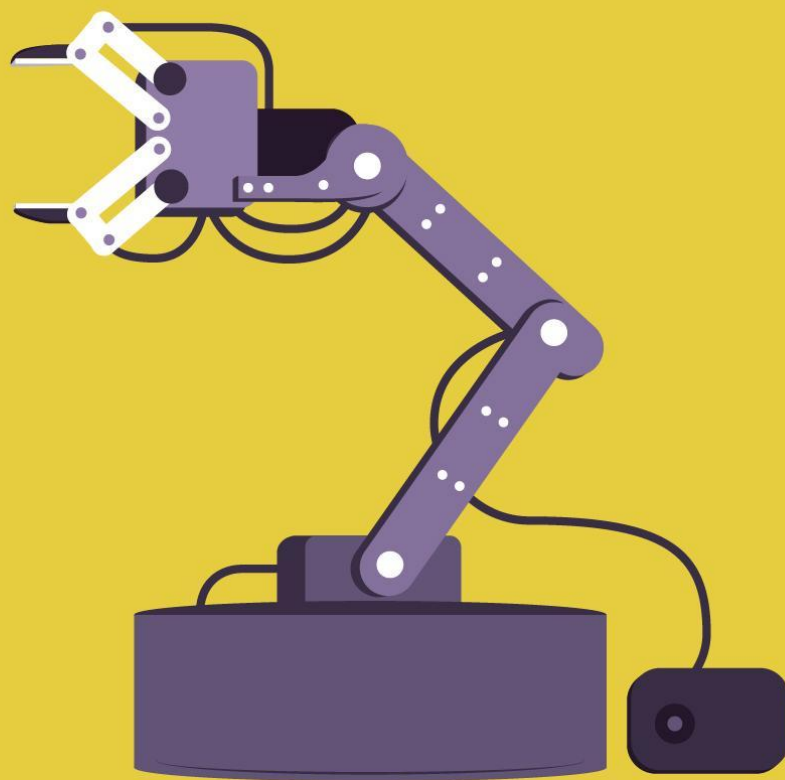


SABERES DIGITALES



BRAZO ROBÓTICO

AUTORIDADES

Presidente de la Nación

Mauricio Macri

Vicepresidenta de la Nación

Marta Gabriela Michetti

Jefe de Gabinete de Ministros

Marcos Peña

Ministro de Educación, Cultura, Ciencia y Tecnología

Alejandro Finocchiaro

**Titular de la Unidad de Coordinación General del
Ministerio de Educación, Cultura, Ciencia y Tecnología**

Manuel Vidal

Subsecretario de Coordinación Administrativa

Javier Mezzamico

Director Ejecutivo INET

Leandro Goroyesky

Gerenta General de EDUCAR Sociedad del Estado

Liliana Casaleggio

Directora Nacional de Asuntos Federales

María José Licio Rinaldi

Director Nacional de Educación Técnico - Profesional

Fabián Prieto

Coordinador de Secundaria Técnica

Alejandro Anchava

Responsable de Formación Docente Inicial y Continua INET

Judit Schneider

Coordinador General En FoCo

Pablo Trangone

AUTORIDADES	¡Error! Marcador no definido.
BRAZO ROBÓTICO	4
Ficha técnica	4
Presentación	5
Desarrollo	6
Nivel Inicial	6
Paso 1 - Realizar la conexión de los servomotores	6
Paso 2 - Programar los ángulos de los servomotores	7
Paso 3 - Subir el código a la placa Arduino	13
Paso 4 - Armado del brazo robótico	14
Paso 5 - Programar un movimiento sincronizado de los servomotores	21
Nivel Intermedio	24
Paso 1 - Armar el circuito	24
Paso 2 - Crear variables	26
Paso 3 - Adaptar los valores	27
Paso 4 - Programar los servomotores restantes	27
Paso 5 - Controlar todos los servomotores	29
Nivel Avanzado	35
Paso 1 - Armar el circuito	35
Paso 2 - Programar la secuencia de movimiento	36
Cierre	42
Glosario	43
Reconocimientos	¡Error! Marcador no definido.

BRAZO ROBÓTICO

Ficha técnica

Nivel educativo	Secundario. Ciclo Básico.
-----------------	---------------------------

Descripción general	Construcción, programación y montaje de un brazo robótico.
Niveles de complejidad	<p>Nivel inicial: Ensamblar las partes del <i>kit</i> del brazo robótico y programar su Placa Arduino para automatizar una operatoria simple.</p> <p>Nivel intermedio: Programar el brazo robótico para controlar sus movimientos por medio de potenciómetros simulando un mando a distancia.</p> <p>Nivel avanzado: Automatizar el brazo robótico para simular una serie de operaciones de un proceso productivo.</p>

Insumos	<ul style="list-style-type: none">• 6 x Potenciómetros• 1 x Fuente 5V 2A• 1 x Cable USB Tipo B• 20 x cables dupont macho hembra• 1 x Protoboard 830 puntos• PLA
Equipamiento	<ul style="list-style-type: none">• Computadora• Kit Brazo Robótico• Destornilladores philips• Llave francesa• Computadora• Soldador• Estaño• Alicata• Pinza de punta• Brusela
Otros requisitos	<ul style="list-style-type: none">• Conexión a internet• Descargar el programa “mBlock3” http://www.mblock.cc/software-1/mblock/mblock3/

	<ul style="list-style-type: none"> • Descargar <i>driver</i> para controlar la placa DFRobot https://www.geekfactory.mx/download/driver-ch340-windows/#
--	---

Presentación

Descripción ampliada del proyecto

El proyecto propone el armado y la programación de un brazo robótico que pueda desarrollar movimientos simples simulando operaciones de procesos productivos reales. El brazo robótico contará con 6 motores (servomotores) que le permitirán moverse sin ninguna restricción en sus grados de libertad. El último de los servomotores, ubicado en el extremo del brazo, estará encargado de comandar una pinza capaz de sujetar objetos, permitiendo cambiar su posición o manipularlos por un tiempo determinado. Esta funcionalidad puede resultar útil, por ejemplo, cuando se necesita mezclar dos reactivos peligrosos para el contacto humano.

En el nivel inicial se desarrollarán el montaje, la calibración y la puesta en marcha del brazo robótico. Mediante la programación por bloques, se generarán las instrucciones necesarias para llevar a cabo los primeros movimientos del brazo.

En el nivel intermedio se propone comandar el brazo robótico a través de una serie de potenciómetros, imitando un sistema de mando a distancia.

En el nivel avanzado, se propone generar una secuencia de operaciones compleja para que realice el brazo. Se introducirá a los alumnos en la programación para que puedan dar respuesta a diferentes problemáticas.

Objetivos

- Aproximarse al conocimiento de dispositivos automatizados que poseen una programación similar a la que se puede encontrar en insumos y máquinas pertenecientes a diferentes sectores de la industria.
- Analizar los diferentes escenarios donde la robótica pueda generar beneficios, tanto para la optimización de procesos como para la seguridad de los operarios.
- Conocer los componentes mecánicos que interactúan en un brazo robótico (nivel inicial).
- Comandar un dispositivo automatizado mediante diferentes interfases (nivel intermedio).
- Analizar la programación de la estructura secuencial de un programa que permita realizar operatorias complejas (nivel avanzado).

- Entender cómo se genera un movimiento articulado que imite el de un brazo humano mediante la utilización de servomotores y diferentes piezas mecánicas.

Desarrollo

Nivel Inicial

En una fábrica se detectó que varios operarios presentaban lesiones musculares después de haber pasado un tiempo extenso haciendo acopio de la materia prima para la producción. Por esta razón se decidió desarrollar un dispositivo mecánico automatizado que permitiera reemplazar la intervención de los empleados en tareas de trabajo forzado que pudieran ser perjudiciales para su salud.

En esta actividad se propone al grupo que construya un brazo robótico de 6 motores (servomotores) utilizando los componentes del kit DFRobots.

Paso 1 - Realizar la conexión de los servomotores

El brazo robótico contará con 6 motores (servomotores) que le permiten realizar movimientos de rotación y traslación en el espacio imitando el movimiento de un brazo humano. Cada uno de los servomotores permitirá rotar el brazo robótico sobre uno de sus ejes creando así 6 articulaciones diferentes, que le permiten moverse sin ninguna restricción en sus grados de libertad (correspondientes a tres componentes de traslación y tres componentes de rotación).

Al realizar el armado del brazo robótico debemos asegurarnos de que la posición de los servomotores sea de 90°. De esta manera, el brazo podrá girar en ambos sentidos. Para conectar los servomotores a la placa Arduino utilizaremos el IO *shield* de DFRobot.

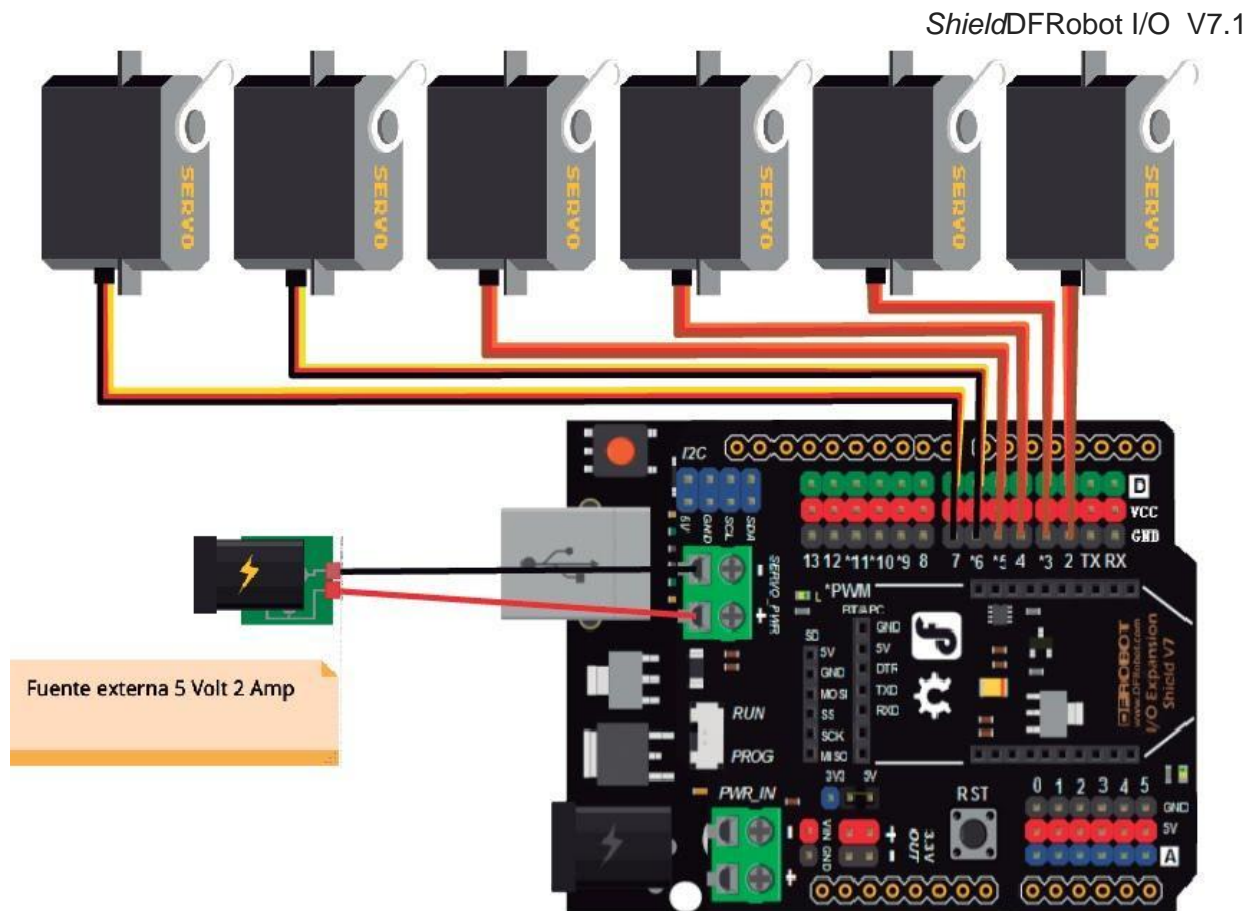
Los shields son placas de circuitos que se monta encima de la placa Arduino para expandir sus funcionalidades. Los mismos pueden ser para diferentes usos como: comunicaciones, sensores, actuadores, interconexión con otros sistemas, sonido, protoboard, etc.

Para ensamblar el *shield* hay que ubicarlo arriba de la placa Arduino haciendo coincidir todos sus pines. Al tener dos grupos de 5 pines de un lado y dos grupos de 6 pines del otro, hay un solo modo de conectarlos adecuadamente sin forzar las piezas.

Un servo es un dispositivo que se compone de un motor y un sistema de control que le permite ubicarse en una posición específica. Los servos más comunes pueden moverse en un rango de 0° a 180°, sin poder girar de forma continua. Se suelen utilizar en aplicaciones tipo barreras o brazos mecánicos. Los servos tienen 3 pines de conexión, dos de ellos se

ocupan en alimentación eléctrica (VCC marrón y GND negro) y un tercer pin que se conecta a una salida digital del Arduino (cable naranja).

Conectaremos los servomotores como indica la figura. De cada servomotor se deberán conectar sus tres cables. El amarillo o naranja debe conectarse al *pin* de dato (D), el rojo al *pin* positivo (VCC) y el marrón o negro al *pin* negativo (GND).



Como el consumo de corriente de los servomotores es alto, necesitamos conectar una fuente de alimentación externa a nuestro sistema. En este caso, utilizaremos una fuente de 5V y 2A.

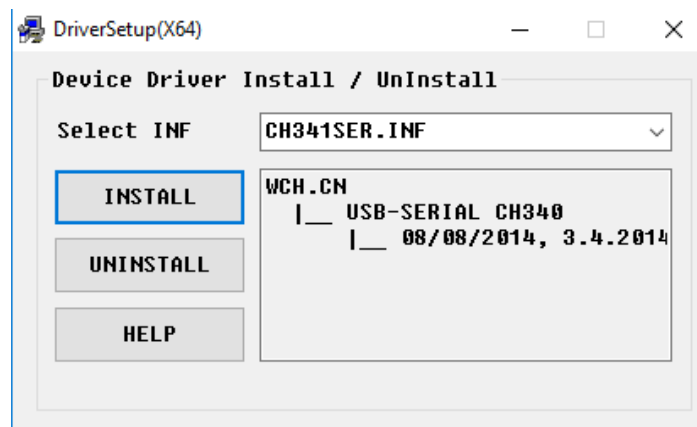
Paso 2 - Programar los ángulos de los servomotores

La programación la realizaremos con mBlock3, un entorno de programación basado en Scratch2 que permite programar proyectos de Arduino utilizando bloques. Pueden descargarlo siguiendo el siguiente enlace: <http://www.mblock.cc/software-1/mblock/mblock3/>

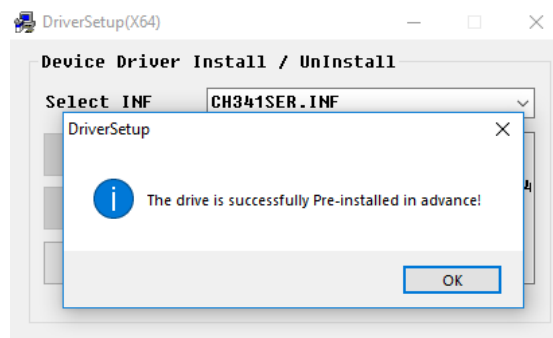
También necesitamos descargar un *driver* para poder utilizar la placa DFRobot. Podemos descargarlo del siguiente enlace: <https://www.geekfactory.mx/download/driver-ch340-windows/#>

Un *driver* es un programa que sirve para controlar un dispositivo de hardware.

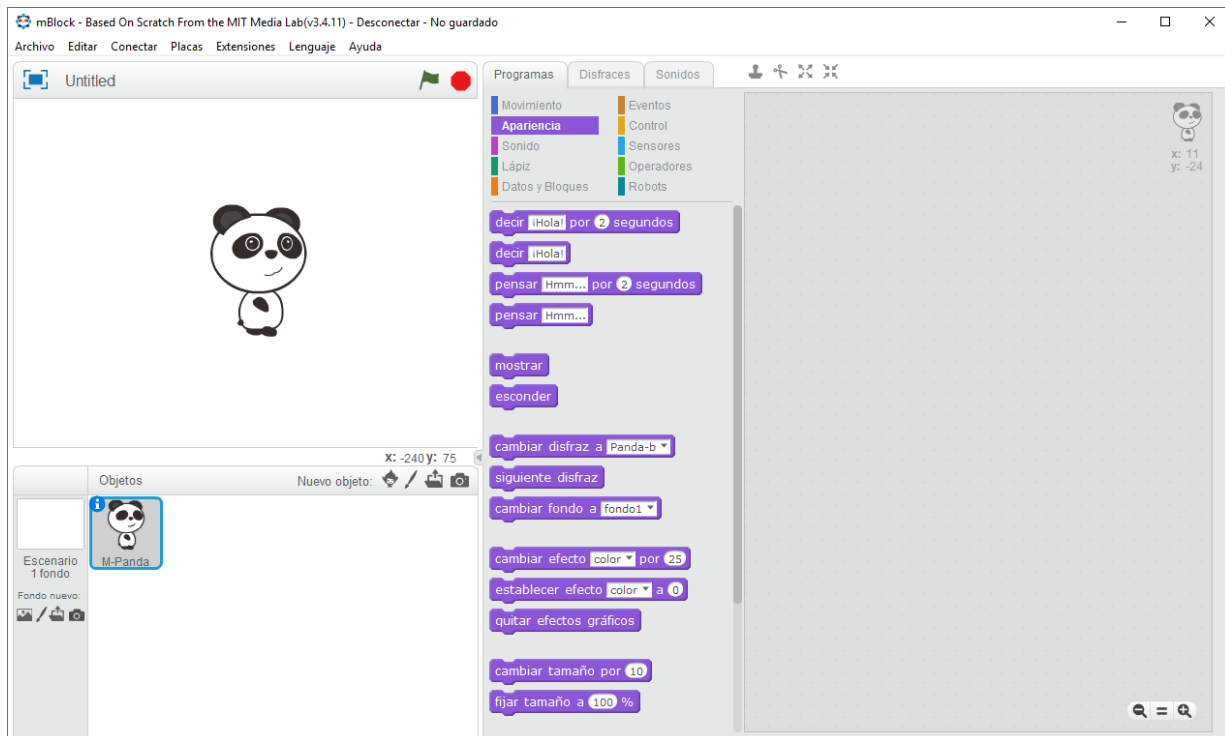
Para instalar el *driver* debemos seleccionar *Install* y esperar hasta que se complete la instalación.



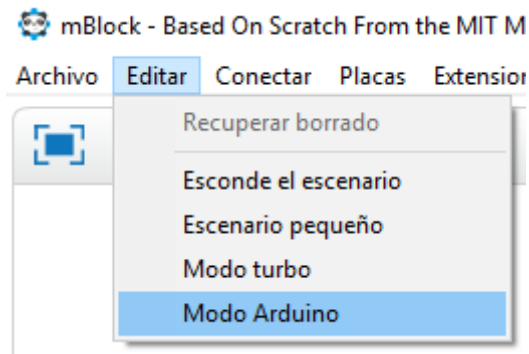
Una vez que se haya completado nos aparecerá la siguiente pantalla. En este momento ya tenemos el *driver* listo para trabajar, sin necesidad de reiniciar la computadora.



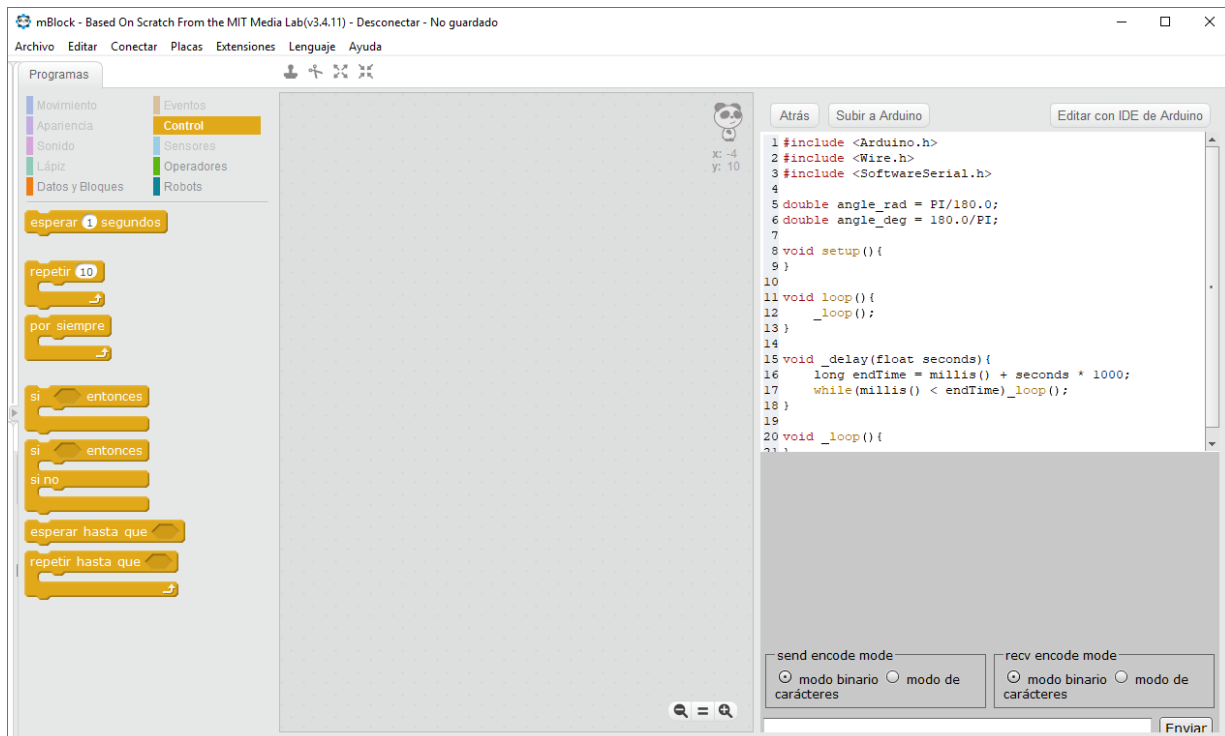
Cuando abrimos mBlock3, veremos una pantalla como la siguiente:



Para programar un proyecto de Arduino con mBlock3 debemos seleccionar el “Modo Arduino” desde el menú.



Al seleccionar este modo, el programa cambiará de aspecto. Se verá un área en el centro que es la que utilizaremos para programar con bloques. A la derecha se verá un campo donde aparecerá el código escrito que le corresponde a los bloques que están en el centro. Este código se irá escribiendo automáticamente a medida que se vaya armando el programa con los bloques.



Los bloques están agrupados por categorías. En este caso, se usarán bloques de las categorías **“Robots”**, **“Control”**, **“Operadores”** y **“Datos y Bloques”**. Cuando seleccionamos una de estas categorías, se pueden visualizar todos los bloques que pertenecen a ese grupo.



Para comenzar el programa necesitamos agregar el bloque “Programa de Arduino” que se encuentra dentro de la categoría **Robots**. Luego programaremos todos los bloques correspondientes a los servomotores para que se muevan formando un ángulo de 90° con respecto a su posición inicial. Es importante acordarse de modificar el número del *pin* correspondiente a cada servomotor (de acuerdo al *pin* de la placa Arduino que se haya utilizado) en los distintos bloques.



Veremos que a la derecha se muestra el código escrito que corresponde a este programa:

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

#include <Servo.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
Servo servo_2;
Servo servo_3;
Servo servo_4;
Servo servo_5;
Servo servo_6;
Servo servo_7;

void setup(){
    servo_2.attach(2); // init pin
    servo_3.attach(3); // init pin
    servo_4.attach(4); // init pin
    servo_5.attach(5); // init pin
    servo_6.attach(6); // init pin
    servo_7.attach(7); // init pin
    servo_2.write(90); // write to servo
    servo_3.write(90); // write to servo
    servo_4.write(90); // write to servo
    servo_5.write(90); // write to servo
    servo_6.write(90); // write to servo
    servo_7.write(90); // write to servo
}

void loop(){
    _loop();
}

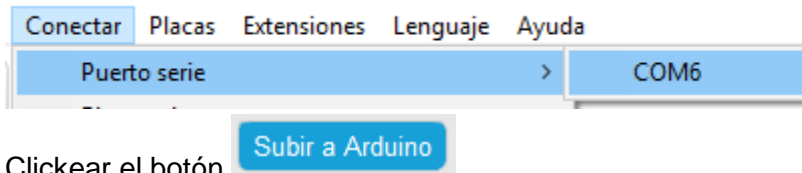
void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}
```

Paso 3 - Subir el código a la placa Arduino

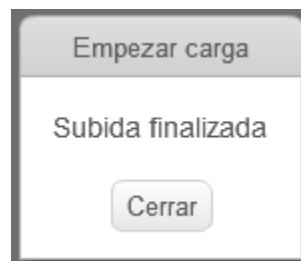
Para subir el código de nuestro programa a la placa Arduino, necesitamos:

1. Conectar la placa Arduino a la entrada USB de nuestra PC.
2. Chequear que en el menu "Placas" esté seleccionado "Arduino Uno".
3. Seleccionar el puerto serie al que está conectada la placa.



4. Clickear el botón

Al terminar de subir nuestro código, veremos este mensaje



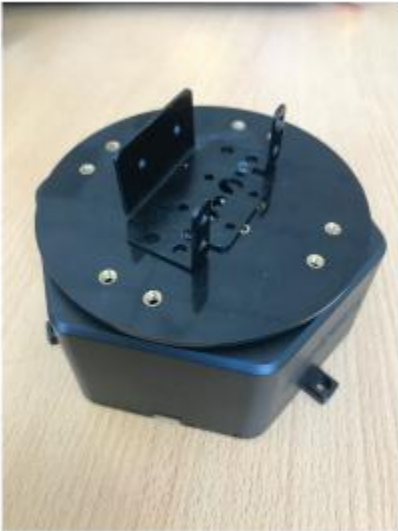



Con la conexión previamente realizada, los servomotores deberán moverse a su nueva posición a 90°.

Paso 4 - Armado del brazo robótico

En el siguiente enlace pueden encontrar un video que muestra cómo realizar el armado del brazo robótico paso a paso: <https://youtu.be/VqmDPwIRkZI>

Estos son los pasos que se deben seguir para su construcción:

<p>1</p>  <p>Fijar el servo DSS M15 a la base</p>	<p>2</p>  <p>Ubicar las pistas del rodamiento de cilindros en la parte superior de la base y en la inferior del plato giratorio. Luego fijamos el plato giratorio mediante el alojamiento central.</p>
<p>3</p>  <p>Vincular el soporte al plato giratorio.</p>	<p>4</p>  <p>Montar el rodamiento en el soporte del servomotor.</p>

5



Montar el vínculo del servomotor a la horquilla.

6



Montar el servomotor DSS M15 al soporte.

7



Vincular la horquilla con el servomotor.

8



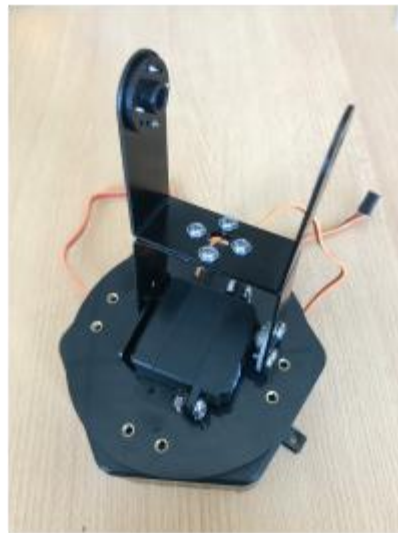
Fijar el vínculo del servomotor al servomotor

9



Montar el vínculo del servomotor a la horquilla.

10



Vincular las horquillas entre sí.

11



Montar el rodamiento en el soporte del servomotor.

12



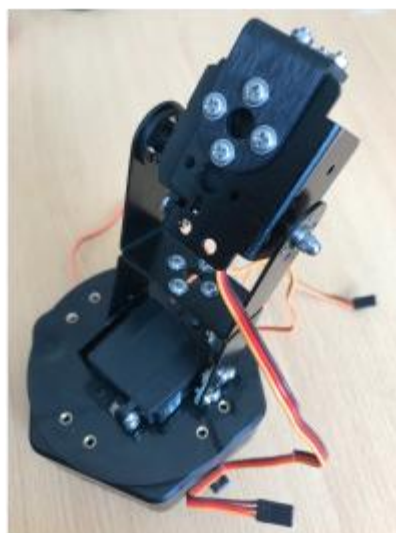
Montar la pieza L al soporte del servomotor

13



Montar el servomotor DSS M15 al soporte.

14



Montar el conjunto en la horquilla.

15



Fijar el vínculo del servomotor al servomotor.

16



Montar el vínculo del servomotor a la horquilla.

17



Fijar la horquilla a la pieza L.

18



Vincular los soportes servomotores entre sí.

19



Montar el rodamiento en el vínculo soporte-soporte.

20



Fijar el Servo DF05BB con el soporte.

21



Fijar el servo HS-311 con el soporte.

22



Fijar el conjunto de servos y soportes a la horquilla.

23



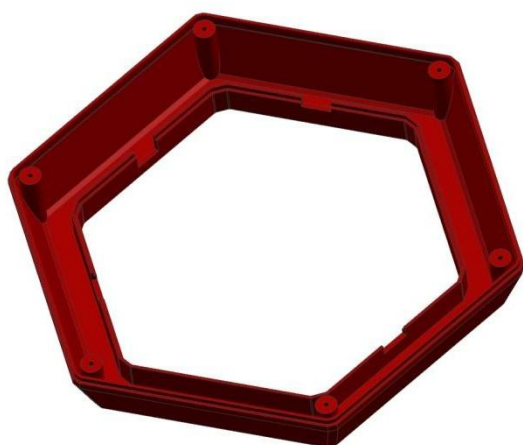
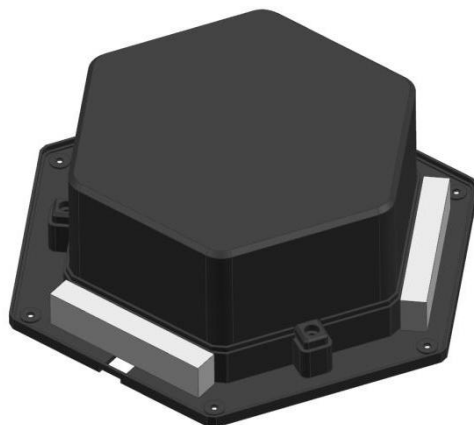
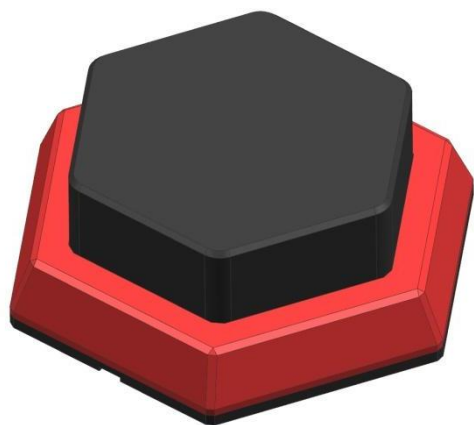
Montar la pinza al servo DF05BB.

24



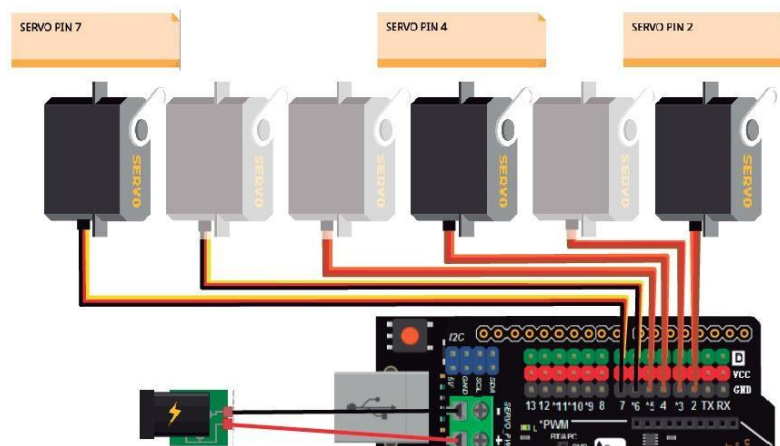
Montar el servo HS-311 la pinza.

En caso de no contar con una superficie de apoyo/mesa en la que se pueda fijar la base del brazo, se sugiere descargar modelos de piezas como las que se muestran en las imágenes que se ven a continuación, imprimirlas y colocar dentro de las carcasas algún tipo de suplemento que les permita obtener la estabilidad necesaria (por ejemplo, material metálico de descarte). Se pueden encontrar ingresando a <https://enfoco.net.ar/sd>.



Paso 5 - Programar un movimiento sincronizado de los servomotores

Una vez que hayamos ensamblado el brazo robótico, programaremos los servomotores para que realicen movimientos de forma sincronizada. Para comenzar accionaremos el servomotor de la base (*pin 2*), el del medio (*pin 4*) y el que permite la apertura/cierre de la pinza (*pin 7*).



Continuamos trabajando sobre el programa que realizamos previamente para calibrar los servomotores, pero le agregamos el bloque “Por siempre”, de manera que esta nueva secuencia de movimientos se repita una y otra vez. Agregamos uno por uno los bloques dejando una espera entre cada movimiento. Nuestro programa debería ser similar al siguiente modelo:



A la derecha de la pantalla podemos ver el código escrito correspondiente, que debería ser similar al siguiente:

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

#include <Servo.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
Servo servo_2;
Servo servo_3;
Servo servo_4;
```

```
Servo servo_5;
Servo servo_6;
Servo servo_7;

void setup(){
    servo_2.attach(2); // init pin
    servo_3.attach(3); // init pin
    servo_4.attach(4); // init pin
    servo_5.attach(5); // init pin
    servo_6.attach(6); // init pin
    servo_7.attach(7); // init pin
    servo_2.write(90); // write to servo
    servo_3.write(90); // write to servo
    servo_4.write(90); // write to servo
    servo_5.write(90); // write to servo
    servo_6.write(90); // write to servo
    servo_7.write(90); // write to servo
}

void loop(){
    servo_2.write(180); // write to servo
    _delay(1);
    servo_4.write(0); // write to servo
    _delay(1);
    servo_7.write(180); // write to servo
    _delay(1);
    servo_4.write(90); // write to servo
    _delay(1);
    servo_2.write(90); // write to servo
    _delay(1);
    servo_4.write(0); // write to servo
    _delay(1);
    servo_7.write(90); // write to servo
    _delay(1);
    servo_4.write(90); // write to servo
    _delay(1);
    _loop();
}
```

```
void _delay(float seconds){  
  long endTime = millis() + seconds * 1000;  
  while(millis() < endTime)_loop();  
}  
  
void _loop(){  
}
```

Nivel Intermedio

En otro sector de la fábrica precisan un brazo robótico para manipular sustancias que resultan peligrosas para la salud de las personas. Por esta razón, necesitan desarrollar un sistema que les permita controlar su accionar a la distancia, evitando al máximo el contacto de los operarios con estas sustancias.

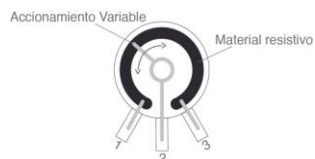
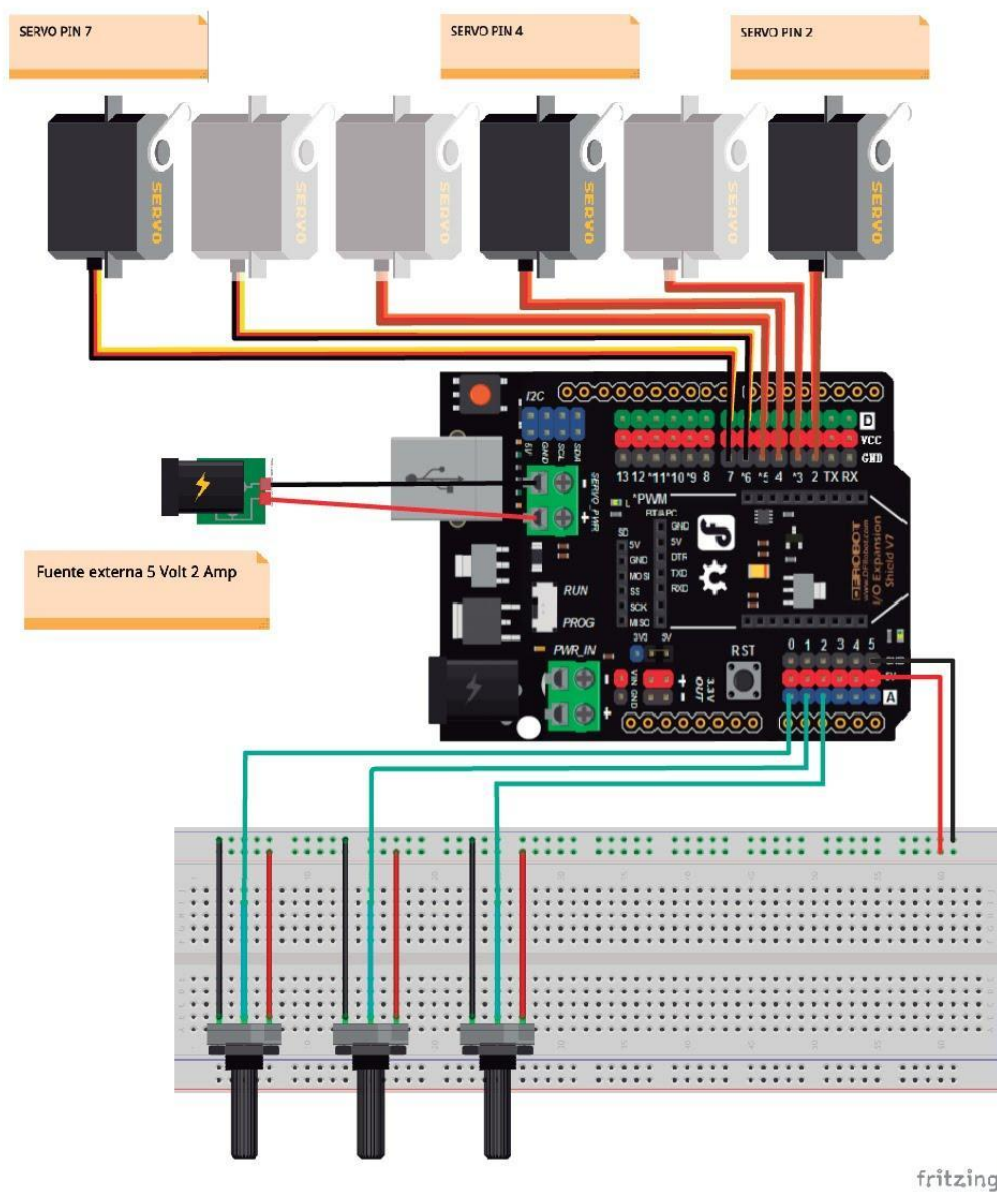
En este momento, se agregan tres potenciómetros al sistema para poder controlar manualmente los servomotores. Luego se completará con otros tres potenciómetros, uno por cada servomotor, lo que permitirá mover el brazo robótico en su totalidad.

Paso 1 - Armar el circuito

Si queremos controlar el brazo robótico a distancia, podemos utilizar potenciómetros para determinar la posición de cada servomotor. Para ello utilizaremos una Protoboard sobre la que realizaremos las conexiones necesarias. Comenzaremos conectando tres potenciómetros para controlar los tres servomotores que programamos anteriormente.

Las entradas analógicas de la placa Arduino (que van desde el pin A0 al A5) se diferencian de las digitales (que van desde el pin 0 al 13) en que estas últimas solo pueden leer dos estados: alto (1) o bajo (0), dependiendo de si la tensión en la entrada está más cerca de **5V** o de **0V**.

Las analógicas, en cambio, nos permiten saber la tensión de la entrada con una resolución mayor, mostrando una escala que va de **0** (cuando hay **0V**) a **1023** (cuando hay **5V**). Si, por ejemplo, la tensión en la entrada es de **1.25V**, leen un valor aproximado de **255**.



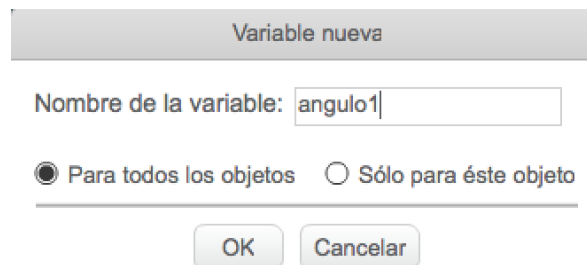
Un potenciómetro es un resistor cuyo valor de resistencia se puede modificar y controlar de forma manual. En muchos dispositivos eléctricos los potenciómetros son utilizados para regular el nivel de tensión. Por ejemplo, en un parlante el potenciómetro se puede utilizar para ajustar el volumen; así como en un monitor se puede utilizar para controlar el brillo.

Paso 2 - Crear variables

A partir de este momento, el ángulo de cada servomotor ya no tendrá un valor fijo, sino que dependerá de la posición relativa de cada potenciómetro. Por eso, para programar el movimiento de cada parte del brazo, lo primero que tenemos que hacer es leer el valor del *pin* analógico en el cual conectamos el servo. Para poder obtener y guardar este dato, crearemos una nueva variable.



Como nombre le pondremos "ángulo1" (presentamos este nombre a modo de ejemplo, pero puede elegirse otro sin problema). Lo escribimos sin tilde ya que en las líneas de código las palabras no pueden llevar tilde ni usar la letra Ñ. A continuación tendremos que crear una variable por cada potenciómetro, por lo cual sugerimos utilizar el mismo nombre para todas modificando el número de acuerdo al servomotor que controle cada una, para que sea más fácil



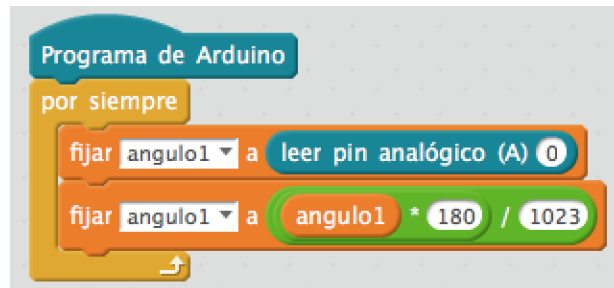
Una variable es un espacio en la memoria que nos permite guardar un dato y ser leído en otra instancia del programa. Es importante tener en cuenta al momento de crear una variable que su nombre no puede comenzar con un número ni contener espacios.

A esta variable le asignaremos el valor del *pin* analógico que corresponda.



Paso 3 - Adaptar los valores

Hay que tener en cuenta que el servomotor se mueve entre 0° y 180° , pero la lectura del *pin* nos entrega valores entre 0 y 1023. Por lo tanto, necesitamos adaptar el valor leído a una escala que varíe entre 0 y 180, que puede resolverse desde la programación con una regla de tres simple. Para esto, debemos programar nuestro sistema para que multiplique el valor obtenido por el *pin* por 180 y luego lo divida por 1023.

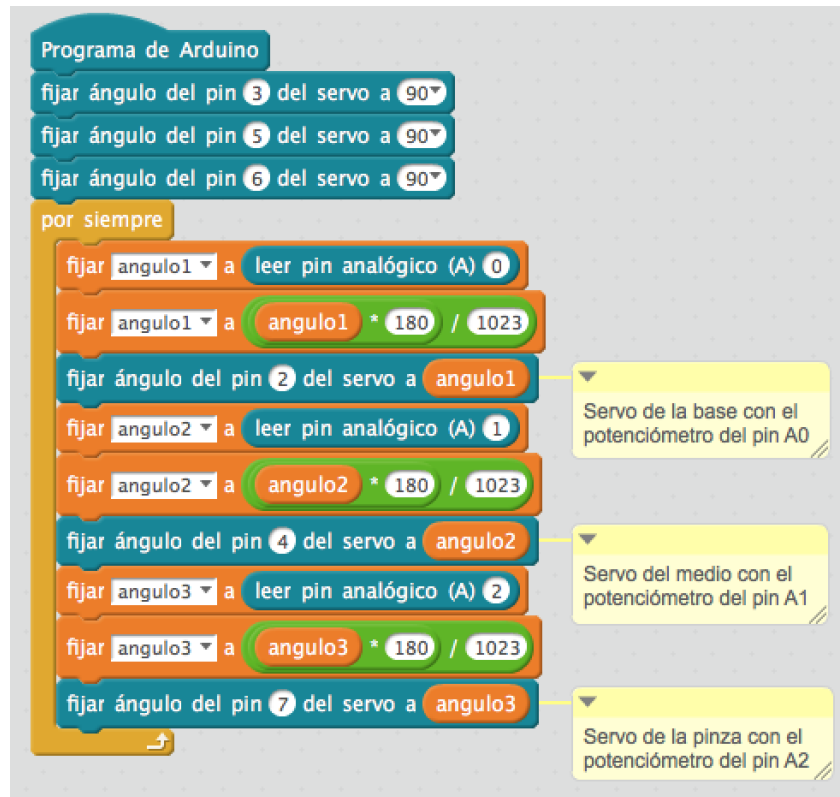


A continuación, agregamos un bloque que fije el ángulo al que debe moverse el servomotor de acuerdo al valor de esa variable.



Paso 4 - Programar los servomotores restantes

Vamos a repetir los pasos que seguimos recién con el resto de los servomotores. Es importante que al principio del programa pongamos todos los servomotores (menos el servomotor con el que trabajamos previamente) en la posición de 90° . El código nos quedaría de la siguiente forma:



A la derecha de la pantalla podemos ver el código escrito correspondiente, que debería ser similar al siguiente:

```

#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <Servo.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double angulo1;
double angulo2;
double angulo3;
Servo servo_3;
Servo servo_5;
Servo servo_6;
Servo servo_2;
Servo servo_4;
Servo servo_7;

```

```

void setup(){
    servo_3.attach(3); // init pin
    servo_5.attach(5); // init pin
    servo_6.attach(6); // init pin
    pinMode(A0+0,INPUT);
    servo_2.attach(2); // init pin
    pinMode(A0+1,INPUT);
    servo_4.attach(4); // init pin
    pinMode(A0+2,INPUT);
    servo_7.attach(7); // init pin
    servo_3.write(90); // write to servo
    servo_5.write(90); // write to servo
    servo_6.write(90); // write to servo
}

void loop(){
    angulo1 = analogRead(A0+0);
    angulo1 = ((angulo1) * (180)) / (1023);
    servo_2.write(angulo1); // write to servo
    angulo2 = analogRead(A0+1);
    angulo2 = ((angulo2) * (180)) / (1023);
    servo_4.write(angulo2); // write to servo
    angulo3 = analogRead(A0+2);
    angulo3 = ((angulo3) * (180)) / (1023);
    servo_7.write(angulo3); // write to servo
    _loop();
}

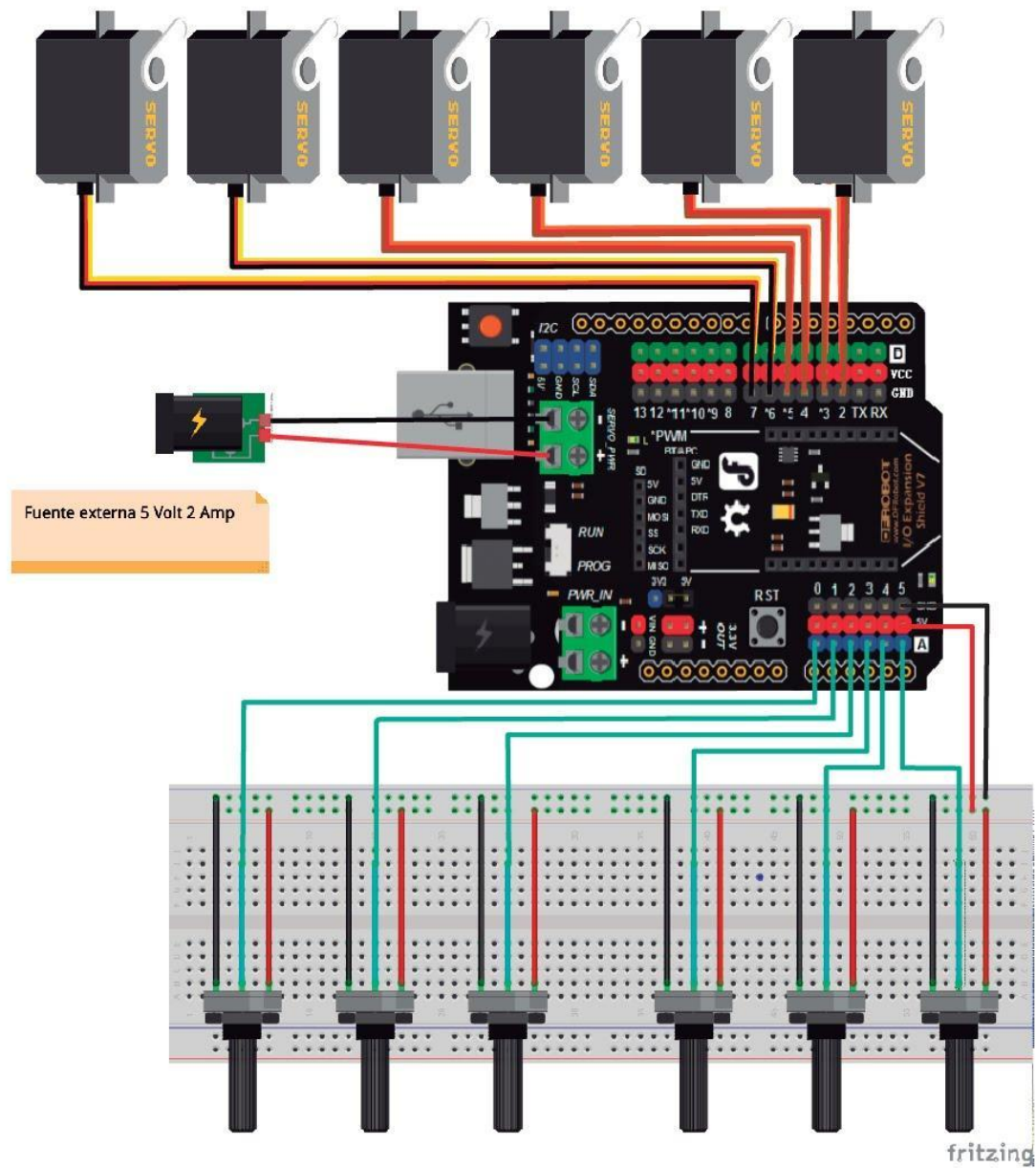
void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}

```

Paso 5 - Controlar todos los servomotores

Si queremos controlar todos los servomotores del brazo robótico, vamos a necesitar conectar tres potenciómetros más.



La programación es similar a la que ya tenemos armada. Simplemente debemos agregar las variables y lecturas correspondientes a estos tres servomotores.



El código escrito que se puede ver a la derecha de la pantalla debería quedar como se ve a continuación:

```

#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

```

```

#include <Servo.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double angulo1;
double angulo2;
double angulo3;
double angulo4;
double angulo5;
double angulo6;
Servo servo_3;
Servo servo_5;
Servo servo_6;
Servo servo_2;
Servo servo_4;
Servo servo_7;

void setup(){
    servo_3.attach(3); // init pin
    servo_5.attach(5); // init pin
    servo_6.attach(6); // init pin
    pinMode(A0+0,INPUT);
    servo_2.attach(2); // init pin
    pinMode(A0+1,INPUT);
    servo_4.attach(4); // init pin
    pinMode(A0+2,INPUT);
    servo_7.attach(7); // init pin
    pinMode(A0+3,INPUT);
    pinMode(A0+4,INPUT);
    pinMode(A0+5,INPUT);
    servo_3.write(90); // write to servo
    servo_5.write(90); // write to servo
    servo_6.write(90); // write to servo
}

void loop(){
    angulo1 = analogRead(A0+0);
    angulo1 = ((angulo1) * (180)) / (1023);
    servo_2.write(angulo1); // write to servo

```



```

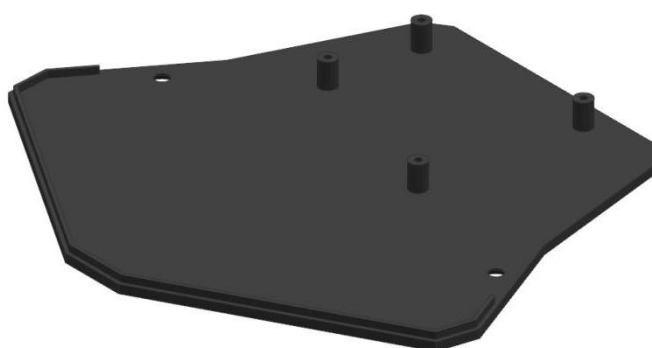
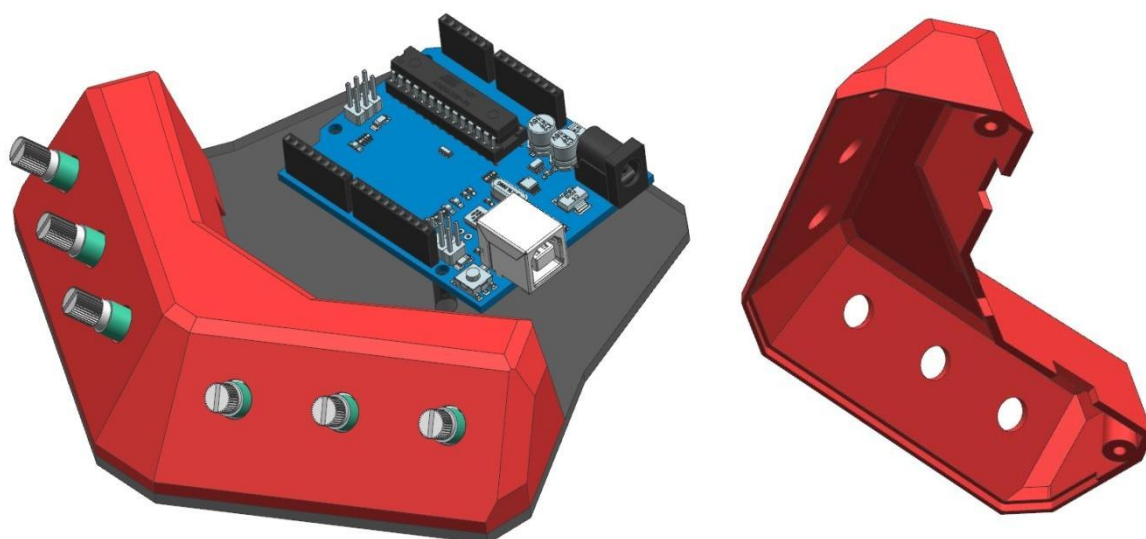
    angulo2 = analogRead(A0+1);
    angulo2 = ((angulo2) * (180)) / (1023);
    servo_4.write(angulo2); // write to servo
    angulo3 = analogRead(A0+2);
    angulo3 = ((angulo3) * (180)) / (1023);
    servo_7.write(angulo3); // write to servo
    angulo4 = analogRead(A0+3);
    angulo4 = ((angulo4) * (180)) / (1023);
    servo_3.write(angulo4); // write to servo
    angulo5 = analogRead(A0+4);
    angulo5 = ((angulo5) * (180)) / (1023);
    servo_5.write(angulo5); // write to servo
    angulo6 = analogRead(A0+5);
    angulo6 = ((angulo6) * (180)) / (1023);
    servo_6.write(angulo6); // write to servo
    _loop();
}

void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}

```

Para poder controlar los potenciómetros sin dificultad necesitaremos traspasar nuestro sistema de la placa Protoboard donde lo probamos a una carcasa definitiva. De esta manera podremos montar y ajustar cada potenciómetro e, incluso, utilizando un cableado más extenso, comandar nuestro brazo a la distancia que queramos. Para armar una carcasa que nos permita operar de manera más cómoda los potenciómetros, se sugiere imprimir los modelos que se presentan a continuación. Pueden descargar los archivos correspondientes del sitio web del proyecto.



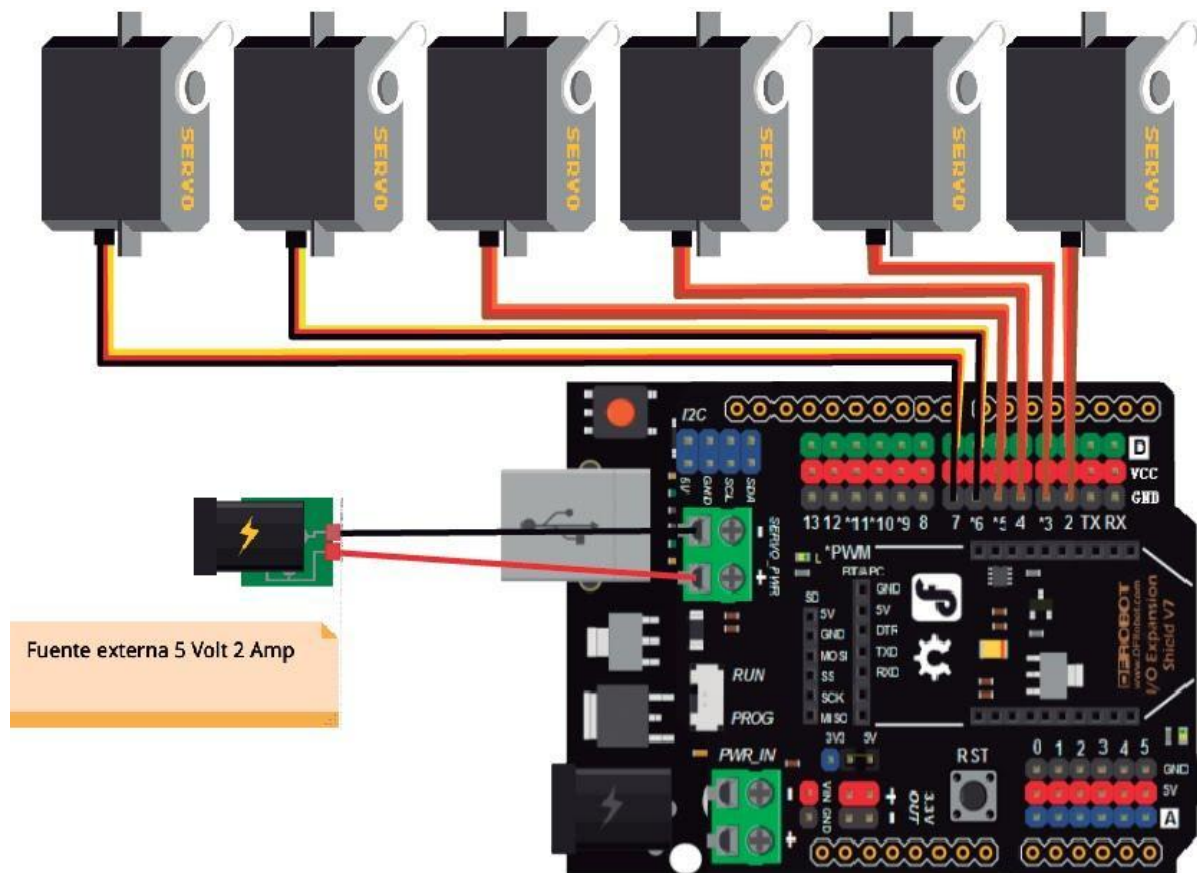
Nivel Avanzado

En el sector de control de calidad también se requiere la implementación de un brazo robótico. Este debe poder tomar los productos defectuosos que se encuentran en una cinta transportadora y colocarlos en otra que los transportará al sector de revisión, donde se evaluará si deben ser desechados o pueden ser reparados.

En esta actividad se trabaja de forma sincronizada con los seis servomotores, generando un movimiento completo del brazo robótico.

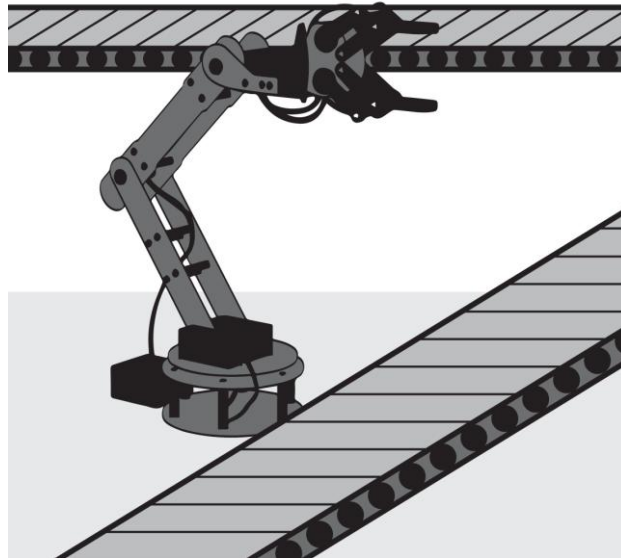
Paso 1 - Armar el circuito

En este caso, el brazo robótico se moverá de forma automática, por lo que tendremos conectados los seis servomotores sin necesidad de usar los potenciómetros.



Paso 2 - Programar la secuencia de movimiento

En esta instancia, vamos a programar el brazo de manera que pueda trasladar los objetos de una cinta transportadora a otra que está ubicada a menor altura y a 90° de la primera, tal como indica la figura.



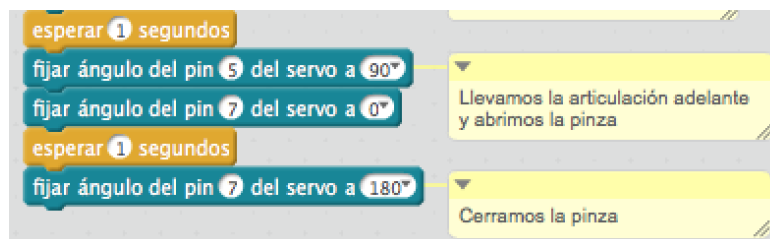
Para tomar los productos de la primera cinta transportadora el brazo deberá extenderse, mover la articulación de la pinza hacia adelante y tomar con la pinza los productos. Luego, para ubicar el producto en la segunda cinta, deberá mover la articulación de la pinza nuevamente hacia atrás, girar a 90° y extenderse para dejar el producto en la posición final. Además, sumaremos como complejidad que los objetos se encuentren inicialmente en posición vertical (en la cinta superior) y el brazo deba dejarlos en posición horizontal, por lo que deberá rotarlos antes de depositarlos en la cinta inferior.

Nuestro sistema debe estar programado, entonces, para operar respetando la siguiente secuencia.

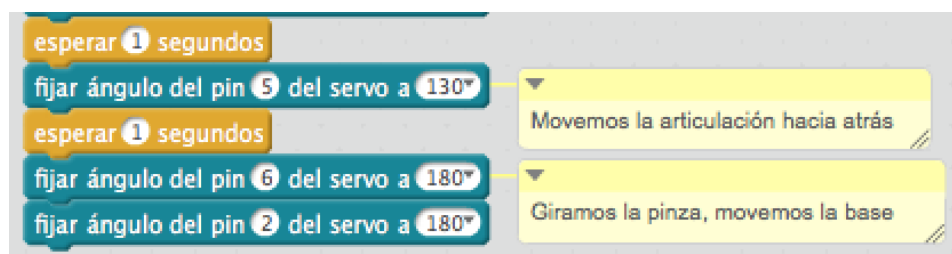
En primer lugar, colocamos todos los servomotores a 90° . Luego, movemos el servomotor de la base y la articulación de la pinza para que el brazo gire en dirección a la cinta sin golpear los objetos.



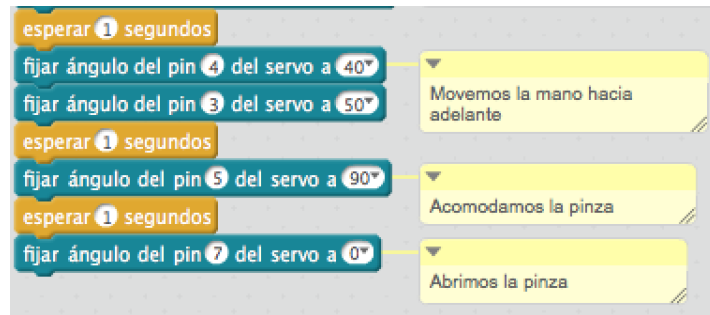
A continuación, necesitamos mover la articulación de la pinza hacia adelante y abrir la pinza para que esta pueda rodear el objeto. Luego de una breve espera cerramos la pinza para agarrar el objeto.



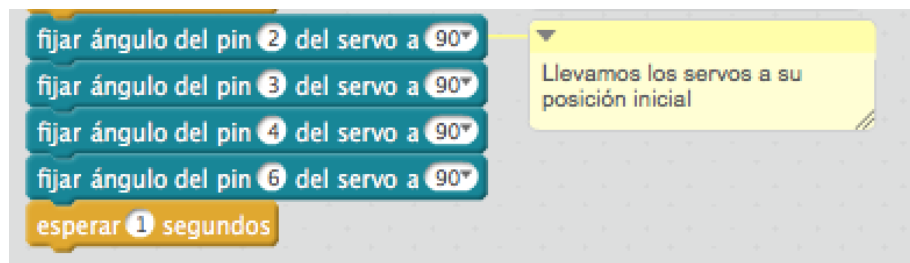
Movemos la articulación nuevamente hacia atrás y, luego, giramos la base y la pinza.



Ahora debemos mover el brazo hacia adelante, acomodar la pinza (para que el objeto se encuentre paralelo a la cinta) y abrirla, dejando un segundo de tiempo de espera entre cada una de las acciones.



Una vez que el brazo soltó el objeto, debemos volver a mover los servos a su posición inicial de 90°.



Para que esto se repita constantemente, sin parar, necesitamos agregar el bloque “Por siempre” en nuestro programa. Finalmente, quedaría de la siguiente manera:



El código escrito que se puede ver a la derecha de la pantalla debería quedar como se ve a continuación:

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
```

```
#include <Servo.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
Servo servo_2;
Servo servo_3;
Servo servo_4;
Servo servo_5;
Servo servo_6;
Servo servo_7;

void setup(){
    servo_2.attach(2); // init pin
    servo_3.attach(3); // init pin
    servo_4.attach(4); // init pin
    servo_5.attach(5); // init pin
    servo_6.attach(6); // init pin
    servo_7.attach(7); // init pin
    servo_2.write(90); // write to servo
    servo_3.write(90); // write to servo
    servo_4.write(90); // write to servo
    servo_5.write(90); // write to servo
    servo_6.write(90); // write to servo
    servo_7.write(90); // write to servo
    _delay(2);
}

void loop(){
    servo_2.write(0); // write to servo
    servo_5.write(130); // write to servo
    _delay(1);
    servo_5.write(90); // write to servo
    servo_7.write(0); // write to servo
    _delay(1);
    servo_7.write(180); // write to servo
    _delay(1);
    servo_5.write(130); // write to servo
    _delay(1);
}
```



```
servo_6.write(180); // write to servo
servo_2.write(180); // write to servo
_delay(1);
servo_4.write(40); // write to servo
servo_3.write(50); // write to servo
_delay(1);
servo_5.write(90); // write to servo
_delay(1);
servo_7.write(0); // write to servo
_delay(1);
servo_2.write(90); // write to servo
servo_3.write(90); // write to servo
servo_4.write(90); // write to servo
servo_6.write(90); // write to servo
_delay(1);
_loop();
}

void _delay(float seconds){
  long endTime = millis() + seconds * 1000;
  while(millis() < endTime)_loop();
}

void _loop(){
}
```

Cierre

Una vez finalizado este proyecto, es posible extenderlo si se quiere continuar. Estas son algunas opciones sugeridas:

- Modificar la trayectoria del brazo robótico.
- Agregar un sensor de color, para poder realizar una selección diferenciada de objetos.
- Incorporar un sistema de control inalámbrico, ya sea por bluetooth o wifi.

El proceso de resolución de problemas como los que se han planteado aquí permite la movilización y la integración de distintos saberes en la búsqueda de soluciones posibles a una situación dada. Si bien la información aquí fue presentada a modo de instructivo, se espera que sean los estudiantes organizados en pequeños grupos quienes vayan encontrando las mejores formas para construir los dispositivos. Esto implica preparar los materiales para que cada grupo cuente con todo lo necesario para la construcción del proyecto. Además, al interior de cada grupo, los estudiantes deben distribuirse los roles y las tareas de acuerdo a las demandas que van teniendo en las actividades.

Es importante que los docentes acompañen las producciones de cada grupo monitoreando los avances de todos los estudiantes y presentando la información que se considere necesaria para continuar la tarea. Pero, al mismo tiempo, es necesario que habiliten espacios para que los alumnos realicen hipótesis, planteen interrogantes, indaguen, prueben y realicen ajustes de acuerdo a lo que ellos mismo van pensando sobre cómo llevar a cabo el proyecto.

En este sentido, registrar lo que se va haciendo, las preguntas de los alumnos, las pruebas, los errores y cómo se fueron construyendo los dispositivos, permite reflexionar sobre la propia práctica, reforzar los aprendizajes construidos a lo largo de este proceso y poder volver a ese material disponible para próximos proyectos que se realicen.

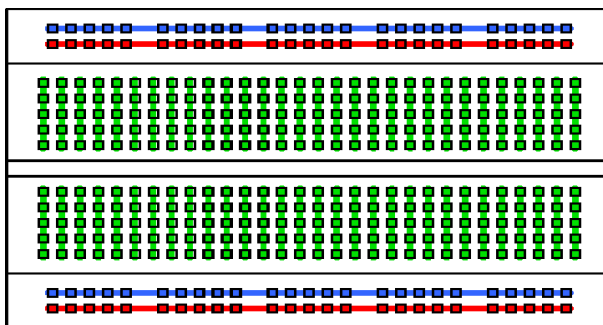
Una vez terminado el proyecto, se sugiere reunir y organizar con el grupo el registro que se hizo del proceso realizado. Esta instancia de sistematización también permite movilizar capacidades vinculadas a la comunicación porque implica tomar decisiones respecto a cómo se quiere mostrar el proyecto a otros (otros grupos, otras escuelas, otros docentes, a la comunidad, etc.).

Glosario

Electrónica y arduino

Protoboard: Es una placa experimental que permite el prototipado rápido de circuitos electrónicos. Tiene orificios para insertar las patas de los componentes permitiendo que se conecten sin tener que recurrir a la soldadura.

El mismo posee una grilla de orificios que se encuentran conectados entre sí siguiendo el esquema de la imagen. Las líneas de conexión superior e inferior recorren la placa de punta a punta y suelen utilizarse para la alimentación del circuito, mientras que las líneas verdes se suelen utilizar para interconectar componentes. Tomar en cuenta que las líneas verdes se interrumpen en el centro de la placa. Generalmente se utilizan cables del tipo dupont para realizar conexiones en la protoboard



Servomotor: Un servo es un dispositivo que se compone de un motor y un sistema de control que le permite ubicarse en una posición específica. Los servos más comunes pueden moverse en un rango de 0° a 180°, sin poder girar de forma continua. Se suelen utilizar en aplicaciones tipo barreras o brazos mecánicos. La programación de los mismos es muy simple, teniendo que especificar únicamente el ángulo al que se lo quiere posicionar.

Existen también servos de “rotación continua” que permiten realizar un control relativamente preciso del movimiento así como también que el eje de giros continuos sin estar acotado a un rango de movimiento como el caso de los servos estándares. Este tipo de servo requiere una lógica de programación un poco más compleja que el caso anterior.

Los servos tienen 3 pines de conexión, dos de ellos se ocupan en alimentación eléctrica (VCC marrón y GND negro) y un tercer pin que se conecta a una salida digital del Arduino (cable naranja). Para controlar el servo el arduino genera una señal con una frecuencia particular y un método de modulación de pulsos cuyo ciclo de trabajo equivale a el ángulo que se desea posicionar el servo, no confundir este método de modulación con el PWM.

Potenciómetro: Un potenciómetro es un resistor cuyo valor de resistencia variable que se controlar de forma manual o mecánica. Sirven para trabajar con bajos niveles de potencia, a modo de señal de control.

Los potenciómetros suelen tener una resistencia fija del valor especificado y un cursor que permite pararse en algún punto intermedio de esta resistencia. Para leer la posición del potenciómetro generalmente conectaremos las dos patas de la resistencia fija (los extremos) a VCC y GND. De esta forma tendremos en la pata central (cursor) un valor de tensión que representa la posición actual de la perilla.

Reconocimientos

Este trabajo es fruto del esfuerzo creativo de un enorme equipo de entusiastas y visionarios de la pedagogía de la innovación, la formación docente, la robótica, la programación, el diseño y la impresión 3D. Les agradecemos por el trabajo en equipo inspirador para traer a la realidad la obra que, en forma conjunta, realizamos INET y EDUCAR del Ministerio de Educación, Cultura, Ciencia y Tecnología de la Nación Argentina.

Contenidos

Equipo INET

Alejandro Anchava
Joreliz Andreyana Aguilera Barragán
Omar Leandro Bobrow
Alejandro Cesar Cáceres
Ezequiel Luberto
Gustavo Roberto Mesiti
Alejandro Palestrini
Judit Schneider
Pablo Trangone

Equipo Educar:

Pablo Aristide
Mayra Botta
Anabela Cathcarth
Eduardo Chiarella
María Laura Costilla
Diego Dorado
Facundo Dyszel
Federico Frydman
Matías Rinaldi
Uriel Rubilar
Camila Stecher
Carolina Sokolowicz
Nicolás Uccello

Para la confección de esta obra se contó con el apoyo de la Universidad Pedagógica Nacional "UNIPE". En particular en el desarrollo de los capítulos 1 y 2, los cuales estuvieron a cargo de los profesores Fernando Raúl Alfredo Bordinon y Alejandro Adrián Iglesias.

Producción y comunicación

Juliana Zugasti

Diseño y edición

Leonardo Frino
Mario Marrazzo

Corrección de estilo

María Cecilia Alegre

Agradecimientos especiales

Mariano Consalvo. Equipo ABP

Damián Olive. Equipo de ABP

María José Licio Rinaldi, Directora Nacional de Asuntos Federales INET, quien siempre acompañó a este equipo en todas las gestiones para su implementación

Estamos comprometidos en instalar la innovación en la escuela secundaria técnica: la robótica, la programación, el pensamiento computacional, los proyectos tecnológicos, el ABP, la impresión 3D, de manera más accesible para todos.

Agradecemos enormemente, docente, tu continua dedicación y compromiso con el futuro de tus estudiantes.

¡Estamos ansiosos por saber qué es lo que vamos a crear juntos!