Servos

7 Servos

Los servos son motores con una reductora, un sensor de posición y un circuito de control del ángulo de giro. Son utilizados en los brazos robóticos y como puedes ver **juegan un importante papel en los rovers** por ejemplo en el actual Perseverance :

https://mars.nasa.gov/layout/embed/video/?v=423

Para saber más de servos te recomendamos el capítulo de servomotores del curso de Arduino de Aularagón..

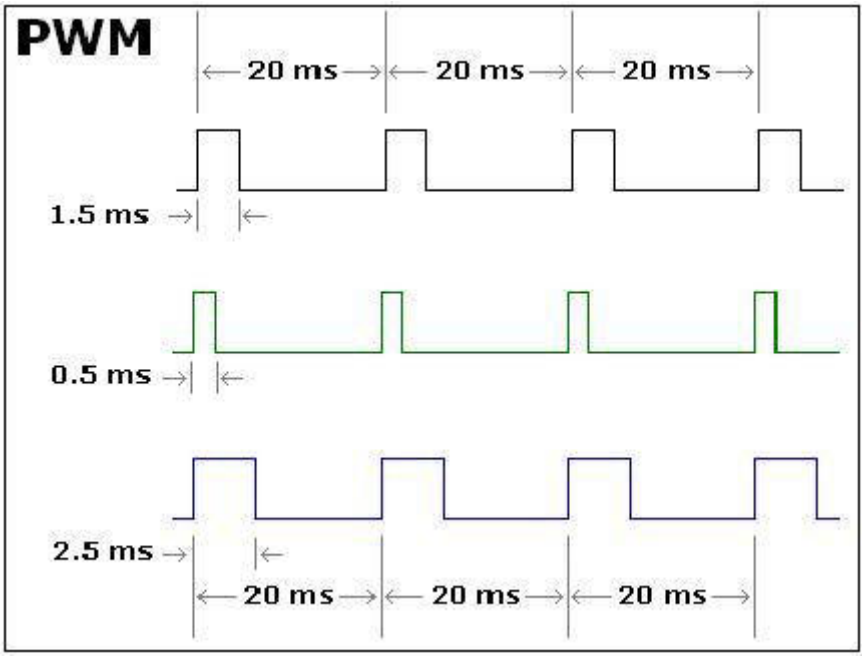
Los que tiene nuestro rover son más baratos:

Tiene 3 cables:

- Marrón a GND
- Naranja a 5V
- Rojo la señal de control

La señal de control tiene que emitir un pulso alto durante un intervalo de al menos 20mseg, que según su duración en estado alto, se traduce en un ángulo de rotación del servo:

Pulse width	Rotation angle
0.5ms	0 degree
1.0ms	45 degree
1.5ms	90 degree
2.0ms	135 degree
2.5ms	180 degree



7.1 BRAZO.py y VARIABLES.py

En Alphabot el servo de abajo del brazo robot (lo llamaremos eje z por ser el responsable del giro del eje vertical) está conectado al GPIO 27 y el servo de arriba (lo llamaremos x) al GPIO 22 luego añadiremos estas líneas en nuestro fichero VARIABLES.py. Lo configuramos como salida y que inicialmente esten no activos para que no se mueva el brazo en el comienzo:

```
#### SERVOS BRAZO ROBOT
SERVOEJEX = 22
SERVOEJEZ = 27

####### SERVOS BRAZO ROBOT
GPIO.setup(SERVOEJEX, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(SERVOEJEZ, GPIO.OUT, initial=GPIO.LOW)
```

BRAZO.py

Realmente el control de un servo se hace con una modulación PWM que ya hemos visto. La función que modula la señal PWM es ChangeDutyCycle y se le da el argumento en % entre 0 y 100. Si queremos 180º necesitamos un pulso de 2.5ms por lo que en 20ms corresponde a 12.5% por lo tanto la fórmula es % = 2.5+10*(angulo/180).

Esta función simplemente le indicamos el ángulo y otro argumento, si es 1 es el eje x y si es 0 es el eje z :

```
import RPi.GPIO as GPIO
import time

from VARIABLES import *

servox = GPIO.PWM(SERVOEJEX,40)
servoz = GPIO.PWM(SERVOEJEZ,40)
servox.start(0)
```

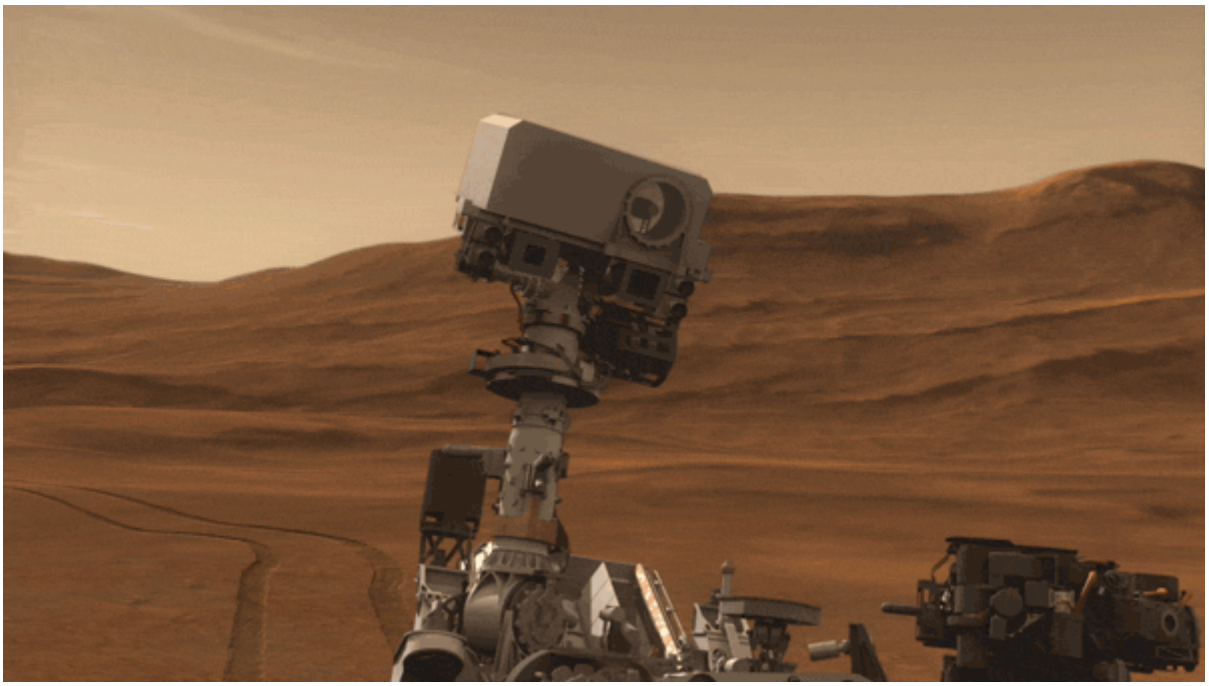
```
servoz.start(0)

def ANGULO(angle,x):
    if (x):
        servox.ChangeDutyCycle(2.5 + 10.0 * angle / 180)
    else:
        servoz.ChangeDutyCycle(2.5 + 10.0 * angle / 180)
```

> Los servos tiemblan algo, es normal, no pienses que un robot barato esté bien calibrado.

7.2 Test Brazo

Vamos a realizar una función que controle con el teclado el brazo robótico, si le pusieramos un diodo laser, podriamos incluso imitar al vaporizador que esta montado en la Perserverante:



NASA's Mars Curiosity rover landed on Mars in 2012. NASA/JPL-Caltech

De momento nos vamos a conformar con hacer esto :

- Teclas 8 y 2 mueven el brazo robot en el eje x arriba y abajo
- Teclas 4 y 6 mueven el brazo robot en el eje z derecha e izquierda.

Fijaremos de antemano un incremento de 10ª cada vez que pulsamos la tecla. Veamoslo con un vídeo:

https://www.youtube.com/embed/S3Z9vRjPtQo

Solución

- Ponemos la librería del fichero BRAZO.py en la misma carpeta que vamos a crear este programa y la incorporamos en el programa con **import**.
- Importamos las variables también con import * from VARIABLES
- Vamos incrementando los ángulos eje x y eje z según la tecla pulsada.

- Todo dentro de un bucle.

¿Te atreves? :

Fichero Test-Brazo.py

```
import RPi.GPIO as GPIO
import time

from VARIABLES import *

import BRAZO

angulox=90
anguloz=90
incremento=20
print("Teclas 8 y 2 SERVOX\n Teclas 4 y 6 SERVOZ")
while True:
    BRAZO.ANGULO(angulox,1)
    BRAZO.ANGULO(anguloz,0)
    tecla=input("Mueve el brazo : ")
    if (tecla=="8"):
        angulox=angulox-incremento
    if (tecla=="2"):
        angulox=angulox+incremento
    if (tecla=="4"):
        anguloz=anguloz+incremento
    if (tecla=="6"):
        anguloz=anguloz-incremento
```