



PROGRAMACIÓN VEHÍCULO ROVER



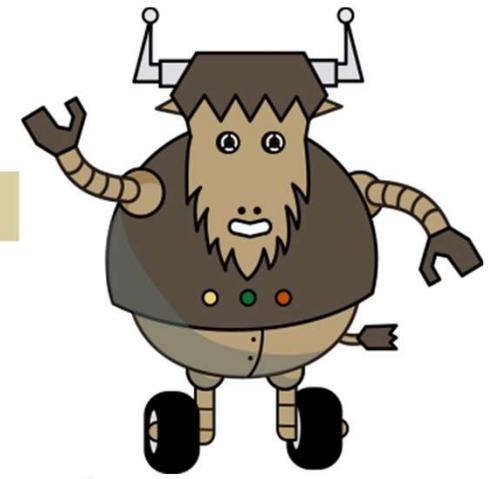


CONTROL WIFI VEHÍCULO ROVER

La programación del vehículo ROVER se realiza con un software de programación gráfico llamado **ArduinoBlocks**.



Una vez logueados y dados de alta en ArduinoBlocks iniciaremos un proyecto personal.





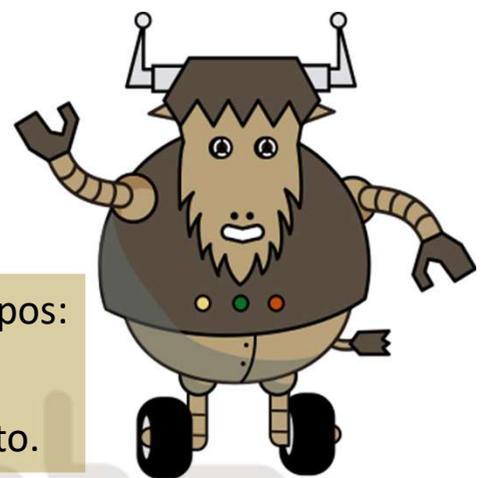
CONTROL WIFI VEHÍCULO ROVER

En TIPO DE PROYECTO seleccionaremos NodeMCU



Después daremos un nombre al programa y opcionalmente rellenaremos el resto de campos: **Descripción, Componentes y Comentarios.**

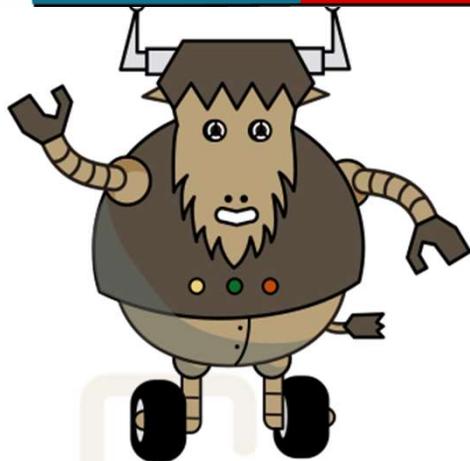
*Estos campos no son obligatorios, pero sí muy recomendables para tener datos del proyecto.





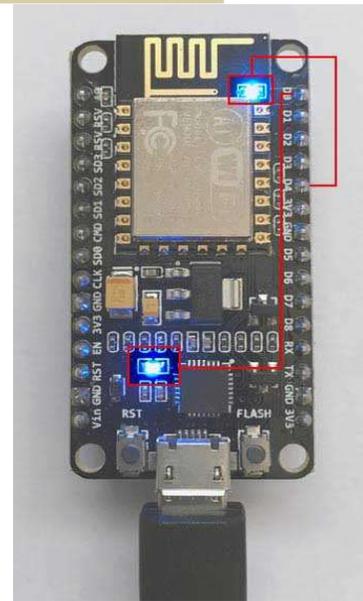
PROGRAMACIÓN VEHÍCULO ROVER

La placa NodeMCU dispone de dos LEDs integrados que podemos programar y que vamos a utilizar para dar nuestros primeros pasos con ArduinoBlocks.



Los pines asociados a esos LEDs son el **D0** y el **D4**. Ver imagen:

IMPORTANTE!!! Estos LEDs tienen la *polaridad invertida*, es decir, para encenderlos hay que poner un estado **LOW** y para apagarlos un estado **HIGH**

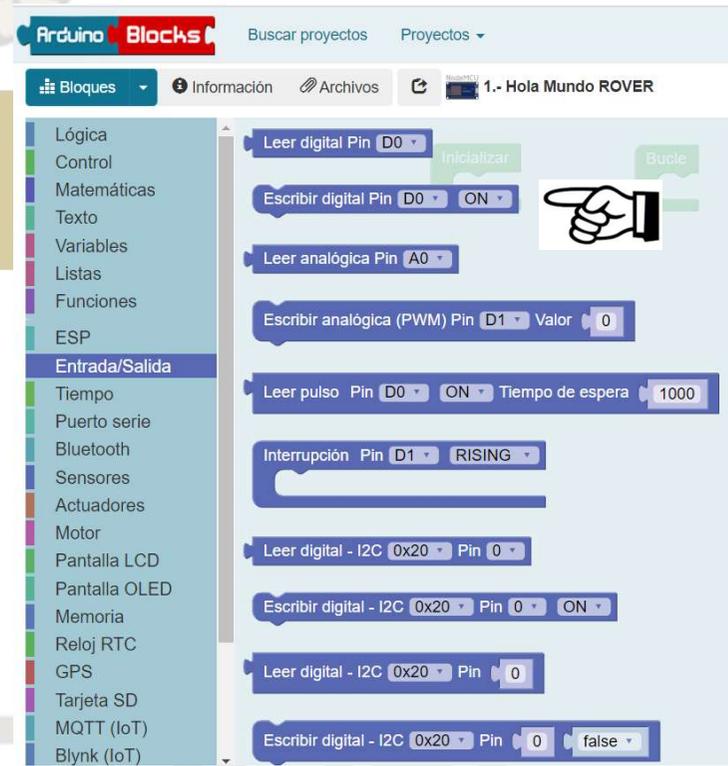




PROGRAMACIÓN VEHÍCULO ROVER

1.- Primer Programa. Hola mundo!!! (ON/OFF LED D0)

Para probar que efectivamente estos LEDs tienen la *polaridad invertida* escribir y cargar el programa A y comprobar que el LED está apagado y después realizar el programa B.



PROGRAMA A



PROGRAMA B



ACTIVIDAD. Prueba el mismo programa con el LED D4.



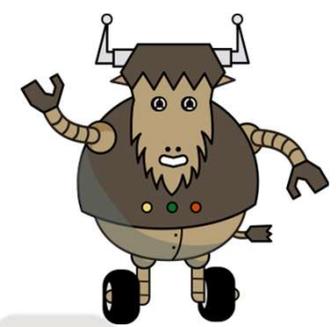
PROGRAMACIÓN VEHÍCULO ROVER

3.- Repetir X veces...

Dentro del grupo de bloques de CONTROL existe el bloque REPETIR__veces hacer__. Vamos a utilizarlo para crear una secuencia de encendido/apagado del LED D0 5 veces seguidas y después que se encienda y apague el LED D4. El programa sería el siguiente:



```
Inicio
  Inicializar
  Bucle
    repetir 5 veces
      hacer
        Escribir digital Pin D0 OFF
        Esperar 250 milisegundos
        Escribir digital Pin D0 ON
        Esperar 500 milisegundos
    Escribir digital Pin D4 OFF
    Esperar 1000 milisegundos
    Escribir digital Pin D4 ON
    Esperar 1000 milisegundos
```



ACTIVIDAD. - Modifica los pines, LEDs y las esperas y comprueba el resultado.



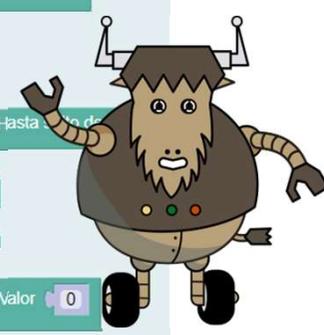
PROGRAMACIÓN VEHÍCULO ROVER

4.- Puerto Serie.

Podemos realizar la lectura de estados o valores utilizando la consola serie que nos ofrece ArduinoBlocks. Dentro del grupo de bloques de **PUERTO SERIE** tenemos los bloques que vamos a necesitar.

Para iniciar la **consola serie**, en **INICIALIZAR**, debemos de iniciar la comunicación con el bloque de Iniciar Baudios _____

***Baudios** es una medida utilizada en telecomunicaciones que representa el número de símbolos por segundo en un medio digital.





PROGRAMACIÓN VEHÍCULO ROVER

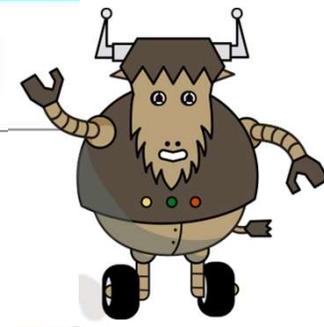
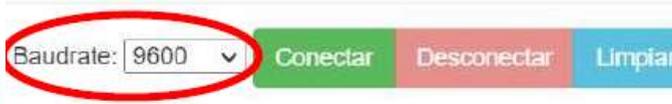
4.- Puerto Serie.

Utilizaremos una velocidad de 9600 que para nuestro proyecto es suficiente. Es importante recordar que esa velocidad debe ser la misma que la seleccionada en la consola.

Con el bloque de enviar escribiremos “LED D4 ENCENDIDO” y “LED D4 APAGADO” y finalizaremos el programa según la imagen.

```
Inicializar
  Iniciar Baudios 9600

Bucle
  Escribir digital Pin D4 OFF
  Enviar "LED D4 ENCENDIDO"
  Esperar 1000 milisegundos
  Escribir digital Pin D4 ON
  Enviar "LED D4 APAGADO"
  Esperar 1000 milisegundos
```



*Para ver los mensajes por la pantalla del PC hay que clicar sobre **CONSOLA** y después en **CONECTAR**.



PROGRAMACIÓN VEHÍCULO ROVER

4.- Puerto Serie. Parte II

También podemos utilizar el PUERTO SERIE para controlar el encendido y apagado de los LED de la placa NodeMCU enviando órdenes desde el PC.

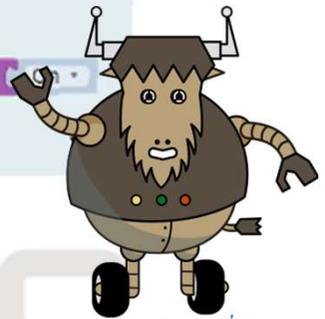
Para ello se debe crear una variable especial tipo TEXTO que llamaremos "TEXTO RECIBIDO". Y esta variable la asociaremos al bloque de **RECIBIR TEXTO**.

```
Establecer TEXTO RECIBIDO = Recibir texto Hasta salto de línea
```

Utilizaremos el siguiente condicional y dentro de él completaremos todo el programa:

```
Bucle  
si  
hacer  
¿Datos recibidos?
```

```
Lógica  
Control  
Matemáticas  
Texto  
Variables  
Listas  
Funciones  
ESP  
Entrada/Salida  
Tiempo  
Puerto serie  
Bluetooth  
Establecer varNum = 0  
varNum  
Establecer varTexto =  
varTexto  
Establecer varBool =  
varBool
```





PROGRAMACIÓN VEHÍCULO ROVER

4.- Puerto Serie. Parte II

El programa completo será el siguiente:

Utilizaremos el bloque de  ya que se trata de variables de textos, no numéricas.

```
Inicializar
  Iniciar Baudios 9600

Bucle
  si ¿Datos recibidos?
  hacer
    Establecer TEXTO RECIBIDO = Recibir texto Hasta salto de línea
    si TEXTO RECIBIDO igual a " ENCENDIDO "
    hacer Escribir digital Pin D4 OFF
    si TEXTO RECIBIDO igual a " APAGADO "
    hacer Escribir digital Pin D4 ON
```

1.- Hola Mundo ROV

- Lógica
- Control
- Matemáticas
- Texto
- Variables
- Listas
- Funciones
- ESP
- Entrada/Salida
- Tiempo
- Puerto serie
- Bluetooth
- Sensores
- Actuadores

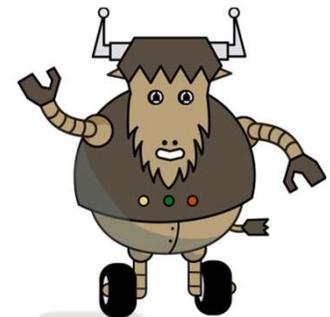




PROGRAMACIÓN VEHÍCULO ROVER

4.- Puerto Serie. Parte II

Una vez cargado el programa en la placa, abrimos la consola serie, **CONECTAMOS** y escribimos los mensajes que queremos enviar en la siguiente casilla y damos a **ENVIAR**.



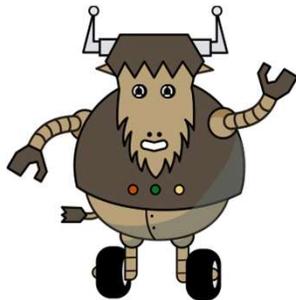
ACTIVIDAD. .- Cambia los textos y los pines y comprueba el resultado.



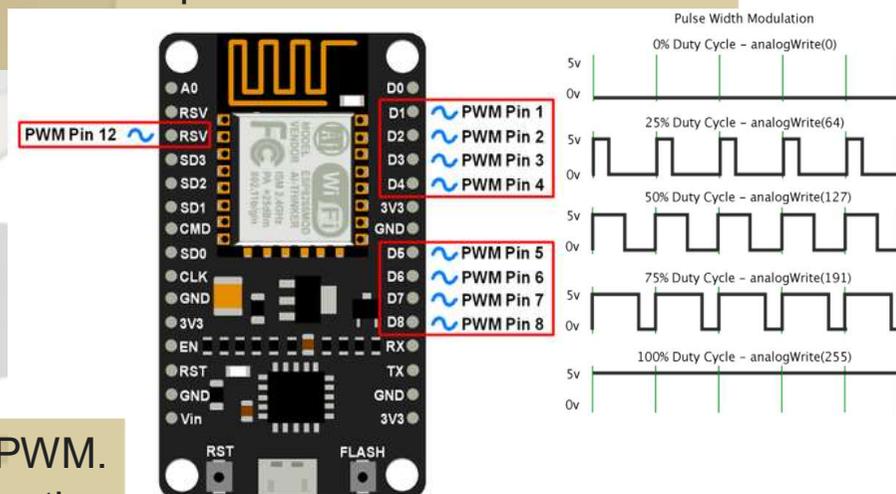
PROGRAMACIÓN VEHÍCULO ROVER

5.- Control PWM.

Un control PWM (*pulse-width modulation*) modulación por ancho de pulso es una técnica en la que se modifica el ciclo de trabajo de una señal periódica que se utiliza para controlar la cantidad de energía que se envía a una carga.



No todos los pines de la placa NodeMCU tienen control PWM. Sólo los pines reflejados en la siguiente imagen, por ese motivo no podemos utilizar los pines D0 y D4 para tal fin.





PROGRAMACIÓN VEHÍCULO ROVER

5.- Control PWM.

ArduinoBlocks tiene un bloque específico para programar los pines como salida PWM. En ese bloque deberemos seleccionar el PIN y asignarle el valor que queramos.



El rango de valores entre los que pueden oscilar los pines PWM en la placa NodeMCU va de 0 a 1023. Con esa variación se puede controlar el brillo de leds y también se puede regular la velocidad de giro de motores cc. Lo veremos en el próximo tutorial.

