

Auxiliares: Isao Parra, Rodrigo Asenjo, Rodrigo Galilea  
 Rodrigo Muñoz, Sebastian Chamas

# Guía de Práctica para la Experiencia 3

## 1. Introducción

En esta última sesión experimentaremos con nuevos elementos, como servomotores, sensores y micro-controladores. Implementaremos un sencillo circuito que nos permitirá controlar un servomotor y acostumbrarnos a la sintaxis de quien será nuestro compañero a lo largo del desarrollo del proyecto, Arduino. Luego, crearemos un algoritmo capaz de controlar una cinta transportadora que clasifique objetos según su tamaño.

## 2. Control de servomotor con Arduino

En esta actividad vamos a controlar la posición de un servomotor con Arduino usando un potenciómetro, para esto utilizaremos el circuito mostrado en la figura 1.

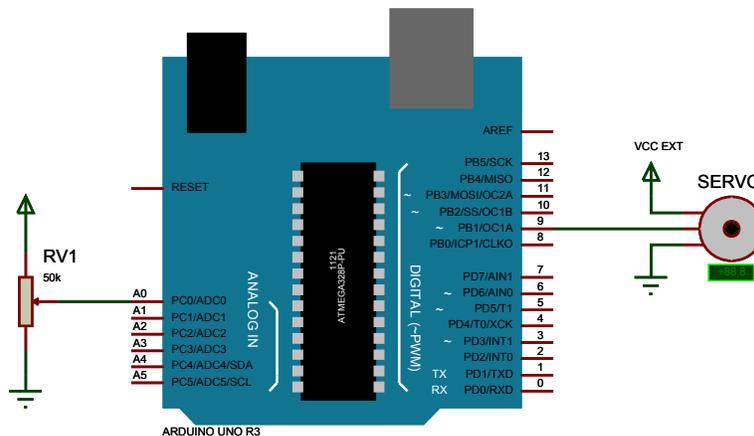


Figura 1: Esquemático de conexión para un servomotor con Arduino.



### Atención

La alimentación del servomotor **debe venir** de una fuente externa, **no use pines de la placa Arduino para alimentar cargas**, pues solo pueden soportar una corriente máxima de 40 mA.

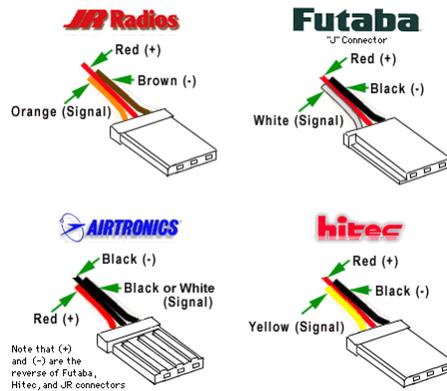


Figura 2: Pines de conexión de un servomotor de distintos fabricantes.

**Paso 1** Con ayuda de la figura 1 arme el circuito en el protoboard. Recuerde conectar la alimentación del servomotor a una fuente externa de 5 V y conectar la tierra de ambos circuitos. **Llame a un profesor auxiliar para que verifique las conexiones.**

**Paso 2** Escriba el código 1 en Arduino IDE, procure respetar la sintaxis para evitar errores. Verifique el código presionando el botón *Verificar* en Arduino IDE. Conecte la placa al computador y seleccione el modelo y puerto de conexión en el menú superior *Herramientas*. Cargue el código en la placa presionando el botón *Cargar*.

**Paso 3** Encienda la fuente externa y observe como responde el servomotor al variar el potenciómetro. Recuerde usar el limitador de corriente de la fuente de poder para evitar problemas.

```

1  /* Control de servomotor mediante potenciómetro
2  Basado en el ejemplo incluido en Arduino IDE de Michal Rinott
3  Laboratorio de Robótica y mecatrónica */
4
5  #include <Servo.h> //Libreria para sermotores
6
7  Servo myservo; //Crear objeto para controlar el servomotor
8
9  int potpin = 0; //Pin analogo de entrada
10 int val; //Variable para leer el valor del pin analogo
11
12 void setup()
13 {
14   myservo.attach(9); //Salida de señal PWM para control del servo por el pin 9
15 }
16
17 void loop()
18 {
19   val = analogRead(potpin); //Lee el valor del pin analogo (valor entre 0 y 1023)
20   val = map(val, 0, 1023, 0, 179); //Escala el valor a angulos (valor entre 0 and 180)
21   myservo.write(val); //Envia el valor de angulo al servo

```

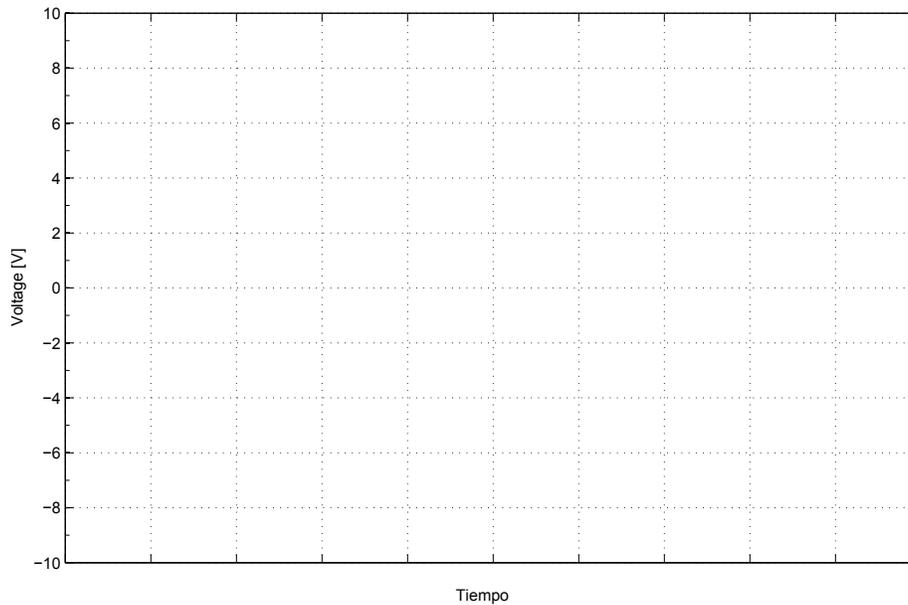
```
22 | delay(15); //Esperar 15 ms
23 | }
```

Código fuente 1: Control de servomotor mediante Arduino.

## 2.1. Preguntas

**P1** Demuestre que el voltaje en pin  $A0$  está dado por  $V_{A0} = V_{cc} \alpha$ , donde  $0 \leq \alpha \leq 1$  denota la posición del potenciómetro. Suponga que no ingresa corriente por el terminal  $A0$ .

**P2** Moviendo el potenciómetro fije el servomotor en la posición central. Con ayuda del osciloscopio bosqueje la señal de salida por el pin 9, mida el ancho del pulso  $\tau$  y el periodo  $T$  de la señal PWM. Calcule el ciclo de trabajo de la señal PWM dado por  $D = \frac{\tau}{T}$ . Observe que sucede con la señal al variar el potenciómetro.



### 3. Cinta transportadora

En esta sección emplearemos Arduino para controlar una cinta transportadora capaz de clasificar cajas de distintos tamaños. La cinta cuenta con una serie de actuadores y sensores para llevar a cabo la tarea de clasificación.

- Los **servomotores** nos permitirán retirar cajas de la cinta transportadora.
- El **sensor de distancia** nos entrega una medida de la distancia al objeto, con él podremos saber si corresponde a una caja grande o pequeña.
- El **encoder** lleva el registro del movimiento de la cinta, nos permitirá conocer la velocidad y distancia que ha recorrido ésta.

#### 3.1. Variables de programación

En este problema el hardware ya está implementado, el objetivo es diseñar un algoritmo capaz de operarlo. Serán útiles las siguientes variables y funciones:

`float beltsorter_distance`

Valor actual del sensor de distancia. Vea los perfiles de medición en las figuras 3 y 4.

`float beltsorter_belt_last_distance`

Valor de la distancia recorrida (acumulada) de la cinta transportadora en [cm].

`float beltsorter_belt_speed`

Valor de la velocidad actual de la cinta transportadora en [cm/s].

`beltsorter_servo1.write(int angulo)`

Mueve el servo 1 en el ángulo `angulo` indicado ( $0^\circ \leq \text{angulo} \leq 180^\circ$ ).

`beltsorter_servo2.write(int angulo)`

Mueve el servo 2 en el ángulo `angulo` indicado ( $0^\circ \leq \text{angulo} \leq 180^\circ$ ).

`delay(int retardo_ms)`

Pausa la ejecución por `retardo_ms` milisegundos.

`Serial.println()`

Imprime datos por el monitor serial del computador.

#### 3.2. Perfiles de medición de altura

Al pasar una caja por el pórtico el sensor de distancia cambia su valor dependiendo del tamaño del objeto. En la figuras 3 y 4 se aprecia la respuesta del sensor al paso de una caja grande y una pequeña respectivamente.

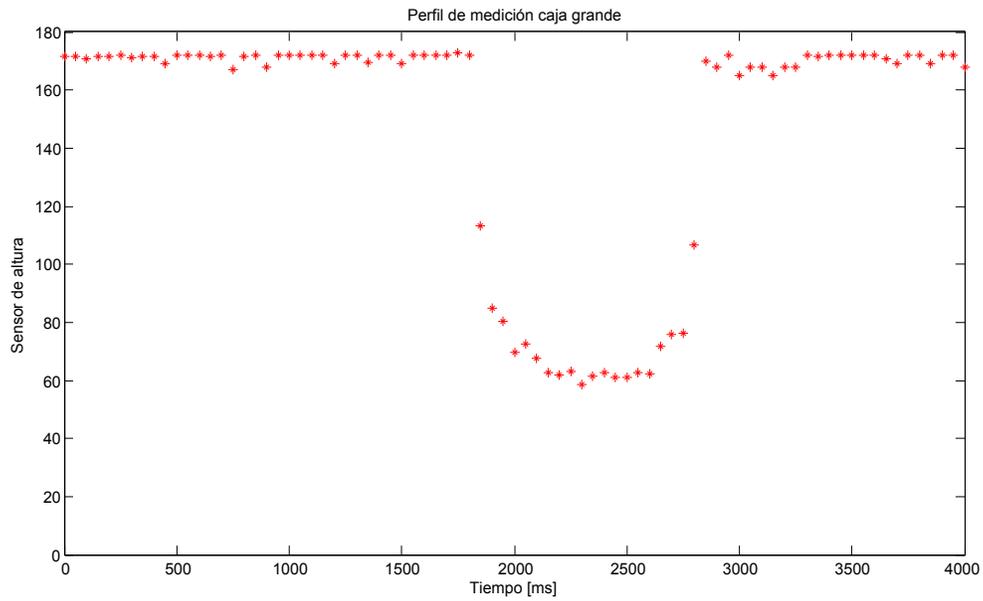


Figura 3: Perfil de medición caja grande.

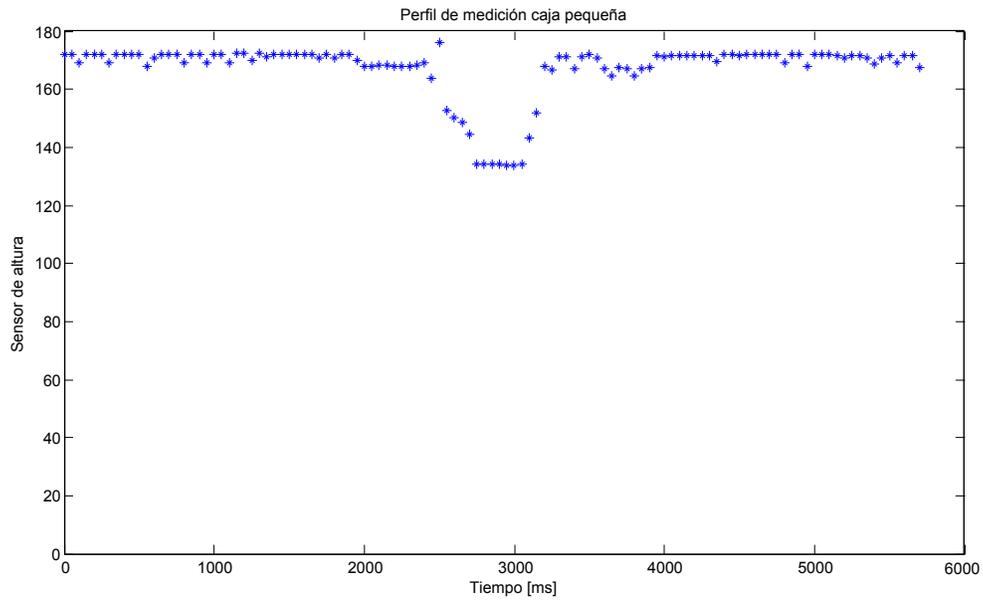


Figura 4: Perfil de medición caja pequeña.

### 3.3. Escritura del algoritmo

Escriba su código en el archivo `BeltSorter.ino` usando Arduino IDE, busque la sección habilitada para escribir que se muestra en el código 2. Esta sección se encuentra al final de la función `loop()`.

```

1  ...
2  beltsorter_last_time = micros();
3
4  //----- CÓDIGO -----
5
6
7  //Escriba su código en esta sección
8
9
10 //-----
11 // NO MODIFICAR FUNCIONES INFERIORES
12 }
13
14 void BeltSorter_set_offset_servo1(int center)
15 ...

```

Código fuente 2: Sección de `BeltSorter.ino` para escribir el algoritmo.

Presione el botón *Verificar* para chequear errores de sintaxis. Recuerde usar las variables y funciones explicadas en la sección 3.1. Si ha terminado su algoritmo llame a un profesor auxiliar para probar su código en la cinta transportadora.



#### Atención

Ordene y limpie su espacio de trabajo al terminar la experiencia.  
 Entregue los componentes al profesor auxiliar desconectados y ordenados.