

Guía Práctica #2

Microcontroladores

1. Introducción

En esta segunda sesión experimentaremos con nuevos elementos, como servomotores, sensores y microcontroladores. 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 deben realizarse las conexiones según se muestra en la figura 1.

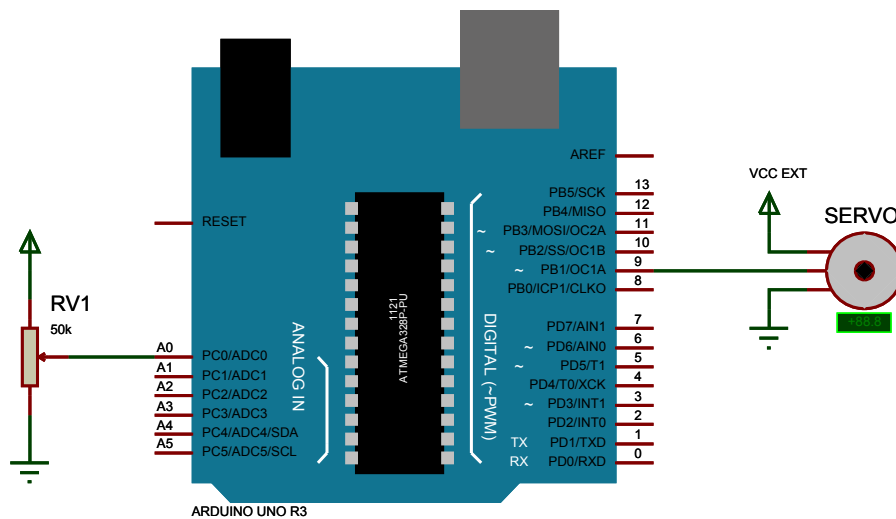


Figura 1: Esquemático de conexión servomotor y potenciómetro en Arduino.



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.

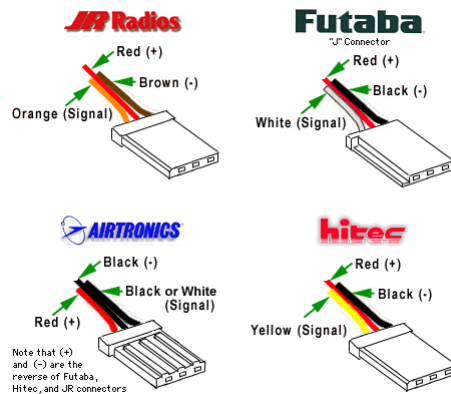


Figura 2: Pines de conexión de un servomotor de distintos fabricantes.

Siga los siguientes pasos para realizar la experiencia.

- 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 Robotica y mecatronica */
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   myservo.attach(9) ; // Salida de senal PWM para control del servo por el pin 9
14 }
15
16 void loop (){
17   val = analogRead ( potpin ) ; // Lee el valor del pin analogo ( valor entre 0 y 1023)
18   val = map ( val , 0 , 1023 , 0 , 179) ;// Escala el valor a angulos ( valor entre 0 and
19   myservo . write ( val ) ; // Envia el valor de angulo al servo
20   delay (15) ; // Esperar 15 ms
21 }

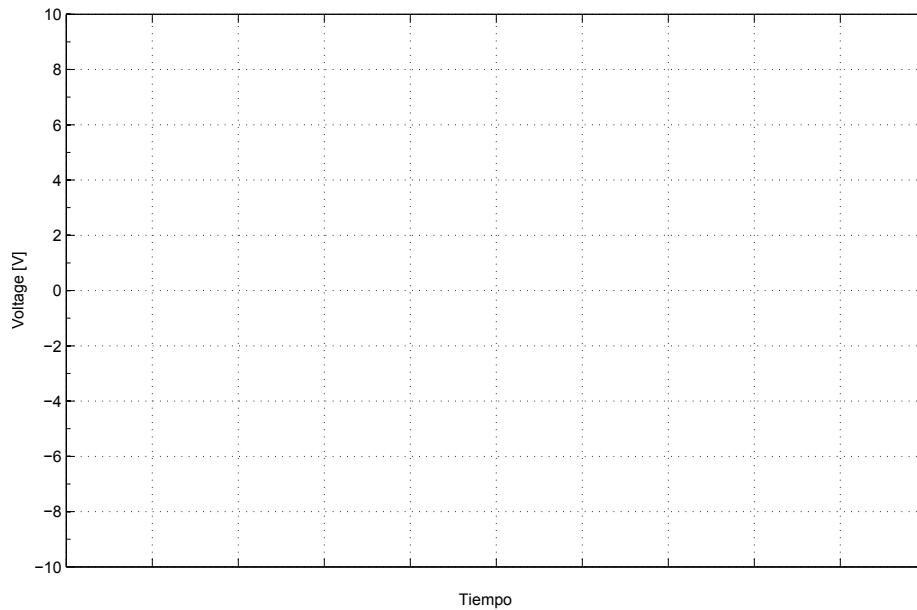
```

Código fuente 1: Control de servomotor mediante Arduino.

2.1. Preguntas

P1 Demuestre que el voltaje en pin A0 esta 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 τ 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. Sensor de Ultrasonido

En esta actividad se conectará un sensor de distancia ultrasónico y se obtendrán las distancias por el puerto serial.

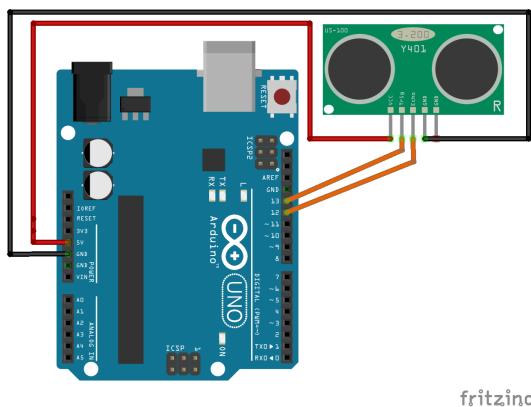


Figura 3: Esquemático de conexión sensor de ultrasonido Arduino.

Siga los siguientes pasos para realizar la experiencia.

Paso 1 Con ayuda de la figura 3 arme el circuito en el protoboard. **Llame a un profesor auxiliar para que verifique las conexiones.**

Paso 2 Escriba el código 2 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*. **Abra la ventana de comunicación serial en el menú superior *Herramientas* > *Serial Monitor***

Paso 3 Acerque distintos objetos al sensor de ultrasonido en el rango de distancia definido en el código y haga las pruebas que estime conveniente para responder la pregunta 1.

```
1 /* Robotica y Mecatronica. Experiencia 2. Sensor de ultrasonido */
2 #include <NewPing.h> // Carga la libreria NewPing.h, permite trabajar con el sensor
3 // Definicion de Pines y constantes:
4 const int pin_trig=13; // Pin para emision de pulso ultrasonico
5 const int pin_echo=12; // Pin para capturar eco producido
6 const int d_max=1000; // Distancia max a medir [cm]
7 int distancia; // variable para distancia
8 // Comienzo del programa
9 NewPing sonar(pin_trig,pin_echo,d_max); // Crea el objeto sonar
10 void setup(){
11   Serial.begin(9600); //iniciar comunicacion serial
12 }
13 void loop(){
14   distancia = sonar.ping_cm(); //entrega distancia al objeto
15   Serial.print("Distancia: "); Serial.println(distancia); //Mostrar datos por puerto
16   delay(500); //esperar 0,5 [s] entre mediciones
17 }
```

Código fuente 2: Medición de distancias con sensor de ultrasonido Arduino.

3.1. Preguntas

P1 Estime el rango de operación y la resolución del sensor. ¿Qué desventajas y limitaciones puede observar del sensor?, ¿qué problemas detectó al operar el sensor a grandes distancias?. Fije un objeto a 5 cm del sensor y calcule el error de medición con respecto a la distancia entregada por el puerto serie.

4. 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.

4.1. Interfaz de programación

En este problema el hardware ya está implementado, el objetivo es diseñar un algoritmo capaz de operarlo. Para ello se han implementado una serie de funciones que permiten obtener información y actuar los distintos elementos de la cinta transportadora.

BeltSorter

Objeto principal que encapsula la implementación de la cinta. Solo debe existir una instancia de este objeto en su código.

```
void BeltSorter::begin()
```

Inicializa la cinta, usualmente este método debería llamarse dentro del bloque `setup()`.

```
void BeltSorter::loop()
```

Actualiza los valores de la cinta, debe ser llamado de forma recurrente para obtener datos actualizados.

```
void BeltSorter::sleep()
```

Bloquea la ejecución del programa por un periodo de 50 ms.

```
void BeltSorter::openPusherA()
```

Libera el paso a través de la cinta abriendo el *pusher* A.

```
void BeltSorter::openPusherB()
    Libera el paso a través de la cinta abriendo el pusher B.
```

```
void BeltSorter::closePusherA()
    Cierra el paso a través de la cinta abriendo el pusher A.
```

```
void BeltSorter::closePusherB()
    Cierra el paso a través de la cinta abriendo el pusher B.
```

```
float BeltSorter::getHeight()
    Obtiene la altura del objeto en [mm]. Vea los perfiles de medición en las Figuras 4 y 5.
```

```
float BeltSorter::getBeltDistance()
    Obtiene la distancia acumulada recorrida por la cinta en [cm].
```

```
float BeltSorter::getBeltSpeed()
    Obtiene el valor de la rapidez actual de la cinta transportadora en [cm/s].
```

```
Serial.println()
    Imprime datos por el monitor serial del computador.
```

4.2. 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 4 y 5 se aprecia la respuesta del sensor al paso de una caja grande y una pequeña respectivamente.

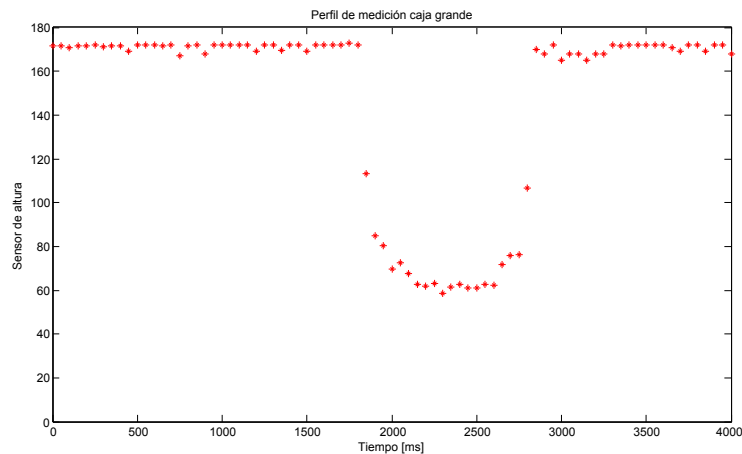


Figura 4: Perfil de medición caja grande.

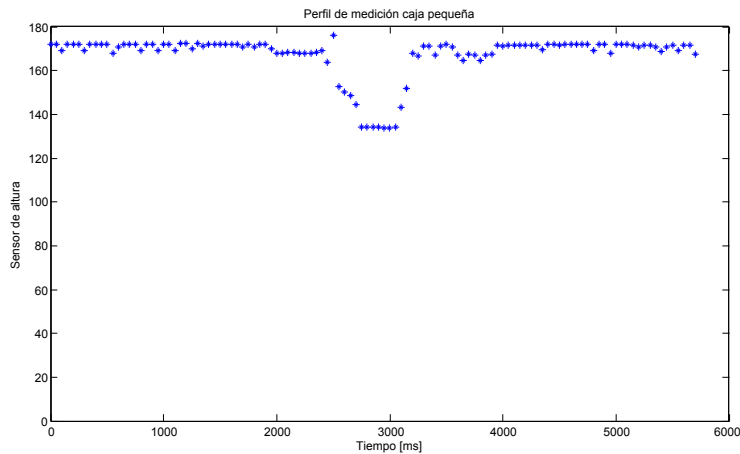


Figura 5: Perfil de medición caja pequeña.

Los valores anteriores corresponden a la medición directa del sensor de ultrasonido utilizado, se recomienda usar la interfaz de programación descrita que entrega al altura del objeto en milímetros de forma directa.

4.3. Escritura del algoritmo

Escriba su código usando Arduino IDE, basandose en el ejemplo disponible en Archivo > Proyecto > libraries > BeltSorter > SimplePusher. Recuerde llamar al método `BeltSorter::loop()` para actualizar el valor de las variables.

```

1  #include <BeltSorter.h>
2  #include <Servo.h>
3
4  BeltSorter bs;
5
6  void setup()
7  {
8      bs.begin();
9      Serial.begin(9600);
10 }
11
12
13 void loop()
14 {
15     // Open pushers
16     bs.openPusherA();
17     bs.openPusherB();
18     Serial.println("Opening pushers");
19
20     while(true)
21     {
22         // Update values
23         bs.loop();

```

```

24 // Get object height and compare
25 if(bs.getHeight() > 10.0)
26 {
27     // Close pusher and wait
28     bs.closePusherA();
29     // Pusher A at 35 cm
30     delay(35.0/bs.getBeltSpeed()*1000.0);
31     // Open Pusher A
32     bs.openPusherA();
33 }
34 bs.sleep();
35 }
36 }

```

Código fuente 3: Ejemplo SimplePusher.ino.

Presione el botón *Verificar* para chequear errores de sintaxis. Si ha terminado su algoritmo llame a un profesor auxiliar para probar su código en la cinta transportadora.

5. Actividad Propuesta (Bonus)

En esta sección se propone implementar la integración del servo y el sensor de distancia. Específicamente, mover el servo según el movimiento de algún objeto detectado por el sensor ultrasónico.

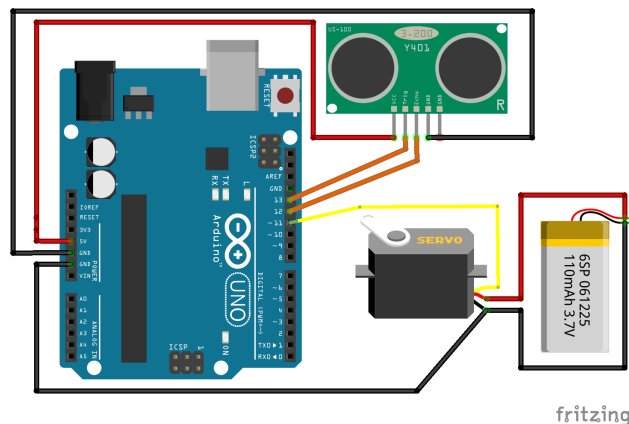


Figura 6: Esquemático de conexión servomotor y sensor de ultrasonido Arduino.

Siga los siguientes pasos para realizar la experiencia.

- Paso 1** Con ayuda de la figura 3 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 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*. **Abra la ventana de comunicación serial en el menú superior *Herramientas* > *Serial Monitor***

Paso 3 Encienda la fuente externa. Acerque distintos objetos al sensor de ultrasonido en el rango de distancia definido en el código para mover el servomotor. Recuerde usar el limitador de corriente de la fuente de poder como medida de seguridad.



Atención

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