

CantabRobots 2021
EXPEDICIÓN A MARTE



PROGRAMACIÓN SENSÓRICA VEHÍCULO ROVER



CantabRobots

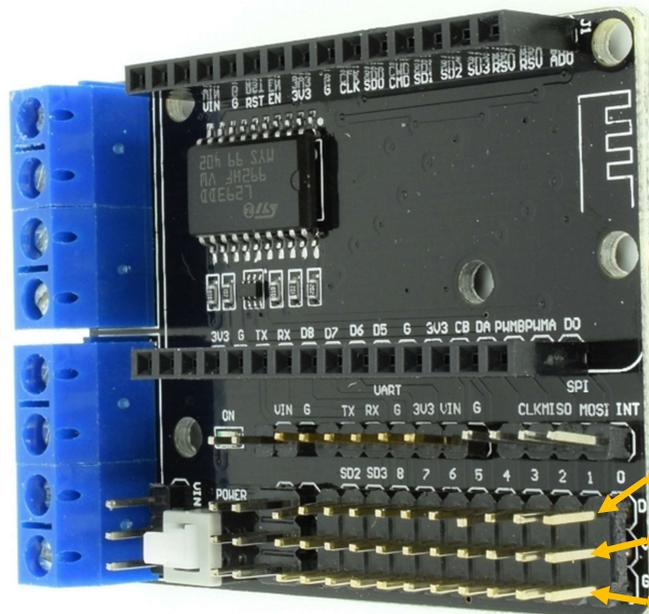




CONEXIONADO VEHÍCULO ROVER

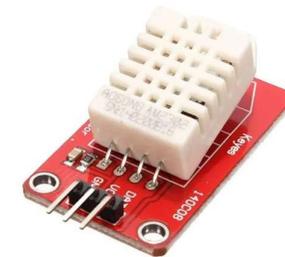
CONEXIONADO SENSORES

Conectar los sensores según la siguiente tabla:



- Señal
- Voltaje (+)
- Tierra (-)

| SENSOR | PIN |
|--------------|-----|
| DHT22 | D5 |
| LDR | A0 |
| EFEECTO HALL | D7 |



IMPORTANTE!!! Comprobar las conexiones G (GND), V (VCC) y D (SEÑAL) entre los sensores y la shield.



PROGRAMACIÓN SENSORIA VEHÍCULO ROVER

1.- Sensor DHT-22

El sensor DHT-22 es un sensor capaz de medir Temperatura y Humedad. En nuestro ROVER el sensor se conectará al PIN Digital D5. Para empezar a realizar el programa lo primero que haremos será crear dos variables numéricas; la variable: Temperatura y la variable: Humedad, y las asignamos al sensor.

The screenshot shows a programming interface with a left sidebar containing categories: Lógica, Control, Matemáticas, Texto, Variables, Listas, Funciones, ESP, Entrada/Salida, Tiempo, Puerto serie, and Bluetooth. The 'Variables' category is selected. A block 'Establecer varNum = 0' is being dragged from the sidebar into a 'Bucle' (Loop) block. A context menu is open over the 'varNum' block, showing options: '✓ varNum', 'Renombrar la variable...', and 'Variable nueva...'. Below the loop, two more blocks are visible: 'Establecer Temperatura = DHT-22 Temperatura °C Pin D5' and 'Establecer Humedad = DHT-22 Humedad % Pin D5'. To the right of the code is a small image of a DHT-22 sensor module and a cartoon rover character.



PROGRAMACIÓN SENSORICA VEHÍCULO ROVER

1.- Sensor DHT-22

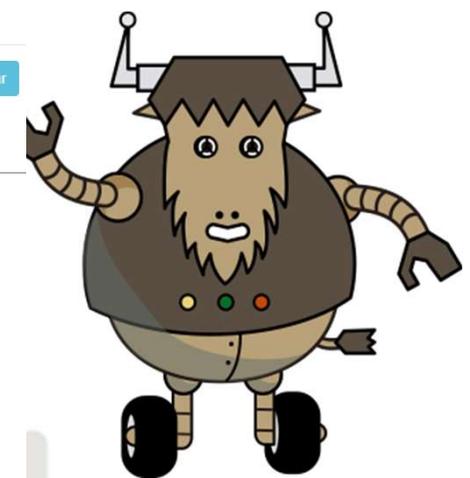
Una vez creadas y asignadas las variables vamos a completar el programa usando el puerto serie para leer sus valores. Primero inicializaremos el puerto serie a 9600 Baudios.

The screenshot shows an Arduino IDE with the following code blocks:

- Inicializar:** Iniciar Baudios 9600
- Bucle:**
 - Establecer Temperatura = DHT-22 Temperatura °C Pin D5
 - Establecer Humedad = DHT-22 Humedad % Pin D5
 - Ejecutar cada 2000 ms
 - Enviar crear texto con "La temperatura es de:" + Temperatura + "°C" (with Salto de línea checked)
 - Enviar crear texto con "La humedad es de:" + Humedad + "%" (with Salto de línea checked)

On the right, the 'ArduinoBlocks :: Consola serie' window shows the following output:

```
Baudrate: 9600 [Conectar] [Desconectar] [Limpiar]
[Enviar]
La temperatura es de: nan C
La humedad es de: nan %
La temperatura es de: 23.00 C
La humedad es de: 34.00 %
La temperatura es de: 23.00 C
La humedad es de: 34.00 %
La temperatura es de: 23.00 C
La humedad es de: 34.00 %
La temperatura es de: 23.00 C
La humedad es de: 34.00 %
La temperatura es de: 23.00 C
La humedad es de: 34.00 %
La temperatura es de: 23.00 C
La humedad es de: 34.00 %
La temperatura es de: 23.00 C
La humedad es de: 34.00 %
```





PROGRAMACIÓN SENSORICA VEHÍCULO ROVER

2.- Sensor LDR.

El sensor LDR (*Light Dependent Resistor*) basa su funcionamiento en una resistencia cuyo valor varía con la incidencia de la luz sobre ella. A más luz menor resistencia y viceversa. Es un sensor analógico por lo que, en nuestro ROVER, se conectará al único Pin analógico que existe, el A0. Crearemos una variable y la llamaremos LUZ.

```
Bucle
  Establecer LUZ = Nivel de luz (LDR) Pin A0 0.1023
```



CantabRobots



PROGRAMACIÓN SENSÓRICA VEHÍCULO ROVER

2.- Sensor LDR

De manera análoga al apartado anterior, realizaremos una lectura por el puerto serie del valor del sensor LDR.

```
Inicializar
  Iniciar Baudios 9600

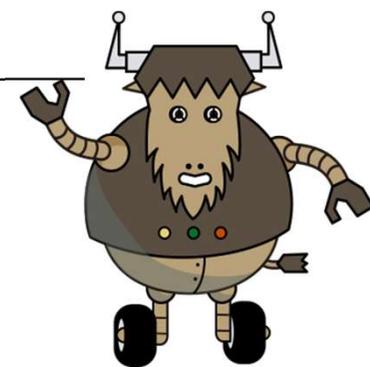
Bucle
  Establecer LUZ = Nivel de luz (LDR) Pin A0 0..1023
  Ejecutar cada 2000 ms
  Enviar crear texto con " El valor de la LDR es: " LUZ
```

ArduinoBlocks :: Consola serie

Baudrate: 9600 Conectar Desconectar Limpiar

Enviar

```
El valor de la LDR es:894.00
El valor de la LDR es:893.00
El valor de la LDR es:765.00
El valor de la LDR es:376.00
El valor de la LDR es:889.00
El valor de la LDR es:293.00
El valor de la LDR es:106.00
El valor de la LDR es:61.00
El valor de la LDR es:394.00
El valor de la LDR es:70.00
El valor de la LDR es:893.00
El valor de la LDR es:893.00
El valor de la LDR es:890.00
```



Ve tapando y destapando el sensor con la mano y comprueba los distintos valores que va dando.

Los valores medidos no son **Lumenes** (medida en el SI del flujo luminoso), ni tampoco son **Ohmios**.

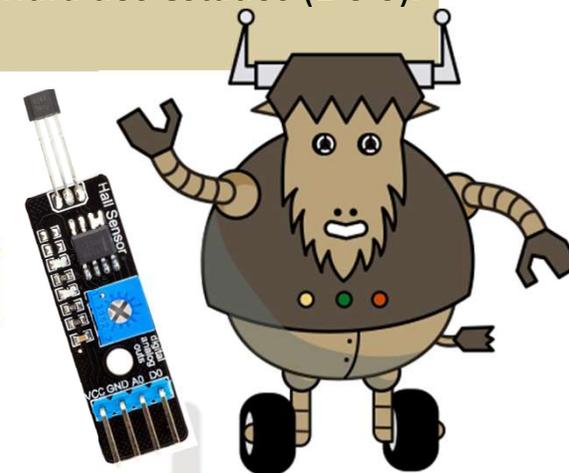
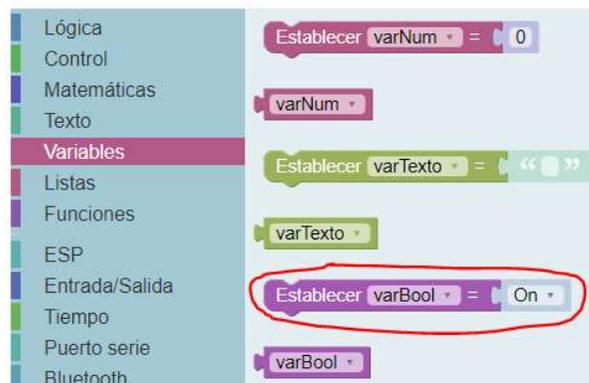


PROGRAMACIÓN SENSORIA VEHÍCULO ROVER

3.- Sensor de efecto HALL.

El sensor de efecto HALL se utiliza para medir campos magnéticos. En nuestra expedición lo utilizaremos para detectar rocas magnéticas y se utilizará como sensor digital conectado al PIN D7.

Crearemos una variable, pero en este caso será booleana, ya que este sensor sólo tendrá dos estados (1 ó 0). La llamaremos MAGNETICO.



Fíjate como el propio sensor tiene un led rojo que se ilumina cuando detecta un campo magnético.



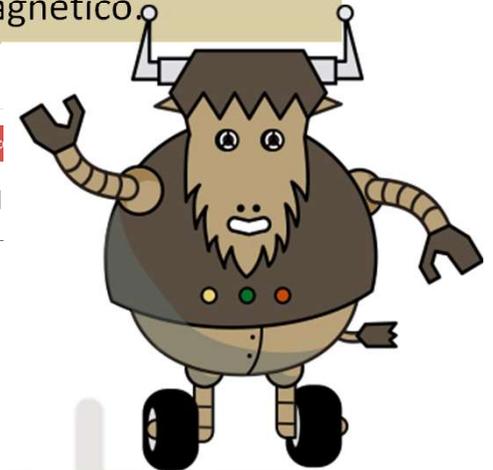
PROGRAMACIÓN SENSORICA VEHÍCULO ROVER

3.- Sensor de efecto HALL.

Vamos a realizar la lectura por el *puerto serie*, pero con algunos cambios con respecto a los sensores anteriores. En esta ocasión utilizaremos un condicional para que, en caso de detectar campo magnético, se envíe el mensaje de “ROCA MAGNÉTICA” y en caso contrario “ROCA NO MAGNETICA”. Paralelamente haremos que se encienda el LED D0 de la placa NodeMCU cuando se detecte el campo magnético.

```
graph TD
    subgraph Inicializar
        A[Iniciar Baudios 9600]
    end
    subgraph Bucle
        B[Establecer MAGNETICO = Sensor de campo magnético Pin D7]
        C[Ejecutar cada 2000 ms]
        D[Enviar crear texto con "DETECTADO:" Salto de línea MAGNETICO]
    end
    A --> B
    B --> C
    C --> D
    D --> B
```

```
ArduinoBlocks :: Consola serie
Baudrate: 9600 [Conectar] [Desc]
DETECTADO: 1
DETECTADO: 1
DETECTADO: 1
DETECTADO: 1
DETECTADO: 1
DETECTADO: 1
DETECTADO: 0
DETECTADO: 0
DETECTADO: 0
DETECTADO: 1
DETECTADO: 1
DETECTADO: 1
DETECTADO: 1
DETECTADO: 0
DETECTADO: 0
```



El sensor presenta una ***lógica inversa*** así que cuando detecta un campo magnético envía un 0 y cuando no detecta envía un 1.



PROGRAMACIÓN SENSORIA VEHÍCULO ROVER

3.- Sensor de efecto HALL.

Se podría mejorar el programa anterior de la siguiente manera:

The screenshot shows the ArduinoBlocks programming environment. On the left, the code is written in a block-based style. It starts with an 'Inicializar' block containing 'Iniciar Baudios 9600'. This is followed by a 'Bucle' (loop) block. Inside the loop, there is an 'Establecer' block for a 'Sensor de campo magnético' on 'Pin D7'. Below this is an 'Ejecutar cada' block set to '2000 ms'. A 'si' (if) block checks if the 'MAGNETICO' sensor is 'verdadero'. If true, it executes a 'hacer' (do) block with two steps: 'Enviar' the string 'ROCA NO MAGNETICA' and 'Led' on 'Pin D0' set to 'ON'. If false, it executes a 'sino' (else) block with two steps: 'Enviar' the string 'ROCA MAGNETICA' and 'Led' on 'Pin D0' set to 'OFF'. On the right, the 'Arduinoblocks :: Consola serie' window shows the serial output. The baudrate is set to 9600. The console displays a sequence of messages: 'ROCA NO MAGNETICA', 'ROCA NO MAGNETICA', 'ROCA MAGNETICA', 'ROCA MAGNETICA', 'ROCA NO MAGNETICA', 'ROCA NO MAGNETICA', 'ROCA MAGNETICA', 'ROCA NO MAGNETICA'.



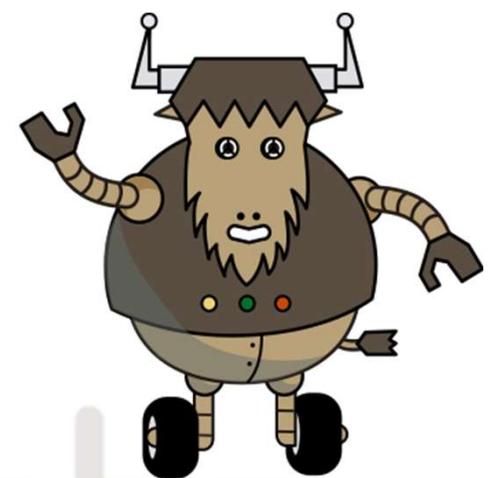


PROGRAMACIÓN SENSORICA VEHÍCULO ROVER

3.- Sensor de efecto HALL.

Podríamos hacer una pequeña modificación en el programa anterior e incluir una negación en la asignación de la variable. De esa manera el programa nos parecerá "más lógico" para nuestro entendimiento.

```
graph TD
    subgraph Inicializar
        A[Iniciar Baudios 9600]
    end
    subgraph Bucle
        B[Establecer MAGNETICO = no Sensor de campo magnético Pin D7]
        C[Ejecutar cada 2000 ms]
        subgraph Si
            D[MAGNETICO == verdadero]
            E[Enviar "ROCA MAGNETICA" Salto de línea]
            F[Led Pin D0 Estado OFF]
        end
        subgraph Sino
            G[MAGNETICO == falso]
            H[Enviar "ROCA NO MAGNETICA" Salto de línea]
            I[Led Pin D0 Estado ON]
        end
    end
    A --> B
    B --> C
    C --> D
    D --> E
    E --> F
    D --> G
    G --> H
    H --> I
```



Recuerda que los LEDs de la placa NodeMCU también funcionan con **lógica inversa**.