

DRONE



STEAM

DRONES@STEAM

Fomentar la transformación digital en los centros de FP y crear nuevas ofertas de empleo en el mercado laboral

Resultado del proyecto 2

Actividad 3: MATERIAL EDUCATIVO: MATERIAL DIDÁCTICO Y
EVALUACIÓN

UNIDAD 2, Capítulo 2.4, Ficha 2.4.2

Socio principal: Universidad de Creta



Co-funded by
the European Union



This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein. Project number: 2021-1-EL01-KA220-VET-000034686

MARCO

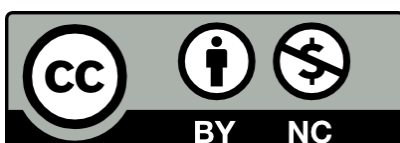
Acuerdo de subvención	2021-1-EL01-KA220-VET-000034686
Programa	Erasmus+
Acción clave	Cooperación para la innovación y el intercambio de buenas prácticas
Acción	Asociaciones estratégicas
Acrónimo del proyecto	DRONES@STEAM
Título del proyecto	DRONES@STEAM: Fomentar la transformación digital en los centros de FP y crear nuevas ofertas de empleo en el mercado laboral
Fecha de inicio del proyecto	28/02/2022
Duración del proyecto	28 meses
Fecha de finalización del proyecto	27/06/2024

PÁGINA WEB:

<https://dronesteam.eu/>

CONSORCIO: LISTA DE SOCIOS

- Universidad de Creta (UoC) - Grecia
- ECAM-EPMI (ECAM) - Francia
- Cyprus Computer Society (CCS) - Chipre
- Politeknika Ikastegia Txorierrri S. Coop (PIT) – España
- Centro Nacional de Investigaciones Científicas “Demokritos” (NCSR) - Grecia
- A & A Emphasys Interactive Solutions Ltd (EMP) – Chipre
- Dirección Regional de Educación Primaria y Secundaria de Ática (RDPSEA) – Grecia



Attribution-NonCommercial
4.0 International ([CC BY-NC 4.0](https://creativecommons.org/licenses/by-nc/4.0/))

Índice

Plan de estudio 2.4.2	4
Ficha 2.4.2 (versión para estudiantes)	5
Ficha 2.4.2.1 Transmisión y recepción de datos	5
Radiocomunicación entre micro:bits	5
Ejercicio 1: Intercambiar mensajes	5
Ejercicio 2: Enviar datos de temperatura	6
Ejercicio 3: Alerta de proximidad	7
Comunicación Bluetooth con el micro:bit	9
Ficha 2.4.2.2 Programa de control remoto	10
Inicializar el radioGroup	10
Arm (Montado)	10
Throttle (Aceleración)	11
Pitch (cabeceo), Roll (balanceo), display (visualización) y envío por radio de los valores	11
Ejercicio 4: Comprobación y depuración del código	12
Ejercicio 5 (deberes)	13

Plan de estudio 2.4.2

UNIDAD 2	
Capítulo 2.4	Vuelo de drones y programación de control remoto
Equipo, software, consumibles	PC con acceso a Internet, una segunda placa micro:bit, Kit de drones programables Air:bit
Duración	3 horas lectivas
Breve descripción	Programar las funciones necesarias del control remoto para conectar y pilotar el dron Air:bit DIY.
Resultados del aprendizaje	Comprender los principios básicos de la transmisión y recepción de datos.
	Aprender a enviar y recibir señales con el micro:bit.
	Reconocer y programar las funciones principales (código del radiotransmisor): arrancar/detener las hélices, cabeceo, balanceo, guiñada, acelerador.
Actividades	
Actividad 1	Ficha 2.4.2.1 Transmisión y recepción de datos
Objetivo	Comunicación por radio micro:bit y Bluetooth.
Duración	1 hora
Tipo	Ficha
Objetivos pedagógicos	Aprender cómo dos (o más) placas micro:bit pueden comunicarse entre sí, con algunas aplicaciones útiles (intercambio de mensajes, alerta de proximidad, etc.).
Recursos	Ficha 2.4.2.1 / Ejercicio 1, Ejercicio 2
Actividad 2	
Actividad 2	Ficha 2.4.2.2 Programa de control remoto
Objetivo	Desarrollar un programa básico de radiocontrol utilizando una segunda placa micro:bit como controlador.
Duración	2 horas
Tipo	Ficha
Objetivos pedagógicos	Aprender a codificar las funciones principales necesarias en un radiotransmisor para arrancar/detener las hélices y controlar el cabeceo, el balanceo, la guiñada y el acelerador.
Recursos	Ficha 2.4.2.2 / Ejercicio 3, Ejercicio 4
Más información	

Ficha 2.4.2 (versión para estudiantes)

Capítulo 2.4: Vuelo de drones y programación de control remoto

Nivel: Intermedio

Ficha 2.4.2.1 Transmisión y recepción de datos

Radiocomunicación entre micro:bits

Para enviar y recibir mensajes o datos, los micro:bits utilizan señales de radio para comunicarse entre sí de forma inalámbrica. Cada micro:bit tiene una antena que puede detectar o generar ondas de radio.



Para utilizar la función de radio de forma más eficaz y evitar interferencias o mezclar señales enviadas desde otros micro:bits, se puede programar el micro:bit para enviar/recibir mensajes a o desde un canal específico. De esta forma, dos o más micro:bits pueden pertenecer al mismo grupo. Los grupos son como los canales de una radio FM o walkie-talkie y hay 256 canales diferentes disponibles (un número entre 0 y 255). El número no es importante, siempre y cuando todos los micro:bits estén utilizando el mismo número de grupo y nadie más cerca esté utilizando el mismo número de grupo para otras aplicaciones.

Ejercicio 1: Intercambiar mensajes

En este ejercicio, se enviará un mensaje sencillo a cada micro:bit del mismo grupo. El o los micro:bits receptores mostrarán el mensaje en la pantalla LED.

En la categoría "Basic" (básico) de la caja de herramientas, utilizar un bloque "on start" (al inicio) y arrastrar hasta el bloque "radio set group" (grupo de radio) de la categoría "Radio" de la caja de herramientas para definir el canal para enviar o recibir mensajes. Establecer el número de grupo en 14.

Arrastrar el bloque "on button A pressed" (al presionar el botón A) de la categoría "Input" (entrada) de la caja de herramientas al área de trabajo y escribir "radio send string" (cadena de envío por radio) con un mensaje específico entre comillas (por ejemplo, "DRONES@STEAM").

Arrastrar el bloque "on radio received received string" (en cadena recibida por radio) de la categoría "Radio" de la caja de herramientas al área de trabajo y el bloque "show string" (mostrar cadena) que hay dentro. Arrastrar la variable "received String" (cadena recibida) desde el bloque "on radio received received string" e introducirla en el bloque "show string".

Probar primero el código en el simulador en pantalla pulsando el botón A. Se debería ver el mensaje "Hola" desplazándose por el segundo micro:bit. Ambos micro:bits son capaces de enviar y recibir el mismo mensaje.

Ahora también se puede descargar el código en el micro:bit. Hay que descargar el mismo programa en todos y cada uno de los dispositivos micro:bit (dos o más).

Ejercicio 2: Enviar datos de temperatura

En este ejercicio, se enviarán lecturas de temperatura desde un micro:bit a un segundo micro:bit y que recibirá y mostrará los datos en su pantalla LED.

Código para el transmisor (emisor):

Utilizar el bloque "on start" y fijar el número de grupo en 14. Crear una variable llamada "myTemp" y utilizar el sensor de temperatura interno para fijar el valor (ver Ficha 1.3.1). Mostrar el valor de "myTemp" en la pantalla LED y enviar el mismo valor a un segundo micro:bit. Dejar transcurrir al menos tres segundos entre la lectura y la transmisión de la temperatura.

Código para el receptor:

Utilizar el bloque "on start" y ajustar el número de grupo a 14. Utilizar el bloque "on radio received" para obtener los datos de temperatura. Si el número recibido (temperatura) está entre -3 y 43 grados, hay que mostrarlo en la pantalla LED. En caso contrario, mostrar un icono "X" en la pantalla.

Probar el código primero en el simulador en pantalla y luego descargarlo en los dos micro:bits (transmisor y receptor).

Ejercicio 3: Alerta de proximidad

En este ejercicio, se utilizará la función de radio para detectar la proximidad de otro micro:bit. Las señales se hacen más fuertes cuanto más cerca está el transmisor, por lo que si la señal es fuerte significa que el otro micro:bit está cerca, y si es débil, el otro micro:bit está más lejos.

Se desarrollarán dos programas diferentes (uno para el transmisor y otro para el receptor) y al menos dos micro:bits. Se almacenará la intensidad de la señal en una variable y se mostrará en la pantalla LED.

Código para el transmisor:

Arrastrar el bloque "radio set group" de la categoría "Radio" al bloque "on start" y fijar el número de grupo en 14. Arrastrar el bloque "radio set transmit power" (se puede encontrar en la categoría Radio -> Más) al bloque "on start" debajo del bloque "radio set group" y establecer el valor de potencia de transmisión en 1, que es bajo, pero funciona mejor en distancias pequeñas. Se puede experimentar con diferentes valores de potencia.

Utilizar el bloque "forever" (para siempre) para seguir enviando un mensaje. Arrastrar "radio send string" y transmitir cualquier mensaje, por ejemplo, el carácter "a" (no importa lo que se envíe). Es mejor añadir un bloque de "pausa" para evitar enviar demasiados mensajes tan rápido. Basta con establecer un valor de 250 ms (es decir, 4 veces por segundo).

En lugar de utilizar el bloque "forever", se puede utilizar un bloque "every 250 ms" de la categoría "Loops" de la caja de herramientas. En este caso, el bloque "pausa" ya no es necesario.

Descargar el código en el transmisor micro:bit.

Código para el receptor:

La idea principal es la siguiente: Cuando el receptor capta un mensaje del emisor, hay que almacenar la intensidad en una variable llamada señal y mostrarla en la pantalla LED.

Arrastrar el bloque "radio set group" de la categoría "Radio" de la caja de herramientas al bloque "on start" y establecer el número de grupo en 14.

Arrastrar el bloque "on radio received received string" de la categoría "Radio" de la caja de herramientas al área de trabajo, crear una nueva variable llamada "señal" (señal) y arrastrar el bloque "set signal to" (fijar señal a) al bloque "on radio received received string". Arrastrar el bloque "received packet signal strength" dentro del área de valores del bloque "set signals to" (cambiar señal a).

La intensidad de la señal puede tener valores de -128 a 28 (-128 significa una señal débil y 28 una fuerte).

Para mostrar la intensidad de la señal en la pantalla LED en un formato más legible, se pueden modificar los valores de la variable "señal" (señal) a un rango entre 0 y 9. Hay que arrastrar el bloque "map" (mapa) de la categoría "Math" (matemática) e insertar los valores "from low":

-100, "high": -40, "to low": 0, "high": 9. Arrastrar el bloque "show number" (mostrar número) de la caja de herramientas "Basic" y colocar el bloque "map" dentro. Colocar el bloque "show number" debajo del bloque "set signal to".

Descargar el código en el micro:bit receptor y probarlo.

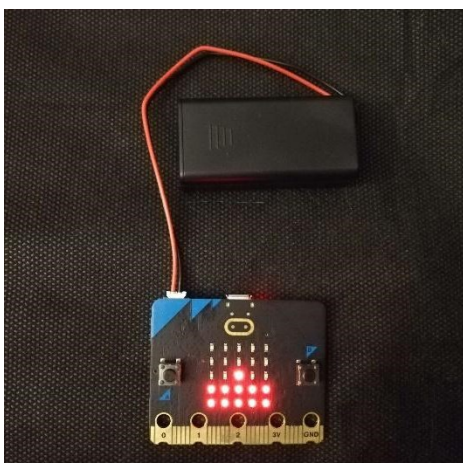
El código anterior solo funciona en el micro:bit, **no** en simuladores de navegador.

Se puede observar que la frecuencia de actualización de la pantalla LED es baja, ya que los números que se desplazan se muestran con lentitud. Para obtener tiempos de respuesta más rápidos, se puede utilizar alternativamente una barra gráfica. Para ello, hay que utilizar el bloque "plot bar graph" de la categoría "Led" en lugar del bloque "show number". Arrastrar todo el bloque "map" dentro del bloque "plot bar graph of" y el valor 9 a la sección "up to".

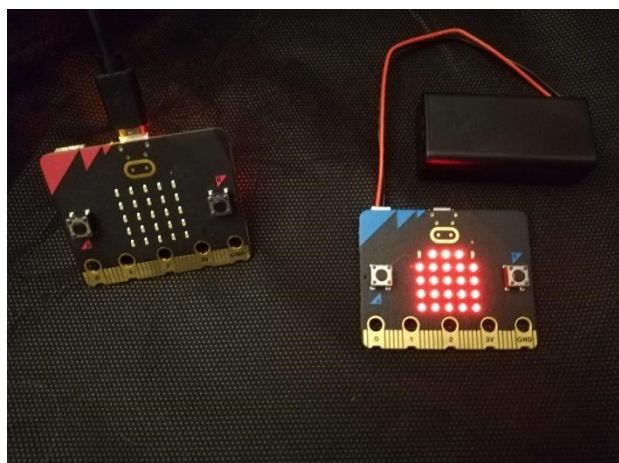


Aparece una barra gráfica que se hace más grande cuanto más fuerte es la señal y cuanto más cercanía hay. Utilizar el bloque de "map" para asignar números de intensidad de señal de radio del rango -100 (más débil) a -40 (más fuerte) a un rango de 0 a 9 para poder utilizarlos para dibujar una barra gráfica.

Descargar el nuevo código en el micro:bit receptor y probarlo.



El receptor cuando el emisor está lejos



El receptor cuando el emisor está cerca

Se puede experimentar cambiando la potencia del transmisor (cualquier número de 0 a 7).

También se puede modificar el código para que el micro:bit receptor emita una alarma de proximidad cuando el micro:bit transmisor se acerque demasiado.

Comunicación Bluetooth con el micro:bit

El micro:bit también se puede programar con MS MakeCode utilizando un dispositivo Android o un iPad/iPhone. Es necesario descargar la aplicación gratuita correspondiente que envía el código al micro:bit mediante conexión Bluetooth.

Se puede descargar la aplicación desde **Google Play** [aquí](#) (requiere Android 5 o posterior) o desde la **App Store** [aquí](#) (requiere iOS 9 o posterior).

Después de descargar la aplicación y empezar a codificar, hay que emparejar el dispositivo (teléfono o tableta) con el micro:bit. La aplicación busca una señal del micro:bit y luego intercambian un código único para asegurarse de que el micro:bit correcto está emparejado con la aplicación. La aplicación contiene instrucciones para este procedimiento.

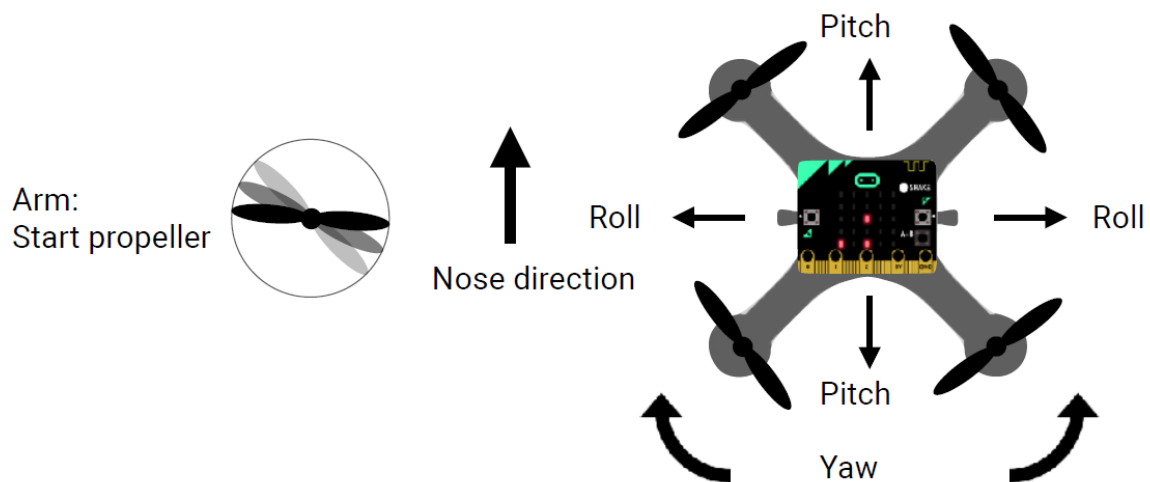
Emparejamiento Bluetooth con el micro:bit

- Emparejar el micro:bit con el dispositivo móvil cada vez que se transfiera un programa de MakeCode.
- Durante el emparejamiento, hay que mantener pulsados los botones A y B a la vez el tiempo suficiente para que se enciendan todos los LED de la pantalla micro:bit.
- En caso de estar utilizando pilas con el micro:bit, hay que asegurarse de que sean nuevas. Aunque parezca que el micro:bit esté funcionando, puede que no tenga suficiente energía para que el Bluetooth funcione correctamente.

Más información y ejercicios en la siguiente ficha 2.4.3.

Ficha 2.4.2.2 Programa de control remoto

Como se ha comentado anteriormente (véase el capítulo 2.2), existen cinco (5) parámetros muy importantes para el vuelo de drones: **Pitch** (cabeceo), **Arm** (armado), **Roll** (balanceo), **Throttle** (aceleración) y **Yaw** (guiñada). Se utiliza el acrónimo **PARTY**.



Para cada uno, hay que escribir un código y controlar los valores de los parámetros correspondientes desde el control remoto (es decir, el segundo micro:bit).

Inicializar el radioGroup

El primer paso es establecer el radioGroup en un número específico (por ejemplo, 14) y mostrar este número en la pantalla LED. Esta acción debe realizarse "on start".

Vamos a utilizar cuatro (4) nuevas variables: "Arm", "Throttle", "Pitch" y "Roll".

Arm (Montado)

Para arrancar y parar las hélices se necesita una función especial. Una buena opción es pulsar los botones A y B a la vez para arrancar o parar las hélices. Se puede utilizar una variable booleana llamada "Arm". Comprobar el valor de "Arm" para ponerlo a 1 si es 0 o a 0 si es 1. En cualquier caso, hay que poner el valor de la variable "Throttle" a 0.

Por motivos de seguridad, es posible implementar una "emergency stop" para apagar el dron, por ejemplo, en caso de accidente inminente. Para ello, hay que poner a 0 las variables "Arm" y "Throttle".

Hay que hacerlo utilizando al menos 3 formas diferentes: a) agitando el control remoto micro:bit (shake), b) poniendo el micro:bit boca abajo (con la pantalla hacia el suelo - screen down), y c) si el micro:bit se cae de las manos (caída libre - free fall).

Throttle (Aceleración)

Para aumentar o disminuir el "Throttle", se pulsarán los botones A y B del micro:bit. Al pulsar el botón A, disminuye el "Acelerador" en 5 (es decir, -5), mientras que al pulsar el botón B el "Throttle" aumenta en 5. Utilizar los bloques "on button pressed" (al pulsar el botón) y "change variable by" (cambiar variable por).

Pitch (cabeceo), Roll (balanceo), display (visualización) y envío por radio de los valores

Dentro del bucle "forever" hay que añadir lo siguiente:

- Ajustar la variable "Pitch" al valor de vuelta de la función "rotación (o)".
- Establecer la variable "Roll" en el valor de vuelta de la función "rotación (o)".
- Despejar la pantalla LED.
- Inicializar el cuadro a $x=0$ e $y=0$ si la variable "Arm" es igual a 1.
- Trazar las variables "Throttle", "Roll" y "Pitch". Para visualizar correctamente estas variables en la pantalla LED, utilizar la función "plot x y" y el bloque "map" de la siguiente forma:

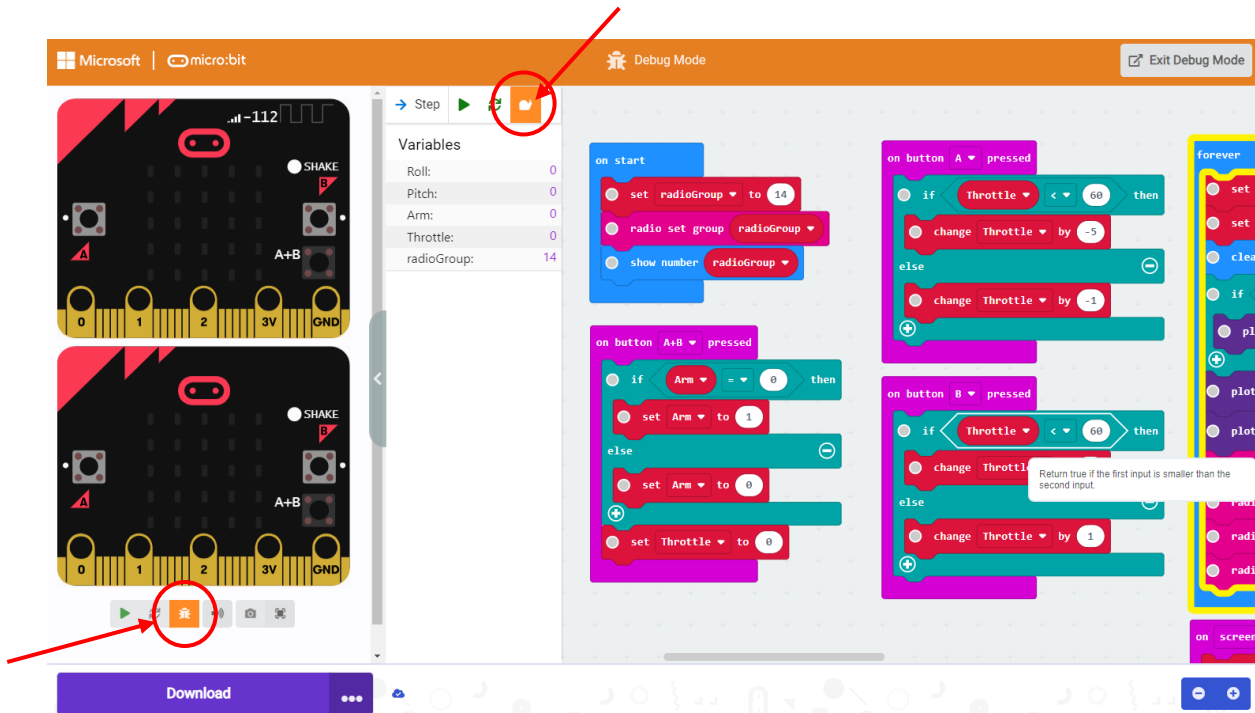


- Enviar los valores de las variables "Arm", "Pitch", "Roll" y "Throttle" por radio, utilizando las letras "A", "P", "R" y "T" y el bloque "radio send value".

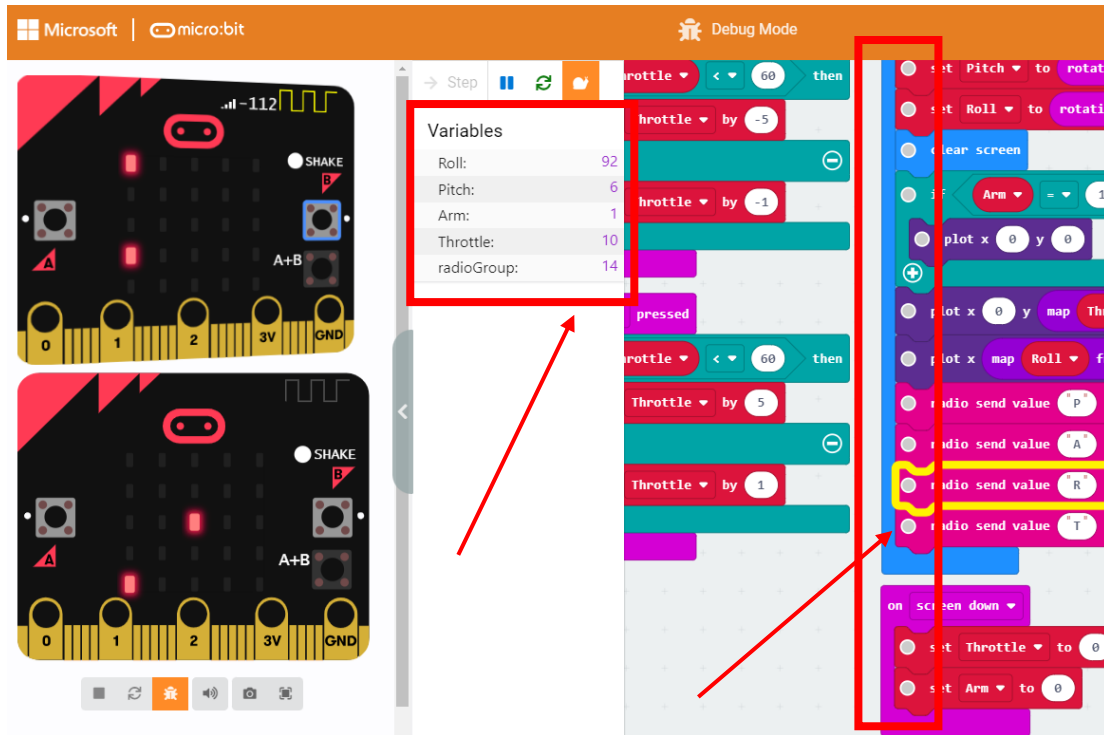
Ejercicio 4: Comprobación y depuración del código

Como primer paso para depurar el código, antes de descargarlo y probarlo en el micro:bit, hay que utilizar el "Debug mode" (Modo depuración) en MakeCode para comprobar los valores de todas las variables.

Hay que hacer click en el pequeño icono de error bajo el micro:bit a la izquierda para entrar en el modo de depuración. A continuación, hay que hacer click en el icono "Slow-Mo" (el botoncito de caracol).



Esta función permite ver paso a paso (a cámara lenta) la secuencia de ejecución de cada comando y bloque. Al cabo de unos segundos, se pueden ver los valores de todas las variables en la columna central. Hay que intentar pulsar algunos botones (por ejemplo, A, B, A+B) y mover el micro:bit para ver cómo cambian los valores. Asegurarse de probar todas las combinaciones de botones y cualquier movimiento para verificar que todo funciona correctamente.



Se puede iniciar, pausar y reiniciar la ejecución, ejecutar uno a uno los comandos (botón "Step"), o incluso establecer un punto de interrupción en cualquier comando seleccionando el botón de opción situado a la izquierda de cada comando (activando y desactivando el punto de interrupción).

¿Qué es un punto de interrupción?

En el desarrollo de software, los puntos de interrupción son marcadores temporales que se colocan en el programa para detener la ejecución en un punto determinado. Es una parada o pausa intencionada en un programa, establecida por el programador, con fines de depuración.

Ejercicio 5 (deberes)

Escribir un programa en un micro:bit receptor para probar y/o depurar el código. Comprobar y visualizar los valores de "A", "P", "R" y "T" transmitidos por el micro:bit de controlador remoto.