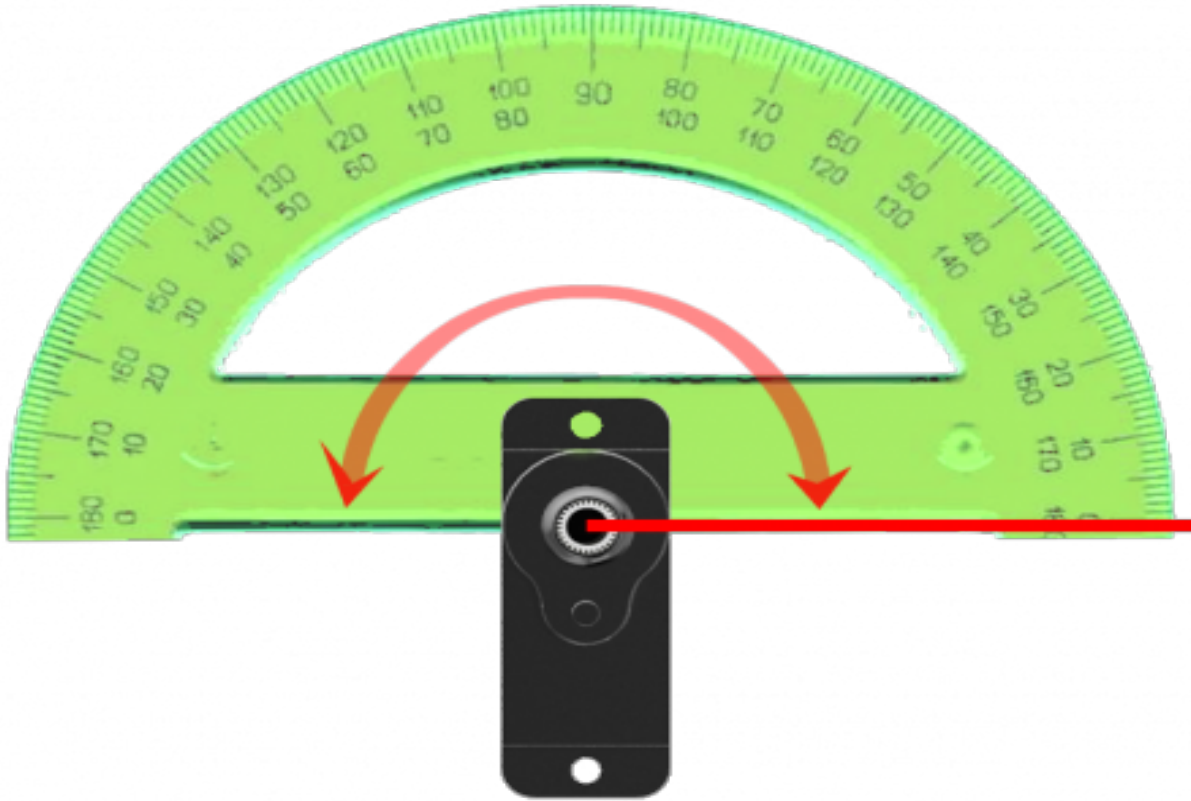


¿Qué es un servomotor?

Un servomotor es un pequeño motor que es capaz de girar entre 0 y 180 grados. Lo interesante de los servomotores es que nosotros decidimos a qué posición o ángulo deben moverse. Durante el curso llamaremos a los servomotores servos de posición, para diferenciarlos de los servos de rotación continua, capaces de girar 360 grados.



IMPORTANTE: Si utilizas muchos motores y el puerto USB puede que no haya suficiente energía para alimentarlos a todos. Siempre que utilices servomotores recuerda alimentarlos mediante el portapilas. Recibirán mucha más energía a la hora de moverse.

Servo de Posición: El objetivo de esta práctica es controlar un servo de posición

El kit Básico posee un servo de posición, es decir, que puede girar media vuelta de 0 a 180°grados.

El Qudino puede hacerlo girar ángulos comprendidos entre 0 y 180°, útil para realizar varios proyectos con mecanismos, por ejemplo: barreras, faros, grúas y otros.

Vamos a hacer un pequeño ejemplo para averiguar cómo funciona el miniservo. Para ello conéctalo a un pin digital S0, por ejemplo.



Servo de Posición, es un actuador compuesto por un servo chico (Micro servo 9 g), que gira un ángulo de terminado por el programador comprendido entre, los valores 0 y 180 Grados.

Insertado en una cajita verde y con tres cables de colores, generalmente marrón (-), rojo(+) y naranja

(señal) y la ficha hembra correspondiente.

La conexión se realiza, con el color naranja ,para el lado de afuera del **Qudino**. Se conectan en las salidas verdes (S0,S1,S2,S3).La conexión se realiza, con el color naranja ,para el lado de afuera del Qudino.

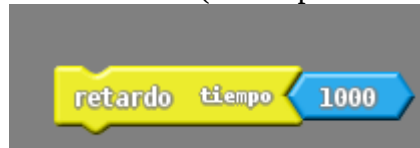
El módulo Servo tiene dos Propiedades **Salida y Ángulo**

Salida, se puede conectar el servo a las salidas S(0,1,2,3)

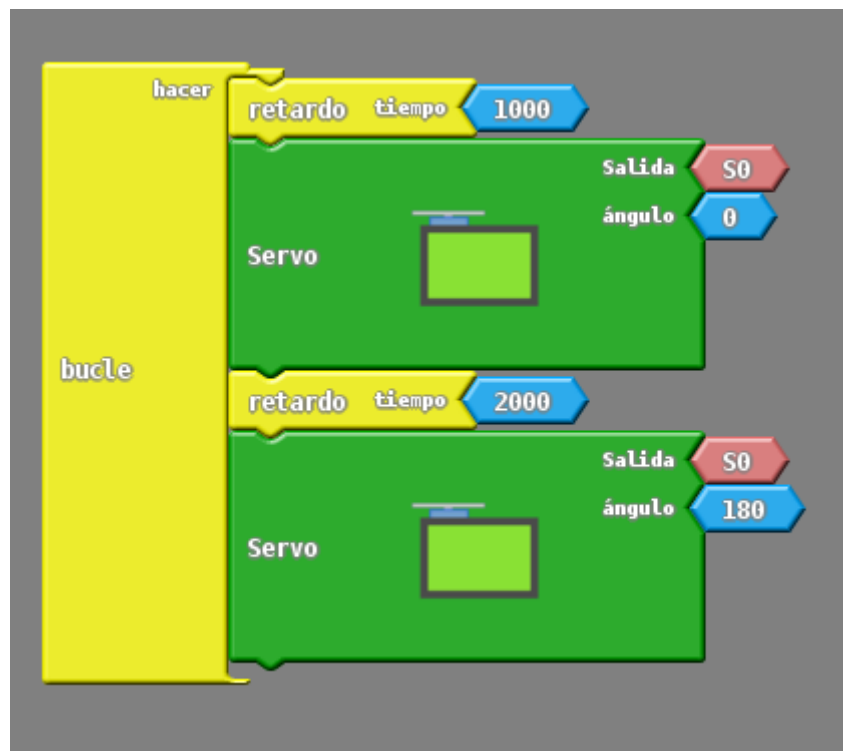
Ángulo, el valor se puede fijar en valores entre 0-180°por ejemplo : 60,90 u otro.

También utilizamos retardo que hace que el programa se tome un tiempo,para luego continuar.

Poner siempre un bloque **retardo** tras mover un miniservo ya que si realizas cambios de dirección a gran velocidad puedes llegar a quemarlo. Una pequeña espera después de cada movimiento protegerá al miniservo de movimientos bruscos. (el tiempo en milisegundos. 1000ms=1s)



Ejemplo de programa:



IMPORTANTE: Todos los servomotores pueden experimentar vibraciones en los ángulos 0° o 180°. Esto se produce debido a las tolerancias de fabricación y puede acortar la vida de tu motor. Si tu servo vibra en estos ángulos programalos para moverse a un ángulo cercano como 5° o 175° para evitar dichas vibraciones.

Ejercicio de practica :

Transportador

Queremos mover el miniservo a los ángulos 0°, 45°, 90°, 135°, 180°. Para ello utilizamos el bloque Servo y escribimos el ángulo al que deseamos que se desplace

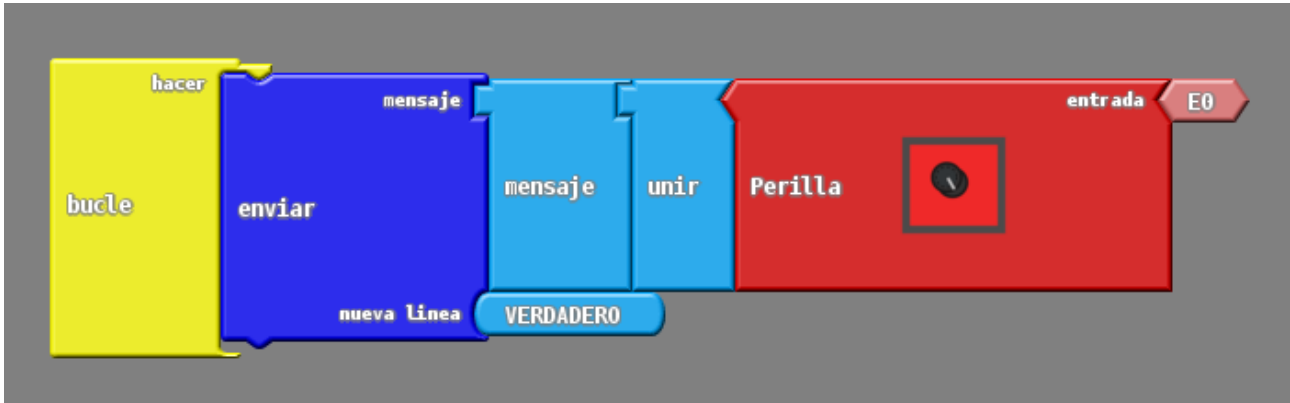
Intenta la solución o revisa la nuestra

Servo y Perilla

Ahora que sabemos cómo utilizar el servo, vamos a probar a controlar su posición utilizando un potenciómetro

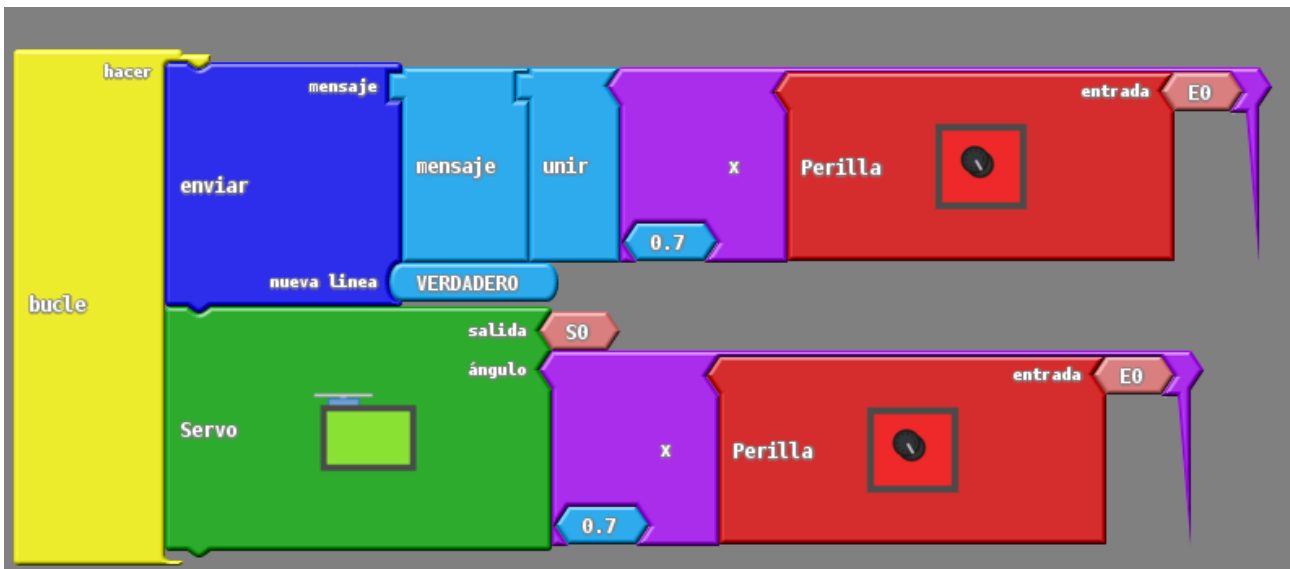
Probando el siguiente programa podemos observar cuales son los valores de E0 que mide la perilla.

Utilizando el Monitor serie, los valores son de 0 a 255 aproximadamente.



Tenemos la necesidad de convertir una escala en otra distinta. Por ejemplo, en el caso del potenciómetro, éste da valores entre 0 y 255, mientras que el miniservo solo puede moverse entre valores de 0 y 180. La solución más sencilla para adaptar el valor del potenciómetro de 0-255 a 0-180 es la fórmula $\text{Ángulo} = (180/255) \times E0 = 0.7 \times E0$, para poder controlar correctamente el miniservo.

$\text{Ángulo} = (180/255) \times E0 = 0.7 \times E0$, que corresponde al siguiente programa.



Para realizar este programa deberás utilizar el bloque genérico del servo. Lo encontrarás en **Componentes->Actuadores**. En el primer hueco deberás poner el número de pin al que has conectado el servo, tal como hemos realizado en otras ocasiones, la variable que almacena dicho número. En avanzados encontrarás dichas variables. ¡Mueve ahora el potenciómetro! Podrás controlar grado a grado la posición del miniservo.

