

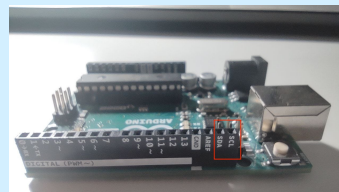
Ejercicio Final: Túnel de lavado con sensores

Programación con Arduino y posterior prototipado de un sistema de lavado automático para coches.



COMPONENTES

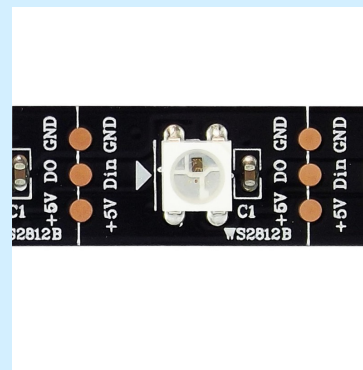
Arduino UNO



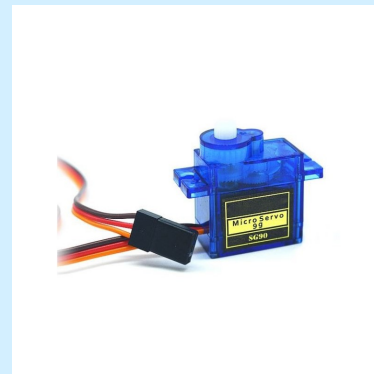
Sensor de Ultrasonidos



Tira leds NeoPixel



ServoMotor



El acceso al túnel estará regulado por un semáforo (2 leds de la tira Neopixel) que permanecerá en **verde** hasta que el vehículo se acerque justo a la posición donde debe parar. En este instante el semáforo cambiará a **rojo**, y comenzará el proceso de lavado.

Durante el tiempo de lavado (6 segundos), el resto de leds de la tira, se irán encendiendo en color **azul**, uno a uno cada segundo, indicando de este modo el progreso del servicio.

Finalizado el proceso de limpiado y secado, la barrera de salida (un servomotor en posición 0 grados), se abrirá (posición 90 grados) para dejar salir al vehículo, y el semáforo volverá a ponerse en verde.

Al tratarse del ejercicio final, únicamente vamos a ver como accionar un servomotor, ya que se pretende ver como el alumnado ha adquirido los conocimientos, y el resto de componentes han sido ya tratados en las experiencias anteriores.

Diseño 2D

Es conveniente utilizar alguna herramienta para diseñar nuestro circuito antes de proceder al montaje. De este modo evitaremos cometer errores con el cableado. Existen muchas herramientas online para este propósito y algunas de ellas como Tinkercad incluso permiten su simulación, pero tienen mayor limitación en las librerías de componentes, por lo que en este caso utilizaremos EasyEDA.

Puedes encontrar un tutorial para crear tus diseños con EasyEDA pinchando [aquí](#).

En este caso, como indicaba al principio, vamos a implementar únicamente el diseño de las conexiones del servomotor. El resto del diseño forma parte de la tarea evaluable de este curso.

Los servomotores tienen 3 pines. Positivo, Gnd y datos. A continuación muestro el servo que he utilizado y como quedan las conexiones.



Project: Tunel de lavado

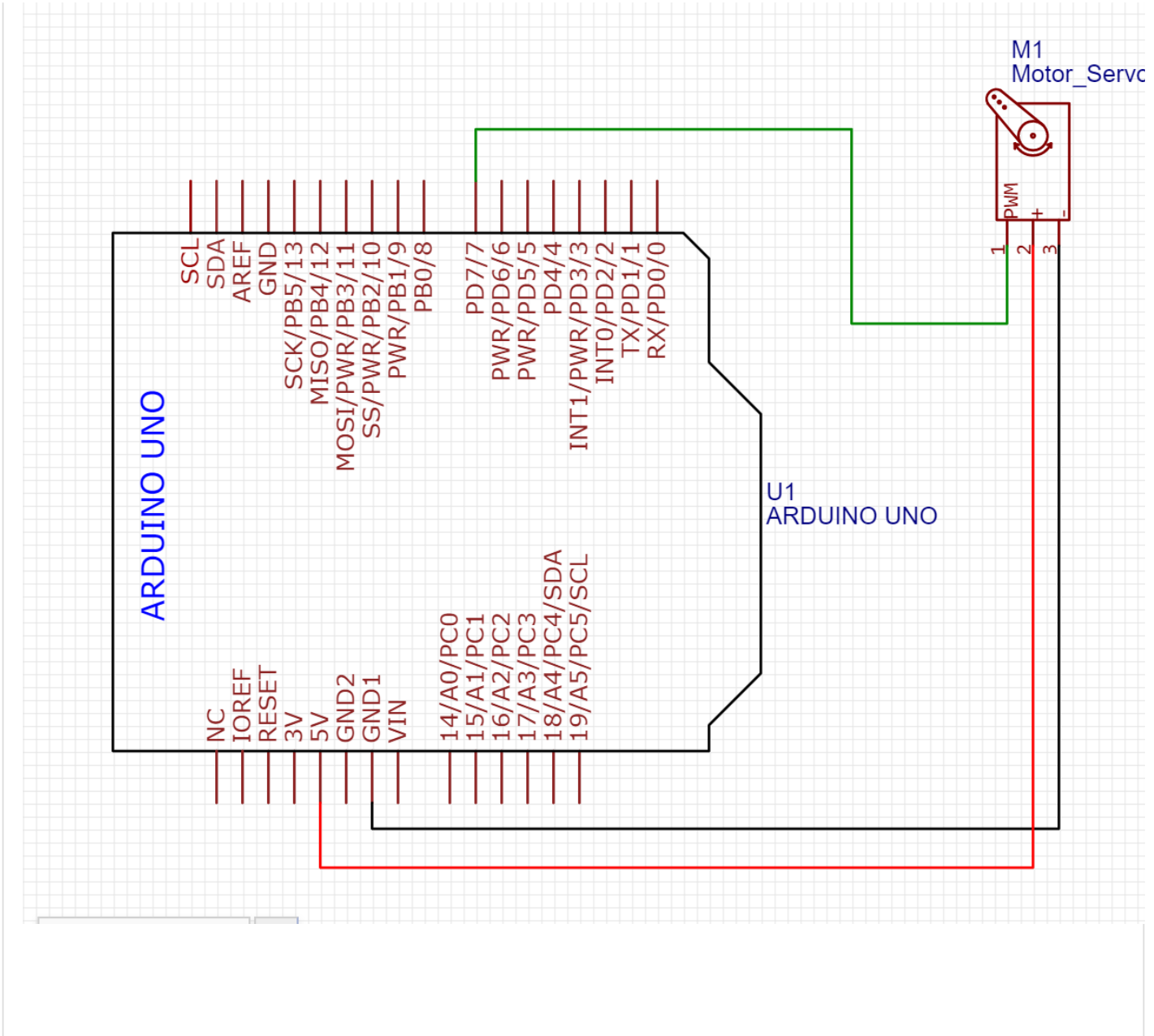
Library Search: servo

Types: Symbol Footprint Spice Symbol SCH Module PCB Module 3D Model

Classes: Work Space(0) LCSC(0) JLCPCB Assembled(0) System(19) Follow(0) User Contributed(990)

Keyword to Filter	Title(PartNO)	Footprint
Company	8-Channel PWM or Servo FeatherWing	8-CHANNEL PWM OR SERVO FEATHERWING
Connectors&Inter...	Mini Maestro 24-Channel USB Servo Controller	MINI MAESTRO 24-CHANNEL USB SERVO CONTROLLE
Diodes	Micro Maestro 6-Channel USB Servo Controller	MICRO MAESTRO 6-CHANNEL USB SERVO CONTROLLE
IC	Micro Maestro 12-Channel USB Servo Controller	MINI MAESTRO 12-CHANNEL USB SERVO CONTROLLE
KiCad SchLibs	Mini Maestro 18-Channel USB Servo Controller	MINI MAESTRO 18-CHANNEL USB SERVO CONTROLLE
Logic/Gates	SMALL SERVO	1X03
Microcontroller	Motor_Servo	
Module Suppliers	Motor_Servo_AirTronics	
Open Source	EL2036CM	SOIC20
Power Management	EL2037ACM	SOIC20
	MC33030DW	SOIC16E
	MC33030P	648C-02
	FI 2027CM	SOIC28

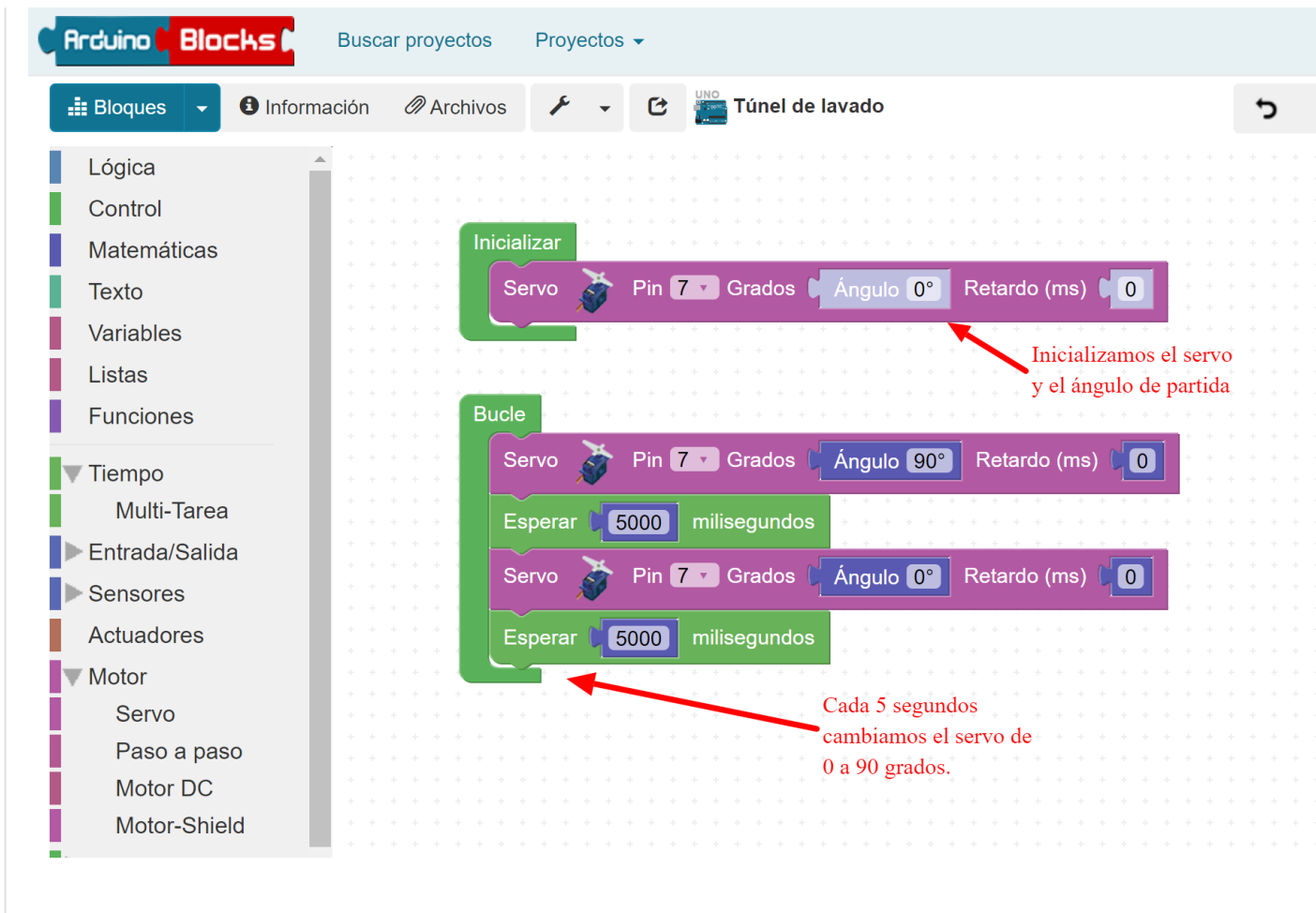
EasyEDA > Symbol > System > Motors > Motor_Servo



A continuación vemos como programar el movimiento de un servo con **ArduinoBlocks**.

Programación - Servomotor

Los servomotores pueden mover su posición de 0 a 180 grados. En ArduinoBlocks, hay un apartado de "Motores" donde podremos encontrar los servos. Lo arrastramos al campo de trabajo y asignamos su pin de datos.



Arduino Blocks Buscar proyectos Proyectos ▾

Bloques ▾ Información Archivos TÚNEL de lavado

- Lógica
- Control
- Matemáticas
- Texto
- Variables
- Listas
- Funciones
- Tiempo
 - Multi-Tarea
- Entrada/Salida
- Sensores
- Actuadores
- Motor
 - Servo
 - Paso a paso
 - Motor DC
 - Motor-Shield

Inicializar

Servo Pin 7 Grados Ángulo 0° Retardo (ms) 0

Bucle

Servo Pin 7 Grados Ángulo 90° Retardo (ms) 0

Esperar 5000 milisegundos

Servo Pin 7 Grados Ángulo 0° Retardo (ms) 0

Esperar 5000 milisegundos

Inicializamos el servo y el ángulo de partida.

Cada 5 segundos cambiamos el servo de 0 a 90 grados.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



Revision #5

Created 2022-10-15 09:14:23 CEST by Ricardo Aguado Vallejo

Updated 2023-01-17 15:53:43 CET by Equipo CATEDU