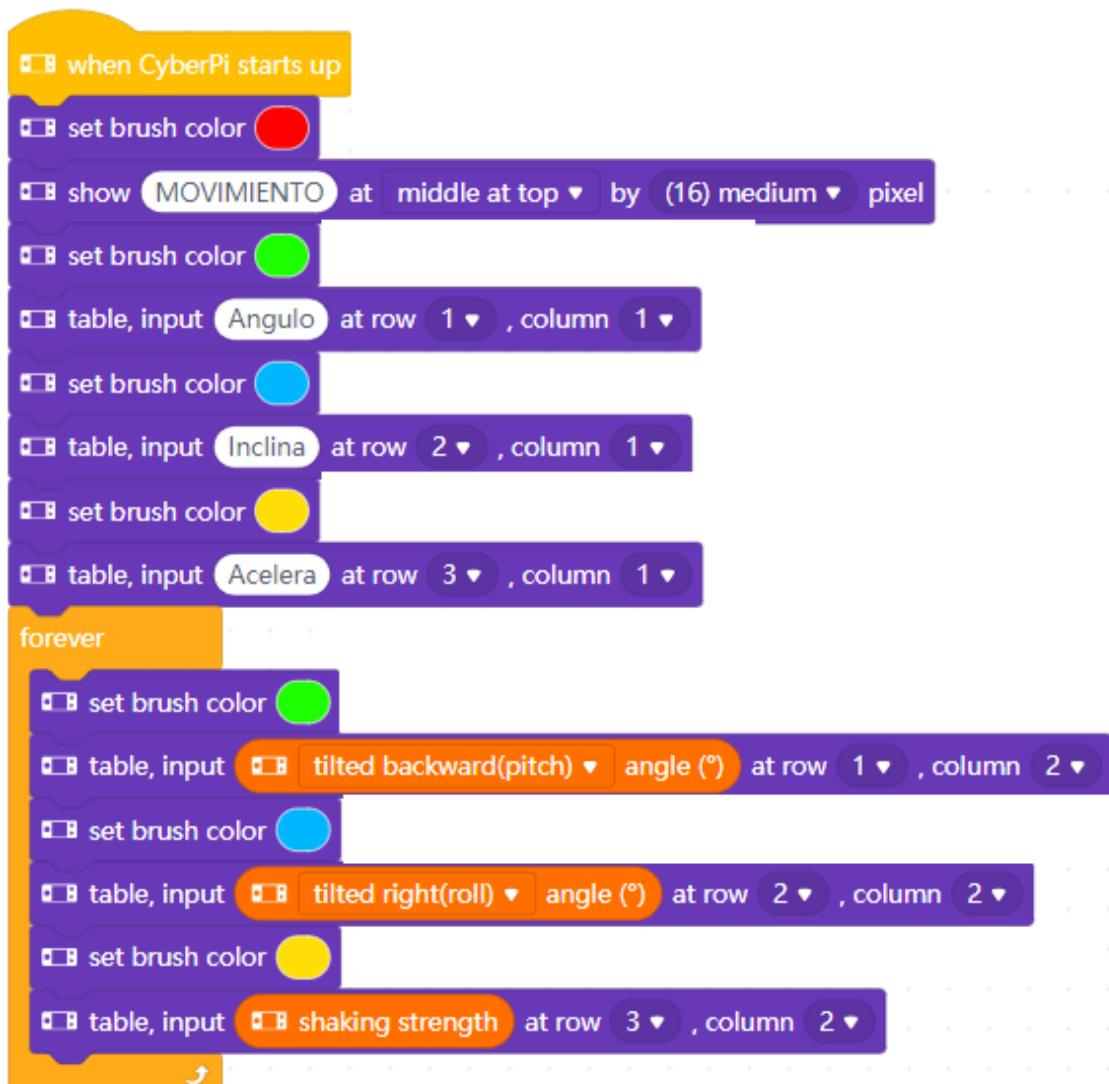
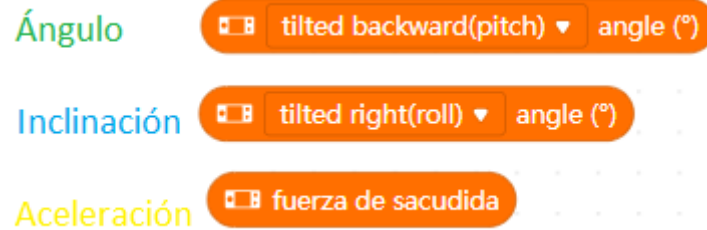
Lo que hacemos es, para empezar, usar el bloque de función “**when CyberPi start up**”. Seguidamente realizamos la tabla a mostrar con las etiquetas de las variables. Para escribir en forma tabulada en el display de **CyberPi** lo hacemos usando las instrucciones “**table, input xxxxx at row x , column x**”



Los colores del texto se definen con la instrucción “**set brush color..**”

A continuación en el bucle de repetición se colocan los valores de las variables a mostrar mediante los comentados bloques de mostrar en texto en forma tabulada así como los colores de cada texto.

Las variables se obtienen de la librería “**Sensores**”



### 8.1.5. Detector de Sonido

Siguiendo con los sensores incluidos en **CyberPi** con este ejemplo vamos a detectar un determinado nivel de sonido.

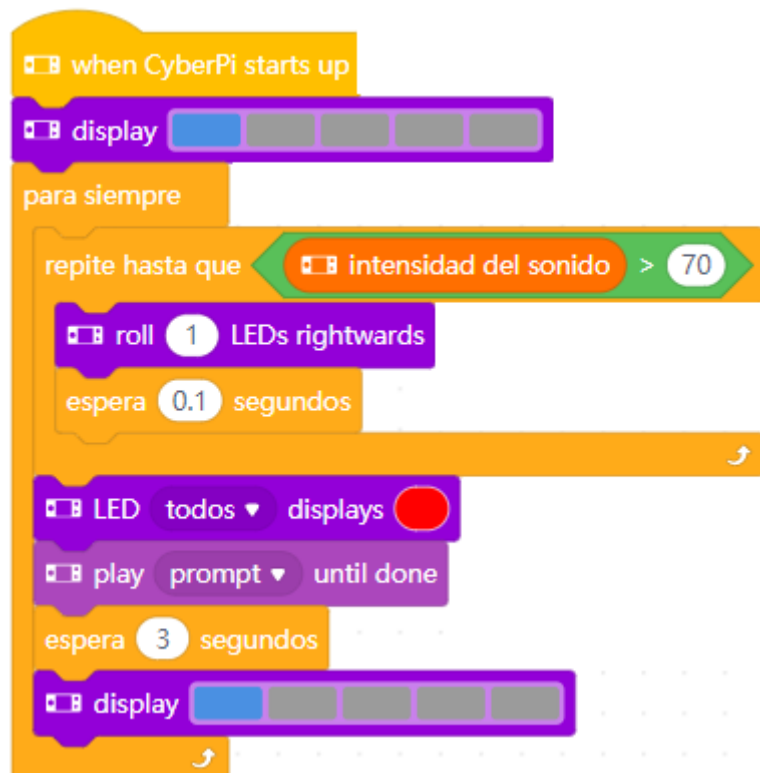
#### Funcionamiento.

Se trata, en este caso, de detectar un nivel determinado de sonido, **loudness (nivel sonido)** < **70**, y visualizarlo mediante el encendido intermitente de los leds de la barra de LEDs de **CyberPi**.

#### Diseño del script del dispositivo

En reposo, cuando el nivel de sonido es menor de 70, se mantiene en movimiento el encendido de los 5 LEDs de **CyberPi** en modo de desplazamiento “**roll x LEDs rightwards**” con un tiempo de 0.1 seg.

Si se supera el nivel, saldremos del bucle “**repite hasta que...**” y se activan los 5 LEDs a la vez que se emite un sonido “**play prompt until done**” durante 3 segundos “**espera 3 segundos**”. Después de este tiempo de encendido en rojo de los LEDs se vuelve a mostrar el movimiento de los LEDs en color azul



### 8.1.6. Funciones Trigonómicas

Una de las capacidades que tiene **CyberPi** es la de poder realizar la representación gráfica de una función matemática en su pantalla. En este ejemplo lo vamos a ver.

#### Descripción de la práctica

Se trata de representar la función seno por triplicado, es decir tres funciones con tres colores distintos y desfasadas entre ellas, es decir que la primera comenzara con el valor angular de “0”, la segunda de “90” y la tercera de “180”. Lo hacemos así para que no se superpongan las tres senoides.

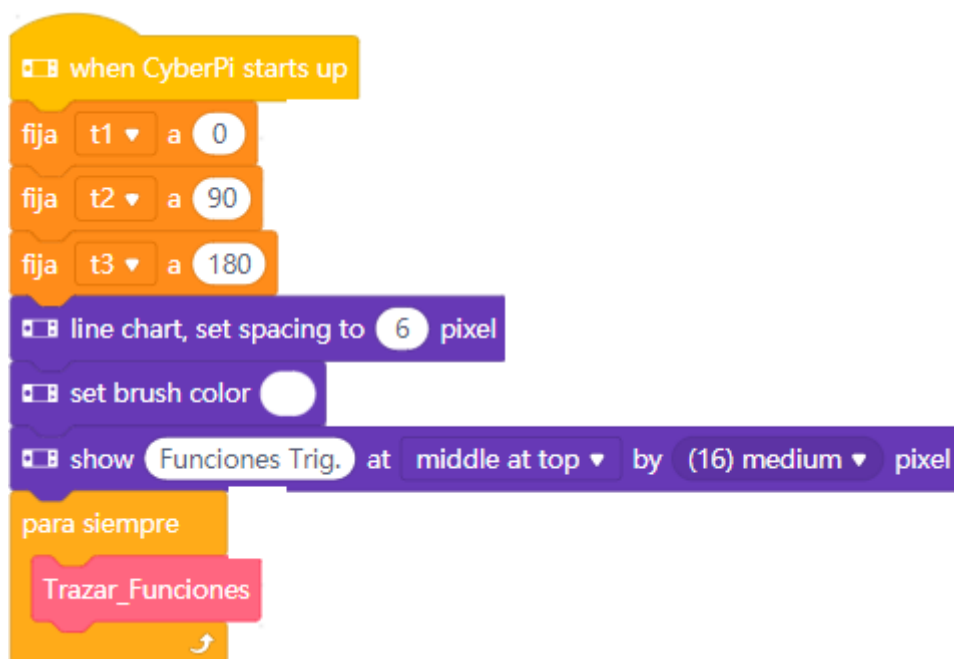
Crearemos tres variables t1, t2 y t3.

Para el trazado se realizará una función nueva llamada “**Trazar\_Funciones**” que se encargará de ello.

#### Script del dispositivo CyberPi

Para comenzar lo hacemos como siempre con el bloque “**when CyberPi starts up**” y seguidamente podemos los valores de las variables **t1=0**, **t2=90** y **t3=180** (recordemos que estas variables representan el valor del ángulo en cada función seno).

Para realizar el trazado en la pantalla fijaremos el espacio entre pixels “**line chart, set spacing to 6 pixel**” y el color de escritura “**set brush color**”, Seguidamente colocaremos una etiqueta de texto en la pantalla con el texto “Funciones Trig,” en el medio de la línea superior de la pantalla mediante el bloque “**show Funciones Trig, at middle at top by (16) médium pixel**”.



Seguidamente entramos en el bucle de ejecución mediante el bloque “**para siempre**”

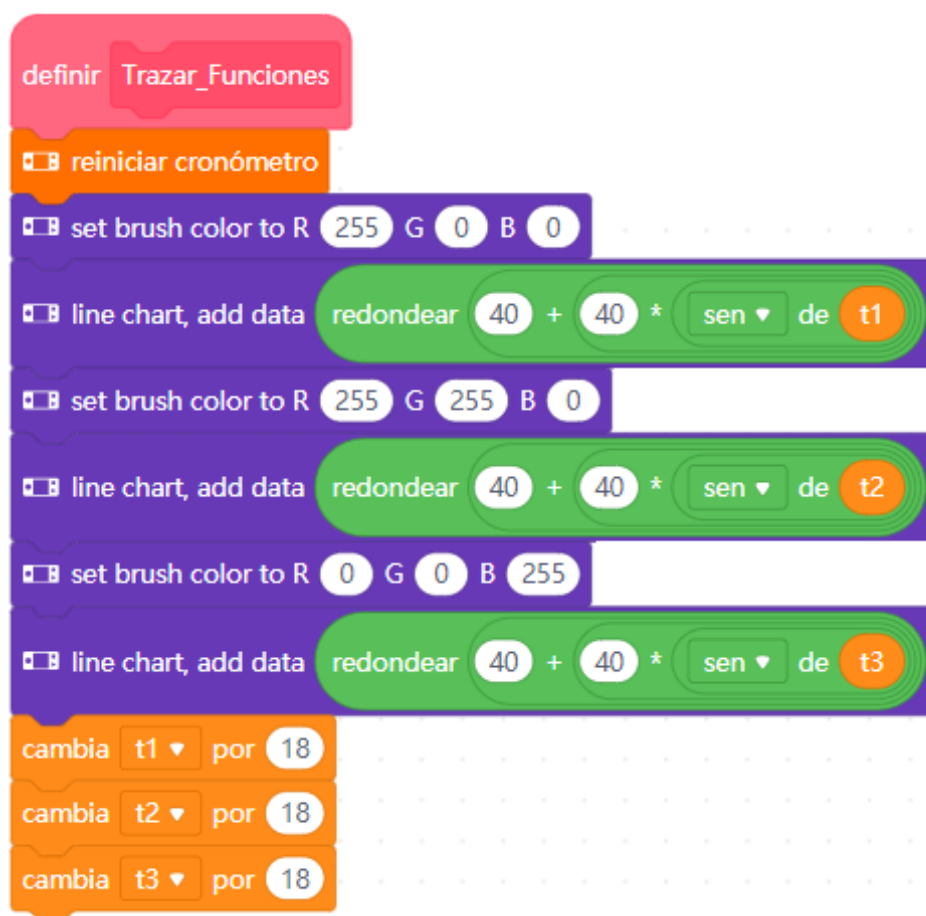
La función “**Trazar\_Funciones**” se crea mediante la función “**Crear un bloque**” de la librería “**Mis Bloques**”

Esta función lo que hace es utilizarla función “**line chart, add data**” que lo que permite es trazar una punto en la posición indicada por el “dato”, El valor del dato puede oscilar entre 0 y 100. En nuestro caso el dato se obtiene mediante la función

**valorredondeado ( $40+40*\text{sen}(t)$ )**

A cada función le ponemos un color distinto:

Función 1	Rojo RGB(255,0,0)
Función 2	Amarillo RGB(255,255,0)
Función 3	Azul RGB(0,0,255)



Vemos que los incrementos de los valores del ángulo son de 18° y la separación entre señales es de 6 pixel tal como hemos definido.

### Funcionamiento

En la siguiente imagen vemos una representación de la aplicación funcionando.



### 8.1.7. Grabación y Reproducción de sonidos

Grabar y reproducir sonidos en una interesante calidad para nuestro dispositivo **CyberPi**. En este ejemplo vamos a implementar un sencillo sistema de grabación y reproducción de sonidos.

#### Funcionamiento

Nuestro dispositivo se va a convertir en una grabadora reproductora de sonidos. Las funciones que vamos a implantar son las siguientes:

1. Grabar un sonido durante un tiempo.
2. Reproducir el sonido grabado
3. Aumentar o disminuir el volumen de reproducción del sonido
4. Aumentar o disminuir la velocidad de reproducción

Todas estas funciones las encontraremos en la librería “Sonido” propia del dispositivo **CyberPi**.

Uso de los botones y Joystick para el control de la aplicación.

- Para iniciar la grabación pulsaremos el botón del Joystick en la posición central



- Para detener la grabación pulsaremos en el Botón A
- Para reproducir la grabación pulsaremos en el Botón B una vez que se haya realizado la grabación.
- Para aumentar el volumen de la reproducción pulsaremos el Joystick hacia la derecha



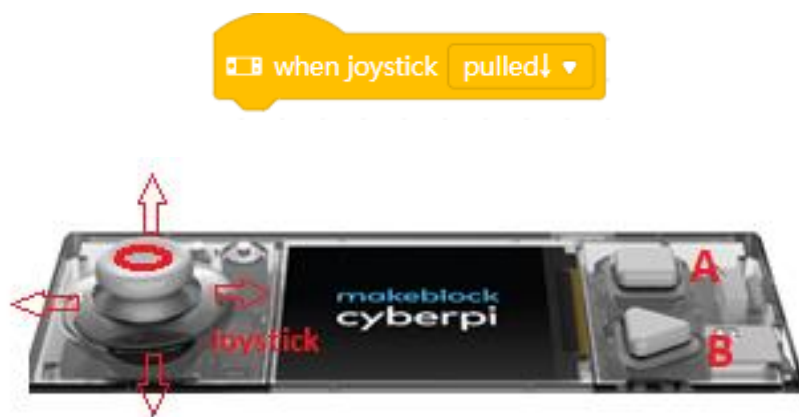
- Para disminuir el volumen de la reproducción pulsaremos el Joystick hacia la izquierda



- Para aumentar la velocidad de reproducción pulsaremos el Joystick hacia arriba.



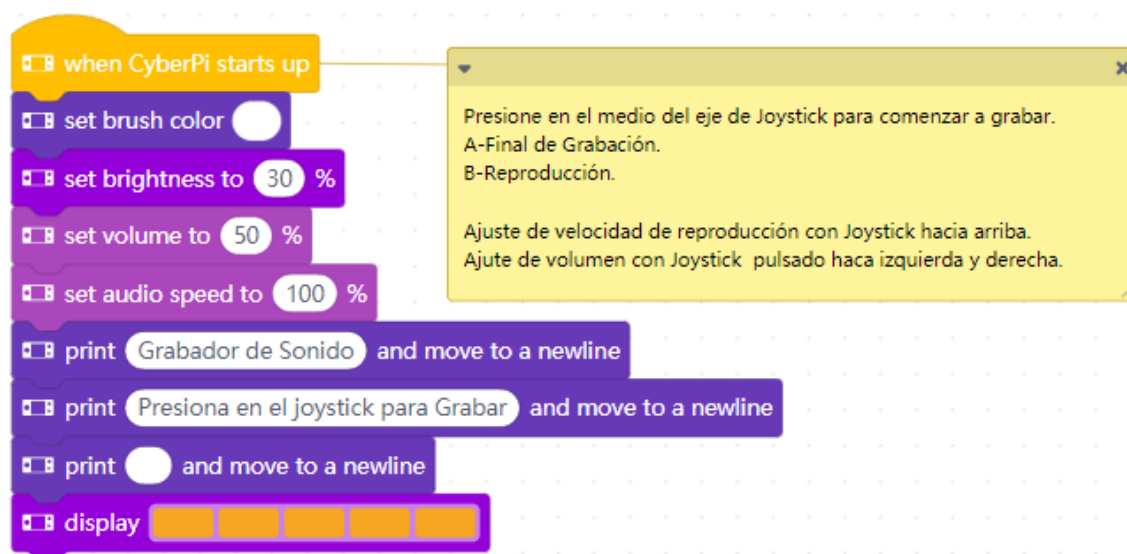
- Para disminuir la velocidad de reproducción pulsaremos el Joystick hacia abajo.



## Realización de la programación

### Inicio del Programa

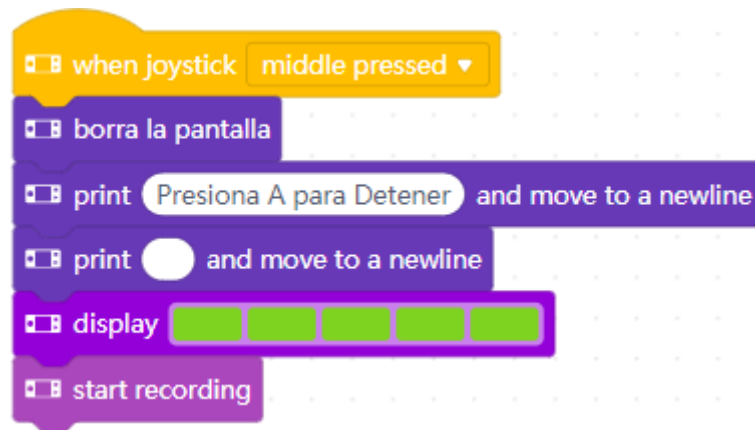
Cuando se inicia el programa se disponen los parámetros color, brillo, volumen y velocidad en los valores que se indican en la imagen. Seguidamente se escribe en el display el texto que se indica: “Grabador Reproductor” y “Presiona en el Joystick para Grabar”. Se imprime una línea en blanco y se ponen los 5 **LEDs** de color en color **naranja**



### Inicia Grabación

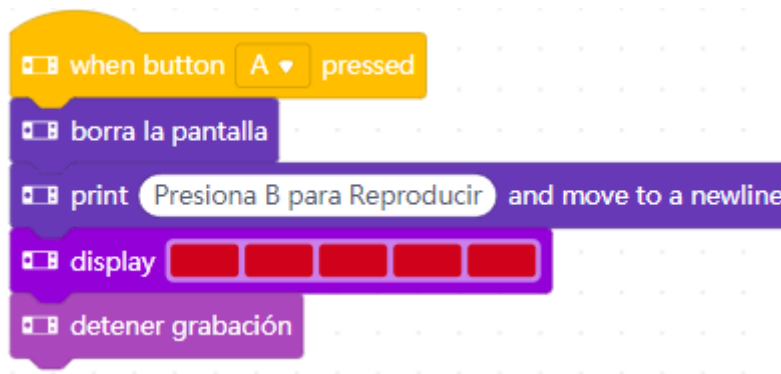
Para iniciar la grabación ya hemos dicho que se hará cuando se pulse en el medio al Joystick, En ese caso el Script sería el de la siguiente imagen.

Se borra pantalla y se escribe “Presiona A para Detener” más una línea en blanco y los **LEDs** se ponen en **verde**. En estos momentos ya se está grabando “**start recording**” quedando pendiente de pulsar el botón A cuando hemos terminado de hablar, para proceder a la reproducción.



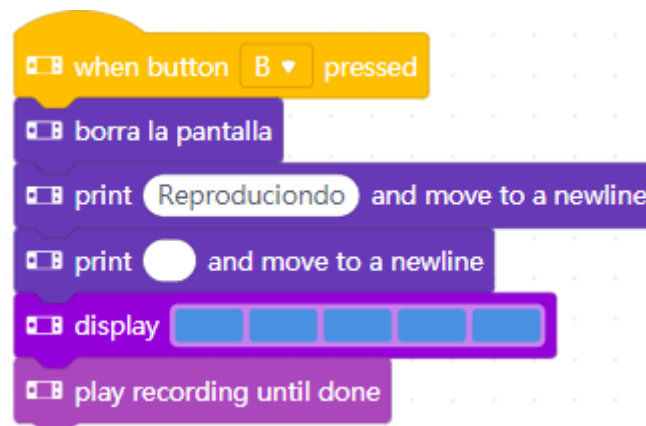
### Detiene Grabación

Cuando pulsamos en el Botón A se procede a entrar en el Script de detención de la reproducción. Se ponen los **LEDs** en color **Rojo** y se ordena la detención de la grabación “**detener grabación**”



### Reproduce grabación

Cuando pulsemos **B** se realizara la reproducción del sonido grabado, escribimos en pantalla “*Reproduciendo*”, imprimimos una línea en blanco, colocamos los LEDs en color Azul y damos la orden “**play recording until done**”



## Incrementa Velocidad de reproducción

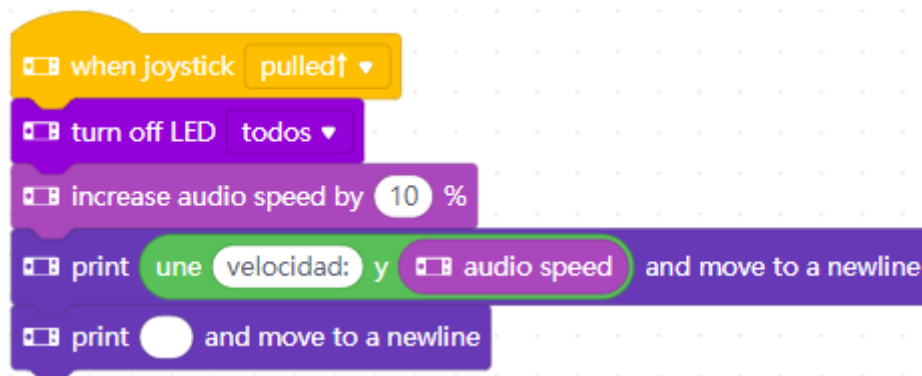
Para el incremento de la velocidad hemos acordado que hay que mover hacia arriba el Joystick. En ese caso se ejecutará el siguiente Script.

Apagamos todos los LEDs “**turn off LED todos**”. Usamos el bloque “**increase audio speed by xx %**” y después mostramos en la pantalla el valor de la velocidad “**audio speed**” colocando el bloque de unión de cadenas (librería Operadores) para poner la “etiqueta”

**Velocidad: + variable (audio speed).**



Finalmente imprimimos una línea en blanco



## Decrementa Velocidad de Reproducción

Para el decremento de la velocidad hemos acordado que hay que mover hacia abajo el Joystick. En ese caso se ejecutará el siguiente Script.

Apagamos todos los LEDs “**turn off LED todos**”. Usamos el bloque “**increase audio speed by - xx %**” (observar que el parámetro se pone negativo porque es un decremento). Se ha colocado un condicional para que, en el caso de que la velocidad sea menor que 40 no permita que baje más.

Después mostramos en la pantalla el valor de la velocidad “**audio speed**” colocando el bloque de unión de cadenas (librería Operadores) para poner la “etiqueta”

**Velocidad: + variable (audio speed).**

Finalmente imprimimos una línea en blanco.

En la siguiente imagen vemos completo el Script



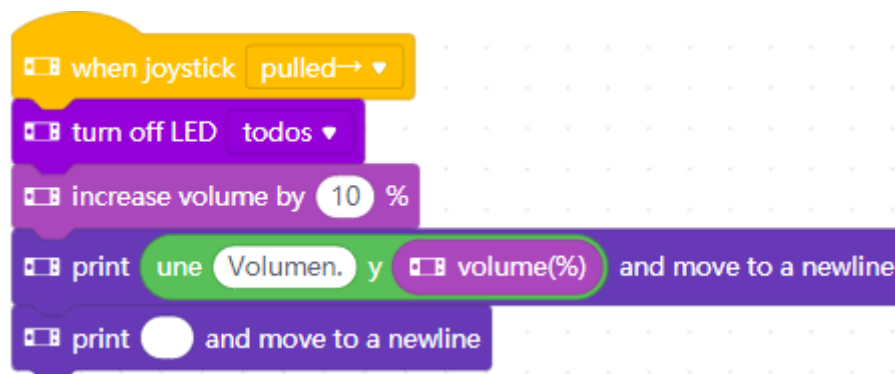
### Incrementa Volumen de Reproducción

Para el incremento del Volumen hemos acordado que hay que mover hacia la derecha el Joystick. En ese caso se ejecutará el siguiente Script.

Apagamos todos los LEDs “**turn off LED todos**”. Usamos el bloque “**increase volumen by xx %**” y después mostramos en la pantalla el valor del volumen que se recoge con el bloque “**volumen(%)**” y colocamos el bloque de unión de cadenas (librería Operadores) para imprimir la “etiqueta”

**Volumen:** + variable (**volumen(%)**)

Después imprimimos una línea en blanco



### Decrementa Volumen de Reproducción

Para el decremento el volumen hemos acordado que hay que mover hacia la izquierda el Joystick. En ese caso se ejecutará el siguiente Script.

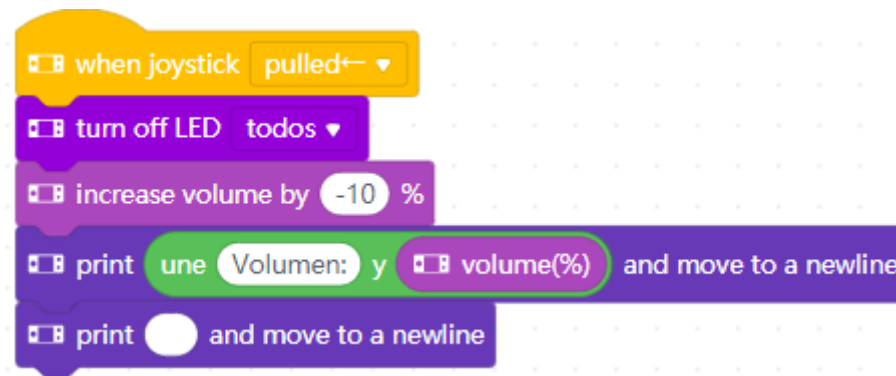
Apagamos todos los LEDs “**turn off LED todos**”. Usamos el bloque “**increase volumen by -xx %**” (observar que el parámetro se pone negativo porque es un decremento).

Después mostramos en la pantalla el valor del volumen “**volume(%)**” colocando el bloque de unión de cadenas (librería Operadores) para poner la “etiqueta”

**Volumen:** + variable (**volume(%)**)

Finalmente imprimimos una línea en blanco.

En la siguiente imagen vemos completo el Script



### 8.1.8. Simulación de Sirena

El sistema de sonido de **CyberPi**, mediante su librería propia de “Sonido” permite emitir tonos de una frecuencia y duración establecida. Vamos a realizar una aplicación usando esta posibilidad.

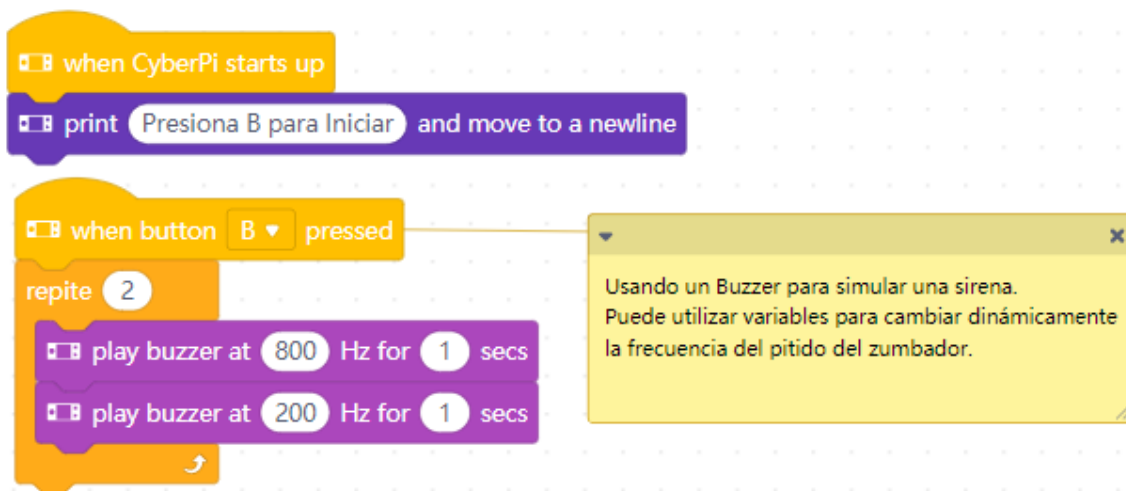
#### Funcionamiento

Se trata de imitar el sonido de una sirena “bitonal” cuando pulsemos el botón B

Deseamos que las dos frecuencias sean de 800 y 200 Hz y de una duración de 1 seg. y además queremos que esto se repita 2 veces.

#### Script

Este sería el Script de esta simulación.



#### Funcionamiento

Basta presionar sobre el botón B para oír dos veces el bitono de 800/600 Hz



### 8.1.9. Toca Canción

**CyberPi** puede tocar notas pertenecientes a la escala musical, En este ejemplo se prueba esta capacidad

#### Funcionamiento

Se trata de que al presionar el botón B se ejecuta una secuencia de notas como la de la figura haciendo uso del bloque de la librería de Sonido “**play note xx for xx beat**”. El parámetro “beat” se refiere a la duración de la nota

La codificación de las notas se realiza de acuerdo a la tabla que ya se comentó en la parte anterior de este libro cuando explicamos los bloques de función de la librería “**Sonido**”

Click con boton derecho para ver Ayuda del bloque seleccionado.  
El standard MIDI se usa para ver las series numericas de notas musicales, en este ejemplo se encadenan varias notas que se corresponden con la secuencia siguiente:

Nota - Numero.

1	C4 60.
2	D4 62.
3	E4 64.
4	F4 65.
5	G4 67.
6	A4 69.
7	B4 71.

## 8.1.10. Traductor de Bolsillo

La posibilidad de reconocimiento de voz y traducción por parte de **CyberPi** nos permite construir un pequeño traductor de mano con el que poder aprender un idioma.

### Funcionamiento

Se trata de realizar la traducción de una palabra o frase que pronunciaremos en este caso en español durante un tiempo dado. El sistema reconoce lo que ha escuchado usando los servicios cognitivos de la librería IA así como los de comunicación **IoT** con el servidor de **mBlok** al que debemos estar dados de alta y conectados, tal como se ha explicado en la primera parte de este libro.

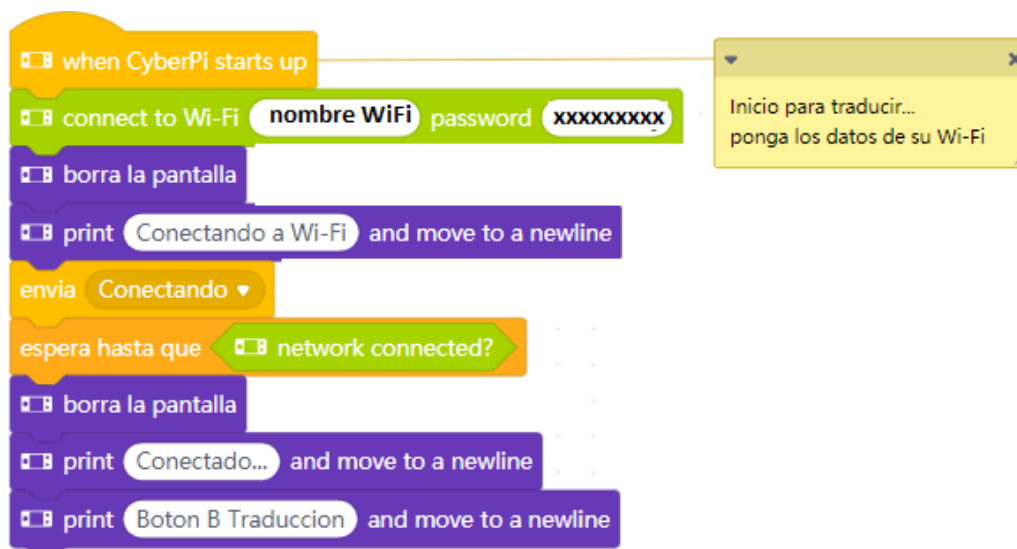
El resultado de la traducción al Inglés de lo que hemos dicho se mostrará junto con lo que hemos dicho en español en la pantalla de **CyberPi**. Así mismo podremos escuchar la traducción en formato voz gracias a la opción “**Speak**” de la librería IA.

### Descripción de los Scripts asociados al dispositivo

#### Arranque

En esta primera etapa se trata de establecer conexión a través de nuestra Wi-Fi con el servidor de **mBlock**, para ello usamos el bloque de la librería **AI** o **IoT** llamado “**connect to Wi-Fi** <nombre> **password** <xxxxxx>”. Imprimimos “*Conectando a Wi-Fi*” y enviamos el mensaje “*Conectando*” que lo que hace es ejecutar un pequeño Script que nos mostrará a través de los LEDs de **CyberPi** que se está realizando la conexión. Esperamos a que se establezca la conexión mediante la instrucción “**espera hasta que** <**network connected?**>.” (*network connected?* es una variable del sistema que se pone en 1 cuando la conexión se ha establecido) y, seguidamente, escribimos en pantalla:

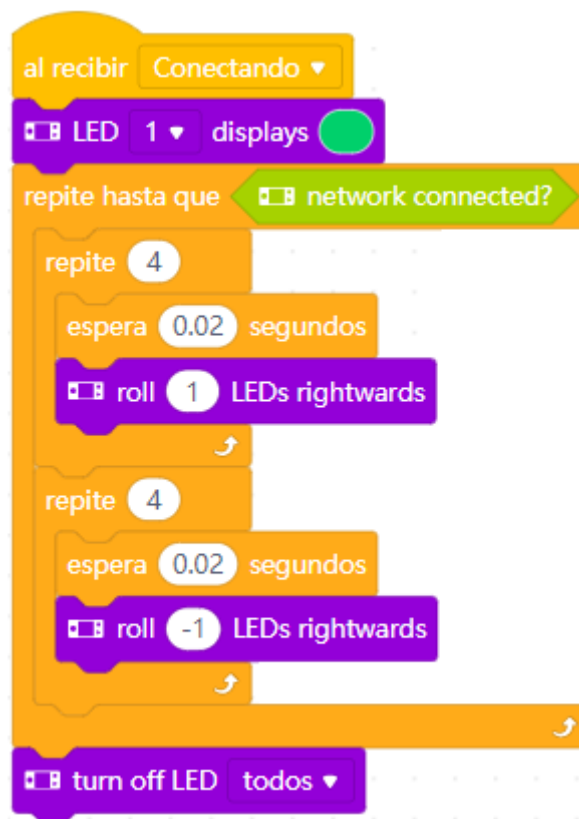
*Conectado,,,*  
*Botón B Traducción*



Con esto indicamos que pulsando el botón B se inicia el proceso de recogida de nuestra voz y posterior traducción.

### Función Conectado

Esta función sirve para indicar mediante los LEDs de **CyberPi** que estamos realizando la conexión. Se trata de que un LED de color azul se vaya iluminando de manera cíclica

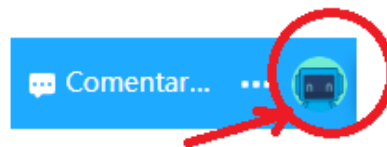


### Función de traducción

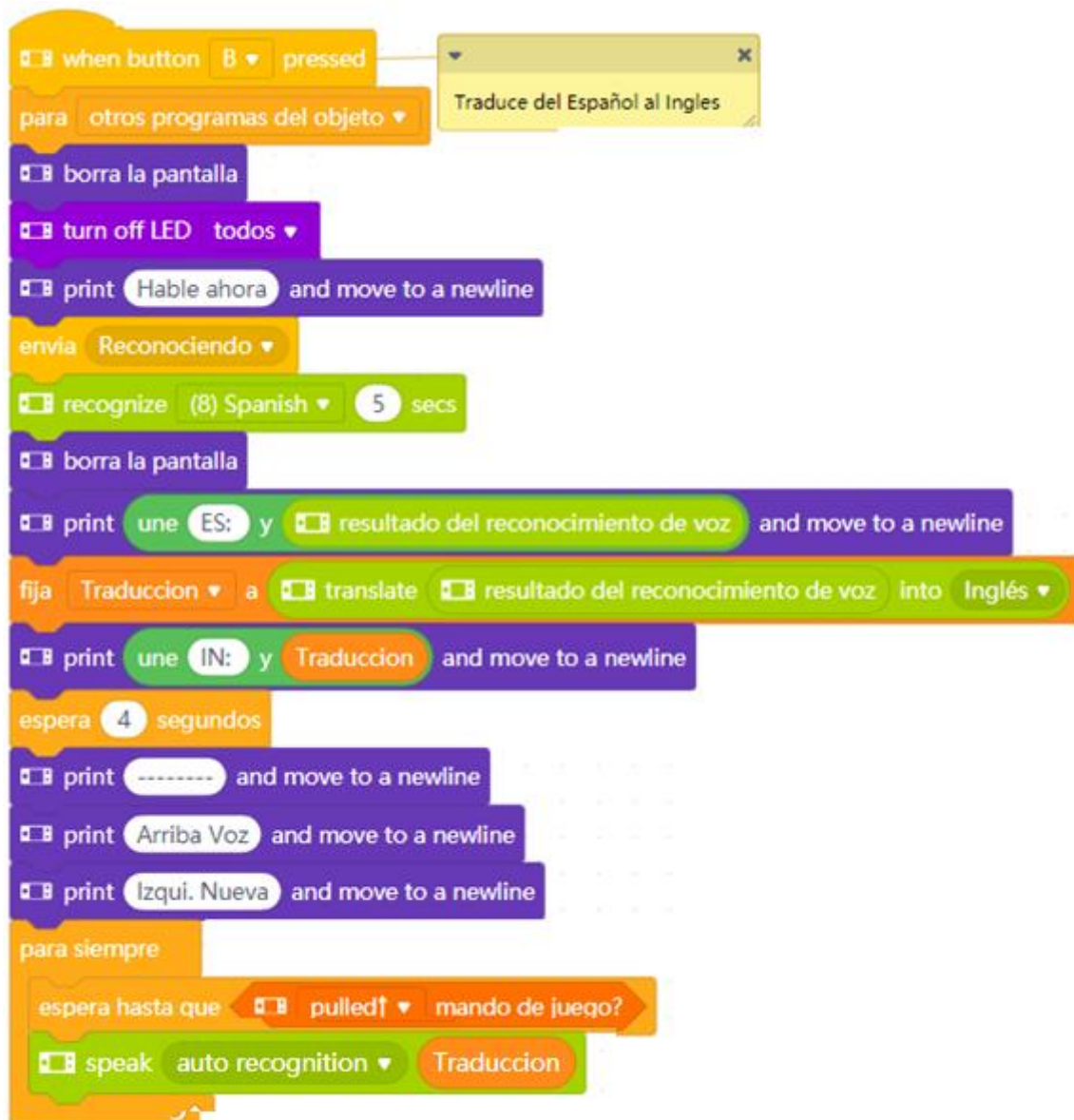
Para este bloque tenemos que crear la variable “**Traducción**”

Esta acción se ejecuta cuando pulsamos el botón B. LO que se hace es detener todos los programas que se estén ejecutando dentro de la aplicación, borrar la pantalla, apagar los LEDs y escribir en la pantalla un mensaje que nos invita a hablar para recoger nuestro mensaje “*Hable ahora*”. A continuación envía el mensaje “**Reconocimiento**” para que los LEDs nos indiquen que estamos en la fase de reconocimiento de nuestra voz.

La función principal de este Script es la que viene a continuación “**recognize (8)Spanish 5 secs**” que ordena escuchar el mensaje y lo envía via Wi-Fi al Servidor de **mBlock** (es importante que tengamos abierta la conexión con este servidor tal como se ha explicado en este libro anteriormente).



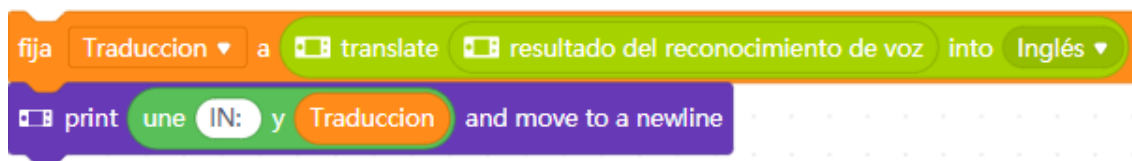
Establecida Conexión  
con el servidor mBlock



Seguidamente borramos la pantalla y escribimos en ella el **“resultado del reconocimiento”**.



Traducimos al Ingles el texto reconocido mediante el bloque **“translate”** de la librería **“AI”** propia de **CyberPi** y lo escribimos a continuación en la pantalla



Seguidamente escribimos en la pantalla los mensajes

-----

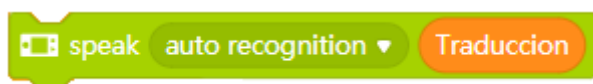
**Arriba Voz**

Pulsando hacia arriba con el Joystick escuchamos la traducción

**Izquierda Nueva**

Pulsando hacia la izquierda el Joystick reiniciamos el ciclo, para una nueva traducción.

Finalmente terminamos con un bucle “**para siempre**” que nos mantiene en el hasta que pulsamos hacia arriba con el Joystick, en cuyo caso **CyberPi** nos pasa a voz el texto traducido.



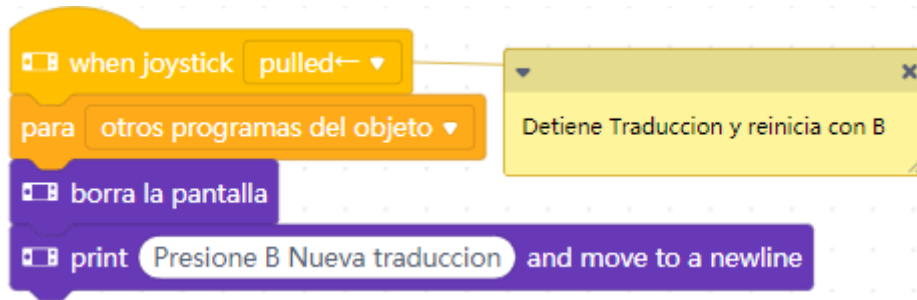
### Indicación luminosa de reconocimiento de voz

Se trata de poner los LEDs a parpadear en forma de movimiento mientras se está reconociendo la voz.



### Iniciar Traducción.

El siguiente Script se activa al pulsar a la izquierda el Joystick y sirve para iniciar una nueva secuencia de traducción. Se trata de para todas las acciones emprendidas en la aplicación, borrar la pantalla e imprimir en ella el mensaje “*Presione B Nueva traduccion*”



### Prueba de funcionamiento



1

Al encender **CyberPi** aparecerá el estado de conexión y los LEDs parpadean en azul moviéndose



2

Una vez establecida la conexión Wi-Fi el sistema espera a que pulsemos B para ponerse en escucha.



3

Cuando aparece este mensaje podemos empezar a decir la frase. Se enciende un LED azul que se mueve



4

Cuando se reconoce la frase se traduce y se muestra en la pantalla.

Nos invita a pulsar hacia arriba el Joystick para escuchar la traducción o a la Izquierda para iniciar otra nueva traducción

### 8.1.11. Control de motor con la voz

Los recursos de **IoT** disponibles en el dispositivo **CyberPi** nos permiten numerosas aplicaciones de control a través de la Web. En esta ocasión vamos a realizar el control de la velocidad de un motor mediante la voz.

#### Funcionamiento.

Trabajaremos en modo off-line es decir en forma de “cargar”.

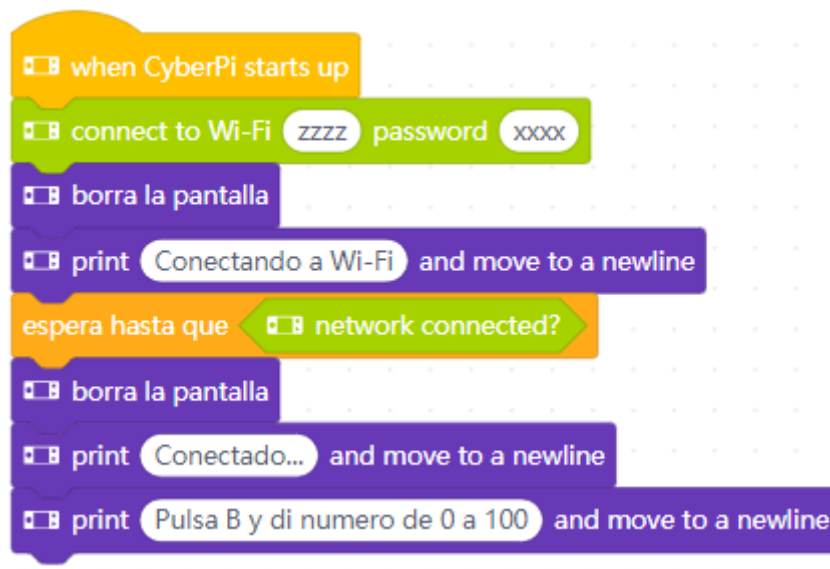
Se trata de que al pulsar el botón **B CyberPi** se ponga a la escucha para recoger el número que le digamos que será la velocidad a la que queremos que gire el motor que hemos conectado en la salida M1 del Shell de Bateria y extensiones. El sistema envía via Wi-Fi la locución, mediante el bloque “**recognize (8)spanish 5 secs**” que está activo durante 5 segundos. El sistema (desde la nube) envía el valor reconocido y este se procesa mediante un bloque personalizado que llamaremos “Control\_motor” que gobernara el motor imponiéndole la velocidad de giro correspondiente al valor recogido.

#### Construcción del Script para el Dispositivo **CyberPi**.

El inicio de la aplicación lo que debe hacer es notificar los datos de nuestra Wi-fi mediante el bloque “**connect to Wi-Fi <nombre wi-fi> password <xxxx>**”. Borramos pantalla y escribimos el texto “Conectando a Wi-Fi” mediante un bloque “**print Conectando a Wi-Fi and move to a newline**”.

Crearemos una variable llamada “valor” que será la recoja el valor numérico de la velocidad.

Cuando se establece la conexión, esperando hasta que se manda la señal de “**network connecte?**” se borra pantalla y se escribe “Conectado...” (ya que estamos conectados a la Wi-Fi, y se escribe en una nueva línea “Pulsa B y di numero de 0 a 100”



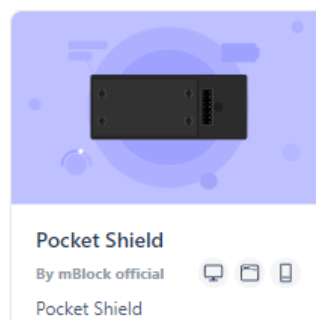
Seguidamente pasamos a montar el Script asociado al evento **“When buton B pressed”**. Se detienen, por si estuvieran ejecutándose otros bloque, todos los programas, con el bloque **“para otros programas del objeto”** de la librería de **“Control”** y se dispone la pantalla para mostrar el mensaje **“Habla ahora”** que nos invita a decir con voz clara el valor numérico de la velocidad.

El proceso del valor pronunciado se recoge mediante el bloque de la librería AI **“recognize (8)Spanish 2 secs”**, se borra la pantalla y se fija la variable **“valor”** al resultado reconocido (que será un número), borramos pantalla y esperamos un tiempo **“espera 2 segundos”** para asegurarnos que el valor del reconocimiento nos llega. Finalmente usamos **“fija valor a resultado del reconocimiento”** para que la variable valor recoja el valor reconocido y seguidamente ejecutar el bloque que hemos de crear con la librería **“Mis Bloques”**



El bloque que creamos **“Control\_Motor”** lo que hace es recoger el valor de la variable **“valor”** y colocarlo como parámetro en el bloque de control del motor.

El motor lo conectaremos a os pines de la salida M1 del Shell de Batería y además debemos cargar la librería asociada a este hardware.



Esta librería nos ofrece el bloque de control de motores.



### IMPORTANTE.

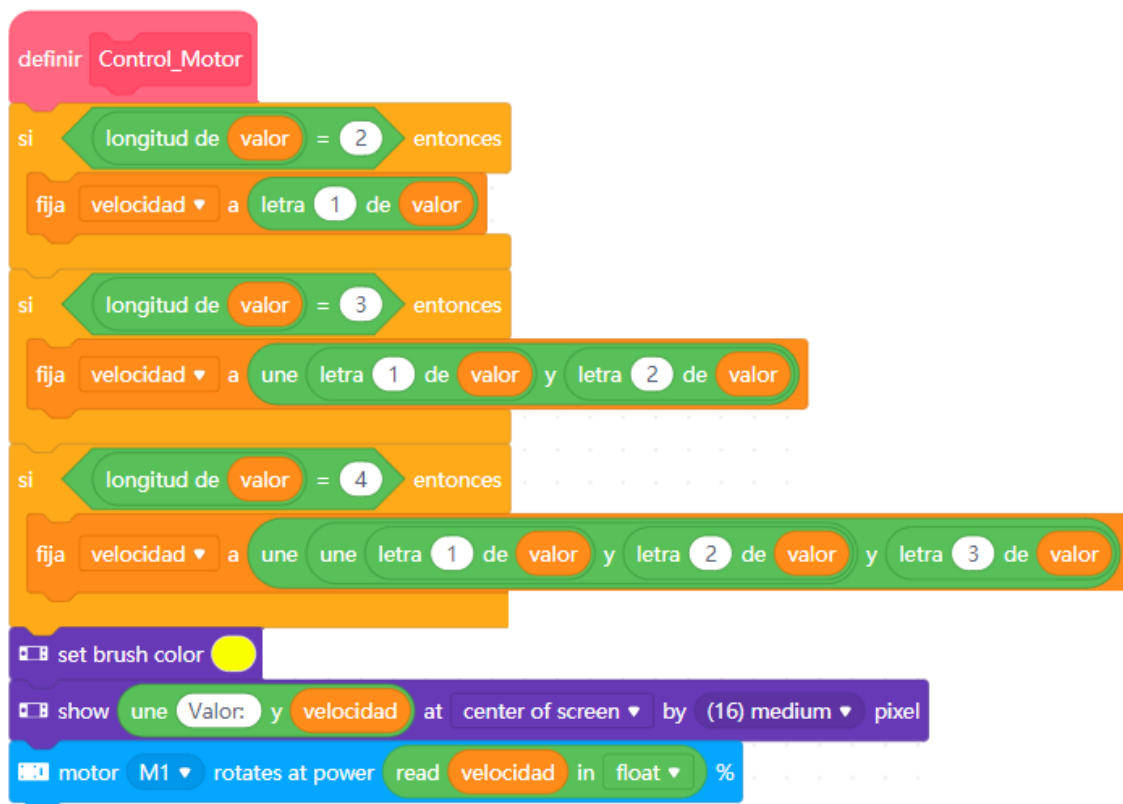
Cuando el sistema reconoce la palabra que indica el número, lo que haces es que añade un "." Al final y este punto debemos quitarlo porque si entregamos un valor numérico con un "." Al final no le sirve este parámetro como velocidad del motor.

Por ejemplo si decimos "65" el sistema nos devuelve reconocido: 65.  
Si le decimos "3" nos devuelve 3.

Para quitar el punto lo que hacemos es crear una nueva variable que llamaremos "velocidad" y de esta variable quitaremos el punto "." Para hacerlo hay que crear tres condicionales de filtrado, uno si el número es solo una unidad, otro para si son dos unidades y un tercer filtrado para cuando el número tiene 3 cifras como.

### Filtrado de la variable "valor" y su conversión en "velocidad"

Por ejemplo si el número es "8" el sistema devuelve "8." Y debemos tomar solo la letra número 1 es decir el "8" esta es la condición de **longitud de valor = 2**



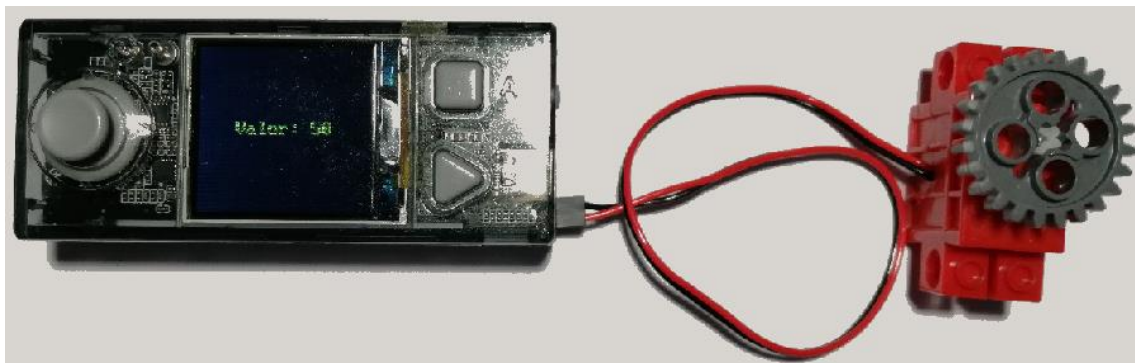
Por ejemplo si el número es "27" el sistema devuelve "27." Y debemos tomar solo las letra 1 y 2, es decir el "27" esta es la condición de **longitud de valor = 3**

Por ejemplo si el número es “**100**” el sistema devuelve “**100.**” Y debemos tomar solo las letra 1,2 y 3, es decir el “100” esta es la condición de **longitud de valor = 4**

Finalmente lo que hacemos es mostrar en la pantalla de **CyberPi** el valor de la velocidad en color “amarillo” con la opción “**show...**”

Terminamos el nuevo bloque con la orden de cambio de velocidad en el motor que ya hemos comentado.

Este sería el aspecto del montaje.

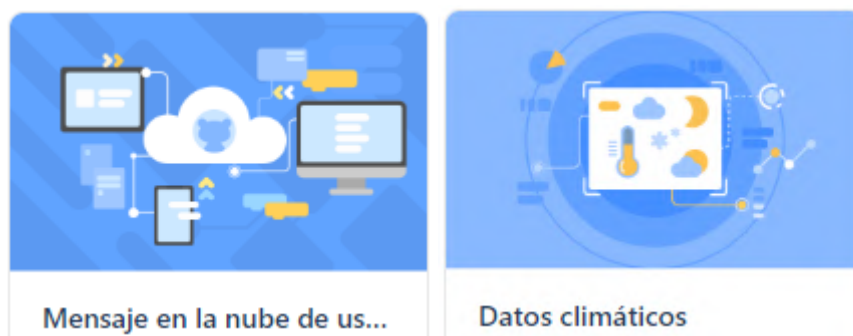


## 8.2. Proyectos IoT con CyberPi

### 8.2.1. Estación Meteorológica Remota

#### Descripción de la práctica.

Vamos a realizar una lectura de datos meteorológicos usando la Extensión “**Datos de Clima**” de **mBlock** disponiendo de un PC que cargará los datos desde la “nube” y los enviara por el método “**mensaje**” disponible en la Extensión “**Mensaje en la nube**” a un terminal remoto que en nuestro caso será la tarjeta **CyberPi** que podrá encontrarse en cualquier lugar.



Para la realización de esta aplicación debemos contar con que estamos conectados a través de nuestro acceso a los servicios de la nube de **mBlock**.

Nuestro diseño contará con una pantalla “escenario” en la que estaremos viendo los datos y desde la cual podremos:

- Enviar datos de modo manual (al pulsar un botón)
- Enviar datos cada cierto tiempo de manera automática (pulsando otro botón)

Fondo de pantalla



Escenario completo con los objetos



Los valores climáticos que vamos a leer y enviar serán:

- Temperatura Máxima
- Temperatura Mínima
- Humedad

De los envíos tendremos un piloto de señalización que parpadeará mientras enviamos datos al terminal **CyberPi**.

### Variables de la aplicación

Las variables que debemos crear en esta aplicación son:

<input type="checkbox"/> Automatico	<i>Automático</i> define el modo de envío Automático
<input type="checkbox"/> Ayuda	<i>Ayuda</i> indica si se muestra o no la imagen de ayuda
<input type="checkbox"/> envio	<i>envio</i> Indica si se envían o no los datos
<input checked="" type="checkbox"/> Humedad	Valor de la <i>humedad</i> (mostrándose en el escenario)
<input type="checkbox"/> Manual	Indica si enviamos en modo manual
<input checked="" type="checkbox"/> Temp. Max	Valor de la <i>Temperatura máxima</i> (mostrándose en el escenario)
<input checked="" type="checkbox"/> Temp. Min	Valor de la <i>Temperatura mínima</i> (mostrándose en el escenario)

## Definición de Objetos y Dispositivos de la aplicación

Los **Objetos** que tenemos que definir y programar con sus Scripts son los siguientes:



- Botón: “**Datos Manual**”
- Botón: “**Datos Automático**”
- LED: “**Indicador Manual**”
- LED: “**Indicador Automático**”
- LED: “**Indicador Transmisión**”
- Botón: “**Ayuda**”
- Texto: “**Texto Ayuda**”

## Disfraces de los Objetos

Los disfraces de estos botones y LEDs serán:

- Para los botones usaremos el disfraz de la librería llamado “**Empty Button1**” a los que les añadiremos el texto del botón
- Los LEDs los creamos con un círculo de color verde y de color negro que se identificarán con el estado **activo=verde** y **desactivado=negro**. En el caso del indicador de Transmisión (cuadrado) los colores serán **transmite=rojo no trasmite=negro**
- El Objeto “**Texto Ayuda**” será creado en el editor de disfraces o se importará como una imagen y será algo parecido a la siguiente imagen:



## Proceso de realización de la aplicación,

Describiremos a continuación el desarrollo de la aplicación.

1. Creación del escenario
2. Carga de las Extensiones necesarias: Desde Objetos debemos cargar las extensiones que hemos comentado al principio: “**Mensajes en la Nube**” y “**Datos Climáticos**”
3. Creación de los Objetos y diseño o selección de los disfraces de cada uno. No olvidar etiquetarlos con los nombres que hemos establecido.
4. Conexión de la unidad **CyberPi** al PC bien con un cable USB o bien mediante Bluetooth. Se trabajara en modo “**Cargar**”
5. Realización del **Script** del Dispositivo **CyberPi**

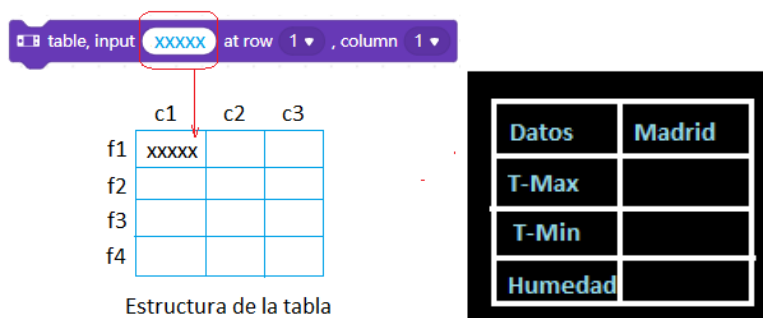
## Script del dispositivo CyberPi



Este script será el que se deposite en la unidad **CyberPi** y se encargará de establecer la conexión con el servidor de datos de **mBlock** para poder recoger de él los datos meteorológicos que hemos mencionado.

La pantalla de **CyberPy** mostrara los datos haciendo uso de las instrucciones de entrada de datos a tabla “**table input xxx at row xx, column xx**”. Los colores se seleccionan con la función “**set brush color xx**”

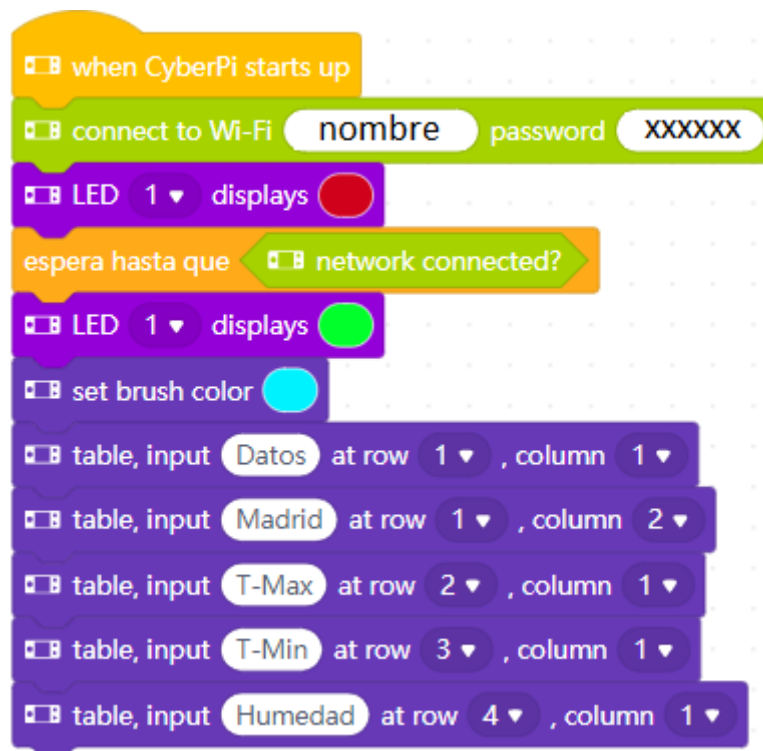
Señalizaremos el estado de conexión con el LED 1 de CyberPi de tal forma que cuando se está estableciendo la comunicación WiFi el LED se pone en color rojo y una vez establecida se pone en color verde. El bloque “**espera hasta que <network connected?>**” lo que hace es detener la ejecución hasta que se establezca la conexión, en cuyo caso el valor *network connected=1*

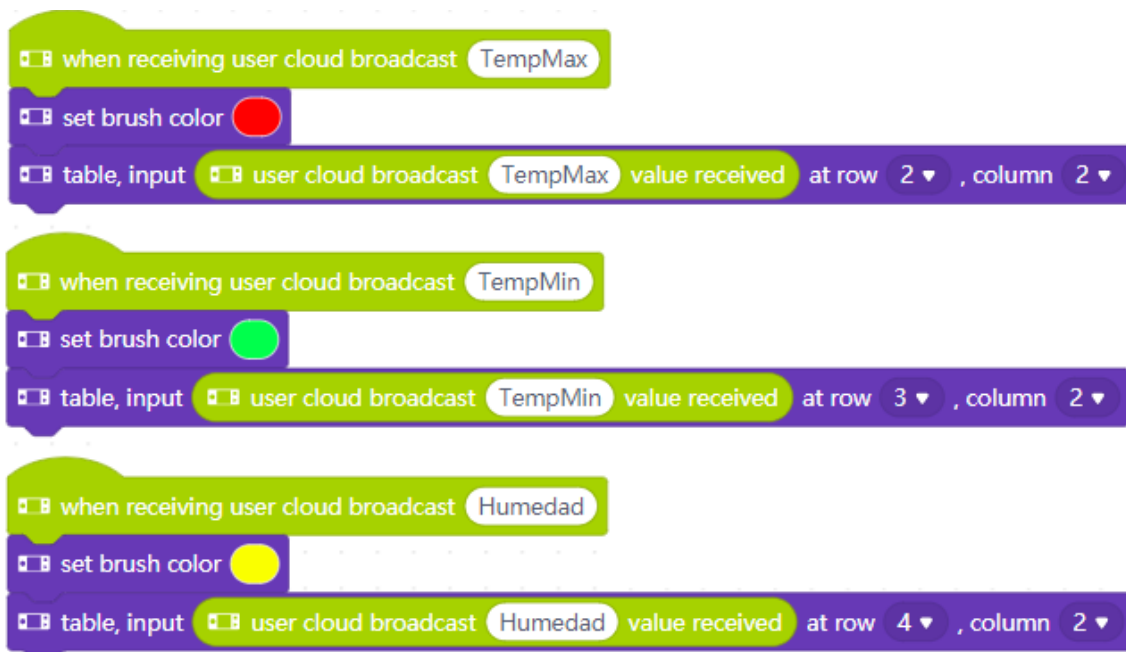


Los datos llegan a la unidad CyberPi cuando se activa el bloque “**when receiving cloud broadcast** <xxx>” que lleva asociado el nombre de la variable a recibir: **Temp.Max**, **Temp.Min** y **Humedad**.

Los datos se escriben en la posición correspondiente de la tabla a mostrar en la pantalla de CyberPi.

- **Temp.Max** columna 2, fila2
- **Temp.Min** columna 3, fila2
- **Humedad** columna 4, fila2





## 6. Realización de los Scripts de cada uno de los Objetos




A continuación vamos a describir los scripts asociados a cada uno de los objetos de la aplicación.

### Envío de Datos Manual

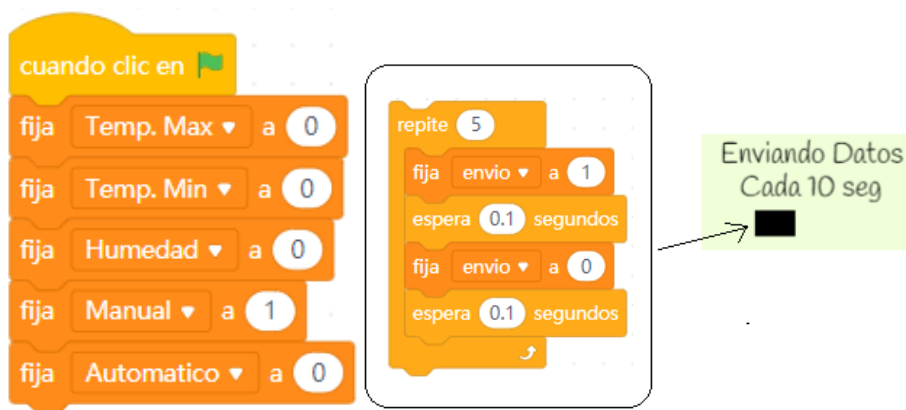


Al presionar este botón lo que hacemos es enviar los datos climáticos leídos a la unidad remota **CyberPi**. Cuando está activado este modo se fija la variable “**envio**” en valor 0 y 1 alternativo, cinco veces lo que hará parpadear el indicador de envío en color rojo/negro

En el script lo que hace es igualar las variables “**Temp.Max**”, “**Temp.Min**” y “**Humedad**” a los datos que obtenemos de la librería “**Datos de clima**.”

Temp. Max =	 Madrid, Madrid, ES temperatura más alta (°C)
Temp. Min =	 Madrid, Madrid, ES temperatura más baja (°C)
Humedad =	 Madrid, Madrid, ES humedad (%)

NO debemos olvidar que todas las variables se deben poner a cero al arrancar la aplicación menos la variable Manual que se pone en el inicio a 1.



Cuando está activado este modo se fija la variable “**envio**” en valor 0 y 1 alternativo.






### Envío Datos Automático



Al presionar este botón lo que hacemos es enviar los datos climáticos leídos a la unidad remota CyberPi de manera automática y continua con intervalos de 5 segundos.

El script lo que hace es igualar las variables que obtenemos de la librería “**Datos de clima**” como en el caso anterior.

Temp. Max =	 Madrid, Madrid, ES temperatura más alta (°C)
Temp. Min =	 Madrid, Madrid, ES temperatura más baja (°C)
Humedad =	 Madrid, Madrid, ES humedad (%)

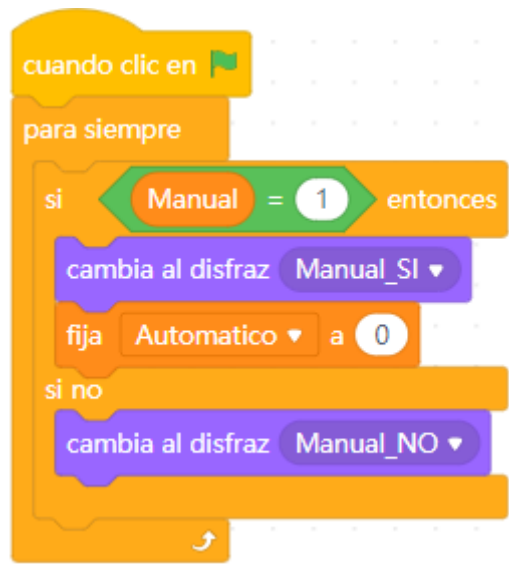
De la misma manera que en el anterior Script se realiza la señalización intermitente del indicador de transmisión.



## Indicador Manual



Este objeto lo que hará será cambiar el color del círculo en función del estado de la variable “**Manual**” para saber que trabajamos en el modo manual. Cuando se activa este estado Manual lógicamente se pone el estado **Automatico=0**



### Indicador Automático



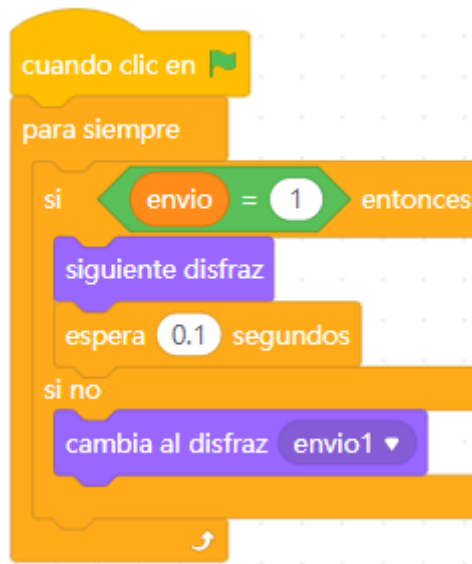
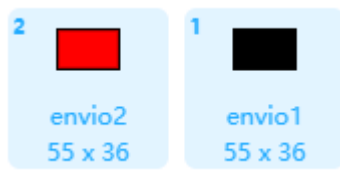
Este objeto se encarga de visualizar el estado “**Automático**” cambiando el disfraz a color negro/rojo y forzando el estado **Manual=0**



### Indicador Transmisión



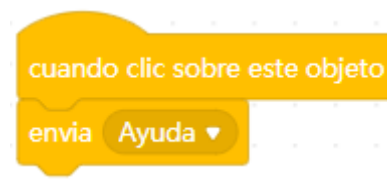
Este script lo único que hace es señalar el envío de datos mediante el cambio alternativo de disfraz (rectángulo en color rojo/negro).



### Botón Ayuda



Al pulsar con el ratón sobre este objeto se mostrara el disfraz del objeto “**Pantalla Ayuda**” mediante el envío del mensaje “**Ayuda**”

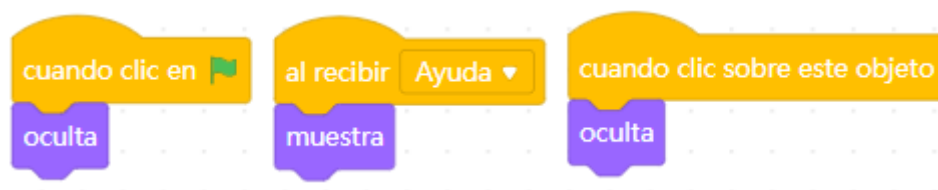


### Pantalla Ayuda



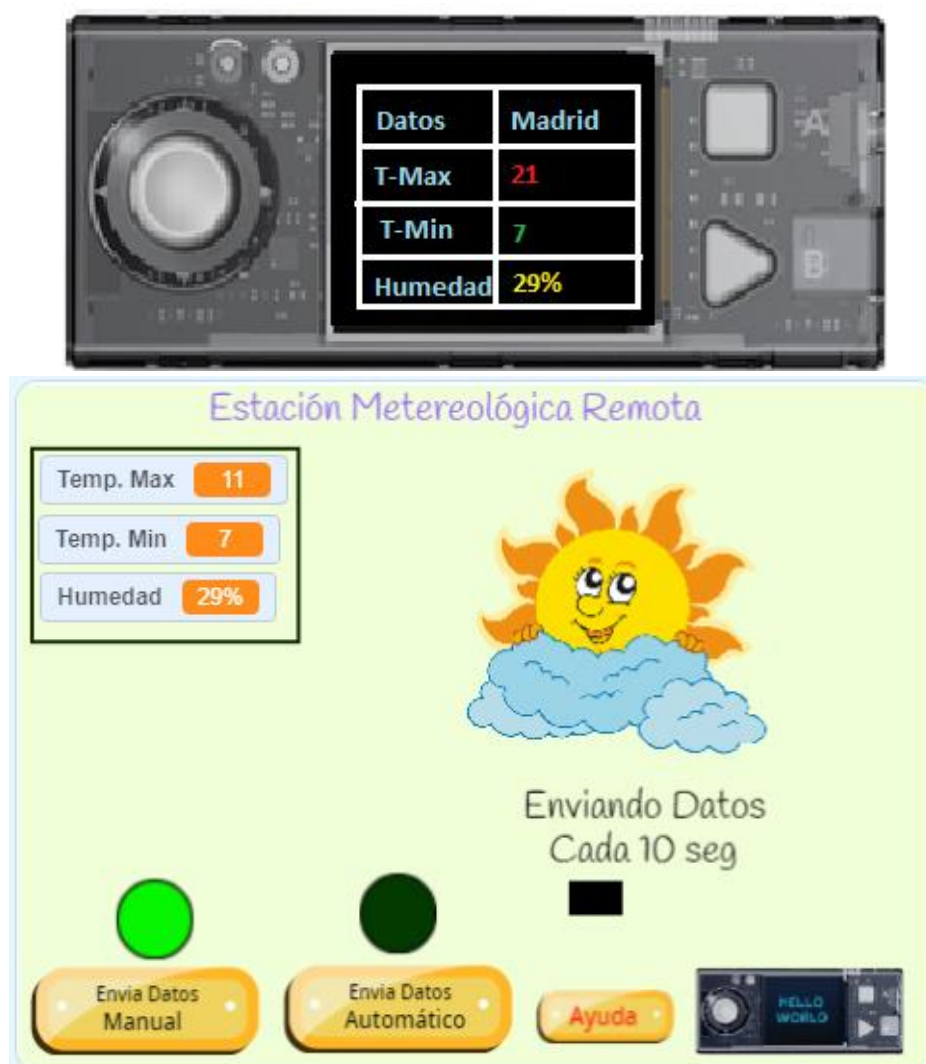
Este objeto tiene un único disfraz que es la pantalla de ayuda que explica el manejo de este ejemplo.

Al iniciar la aplicación esta pantalla permanece oculta. Al pulsar sobre este objeto se oculta.



### Funcionamiento

Para probar la aplicación bastara con descargar sobre CyberPi el programa que esta asociado a su script. No se debe olvidar que la conexión WiFi a través de la que se conecta la unidad este en servicio. Seguidamente se ejecuta el ejemplo en **mBlock** pulsando sobre el icono de inicio (bandera verde). Seguidamente se pulsara sobre uno de los dos botones de envío, el de modo manual o el de modo automático y observaremos como se cargan los datos meteorológicos tanto en el escenario de la aplicación como en el propio dispositivo.



## 8.2.2. Encendido Farolas

La incorporación de las posibilidades de la librería IoT nos permite controlar el estado de una farola ubicada en un punto lejano al lugar en el que nos encontramos. Vamos a realizar un ejercicio que nos permita comprender como usar esta capacidad de **CyberPi**.

### Descripción de funcionamiento

Dispondremos de una unidad **CyberPi** y también de la unidad shield de alimentación llamado “**Pocked Shield**” y su librería de funciones. Usaremos también la librería IoT asociada al Dispositivo y la librería “**Mensaje en la nube de usuario**” que nos permitirá el envío de órdenes de activación

En la imagen siguiente vemos un esquema de lo que pretendemos hacer:



La idea consiste en el envío de un mensaje a la unidad **CyberPi** que se encargará de la activación/desactivación de la señal que mediante una interface (relé de potencia) podremos encender y apagar la farola.

En la parte remota de “mando” solo tendremos un PC con **mBlock** cargado y funcionando en el que pondremos un escenario que simule el proceso con un botón para activar/desactiva y también dispondremos de un botón que mostrara una pantalla de ayuda para facilitar las instrucciones de uso al operador del sistema.

Dispondremos de un escenario que recogeremos de la colección de fondos de pantalla y así mismo los objetos de nuestra aplicación serán los siguientes:



- Dispositivo **CyberPi**

- Botón Activación
- Farola
- Botón de Ayuda
- Texto Ayuda

El fondo pantalla incluye una pequeña imagen de **CyberPi** así como un texto “Gobierno a distancia del Encendido de una Farola.



Crearemos una variable a la que llamaremos “estado” para indicar el estado en el que se encuentra la farola.

Las librerías a incorporar en la aplicación serán:

- Para el Dispositivo **CyberPi**: Pocked Field
- Para los objetos comunes: Mensaje en la nube de usuario

**Para resumir, el funcionamiento consiste en:**

1. Supondremos activados tanto **mBlock** en el PC remoto como el dispositivo **CyberPi** junto a la Farola.
2. Al arrancar la farola se mantendrá apagada.



3. Al pulsar el Botón Activación enviamos a la nube el mensaje de activación-
4. Al recibirse ese mensaje desde **CyberPi** se activa la farola.



5. La desactivación podemos decir lo mismo.
6. Si pulsamos el botón “Ayuda” aparecerá una pantalla o imagen (disfraz) con un texto de ayuda. Para ocultar este mensaje bastará con pulsar con el ratón sobre ese texto.



### Construcción de la aplicación

En la imagen se muestra el diseño de la pantalla de fondo con los objetos colocados en ella,



### Script de CyberPi

Empezaremos por describir el Script asociado al dispositivo **CyberPi**.

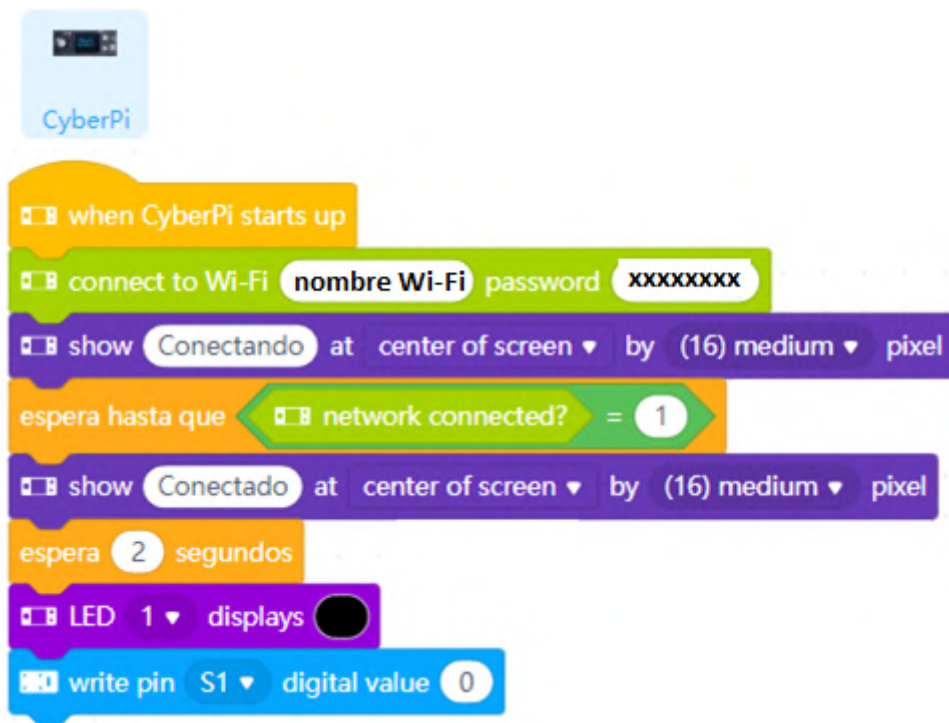
Decir para empezar que nuestro dispositivo trabajara en modo Off-line es decir en modo “cargar” de acuerdo con el entorno **mBlock**. Por este motivo debemos cargar este script en la unidad **CyberPi**.

Bajo el bloque “When **CyberPi** start up” debemos colocar el bloque en el que se definan los datos de la WiFi a la que estar conectado nuestro dispositivo (hablamos de la wifi a la que se accederá desde el parque al pie de la farola –en donde este colocado **CyberPi** - .

Mostraremos un mensaje que nos indica que el dispositivo estas estableciendo la comunicación. Seguidamente, cuándo se establezca la conexión con éxito “*espera hasta que network connected ?*” =1

Una vez conectado se apagan los leds y además se muestra el mensaje “Conectado”

A la vez que ponemos a cero el estado de la salida **S1** de “**Pocked shield**” con la orden “**write pin S1 digital value 0**”



A los efectos de **detectar las órdenes de activación** que le lleguen a **CyberPi** debemos construir este script cuya misión es detectar el valor recibido.

Si el mensaje recibido, asociado a Farola, es *SI*, se activa la salida **S1=1** y se muestra “*Encendido*” en la pantalla de **CyberPi**

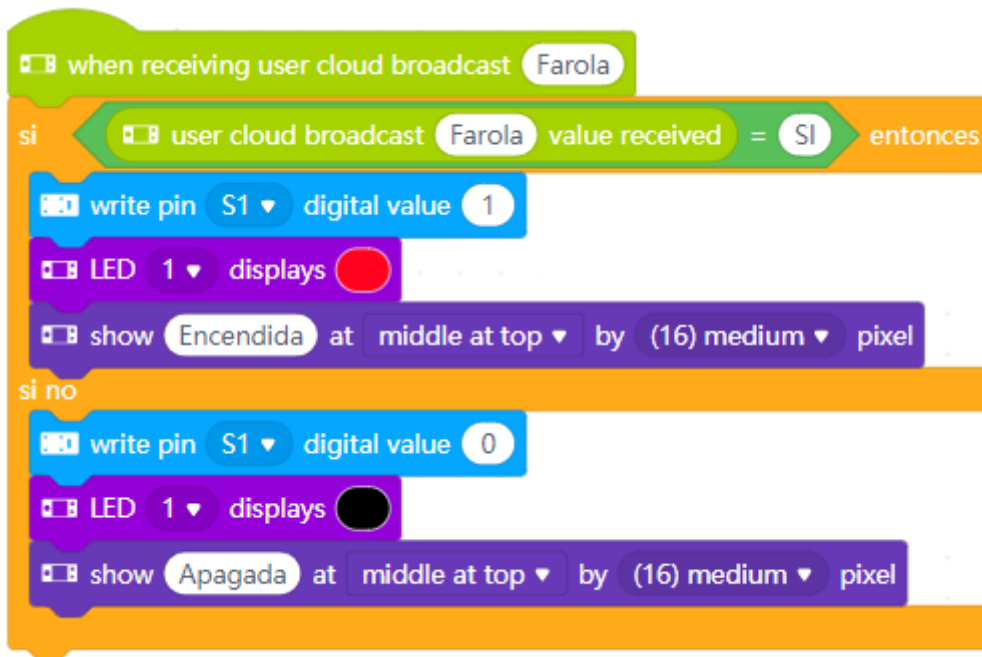


Si el mensaje recibido, asociado a Farola, no es *SI*, se desactivara la salida **S1=0** y se muestra “*Apagado*” en la pantalla de **CyberPi**

Con la activación y desactivación de S1 se enciende y apaga el LED1 de **CyberPi**



A continuación vemos esta parte del conjunto de Script del dispositivo **CyberPi**



### Botón Activación

Este botón presentará dos disfraces, en los que se mostrara la acción de **Encender** y **Apagar** y que activaran o desactivaran de modo “biestable”, por eso definimos una variable llamada “estado” que nos indicará el estado anterior de la Farola:

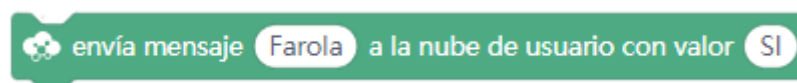
estado=SI => Encendida y estado=NO => Apagada



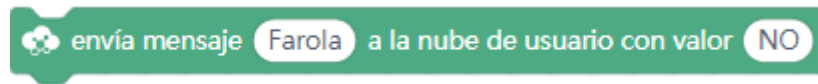
Al iniciar la aplicación nos aseguramos de que el estado de la farola ha de ser apagada es decir **estado=NO**. Esto se consigue simplemente con colocar debajo del bloque de ejecución “**Cuando click en bandera**” la instrucción que asigna NO a la variable valor “**fija estado a NO**”

En cuanto a la orden de activación/desactivación, propiamente dicha, el Script se activara al hacer clic con el ratón sobre el objeto botón “**Activación**”. En este caso lo que hacemos, en función del estado en el que se encuentre la farola, será enviar el mensaje “SI” o el mensaje “NO”

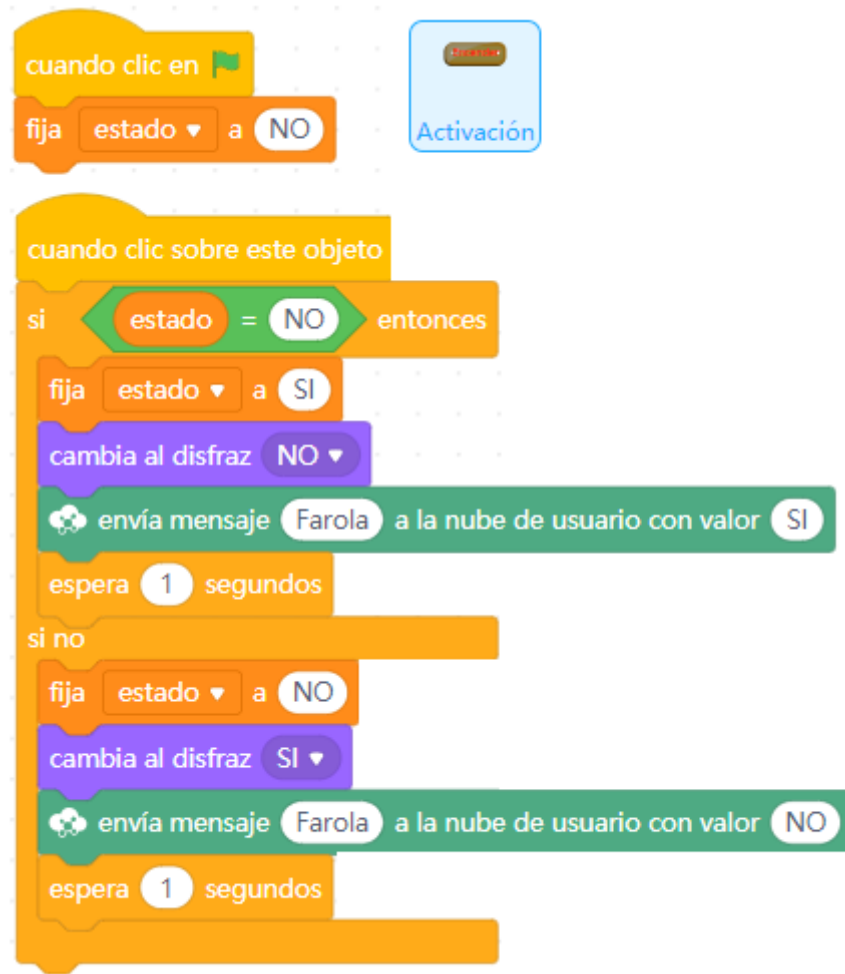
- Si **estado=NO** => Ponemos disfraz “**Encender**” (lo siguiente que podemos hacer es Encender) y enviamos el **mensaje SI a la nube** y ponemos la variable estado=SI “**fija estado a SI**”



- Si **estado=SI** => Ponemos disfraz “**Apagar**” (la siguiente acción deberá ser Apagar) y enviamos el **mensaje NO a la nube** y ponemos la variable estado=NO “**fija estado a NO**”



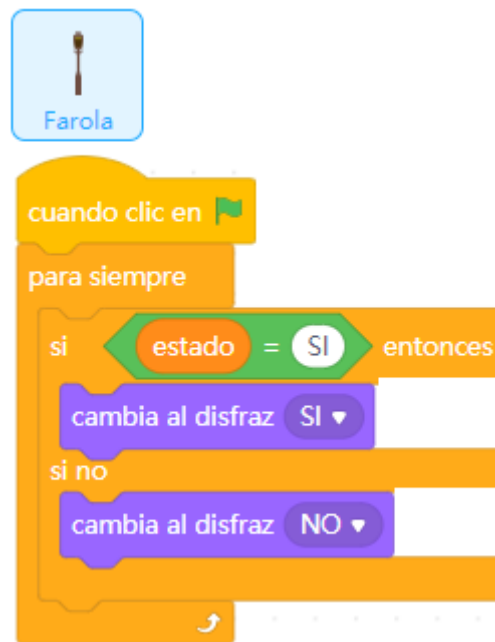
En cada condición ponemos un retardo de 1 seg. Para evitar rebotes “**espera 1 segundo**”



### Script del objeto Farola

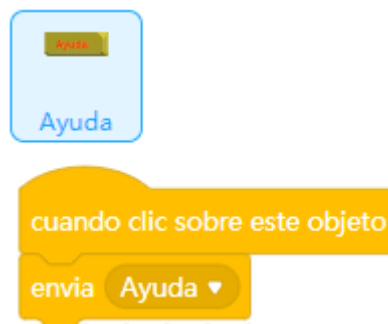
Para la visualización del estado en el objeto Farola bastará que, en función de la variable estado, se ponga uno u otro de los dos disfraces de este objeto. Para conseguir esto, dentro de un bucle “para siempre” colocamos un condicional que se encarga de testear el estado de la variable y cambiar el disfraz.

En la siguiente imagen se muestra el script.



### Script del Objeto Ayuda

En este caso, cuando pulsemos sobre el botón “Ayuda” se envía un mensaje “**envía ayuda**” que será el encargado de que se muestre o no, el siguiente y ultimo disfraz, que es una imagen con el texto de ayuda. Lo veremos en el siguiente objeto.



### Script del Objeto “texto ayuda”

Este es el objeto cuyo disfraz es simplemente la caja de texto de ayuda que se crea como disfraz usando el editor de disfraces o se realiza con el programa de dibujo Paint.

Cuando este objeto recibe el “mensaje ayuda” que emite el botón que hemos explicado antes lo que ocurre es que se hace visible el disfraz mediante la instrucción “**muestra**” de la librería “**Apariencia**” y si queremos ocultar basta que pulsemos sobre el “**cuando clic sobre el objeto**” en cuyo caso se activa la orden “**oculta**” de la librería “Apariencia”.

Nos aseguramos que al iniciar la aplicación se oculte este objeto para no ver la ayuda.



Este sería el disfraz del Objeto “texto\_ayuda”



#### Prueba de funcionamiento

Para probar el sistema debemos separar físicamente **CyberPi** del PC y a ser posible con dos enlaces WiFi diferentes para poder comprobar que funciona tal como se ha definido.

Es posible también cargar la aplicación en un móvil o Tablet y realizar el gobierno desde estos dispositivos-.

Las siguientes imágenes muestran los estados de la pantalla del PC



### 8.2.3. Telemedida y telemando

En los sistemas de comunicación y proceso de datos así como en el telemando y la telemedida es muy frecuente tener que acceder a los sensores y/o actuadores de forma remota. Ya hemos visto algún ejemplo anterior. Ahora veremos otro en el que la comunicación de datos será **Bidireccional**.

#### Explicación de la práctica

Vamos a suponer que tenemos un dispositivo remoto que está leyendo un valor analógico procedente de un sensor y está mandando el valor a una estación receptora (un PC) y que a la vez deseamos enviar desde el PC una orden digital para activación de un receptor que podría ser un motor, un relé, una lámpara, etc...



En nuestro sistema vamos a usar la unidad **CyberPi** acoplada al **Pocket Shield** de Makeblock. Usaremos los pines de **S1** para conectar un potenciómetro con el que simularemos el valor analógico que enviaremos. Utilizaremos la salida **M1** para activar un motor



Vamos a disponer en el terminal remoto (PC) el software **mBlock** para diseñar un escenario en el que visualizaremos la señal analógica que recibamos de la unidad remota

de medida y un botón para la activación del motor que también tendremos en el terminal remoto. Véase la imagen anterior.

Los objetos que forman esta aplicación son los que se muestran en la imagen siguiente:

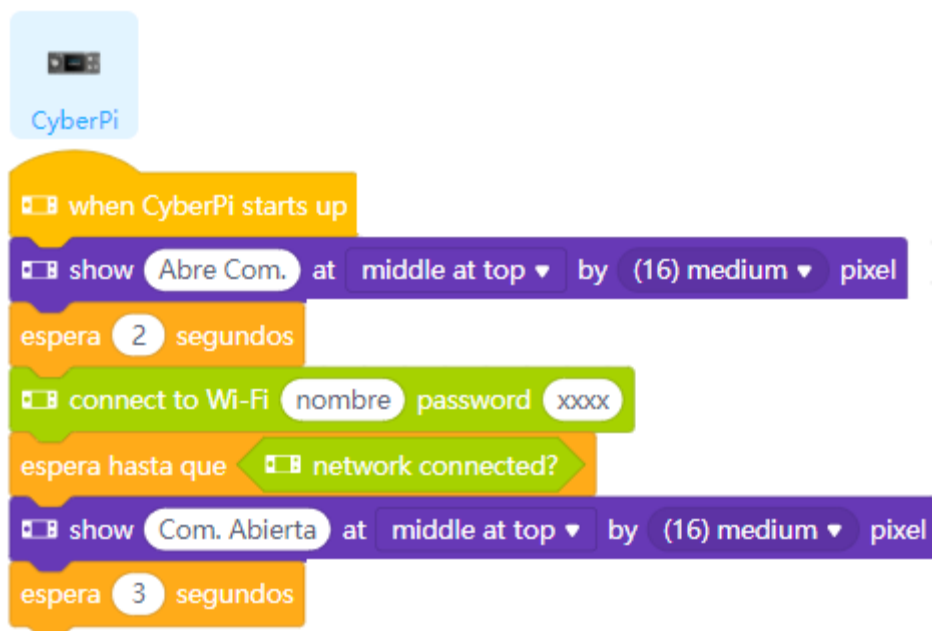


Crearemos dos variables a las que llamaremos:

<b>valor</b>	Para recoger el valor analógico que nos envíe CyberPi
<b>digital</b>	Para almacenar el valor de la señal con la que designamos el estado del motor

### Script del Objeto CyberPi

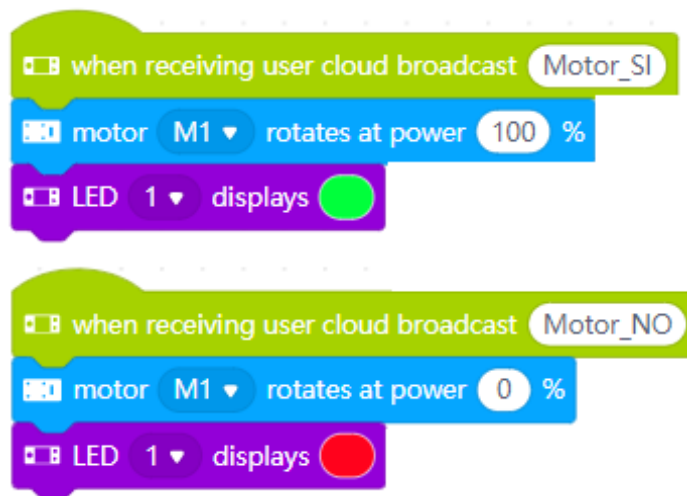
Por tener que enviar y recibir datos a través de la nube debemos designar los datos del servidor WiFi. Al iniciarse el dispositivo mostrara un mensaje “*Abre Com.*” Que nos indica que se procede a abrir la comunicación. Seguidamente se detiene la la ejecución hasta se realiza la conexión “**espera hasta que** <network conected?>”. Seguidamente se muestra el mensaje “*Com. Abierta*” “**show** <Com.Abierta> **at middle at top by (16) médium pixel**” y se espera un tiempo de 3 seg. “**espera 3 segundos**”



### Recepción de mensajes para control del motor

Para gobernar el motor que hemos conectado a la **salida M1** de la unidad lo hacemos mediante bloques de la librería “IoT” “**when receiving user broadcast** <Motor\_SI>” para activar el motor. Para detener el motor se trataría de recibir la palabra “*Motor\_NO*”.

Para cada uno de los mensajes ponemos el LED1 de la unidad CyberPi en color Verde o Rojo dependiendo del estado en el que se pone el motor.



Para el envío del valor digital

El Script que controla el envío del **valor** digital leído en el puerto **S1** de la unidad se procederá a establecer un bucle “**para siempre**” en el que una vez asegurado que la red está conectada “**network connected?**”.

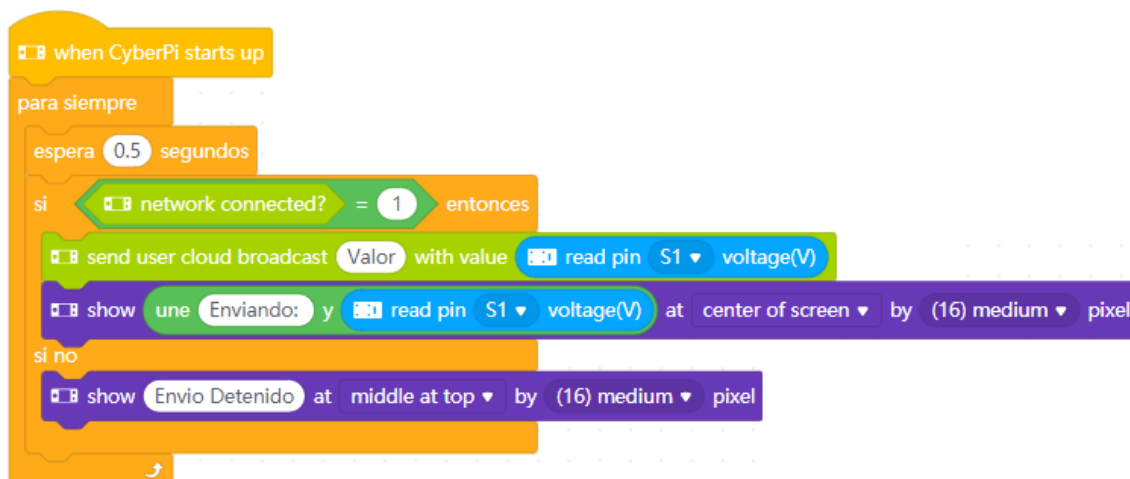
El envío del valor se realiza mediante la función “**send user cloud broadcast <valor> witch value <read pin S1 voltage(v)>**”



a la vez se muestra el valor leído en la pantalla de **CyberPi** con la palabra Enviando:



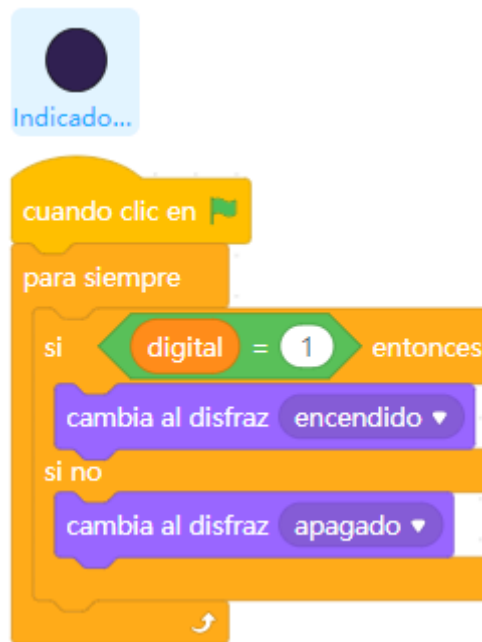
Si la red no está conectada se escribirá el mensaje “**Envío Detenido**”



### Script del indicador de estado del Motor

Con este objeto lo que hacemos es visualizar el estado del motor que estamos controlando de modo remoto.

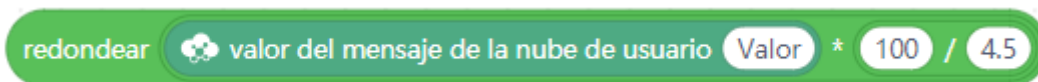
Testeamos permanentemente el estado de la variable digital que es la que recoge el estado del motor, y en función de esta mostramos uno u otro disfraz: Círculo Rojo o Círculo Negro



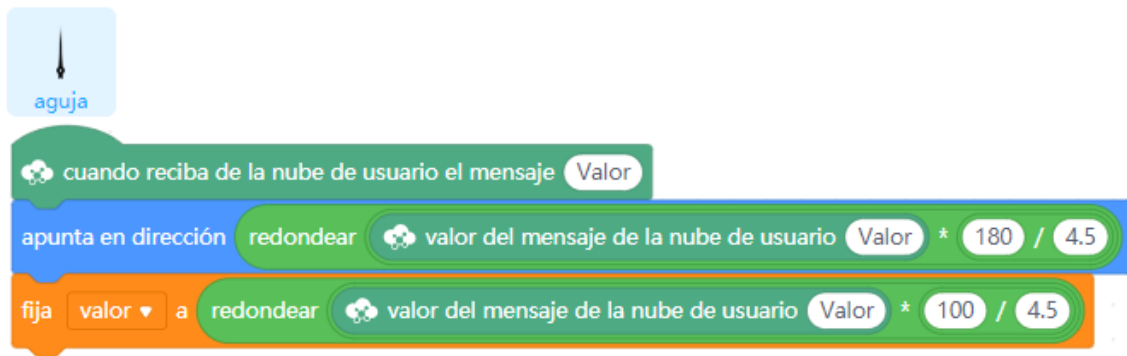
### Script de la “aguja”

La aguja será el indicador, mediante su giro, del valor que se recibe. El valor recibido lo escalamos de 0 a 180 ya que serán 180 los grados que recorrerá esta desde el valor mínimo al máximo. Indicamos el valor simplemente mostrando la variable “valor” igualado este a al valor escalado.

#### Valor escalado



Tomamos **4.5** como valor máximo ya que este es el valor que leemos cuando el potenciómetro está al máximo.

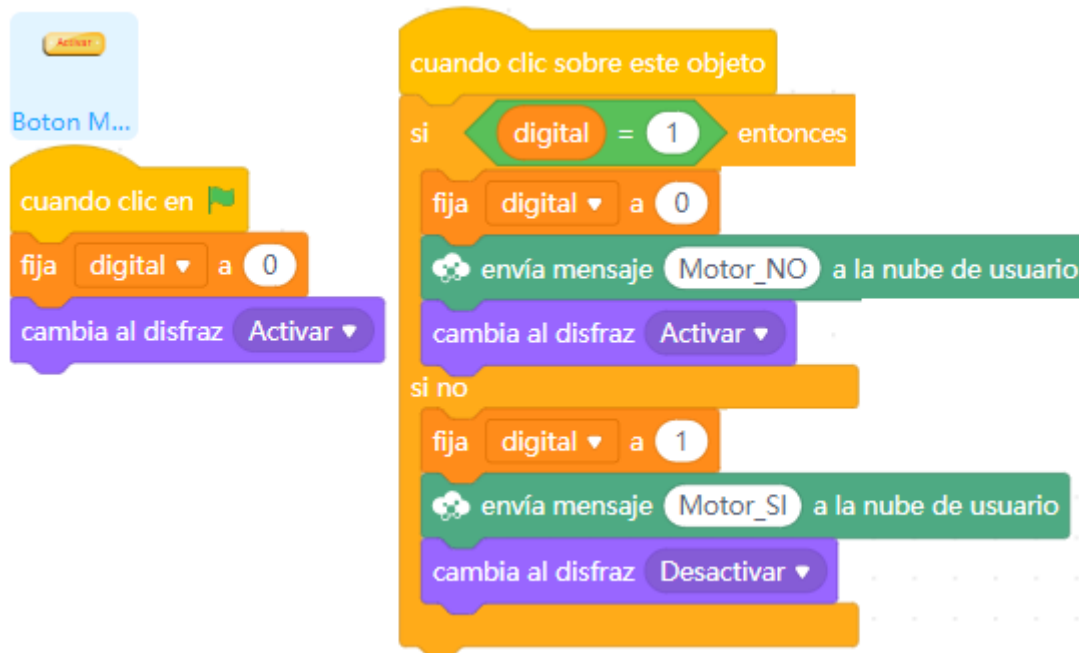


### Script Botón de control para el Motor

Con este botón, al pulsar sobre él, activamos y desactivamos el motor que tenemos conectado remotamente. Nos aseguramos de que al arrancar el valor digital=0 “**fija digital a 0**” y colocamos el disfraz “**activar**”.

Al pulsar en el botón testeamos el estado de la variable “**digital**”, de acuerdo a la ejecución de un condicional compuesto:

- SI **digital=1** Poner digital=0, Enviar **Motor\_NO** Cambia disfraz a “Activar”
- SI **digital=0** Poner digital=1, Enviar **Motor\_SI** Cambia disfraz a “Desactivar”



### Prueba de funcionamiento

La prueba de funcionamiento se realizara con el PC y **CyberPi** alejados o con distintos accesos WiFi con objeto de demostrar que ambos elementos podría estar separados.

No debemos olvidar tener activo el acceso a nuestra cuenta con **mBlock**.

Las imágenes nos muestran el conexionado de un motor y de un potenciómetro a la unidad.

