

III LIGA MAKER DRONE

FUNDACIÓN BARRIÉ
INSTITUTO TECNOLÓGICO
DE GALICIA

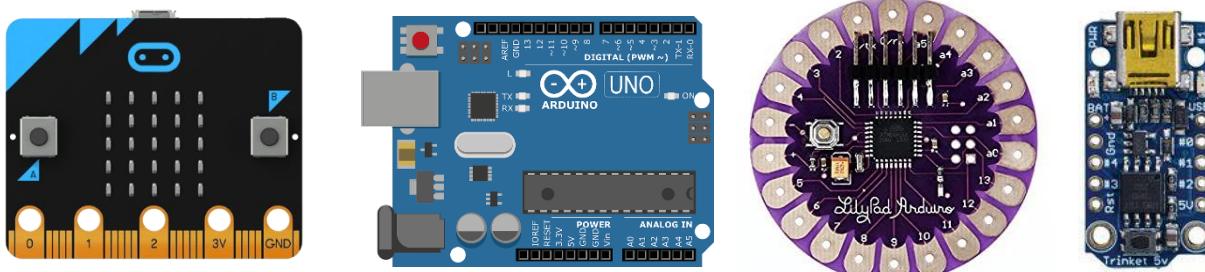


PROGRAMA FORMATIVO

GUÍA DE INTRODUCCIÓN A LOS MICROCONTROLADORES

Para poder controlar nuestro prototipo una vez montado sobre el dron, tenemos dos opciones: conectar nuestro actuador directamente a la controladora del dron (recomendable para casos simples en los que se usa un solo actuador), o conectar el sistema a un microcontrolador externo (recomendable para casos con varios actuadores, sensores o sistemas que requieran un control más preciso o complejo).

Un microcontrolador es un componente electrónico capaz de ejecutar las órdenes que le programamos. Podemos pensar en él como un ordenador de bolsillo, y funciona como un sistema de entradas y salidas: lee información de las entradas (sensores) y manda información a las salidas (actuadores). Existen una gran cantidad de modelos diferentes en el mercado. Cuál elegir dependerá de varios factores: peso, necesidades del sistema, conocimientos de programación... En esta guía se recogen los más habituales, así como recursos para programarlos y conectarlos tanto al prototipo como al dron.



De izquierda a derecha: micro:bit, Arduino UNO, Arduino LilyPad, Adafruit Trinket

CONEXIÓN AL PROTOTIPO

Los diferentes componentes electrónicos del prototipo se conectan al microcontrolador elegido a través de elementos conductivos. Tenemos diferentes tipos:

X CABLES COCODRILO

Son cables con pinzas en los extremos. Muy útiles para prototipar o hacer conexiones simples en microcontroladores como micro:bit.

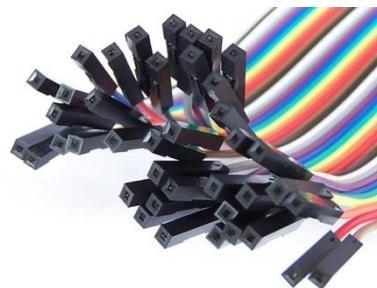


X CABLES DUPONT

Son cables con un conector en cada punta. Estos conectores pueden ser macho o hembra, y un mismo cable puede tener dos extremos iguales (por ejemplo, macho-macho) o diferentes (por ejemplo, macho-hembra). Los conectores macho se utilizan en protoboards o para conectar microcontroladores como el Arduino UNO; mientras que los conectores hembra son útiles para elementos con pines macho, como las salidas de la controladora del dron.



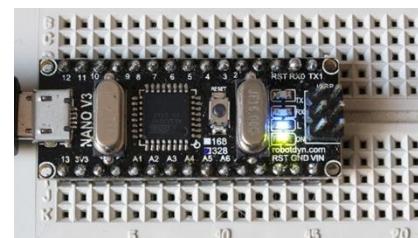
Macho



Hembra

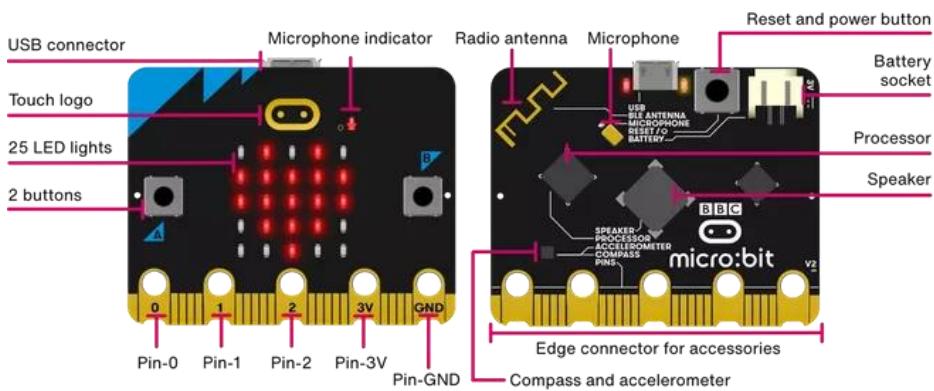
X PROTOBOARD

Algunos microcontroladores vienen soldados a encabezados de pines macho (“patitas”) que les permiten ser montados directamente sobre una protoboard.

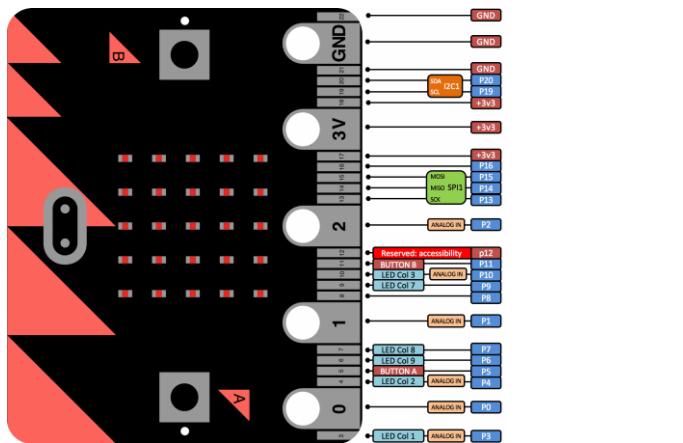


MICRO:BIT

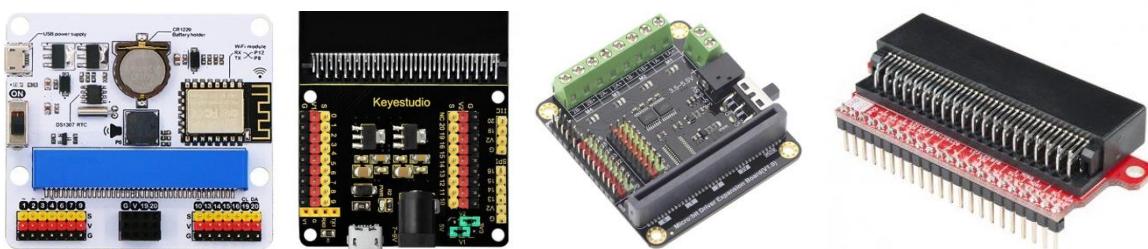
Micro:bit es una placa sencilla de programar que contiene un procesador y una serie de elementos periféricos (ya incluidos en la propia placa), como 2 botones, una matriz de 25 LEDs, sensores de movimiento, temperatura, luz... Permite comunicación por radio y bluetooth. Existen dos versiones de la placa. En esta guía nos referiremos a la v2, más avanzada, aunque la v1 sigue un esquema de funcionamiento y programación similar.



Pesa alrededor de 10 gramos, se alimenta con pilas (3V), batería o por USB, y podemos conectarla a los actuadores y sensores de nuestro prototipo a través de un sistema de pines.



Como se observa en la imagen superior, algunos pines son digitales, es decir, sólo pueden tomar dos valores: 0 y 1, apagado y encendido, para usar por ejemplo con un botón; y otros analógicos, pueden tomar cualquier valor dentro de un rango entre 0 y 1023, para usar por ejemplo con un potenciómetro. Algunos están reservados para las columnas de la matriz de LEDs, por lo que es preferible evitarlos. Los pines 0, 1 y 2, además de 3V y GND (los que utilizamos como fuente de alimentación, positivo y negativo respectivamente) son directamente accesibles a través de una conexión con cables cocodrilo; pero si queremos aprovechar todas sus funcionalidades, debemos extraer el resto de pines a través de una placa externa denominada **SHIELD**. Existen varias *Shields* en el mercado. En función de los actuadores que se quieran utilizar o si se quieren añadir funcionalidades como WiFi deberá elegirse una u otra.



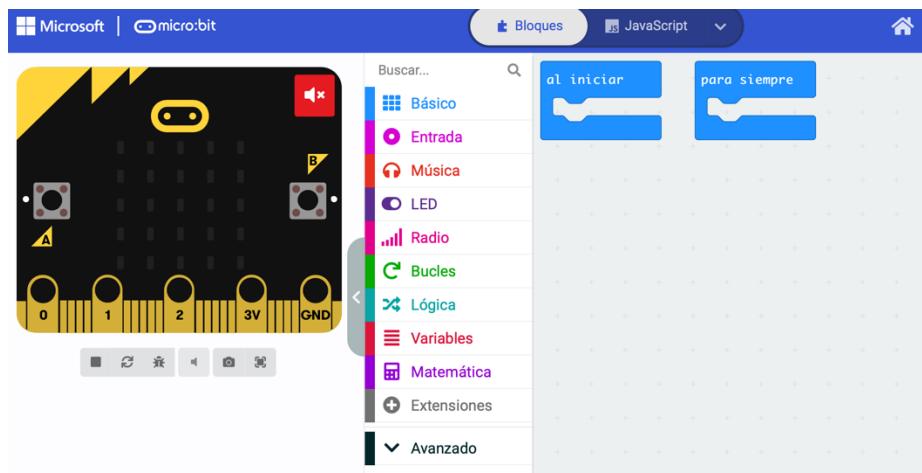
De izquierda a derecha: IoT:bit de Elecfreaks, Sensor Shield V2 de Keyestudio, micro:bit Motor Expansion de DFRobot, micro:bit Breakout de Sparkfun

✗ ENTORNOS DE PROGRAMACIÓN

✗ MakeCode

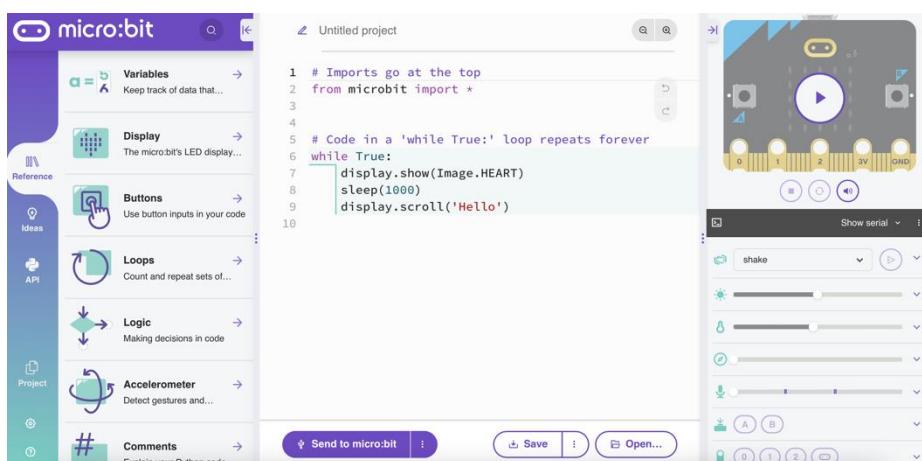
Se trata de una plataforma gratuita para programar online. Permite conectar el micro:bit de forma sencilla y programarlo con Bloques, JavaScript o Python, aunque para este último es preferible otro entorno. Presenta también un simulador para probar nuestra programación sin necesidad de enviarla a la placa física. Es la herramienta recomendada para la mayoría de los casos, especialmente si es nuestro primer contacto con la programación.

Es accesible desde la página <https://makecode.microbit.org>, que incluye también ejemplos y tutoriales.



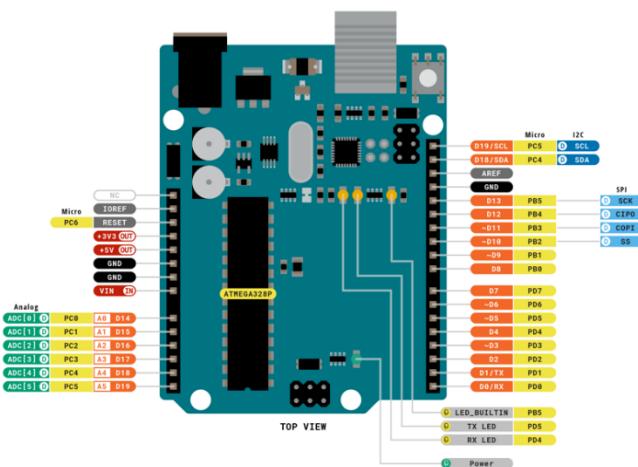
✗ Python Editor

Entorno para desarrollar programas para micro:bit con el lenguaje de programación Python. También online y con simulador. Muy intuitivo y con recursos de aprendizaje disponibles. Recomendado para personas con conocimientos previos de Python y ya experimentadas en el uso de la placa micro:bit. <https://python.microbit.org/v/3>



ARDUINO

Arduino es una plataforma de prototipado basada en una placa con pines configurables como entradas o salidas, mediante la programación de su microcontrolador. Existen varios modelos de placas Arduino (Uno, Zero, Mega, Nano, Robot, Leonardo...), de diferentes pesos, número de pines y forma de conexión. Por ejemplo, la placa Arduino Nano pesa 7 gramos, tiene 20 pines y se conecta a otros componentes mediante cables DuPont o una protoboard; mientras que las placas LilyPad, pensadas para proyectos de electrónica textil, se conectan mediante hilo conductor, cables cocodrilo o broches. En esta guía se explica **Arduino UNO**, la placa más ampliamente utilizada por su simplicidad, pero el resto de placas siguen un esquema de funcionamiento y programación similar. Es posible encontrar fácilmente la hoja de especificaciones de cualquiera de ellas en docs Arduino, buscando las palabras clave ‘*arduino datasheet nombre_placa*’. Por ejemplo, [Arduino datasheet UNO](#).



La placa Arduino UNO, de dimensiones 68.6mm x 53.4mm y 25g de peso, se compone del circuito integrado ATmega328P y de los siguientes pines:

Entradas

Son los pines de la placa que podemos configurar para leer señales de sensores.

- Entradas digitales → pines **0 a 13**
- Entradas analógicas → pines **A0 a A5**

Salidas

Son los pines de la placa que podemos configurar para enviar señales a actuadores.

- Salidas digitales → pines **0 a 13**
- Salidas PWM → pines con símbolo ~ (**3, 5, 6, 9, 10** y **11**)

X Comunicación serie

Son los pines del puerto serie. Estos pines están internamente conectados al puerto USB por lo que es posible utilizar el puerto Serie directamente a través del conector USB. Permiten por ejemplo leer parámetros por el monitor serie.

- | | |
|-------------------------------|---------------------|
| Transmisión de datos en Serie | → pin Tx (0) |
| Recepción de datos en Serie | → pin Rx (1) |

X Alimentación

Existen cuatro formas posibles de alimentar a la placa Arduino UNO:

1. Conexión con cable **USB**
2. Conexión con **Jack** y transformador
3. Conexión con **Jack** y batería
4. Con **Vin** y batería:



Se puede utilizar una batería conectando el terminal positivo al pin **Vin** y el terminal negativo al pin **GND**. El puerto **Vin** permite una entrada de entre 7 y 12 voltios, se recomienda una batería de 9V o varias en serie.

Además, contiene pines que permiten alimentar a sensores y actuadores:

- | | |
|------------------------|--|
| Toma de tierra (0v) | → pines GND (la placa contiene 3) |
| Fuente de tensión 5V | → pin 5V |
| Fuente de tensión 3.3V | → pin 3.3V |

X ENTORNOS DE PROGRAMACIÓN

X Arduino IDE

Entorno integrado de desarrollo instalable, permitiendo de esta forma el trabajo offline. Programación en código mediante sketches. Entorno completo pero que puede resultar complejo si no se tienen conocimientos de código. Instalador disponible para Windows, Linux y MacOS: <https://www.arduino.cc/en/software>

A screenshot of the Arduino IDE interface. The title bar says "Blink | Arduino 1.8.5". The code editor window contains the "Blink" sketch, which blinks an LED connected to pin 13. The code is as follows:

```
This example code is in the public domain.  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

X Arduino Web Editor

Versión en la nube (Arduino Cloud) del IDE. No necesita instalación del entorno en PC, pero sí de un plugin que permitirá la conexión con la placa. Compatible con Windows, Linux, MacOS y ChromeOS. <https://create.arduino.cc/editor>

X Entornos de programación por bloques

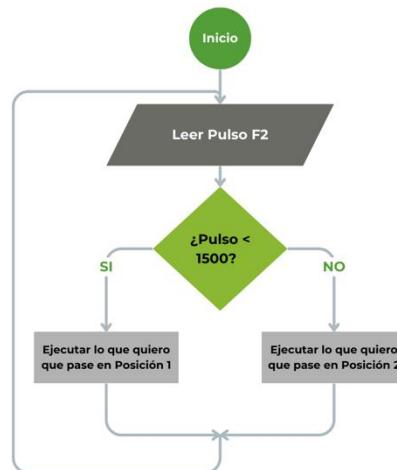
Entornos gráficos menos completos que los anteriores, pero más sencillos de utilizar. Existen varios: **S4A** (Scratch for Arduino), instalable (https://s4a.cat/index_es.html); **mBlock**, instalable u online (<https://www.mblock.cc/>); **Ardublock**, instalable u online (<http://ardublock.ru/index.php>); **Bitbloq** o **Minibloq**.



De izquierda a derecha: S4A, mBlock, Ardublock

CONEXIÓN AL DRON

Para conectar el microcontrolador al dron y poder controlar nuestro sistema desde el radiocontrol, podemos emplear la salida **F2** de la controladora (consultar el documento **S02-01 GUÍA RÁPIDA CONFIGURACIÓN RADIOCONTROL**). Esta salida nos da una señal PWM con un ancho de pulso que va entre $1000\mu\text{s}$ y $2000\mu\text{s}$, aunque podemos configurarla para abarcar un rango menor. Para controlar el prototipo con un switch de 2 posiciones, por ejemplo, y utilizando todo el rango, en una posición tendríamos un pulso de $1000\mu\text{s}$ y en la otra de $2000\mu\text{s}$. En este caso, la programación seguiría el siguiente esquema:



Que, traducido a código de micro:bit:



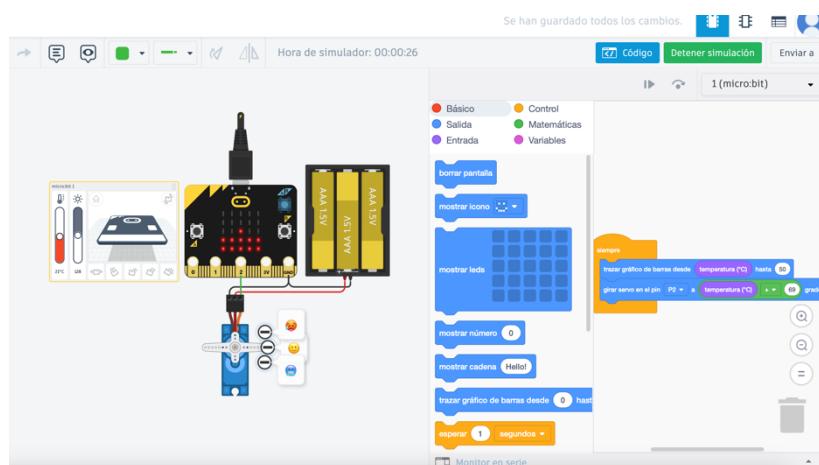
Y para Arduino:

```
6 void loop() {
7   // put your main code here, to run repeatedly:
8   pulso = pulseIn(11, HIGH);
9   if (pulso < 1500){
10     // Qué queremos que pase cuando el switch está en posición 1
11   } else {
12     // Qué queremos que pase cuando el switch está en posición 2
13   }
14 }
```

PARA AMPLIAR INFORMACIÓN

En el caso de querer utilizar otras placas, es recomendable consultar el sitio web del fabricante, así como buscar información con la palabra clave *pinout*, para obtener un esquema detallado de las conexiones. Modelos de otras marcas como **Adafruit** son fácilmente programables con entornos de los mencionados, como MakeCode o Arduino IDE.

Un software interesante para probar microcontroladores con diferentes circuitos es TinkerCAD, donde podemos prototipar nuestro sistema electrónico y simular el funcionamiento de la programación con Bloques, Python o C++ antes de llevarlo al montaje real.



En las siguientes páginas se encuentran tutoriales o información que puede resultar útil:

- <https://www.programoergosum.es/>
- <https://lab.bricogeek.com/>
- <https://bricolabs.cc/wiki/start>
- <https://www.arduino.cc/education/courses/>
- <https://microbit.org/projects/make-it-code-it/>
- <https://projects.raspberrypi.org/en/paths>
- <https://www.tinkercad.com/learn/circuits>
- <https://www.tinkercad.com/lessonplans>