

PROGRAMACIÓN LIMPIAMARES CON ARDUINOBLOCKS Y BLYNK



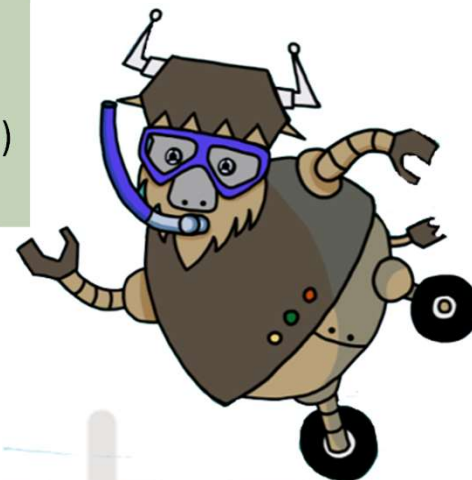
CONTROL WIFI LIMPIAMARES

En este apartado vamos a aprender como controlar nuestro ROBOT LIMPIAMARES vía WIFI utilizando la aplicación BLYNK (<https://blynk.io>) y programado con ArduinoBlocks (<http://www.arduinoblocks.com>)

En estos primeros ejemplos vamos a usar el propio servidor de **BLYNK**, por lo que deberemos de registrarnos y crear una nueva cuenta directamente desde la APP.

Es recomendable crearse un nuevo correo electrónico para utilizar la APP.

Esa cuenta debe de ser válida, ya que recibiremos mensajes en ella con códigos (*tokens*) que vamos a necesitar posteriormente para realizar los programas en **ArduinoBlocks**.



CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

Para esta primera práctica vamos a darnos de alta en BLYNK y hacer nuestro primer programa para encender y apagar el LED DO de la placa NodeMCU.

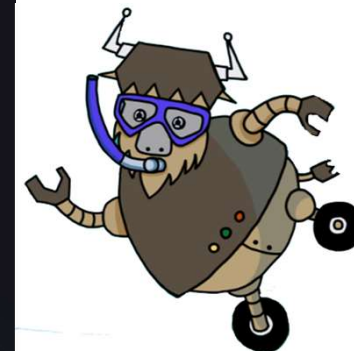
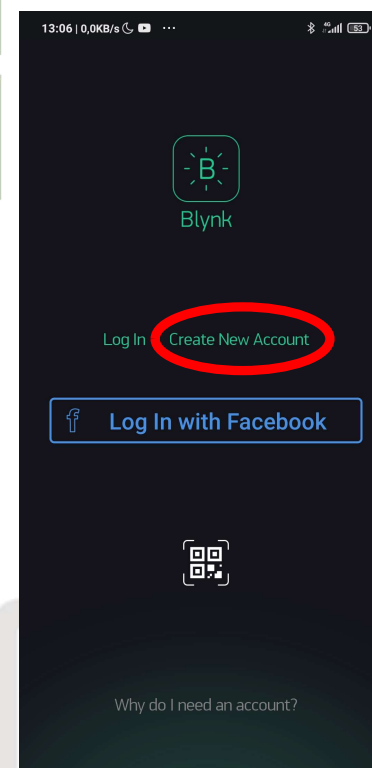
Lo primero será descargar en nuestro dispositivo móvil la aplicación BLYNK.

<https://play.google.com/store/apps/details?id=cc.blynk&hl=es>

<https://apps.apple.com/us/app/blynk-iot-for-arduino-esp32/id808760481>

Una vez descargada e instalada, crearemos una nueva cuenta con una cuenta de correo electrónico real.

Recomendamos crearse una cuenta nueva para este fin.



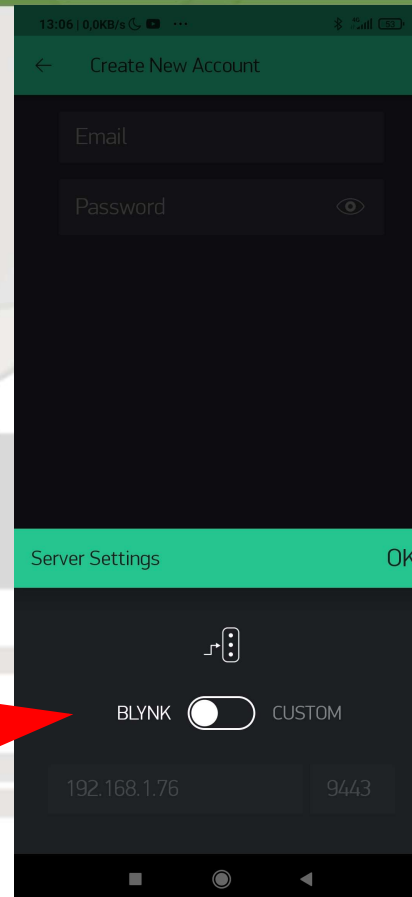
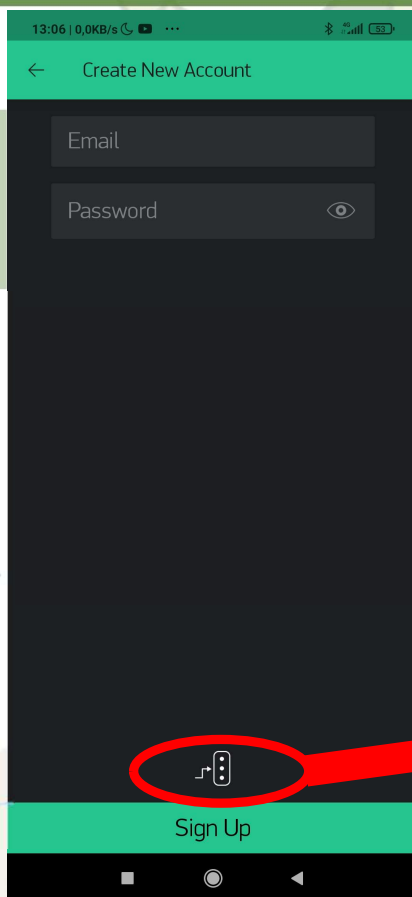
CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

Introducimos nuestro Email y una contraseña y clicaremos sobre **Sign Up**



(CC) BY-NC-SA



En **Server Settings** tenemos dos opciones: **BLYNK** y **CUSTOM**.

Para usar el propio servidor de BLYNK seleccionaremos esta opción.

La opción de **CUSTOM** donde habrá que introducir una dirección IP y una puerta no será necesaria en esta ocasión.

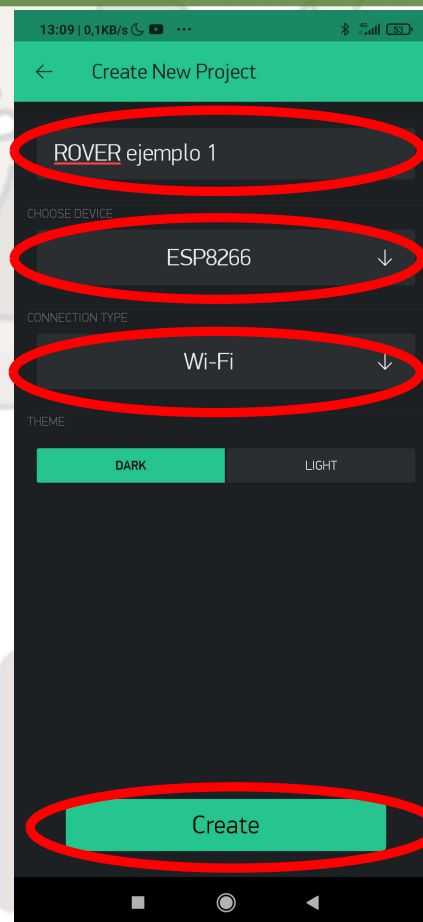
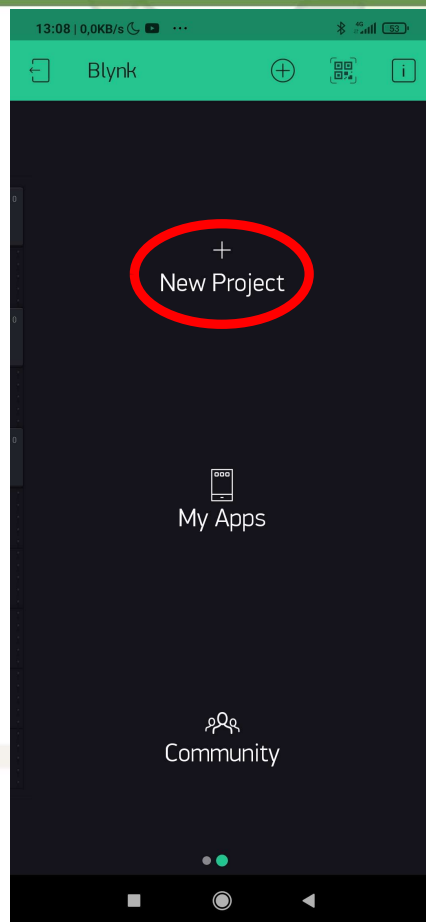
CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

Una vez logueados crearemos un **NUEVO PROYECTO**.



(CC) BY-NC-SA



A ese nuevo proyecto le daremos un nombre identificativo, seleccionaremos el tipo de dispositivo que vamos a usar; en nuestro caso **ESP8266** y por último el tipo de conexión; **WIFI**.

También podremos elegir el tipo de tema; oscuro o brillante.

Clicaremos en **CREATE**.

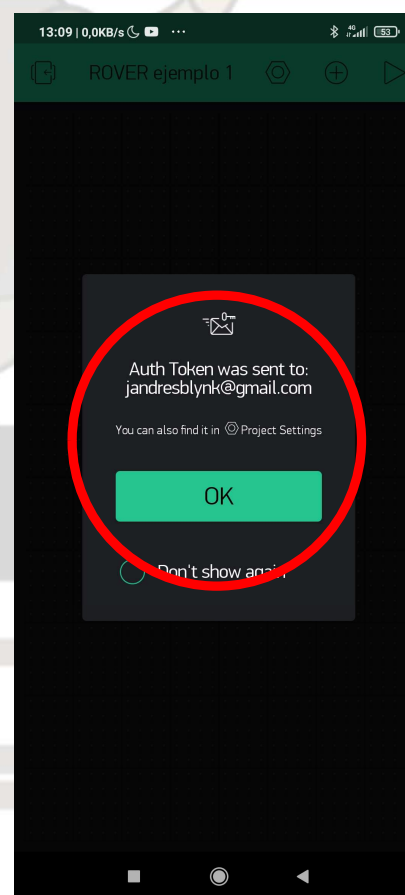
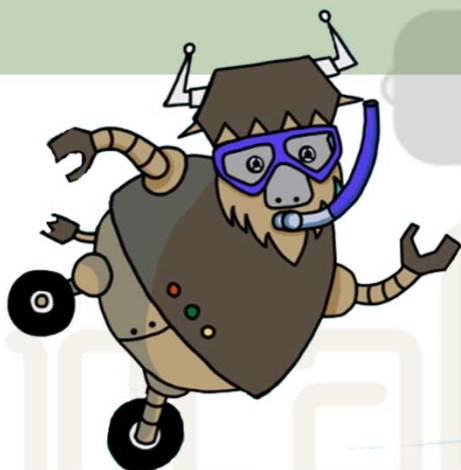
CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

Una vez dado a **CREATE** nos aparecerá el siguiente mensaje en el que nos pide permiso para enviar el **TOKEN** a la cuenta de correo con la que nos hemos logueado.

Ese **TOKEN** es un código único que necesitaremos para realizar el programa con *ArduinoBlock*. Ya se verá.

Clicaremos en **OK**

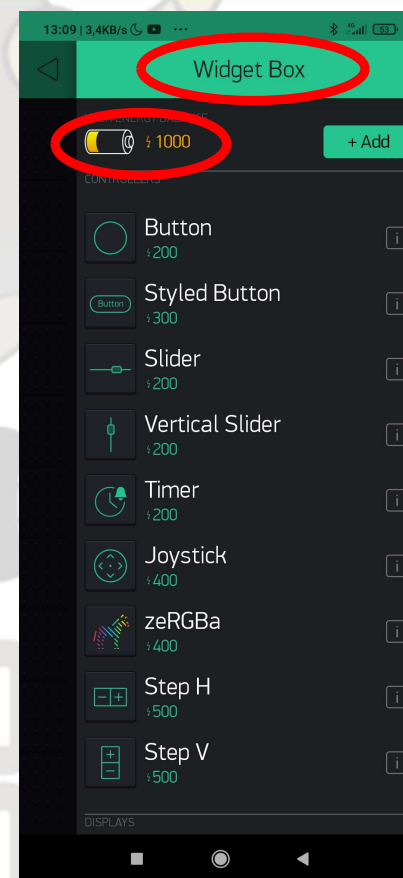
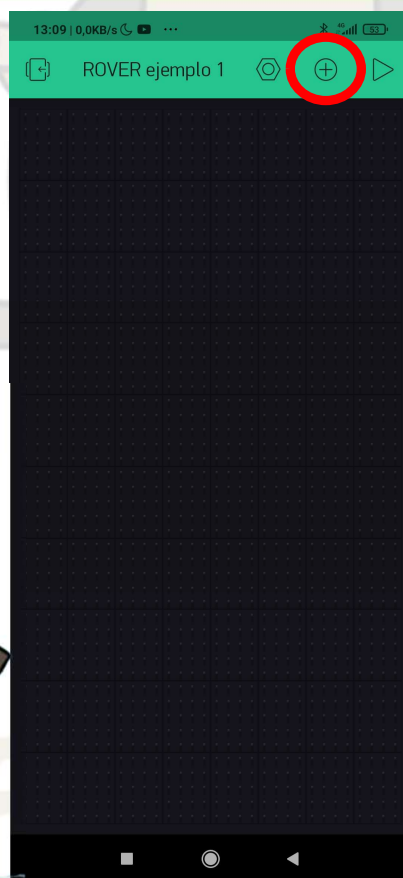


CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

En la siguiente pantalla aparecerá vacío el panel sobre el que añadiremos todos los widget que vamos a utilizar.

Clicaremos en el botón + para abrir el panel **Widget Box**



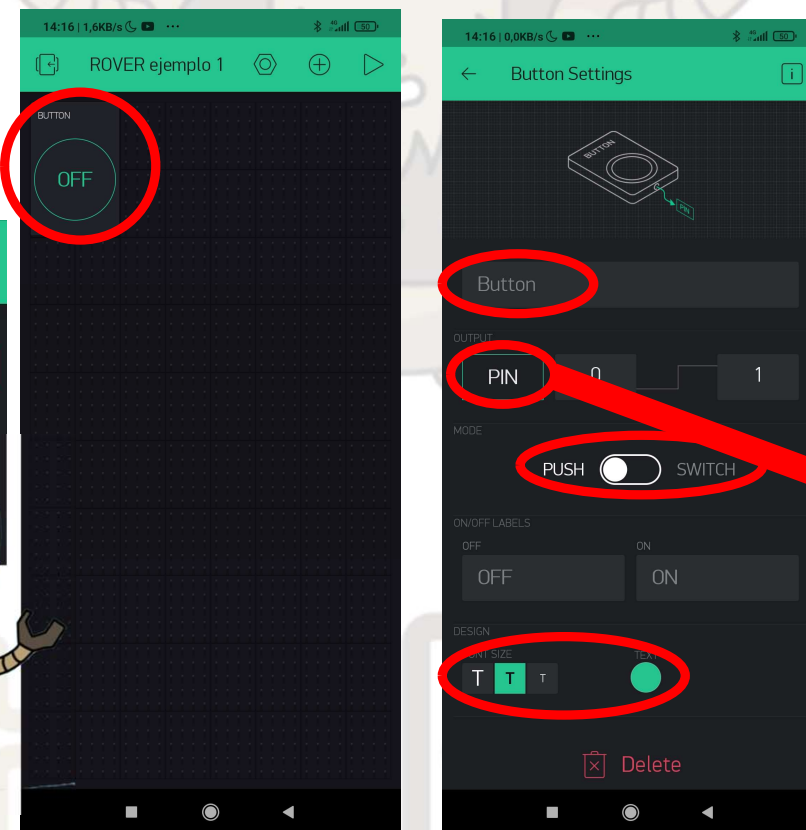
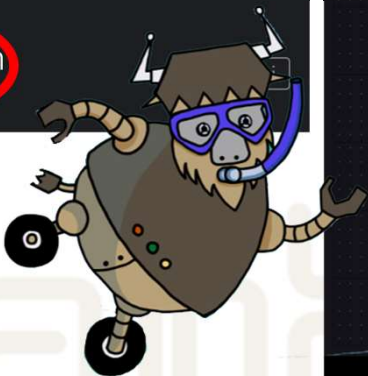
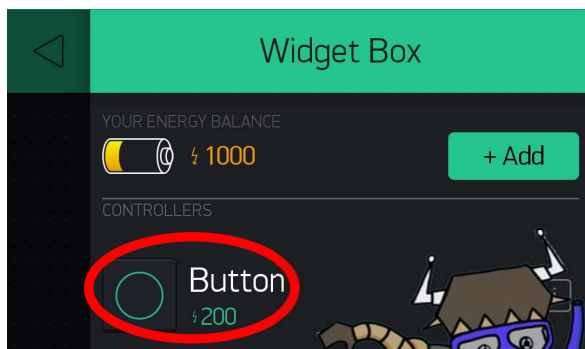
Usando la opción del servidor BLYNK tenemos limitada la "energía", es decir, limitados el número de widget que podemos usar.

Esta limitación en la opción de servidor local no se tiene.

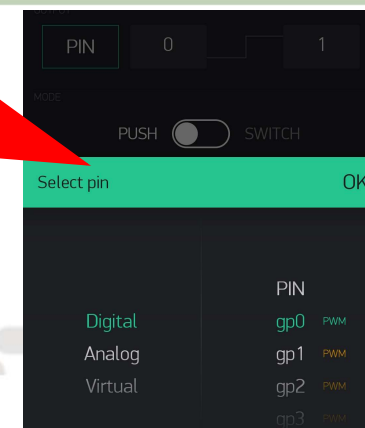
CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

Para este ejemplo elegimos el widget **Button**.



Al clicar sobre el botón entraremos en su pantalla de configuración donde podremos cambiar el nombre, seleccionar el PIN, el color, el tamaño de texto y si queremos que funcione como un botón o como un pulsador.



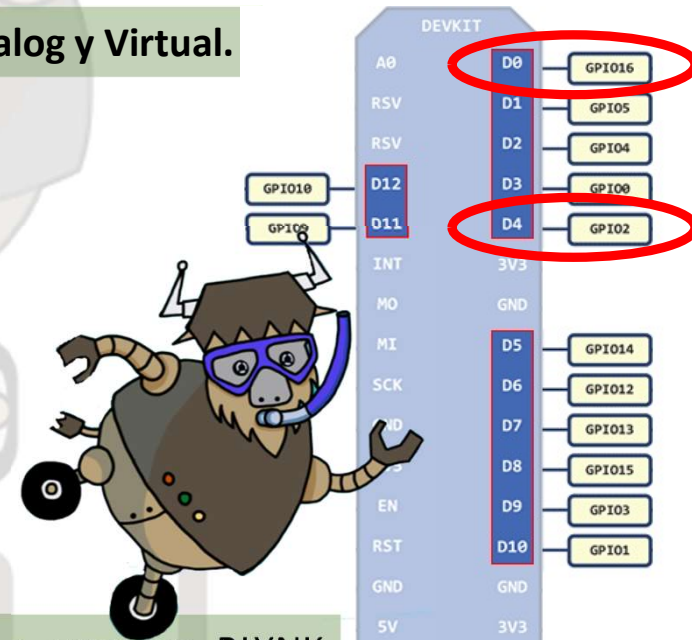
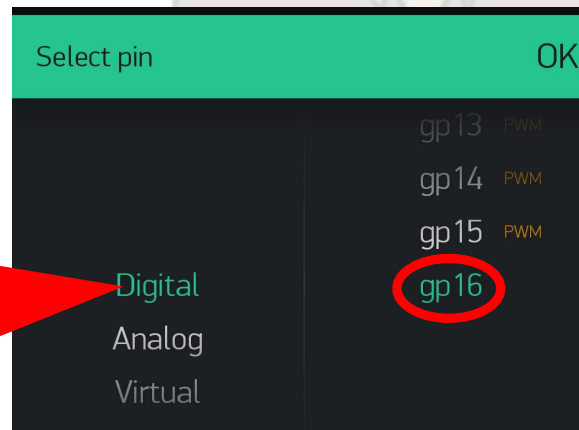
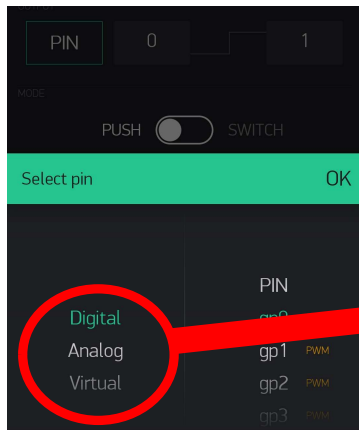
ascentic

CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

Selección del PIN DIGITAL

Existen 3 tipos de PINES; Digital, Analog y Virtual.



IMPORTANTE!!! En este ejemplo el pin DIGITAL que vamos a usar en BLYNK corresponde con el GPIO 16 (gp16) que es el D0

CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

Ya hemos terminado la parte correspondiente a BLYNK, ahora vamos a empezar con el programa de **ArduinoBlocks**.

Vamos a necesitar el TOKEN que BLYNK nos ha mandado a nuestra cuenta de correo. (*Mirar en el SPAM si no se ve*).

☆ Blynk Auth Token for ROVER ejemplo 1 project and device ROVER ejemplo 1 - Auth Token : N6n1C-d7n-F_8FyvPJ8NGPS4MkogaicC Happy Blynking! - Getting Star...



Blynk <dispatcher@blynk.io> Anular suscripción para mí ▾

Auth Token : N6n1C-d7n-F_8FyvPJ8NGPS4MkogaicC

Happy Blynking!

Getting Started Guide -> <https://www.blynk.cc/getting-started>

Documentation -> <http://docs.blynk.cc/>

Sketch generator -> <https://examples.blynk.cc/>

Latest Blynk library -> https://github.com/blynkkk/blynk-library/releases/download/v0.6.1/Blynk_Release_v0.6.1.zip

Latest Blynk server -> <https://github.com/blynkkk/blynk-server/releases/download/v0.41.13/server-0.41.13.jar>

<https://www.blynk.cc>

twitter.com/blynk_app

www.facebook.com/blynkapp

CONTROL WIFI LIMPIAMARES

1.- ON/OFF LED PIN D0

El programa de **ArduinoBlocks** es tremendamente sencillo ya que sólo habría que inicializar la placa NodeMCU con la wifi en la que estemos trabajando y pegar el TOKEN.

Inicializar

B Iniciar

WiFi SSID

WiFi clave

IP servidor 0 . 0 . 0 . 0 Puerto 8080

Código (Auth) N6n1C-d7n-F_8FyvPJ8NGPS4MkogaicC

Bucle

WIFI de conexión del móvil y de la NodeMCU (misma wifi)

Clave de la conexión WIFI

IP y Puerto; dejar en 0.0.0.0 y 8080

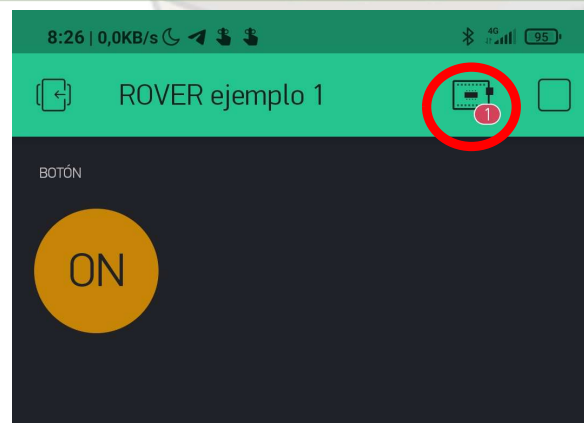
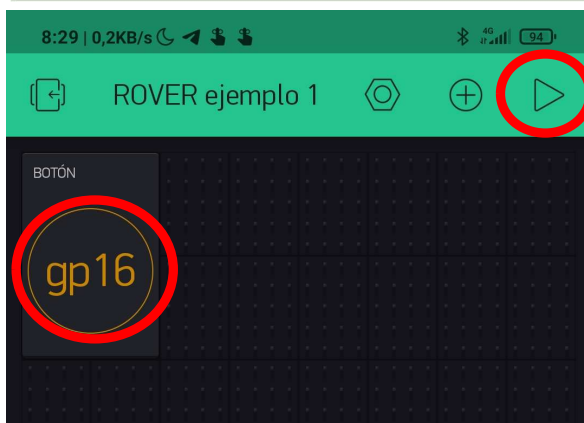
TOKEN recibido por email.



CONTROL WIFI LIMPIAMARES

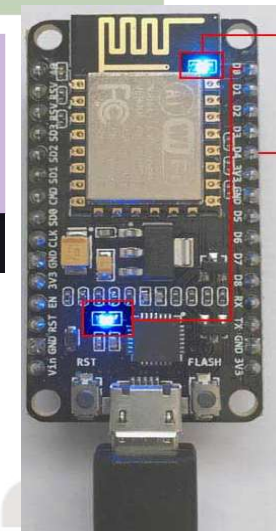
1.- ON/OFF LED PIN D0

Una vez cargado el programa en la placa está todo preparado para funcionar. Vamos a la aplicación BLYNK y clicamos en **PLAY**. En ese momento ya podremos controlar el LED D0 con el botón que hemos creado en BLYNK



Si aparece este mensaje de error es que la placa no está conectada.

ROVER ejemplo 1
Offline since 10:41 Apr 8, 2021



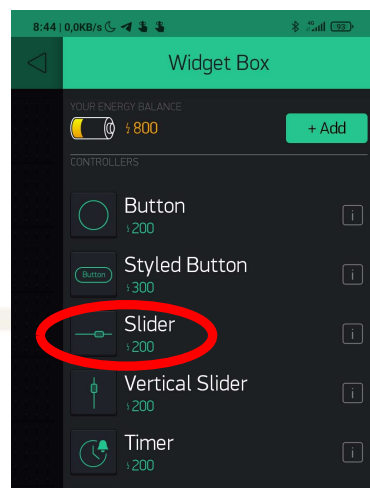
RECUERDA!!! El **gp16** corresponde al **D0** y el **gp2** al **D4**.

ACTIVIDAD: - Crea otro botón para controlar el LED D4 (gp2).

CONTROL WIFI LIMPIAMARES

2.- CONTROL DE BRILLO LED PIN D4

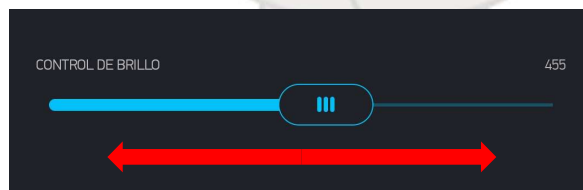
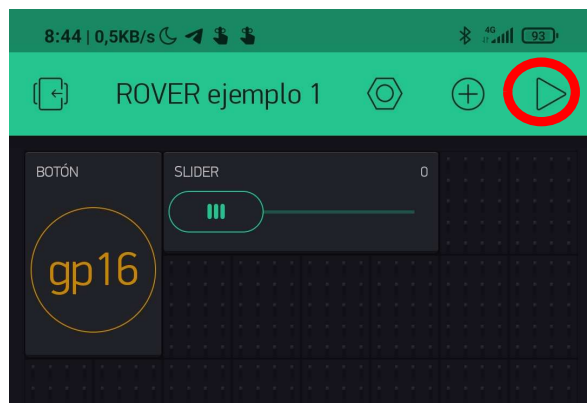
Continuamos con BLYNK y en este caso vamos a controlar el brillo del LED D4. Vamos a utilizar el mismo programa anterior ampliándolo con un widget **SLIDER**. Al entrar en propiedades elegimos **GP2** y cambiamos de **0 a 1023** por de **1023 a 0** ya que estos LEDs tienen la *polaridad invertida*, es decir, para encenderlos hay que poner un estado LOW y para apagarlos un estado HIGH.



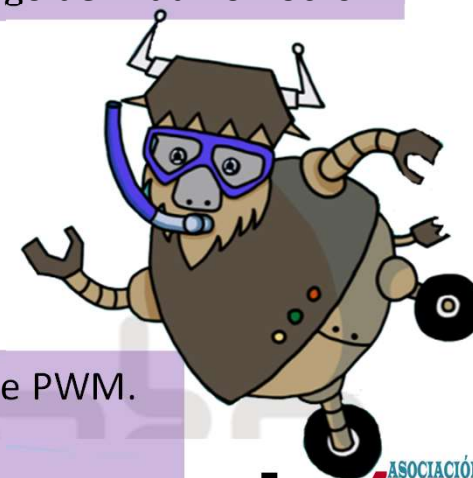
CONTROL WIFI LIMPIAMARES

2.- CONTROL DE BRILLO LED PIN D4

Una vez configurado el widget SLIDER clicamos en PLAY y comprobamos el funcionamiento deslizando el widget hacia un lado y otro.



OBSERVA!!! Como no has tenido que modificar el código de ArduinoBlocks.



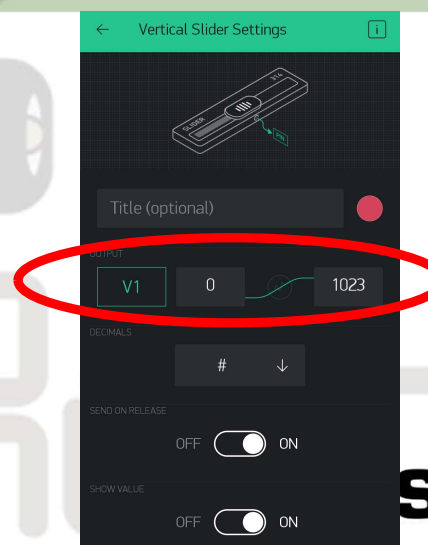
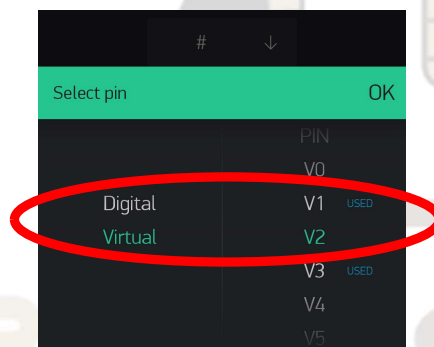
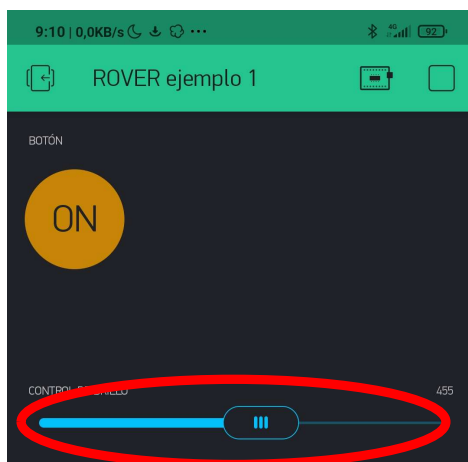
RECUERDA!!! Con el **gp16** (LED D0) este control no se puede hacer ya que no dispone de PWM.

ACTIVIDAD: .- Prueba a utilizar un SLIDER VERTICAL.

CONTROL WIFI LIMPIAMARES

3.- CONTROL DE BRILLO LED PIN D4 CON PINES VIRTUALES

Utilizando los pines virtuales se puede intercambiar información de forma muy sencilla entre la APP y la placa **NodeMCU**. En este ejemplo utilizaremos los pines virtuales para controlar el brillo del LED (D4). Aparentemente el funcionamiento va a ser el mismo que en el ejemplo anterior, pero más adelante veremos su verdadera utilidad. Usando el ejemplo anterior cambiaremos el Pin DIGITAL por un Pin VIRTUAL y seleccionaremos el V1



CONTROL WIFI LIMPIAMARES

3.- CONTROL DE BRILLO LED PIN D4 CON PINES VIRTUALES

El programa que necesitamos cargar en la NodeMCU con ArduinoBlocks es el siguiente;

The screenshot shows an ArduinoBlocks program with the following structure:

- Inicializar:** A block labeled 'Iniciar' with fields for WiFi SSID, WiFi clave, IP servidor (0.0.0.0), Puerto (8080), and Código (Auth) (N6n1C-d7n-F_8FyvPJ8NGPS4MkogaicC).
- Bucle:** A loop block containing:
 - A 'Valor recibido por V1' block (circled in red).
 - An 'Establecer INTENSIDAD LED = Valor del pin virtual (numérico)' block.
 - A 'Led intensidad (PWM)' block with 'Pin D4' (circled in red) and 'Valor INTENSIDAD LED'.

A cartoon robot character is positioned to the right of the code blocks.

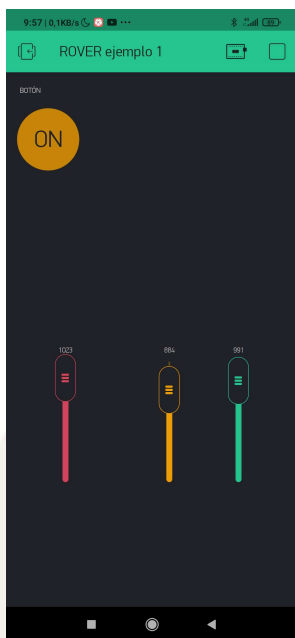
RECUERDA!!! Al estar reutilizando el mismo programa no tenemos que cambiar el TOKEN. En caso de crear un programa nuevo tendremos que actualizar el TOKEN.

Prueba el funcionamiento de todo.

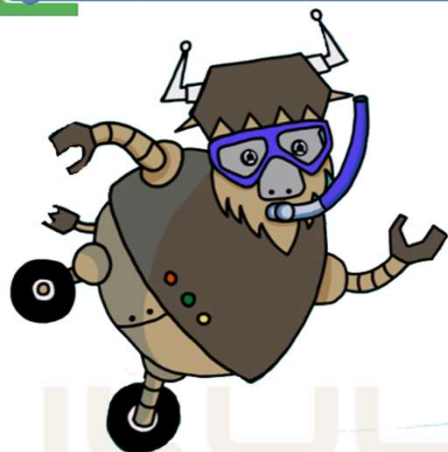
CONTROL WIFI LIMPIAMARES

4.- CONTROL DE BRILLO LEDS D1, D2 y D3 CON PINES VIRTUALES

Ampliando el ejercicio anterior podríamos conectar 3 LEDs a la placa NodeMCU y controlarlos con pines Virtuales desde BLYNK. Este sería el ejemplo:



```
Inicializar
B Blynk Iniciar
WiFi SSID [ ]
WiFi clave [ ]
IP servidor 0 . 0 . 0 . 0 Puerto 8080
Código (Auth) N6n1C-d7n-F_8FyvPJ8NGPS4MkogaicC
```



```
Bucle
B Blynk Valor recibido para V1
  Establecer INTENSIDAD LED = B Blynk Valor del pin virtual (numérico)
  Led intensidad (PWM) Pin D1 Valor INTENSIDAD LED

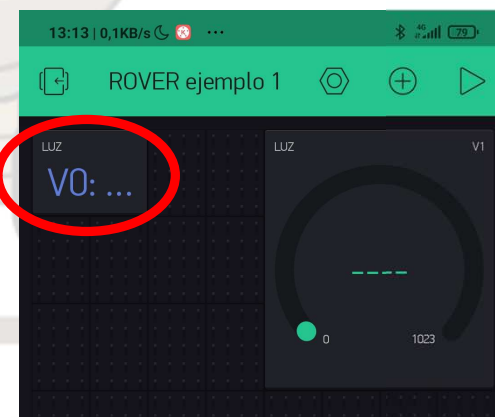
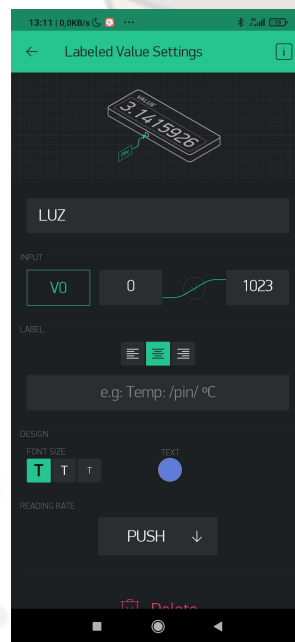
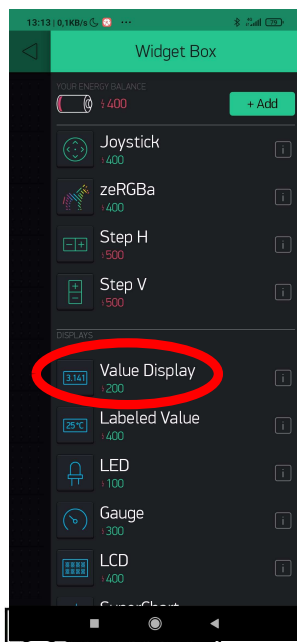
B Blynk Valor recibido para V2
  Establecer INTENSIDAD LED = B Blynk Valor del pin virtual (numérico)
  Led intensidad (PWM) Pin D2 Valor INTENSIDAD LED

B Blynk Valor recibido para V3
  Establecer INTENSIDAD LED = B Blynk Valor del pin virtual (numérico)
  Led intensidad (PWM) Pin D3 Valor INTENSIDAD LED
```


CONTROL WIFI LIMPIAMARES

6.- RECEPCIÓN CON PINES VIRTUALES

En esta ocasión vamos a enviar valores de la placa NodeMCU para ser visualizados por pantalla por BLYNK. Creamos un nuevo programa y copiamos el nuevo TOKEN enviado al correo. En BLINK elegiremos el widget **Value Display**. En su configuración seleccionaremos Pin **Virtual V0** y le daremos el nombre de **LUZ**.

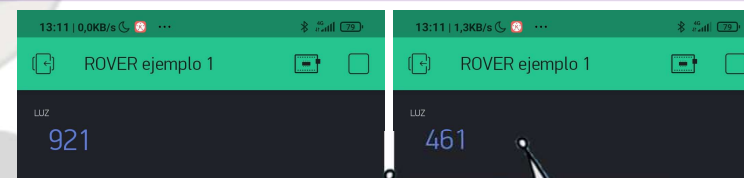


CONTROL WIFI LIMPIAMARES

6.- RECEPCIÓN CON PINES VIRTUALES

En **ArduinoBlocks** crearemos el siguiente programa y lo cargaremos en la placa **NodeMCU**. Este programa enviará el valor una LDR cada segundo. Damos al PLAY en la APP y veremos los resultados.

Para realizar esta práctica es necesario la conexión de un sensor LDR en la entrada analógica, la A0.

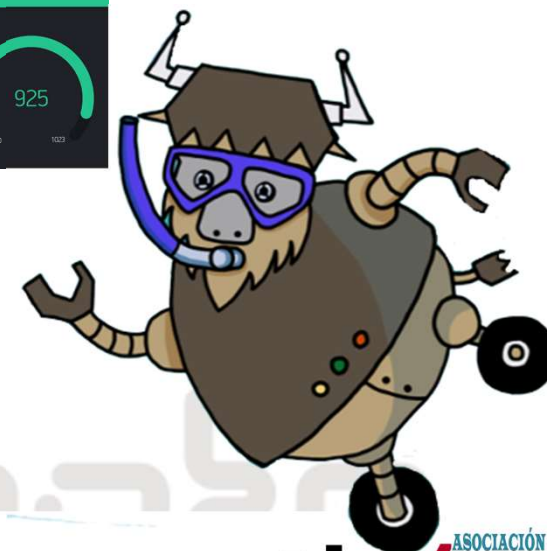
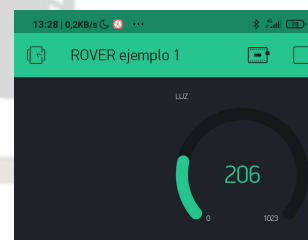
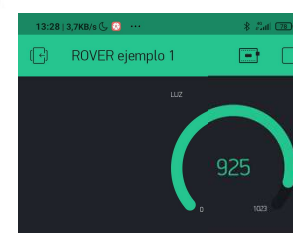
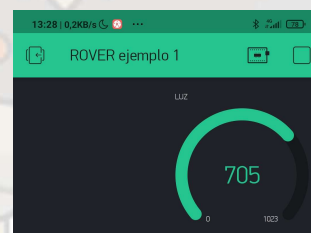
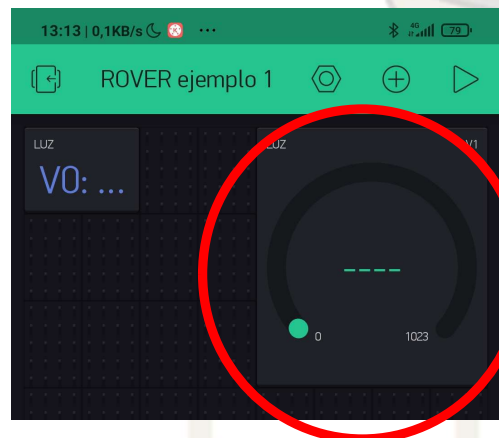
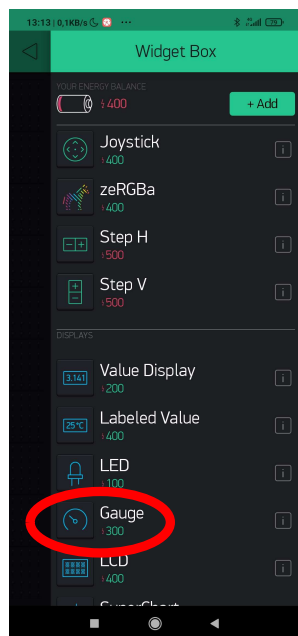


RECUERDA!!! La placa NodeMCU sólo tiene una entrada analógica, la A0.

CONTROL WIFI LIMPIAMARES

6.- RECEPCIÓN CON PINES VIRTUALES

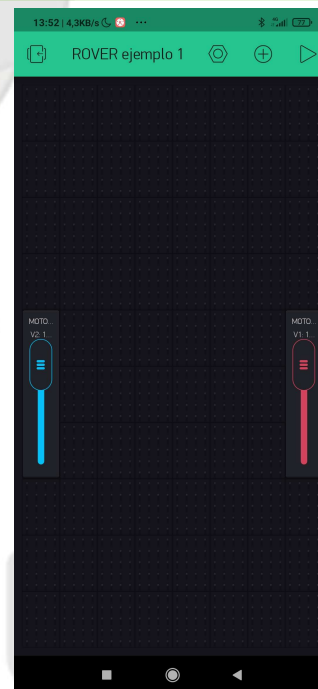
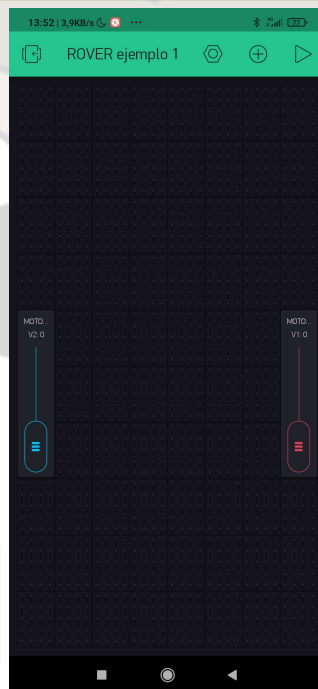
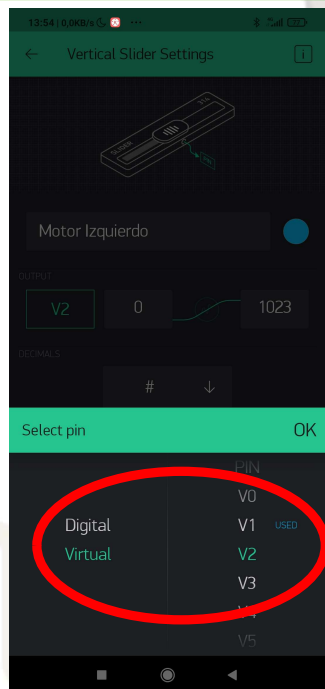
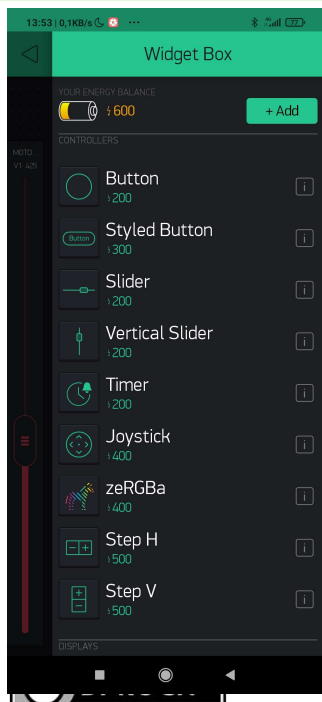
Podríamos hacer lo mismo, pero utilizando otro widget, el **Gauge**. Configúralo para el Pin **Virtual V0**.



CONTROL WIFI LIMPIAMARES

7.- CONTROL DE MOTORES

En esta actividad vamos a controlar los motores de nuestro LIMPIAMARES utilizando los pines **VIRTUALES** de BLYNK. Vamos a usar dos widget **Vertical Slider**. Y los configuramos como **V1** y **V2**.



CONTROL WIFI LIMPIAMARES

7.- CONTROL DE MOTORES

Ahora realizaremos el programa en **ArduinoBlocks**. Primero recordar los pines de conexión de los motores en el driver de motores.

MOTOR A	D1, D3
DIRECCIÓN MOTOR A (DA)	D3
PWM MOTOR A	D1
MOTOR B	D2, D4
DIRECCIÓN MOTOR B (DB)	D4
PWM MOTOR B	D2

```

Inicializar
  B Iniciar
  WiFi SSID
  WiFi clave
  IP servidor 0 . 0 . 0 . 0 Puerto 8080
  Código (Auth) N6n1C-d7n-F_8FyvPJ8NGPS4MkogaicC
  
```

```

Bucle
  B Valor recibido para V1
  Establecer MOTOR DERECHO = B Valor del pin virtual (numérico)
  Motor DC Pin D1 ON Pin D3 ON Pin-PWM D1 MOTOR DERECHO
  B Valor recibido para V2
  Establecer MOTOR IZQUIERDO = B Valor del pin virtual (numérico)
  Motor DC Pin D2 ON Pin D4 ON Pin-PWM D2 MOTOR IZQUIERDO
  
```



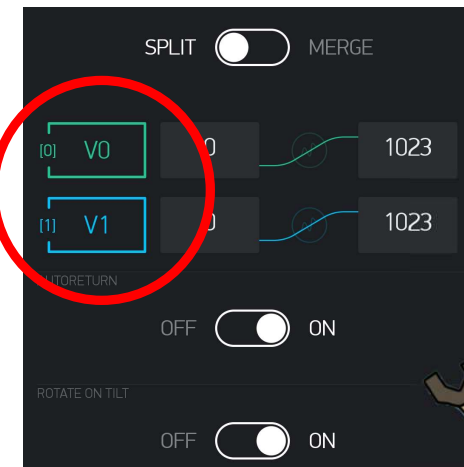
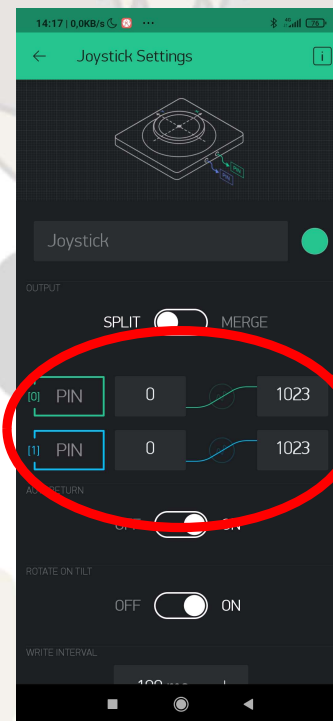
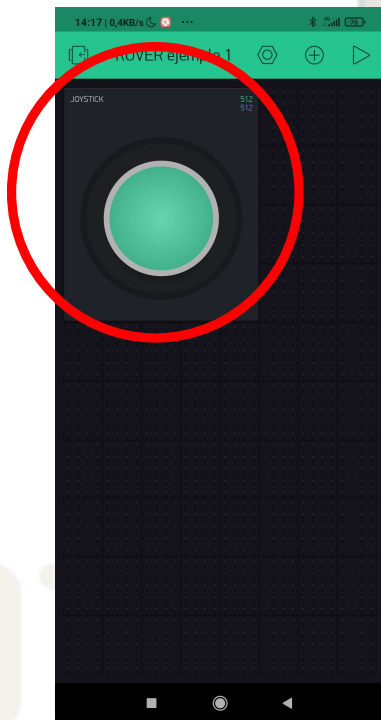
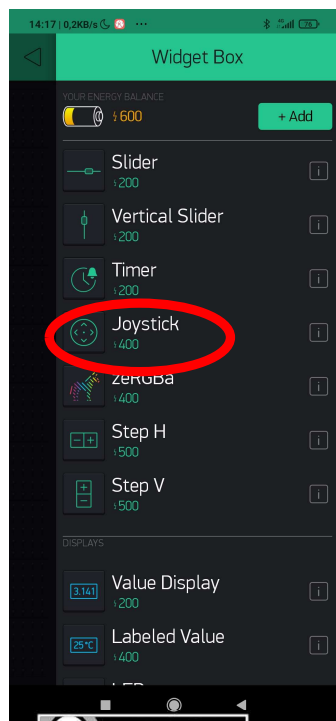
Ya puedes probar los motores deslizando los controles arriba y abajo.



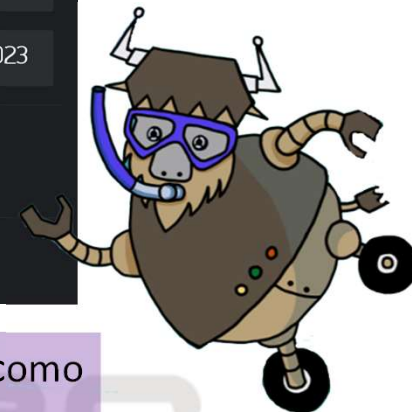
CONTROL WIFI LIMPIAMARES

8.- CONTROL DE MOTORES II.

En esta ocasión controlaremos los motores utilizando el widget **JOYSTICK**.



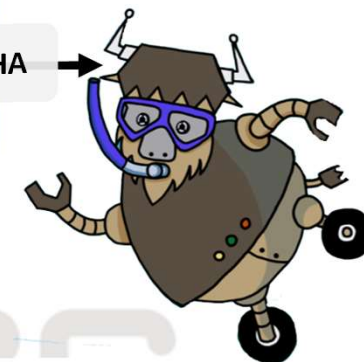
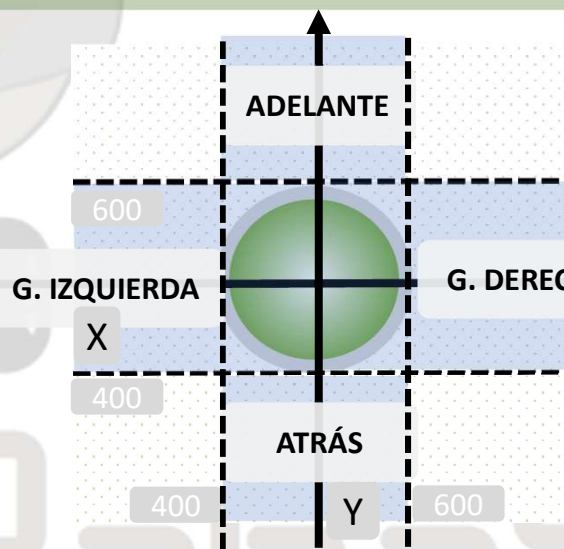
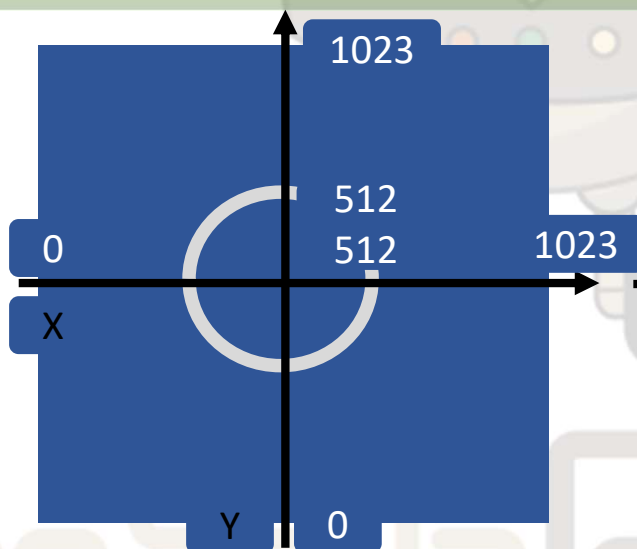
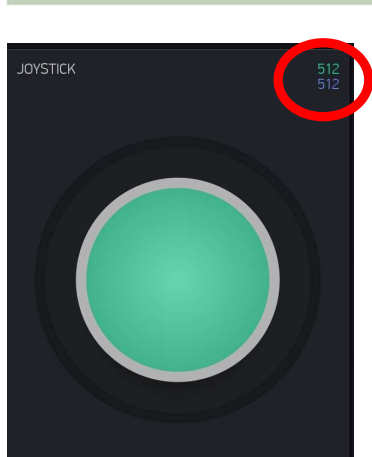
Configuramos los pines como pines **VIRTUALES V0 y V1**.



CONTROL WIFI LIMPIAMARES

8.- CONTROL DE MOTORES II.

El widget **JOYSTICK** tiene dos ejes (X e Y) cuyo valor oscila entre 0 y 1023. En el punto de reposo, en el centro, el valor de X = 512 e Y = 512.

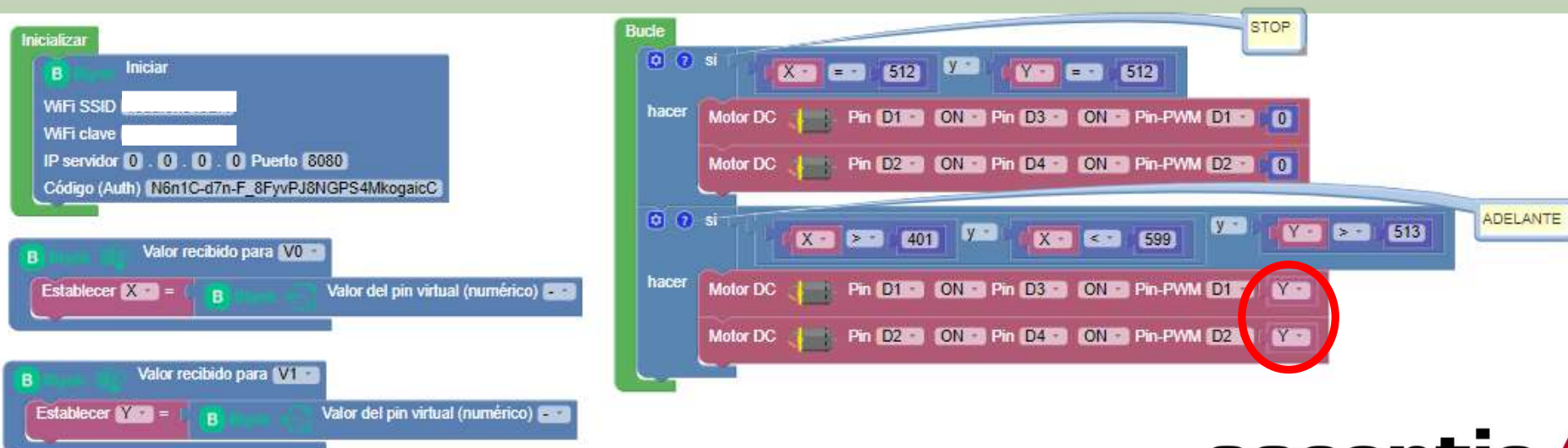


Para hacer el programa en ArduinoBlocks utilizaremos rangos de valores entre 400 y 600.
 Lo vemos en la próxima diapositiva.

CONTROL WIFI LIMPIAMARES

8.- CONTROL DE MOTORES II.

Este sería el programa en **ArduinoBlocks**, para simplificarlo sólo están reflejadas dos acciones; *motores parados* y *motores hacia adelante*. Para estar los motores parados **X** e **Y** deben tener un valor de 512. Para ir hacia adelante **X** debe ser *mayor de 401 y menor de 599* e **Y** *mayor de 513*. El valor PWM de los motores será directamente el valor de **Y**.



CONTROL WIFI LIMPIAMARES

8.- CONTROL DE MOTORES II.

El programa de control de motores completo sería el siguiente;

