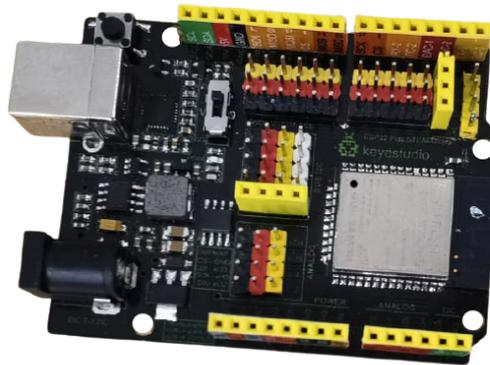


ESP32 STEAMakers



+



Telegram

ArduinoBlocks



Juanjo López
Versión 1.0

Introducción

Telegram es una plataforma de mensajería y VOIP, desarrollada por los hermanos Nikolái y Pável Dúrov. La aplicación está enfocada en la mensajería instantánea, el envío de varios archivos y la comunicación en masa. Telegram es hoy por hoy una de las grandes alternativas a WhatsApp, una auténtica navaja suiza que ofrece todo tipo de opciones que van más allá de las que suelen dar la mayoría de apps de este tipo.

Telegram es una aplicación que podemos instalar en cualquier dispositivo móvil, y posee una plataforma web desde la que podrás acceder desde tu ordenador.

El envío y recepción de mensajes cuenta con un protocolo de seguridad que cifra cada texto y asegura tus mensajes.

En Telegram podemos crear grupos o canales de difusión.

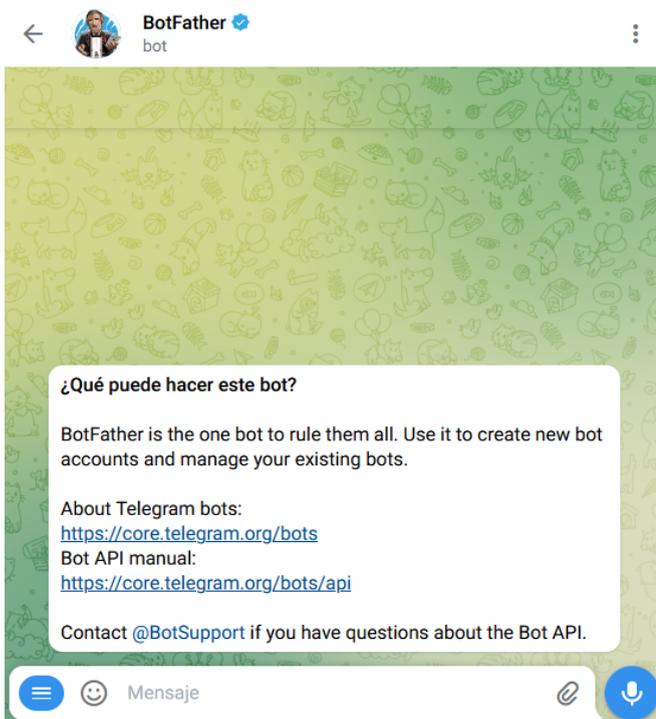
Crear un bot en Telegram

Para crear un Bot en Telegram lo hacemos desde el propio Telegram (app o versión web, como más te guste).

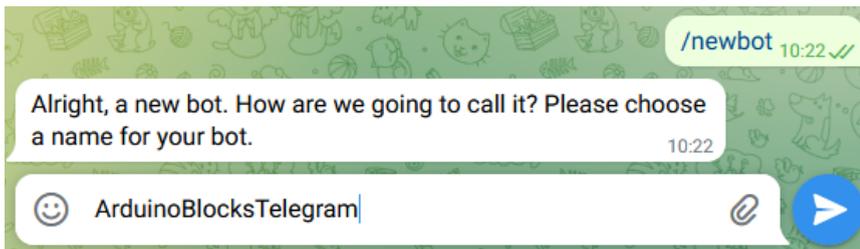
Via web: <https://web.telegram.org>

Debemos “chatear” con el bot de los bots... el **@botfather**

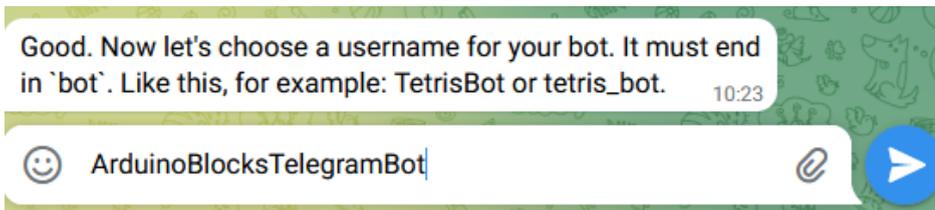
Una vez iniciada la conversación con el **@botfather** nos aparece un listado de comandos para crear el nuevo Bot y poder configurarlos (*¡muy sencillo!*)



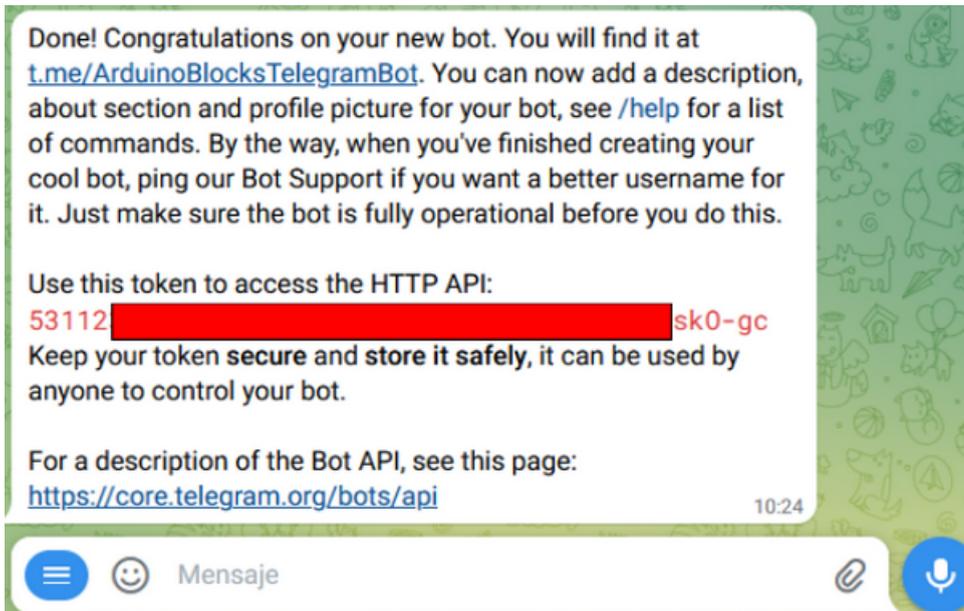
El comando para crear un nuevo Bot es **/newbot** y seguidamente nos pregunta el nombre del nuevo Bot:



Seguidamente nos pregunta por un nombre de usuario para este Bot recién creado (el nombre debe acabar en **Bot** o **_bot**):

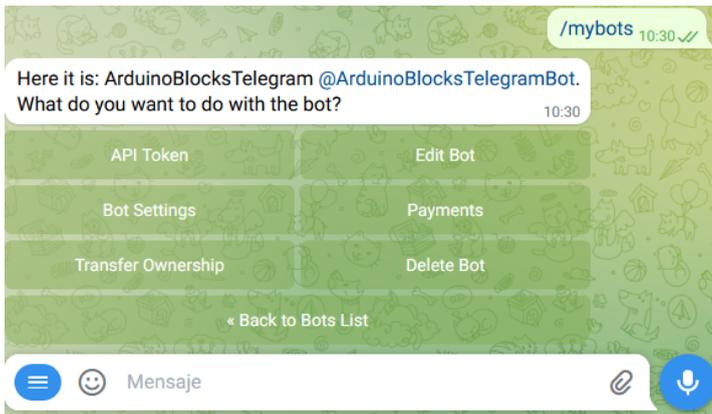


¡Ya lo tenemos! **@botfather** ya ha creado el nuevo Bot y ya nos da un **Token** para conectar con la API de Telegram (bloque Iniciar Telegram Bot en ArduinoBlocks)



El Token para la API es muy importante, debes guardarlo a buen recaudo pues nos permitirá tener el control total de nuestro Bot (no lo difundas)

Si queremos modificar o completar la información de nuestro Bot, lo más sencillo es mediante el comando **/mybots** seguir las opciones que nos muestra en forma de menú y botones para ir completando la información.

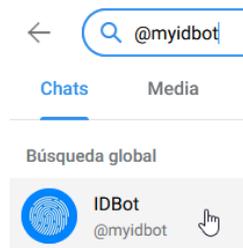


¿Algo más?

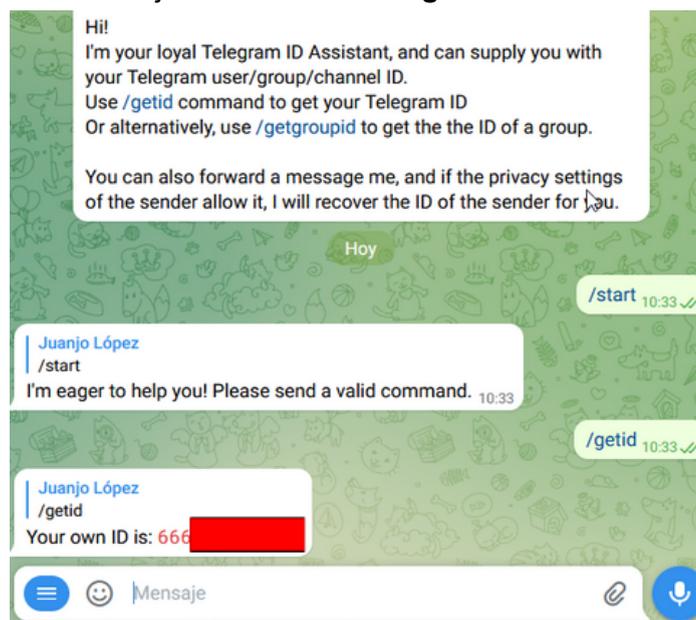
Con ésto ya podríamos empezar a “jugar”, pero si queremos que nuestro Bot pueda enviar mensajes a un chat directamente (para responder no haría falta), necesitamos el ID del Chat en cuestión. Para obtener el Chat-ID podemos preguntárselo a otros Bots.

- **Obtener el ID de mi chat privado** (para que mi Bot envíe mensajes a mi usuario de Telegram directamente)

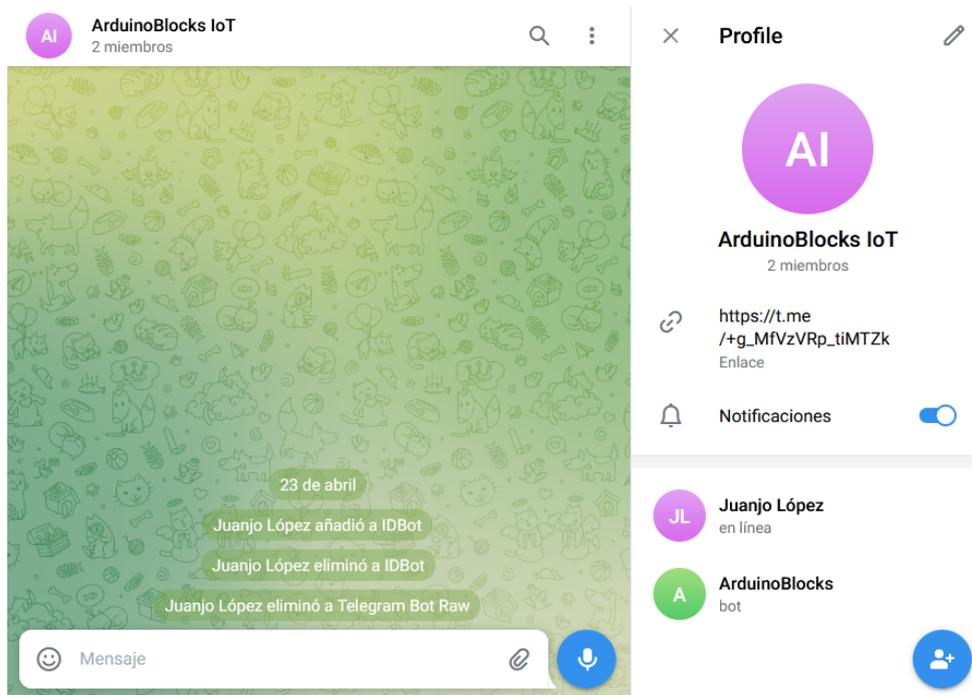
Busco al Bot **@myidbot** en Telegram y le pregunto mi ID



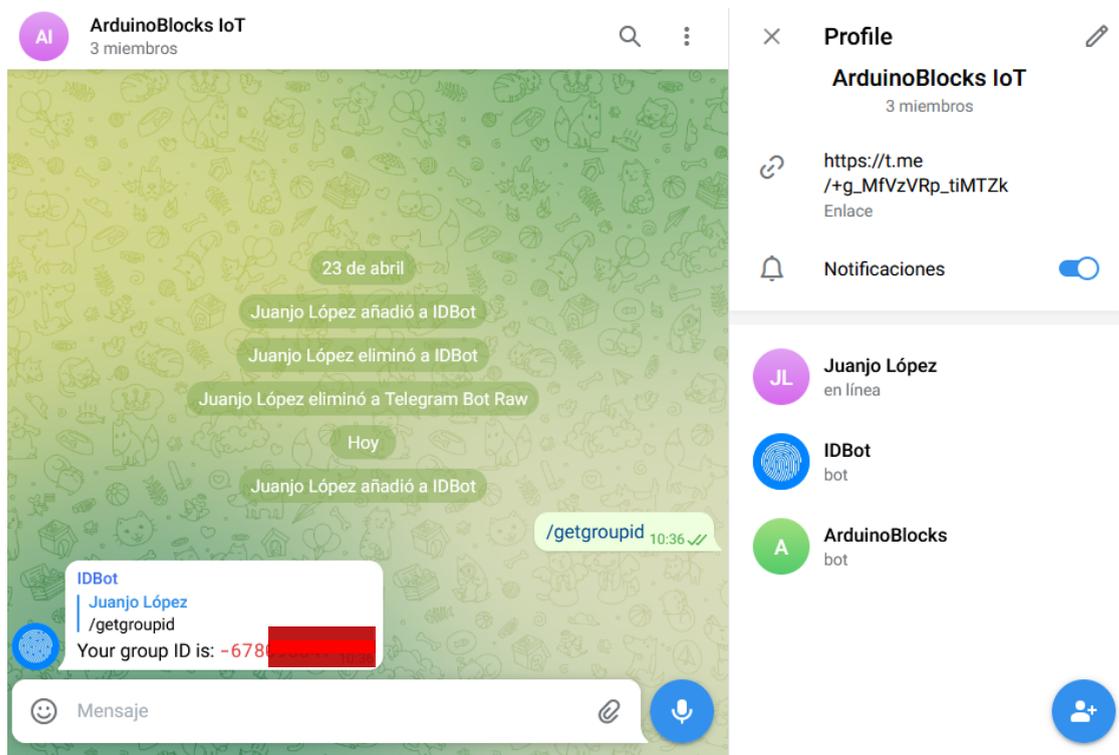
Debemos ejecutar el comando **/getid**



- **Obtener el ID de un grupo:** Para ello debemos añadir a **@myidbot** al grupo que queremos obtener su ID



Y ejecutar el comando **/getgroupid** (saldrá un número de ID negativo)



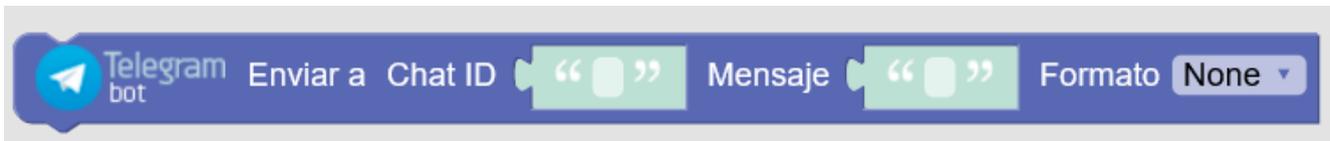
Posteriormente eliminamos al Bot @myidbot del grupo

Bloques Telegram-Bot en ArduinoBlocks

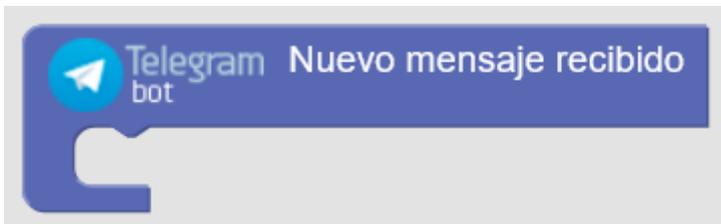
- **Iniciar Telegram Bot:** permite iniciar el sistema Telegram en la ESP32, debemos indicar el Token obtenido para la API al crear el bot (con **@botfather**)



- **Enviar un mensaje:** permite enviar mensajes a cualquier chat de telegram directamente
 - **Chat-ID:** puede ser el ID de una conversación privada a un usuario en concreto, de un grupo o de un canal).
 - **Texto del mensaje:** el texto con el mensaje que se quiere enviar
 - **Formato:** Se puede indicar tres tipos de formato para aplicar estilos al mensaje (negrita, títulos, etc.). Formato: [Markdown](#), MarkdownV2 o HTML



- **Evento de cuando se recibe un nuevo mensaje desde Telegram:**
 - Dentro de este bloque procesaremos el mensaje recibido, pudiendo acceder al texto del mensaje, al remitente (nombre) y al Chat-ID de donde procede el mensaje.
 - Aquí responderemos o actuaremos según el mensaje que nos llegue.



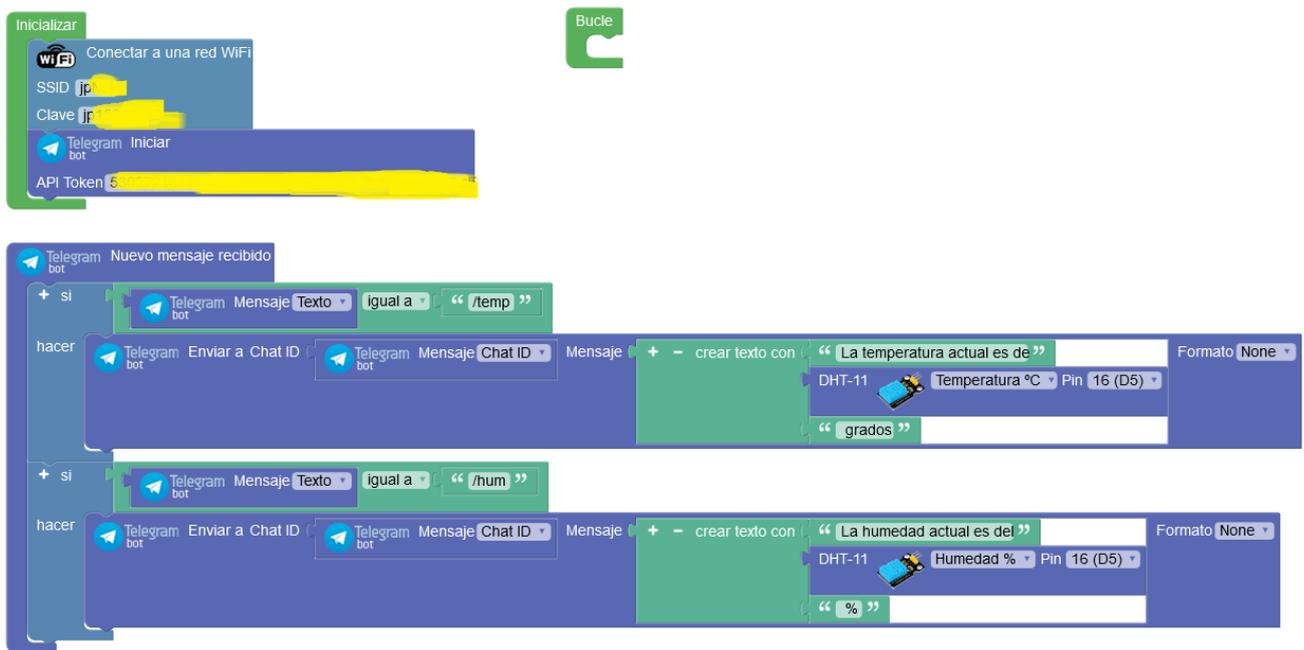
- **Mensaje recibido:** Este bloque sólo tiene sentido usarlo dentro del evento de "nuevo mensaje recibido", si se usa fuera de él su contenido estará vacío.
 - Texto: texto del mensaje recibido
 - Chat ID: ID del chat de donde procede el mensaje. Muy útil para responder a ese mismo chat (pueden llegar mensajes de distintos chats, grupos, etc. por lo que de esta forma identificamos el origen y nos sirve para responder al chat correcto)
 - Remitente: El nombre del usuario remitente del mensaje.





Ejemplo 1: Bot de Temperatura y humedad con DHT11

El Bot responderá con la temperatura y humedad de un sensor DHT-11 conectado a la ESP32 STEAMakers.



Desde telegram “chateamos” con el Bot de la ESP32 STEAMakers y obtenemos la temperatura y humedad del sensor en tiempo real.



Ejemplo 2: Situar un servo en la posición enviada desde Telegram

Mediante el comando **/servo=GRADOS** situaremos el servo en la posición deseada con el valor GRADOS

