Movimiento

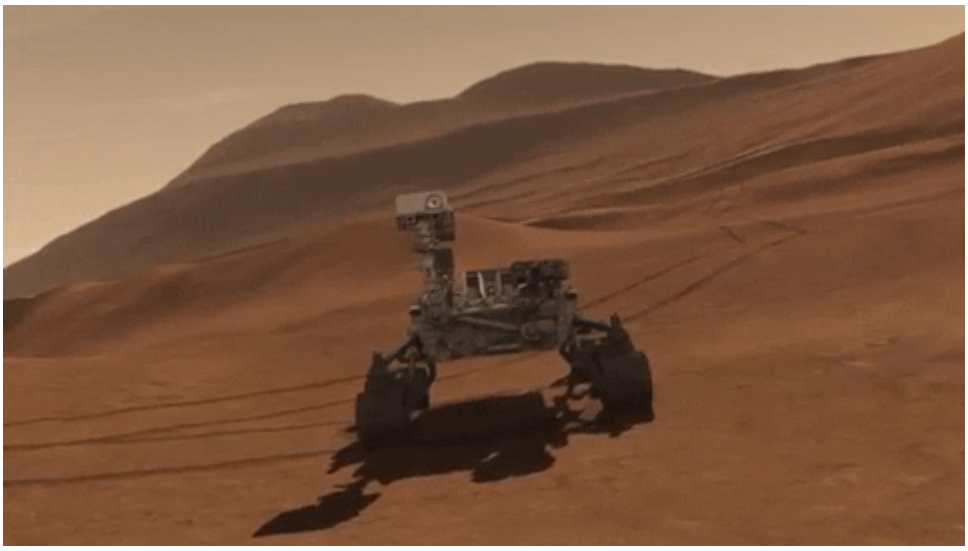
- 2 Movimiento
- 2.1 Motores
- 2.2 Fichero VARIABLES.py
- 2.3 Libreria MOVIMIENTOS.py
- 2.4 Baile
- 2.5 Movimientos con tecla

Copyright 2025

2 Movimiento

El primer contacto con este robot va a ser controlar el movimiento.

> Nomenclatura: Hacia delante es donde está la cámara.



Giphy GIF by NASA

Copyright 2025

2.1 Motores

Vamos a empezar por la parte importante de un rover:

NASA/JPL-Caltech

Los motores estan gobernados con los siguientes GPIO

Interfaces	Puertos GPIO de la Raspberry Pi
IN1	P12
IN2	P13
ENA	P6
IN3	P20
IN4	P21
ENB	P26

Luego una de las primeras líneas que hay que poner en nuestros programas es traducir esos números a letras para que sea más facil utilizarlos en el código, y definir esos pines como pines de salida que van a gobernar a los motores:

```
import RPi.GPIO as GPIO
```

```
IN1=12;IN2=13;ENA=6;IN3=20;IN4=21;ENB=26

GPIO.setmode(GPIO.BCM);GPIO.setwarnings(False)
GPIO.setup(IN1,GPIO.OUT);GPIO.setup(IN2,GPIO.OUT);GPIO.setup(IN3,GPIO.OUT);GPIO.setup(IN4,GPIO
.OUT)
GPIO.setup(ENA,GPIO.OUT);GPIO.setup(ENB,GPIO.OUT)
```

¿Qué significan IN1 IN2 IN3 IN4 ?

IN1	IN2	IN3	IN4	Descripción
1	0	0	1	Motores hacia delante
0	1	1	0	Motores hacia atrás
0	0	0	1	Giro derecha
1	0	0	0	Giro izquierda
0	0	0	0	Stop

¿Y qué significa ENA ENB?

ENA y ENB es la velocidad de los motores A y B respectivamente.

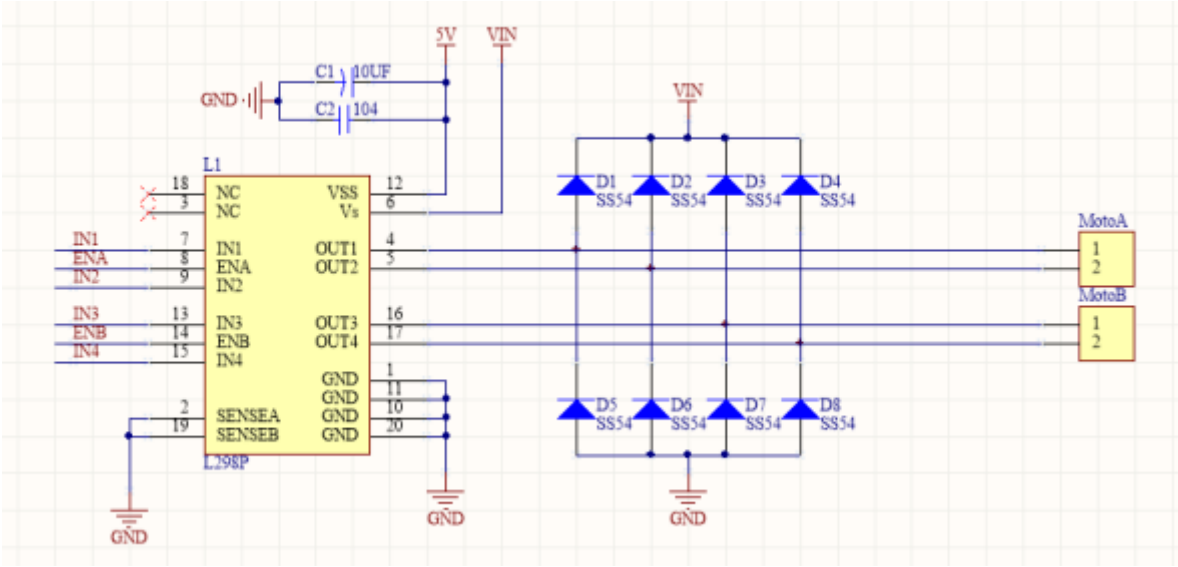
Su valor tiene que ser analógico pero los GPIO son digitales, así que tienen que ser señales PWM.

Si vamos a poner una frecuencia de 500Hz y una velocidad media, el código que tenemos que poner al principio de nuestro programa es:

```
PWMA = GPIO.PWM(ENA,500);PWMB = GPIO.PWM(ENB,500)
PWMA.start(50);PWMB.start(50)
```

Bueno, pero ... ¿Cómo son las conexiones?

En el AlphaBot están conectados los pines IN1 IN2 IN3 IN4 ENA ENB en los pines de un chip L298P que hace de driver a los motores (nunca conectes un motor a un GPIO de la Raspberry ya lo sabes)



Vale... ¿Y cómo se utiliza?

Podemos definir en nuestros programas unas funciones para simplificar código para utilizar los motores hacia delante, detrás y giros:

```
def FORDWARD():

GPIO.output(IN1,GPIO.HIGH);GPIO.output(IN2,GPIO.LOW);GPIO.output(IN3,GPIO.LOW);GPIO.output(IN4
,GPIO.HIGH)
def BACKWARD():

GPIO.output(IN1,GPIO.LOW);GPIO.output(IN2,GPIO.HIGH);GPIO.output(IN3,GPIO.HIGH);GPIO.output(IN
4,GPIO.LOW)
def LEFT():

GPIO.output(IN1,GPIO.LOW);GPIO.output(IN2,GPIO.LOW);GPIO.output(IN3,GPIO.LOW);GPIO.output(IN4,
GPIO.HIGH)
def RIGHT():

GPIO.output(IN1,GPIO.HIGH);GPIO.output(IN2,GPIO.LOW);GPIO.output(IN3,GPIO.LOW);GPIO.output(IN4
,GPIO.LOW)
def STOP():
```

```
GPIO.output(IN1,GPIO.LOW);GPIO.output(IN2,GPIO.LOW);GPIO.output(IN3,GPIO.LOW);GPIO.output(IN4,
GPIO.LOW)
```

2.2 Fichero VARIABLES.py

Debido a que vamos a utilizar varias variables que serán comunes a varias librerías que también vamos a crear, vamos a crear un fichero común a todos, de momento será este:

Cuando queramos incorporar estas variables pondremos esta instrucciónde Python **from VARIABLES import ***

VARIABLES.py

```
import RPi.GPIO as GPIO

DataMotorR = 7
DataMotorL = 8

IN1=12
IN2=13
ENA=6
IN3=20
IN4=21
ENB=26

##############CONFIGURACION GPIO ENTRADAS SALIDAS ####
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(IN1,GPIO.OUT)
GPIO.setup(IN2,GPIO.OUT)
GPIO.setup(IN3,GPIO.OUT)
GPIO.setup(IN4,GPIO.OUT)
GPIO.setup(ENA,GPIO.OUT)
GPIO.setup(ENB,GPIO.OUT)

GPIO.setup(DataMotorR,GPIO.IN)
GPIO.setup(DataMotorL,GPIO.IN)
```

```
########################## VELOCIDAD DE LOS MOTORES
PWMA = GPIO.PWM(ENA,500)
PWMB = GPIO.PWM(ENB,500)
```

2.3 Libreria MOVIMIENTOS.py

Para simplificar nuestros programas podemos hacer una librería propia.

Esta librería la vamos a llamar **MOVIMIENTOS.py** y su contenido sería lo visto en las páginas anteriores, añadiendo las variables definidas en **VARIABLES.py**:

```
import RPi.GPIO as GPIO

from VARIABLES import *



##########################FUNCIONES######################
def FORDWARD(vel):
    GPIO.output(IN1,GPIO.HIGH)
    GPIO.output(IN2,GPIO.LOW)
    GPIO.output(IN3,GPIO.LOW)
    GPIO.output(IN4,GPIO.HIGH)
    PWMA.start(vel)
    PWMB.start(vel)

def BACKWARD(vel):
    GPIO.output(IN1,GPIO.LOW)
    GPIO.output(IN2,GPIO.HIGH)
    GPIO.output(IN3,GPIO.HIGH)
    GPIO.output(IN4,GPIO.LOW)
    PWMA.start(vel)
    PWMB.start(vel)

def LEFT(vel):
    GPIO.output(IN1,GPIO.LOW)
    GPIO.output(IN2,GPIO.LOW)
    GPIO.output(IN3,GPIO.LOW)
    GPIO.output(IN4,GPIO.HIGH)
```

```
    PWMA.start(vel)
    PWMB.start(vel)

def RIGHT(vel):
    GPIO.output(IN1,GPIO.HIGH)
    GPIO.output(IN2,GPIO.LOW)
    GPIO.output(IN3,GPIO.LOW)
    GPIO.output(IN4,GPIO.LOW)
    PWMA.start(vel)
    PWMB.start(vel)

def STOP():
    GPIO.output(IN1,GPIO.LOW)
    GPIO.output(IN2,GPIO.LOW)
    GPIO.output(IN3,GPIO.LOW)
    GPIO.output(IN4,GPIO.LOW)
```

Copyright 2025

2.4 Baile

Vamos a realizar un sencillo programa para romper el hielo, vamos a imitar los movimientos de la Sojourner

Crédito de imagen: Proyecto Mars Pathfinder

Unos movimientos delante, atrás, derecha, izquierda y paro utilizando la librería anterior:

https://www.youtube.com/embed/WAycuDaKB0Q

Solución

- Ponemos el fichero MOVIMIENTOS.py que hemos visto en la misma carpeta que vamos a crear este programa.
- En este programa importamos la librería de MOVIMIENTOS.py.
- Vamos llamando a las distintas funciones de movimientos, fijamos la velocidad al 50%, insertando entre ellas un tiempo de retraso de la librería time de 1 segundo.

¿Te atreves? Sino, mira la solución:

```python
import RPi.GPIO as GPIO
import time

import MOVIMIENTOS

MOVIMIENTOS.FORDWARD(50)
time.sleep(1)
MOVIMIENTOS.BACKWARD(50)
time.sleep(1)
MOVIMIENTOS.LEFT(50)
time.sleep(1)
MOVIMIENTOS.RIGHT(50)
time.sleep(1)
MOVIMIENTOS.STOP()
```

Fichero 2-6-Baile.py

2.5 Movimientos con tecla

Ahora vamos a hacer lo mismo, pero gobernado por el teclado:

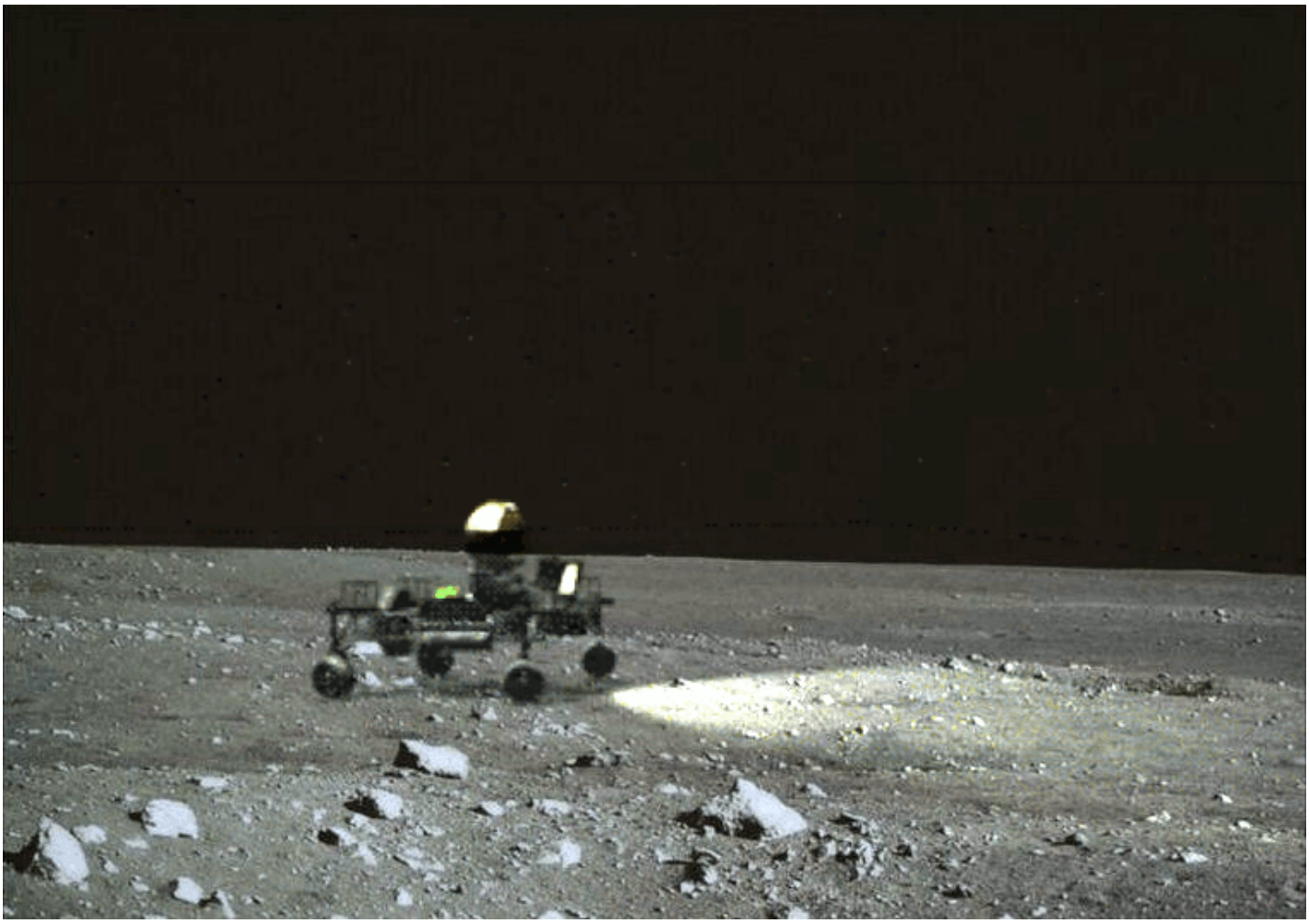
- PARAR = tecla ESPACIO
- ADELANTE=FORDWARD = f
- ATRAS=BACKWARD = b
- DERECHA=RIGHT = r
- IZQUIERDA=LEFT = l

https://www.youtube.com/embed/fb6w5yQB_AM

Solución

- Ponemos el fichero MOVIMIENTOS.py que hemos visto en la misma carpeta que vamos a crear este programa.
- En este programa importamos la librería de MOVIMIENTOS.py.
- Vamos llamando a las distintas funciones de movimientos según la tecla pulsada, fijamos la velocidad al 30% para que nos de tiempo de gobernarlo, por pantalla va saliendo el mensaje del estado.
- Todo dentro de un bucle de manera que si pulsamos la tecla espacio sale del buble no sin antes parar el robot.

¿Te atreves a hacerlo tú solo? Venga!! no le des tantas vueltas



From Gifer

Sino, mira la solución:

```python
import RPi.GPIO as GPIO
import time

import MOVIMIENTOS

print("""TECLAS ¡en minúscula!:
PARAR = tecla ESPACIO
ADELANTE=FORDWARD = f
ATRAS=BACKWARD = b
DERECHA=RIGHT = r
IZQUIERDA=LEFT = l""")

tecla='x'
while tecla!=' ':
    tecla = input('\nPresiona una tecla y después enter : ')
    if tecla != ' ':
        print ('\nHas presionado ', tecla)
        if tecla=='f':
            print ('\nadelante')
            MOVIMIENTOS.FORDWARD(30)
        if tecla=='b':
            print ('\natrás')
            MOVIMIENTOS.BACKWARD(30)
        if tecla=='r':
            print ('\nderecha')
            MOVIMIENTOS.RIGHT(30)
        if tecla=='l':
            print ('\nizquierda')
            MOVIMIENTOS.LEFT(30)

    else:
        print ('\nFin, has apretado STOP')
        MOVIMIENTOS.STOP()
```

Fichero [2-7-Movimientos-Teclas.py]