Laboratorio de Sensores y Actuadores

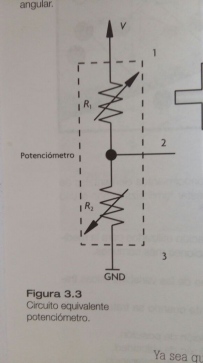
Practica 2

Potenciómetro

Objetivo

Marco Teórico

Un Potenciometro es un transductor entre la posicion de un objeto, ya sea lineal o angular, y un cambio de resistencia. Este tipo de elementos resistivos se utilizan normalmente con un voltaje de CD. Constan de 3 terminales, una en cada extremo del material, y una tercera terminal que recorre el cuerpo del elemento resistivo, de tal manera que la resistencia entre la terminal movil y cada una de las terminales fijas varia cuando el elemento movil cambia de posicion; asi, cuando una resistencia disminuye, la otra nescersariamente aumenta.

Para lograr una correlación entre voltaje de salida y la posición de un determinado objeto, suele ser común acoplar mecánicamente el elemento móvil de potenciómetro al objeto; así, cuando el elemento móvil del potenciómetro s encuentre en uno de los extremos, el voltaje a la salida será máximo, de otra manera cuando el elemento móvil este en la parte proximal, el voltaje a la salida será mínimo. Esta correlación se puede calcular usando la siguiente formula:

Donde la resistencia estará dada entre el nodo de referencia a tierra y el nodo de, mientras que la resistencia estará dada entre el nodo de y el nodo de . Al momento de utilizar este tipo de instrumentación para determinar la posición de un objeto, se debe tener en cuenta cuales son las características del potenciómetro, ya que para estas aplicaciones lo más recomendable es utilizar un potenciómetro cuyo cambio a la salida sea lineal. Cabe hacer la distinción aquí que cuando se habla de linealidad en el comportamiento del sensor se hace referencia a que la entrada sea linealmente proporcional a la salida y no al tipo de movimiento que presenta el elemento móvil. Que tan lineal será el comportamiento de cada potenciómetro dependerá, en mayor medida, del tipo de material con el que esté constituido.

Tipos de potenciómetros

A continuación se muestran en la figura 1 y 2 algunos de los diferentes tipos de potenciómetros.



Figure 1 Potenciometro lineal dual



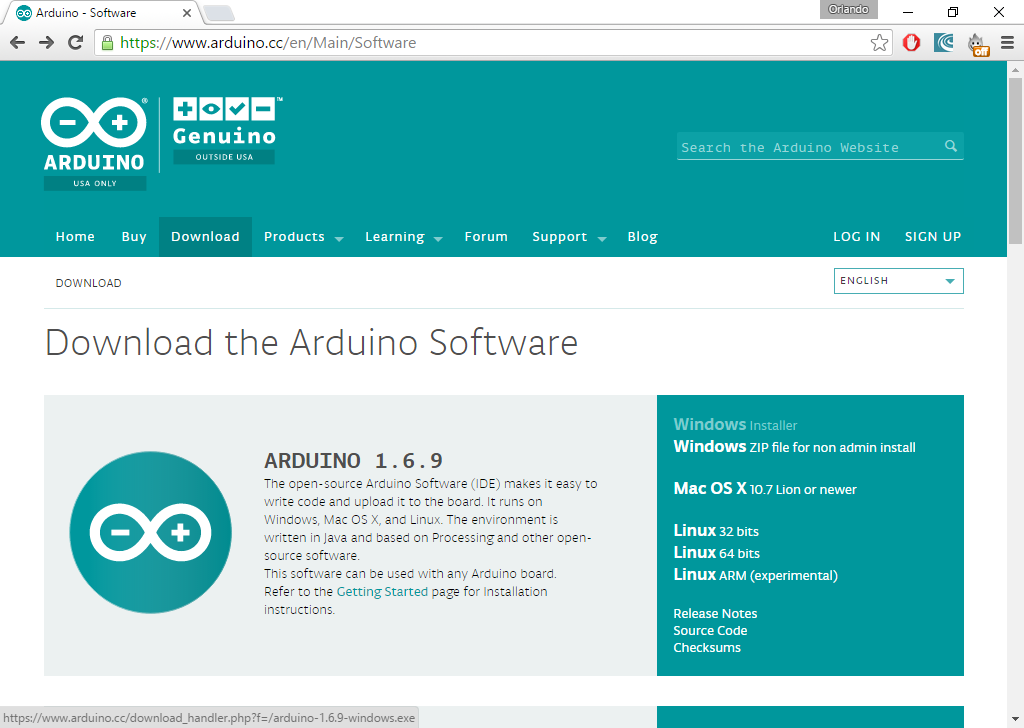
Figure 2 Potenciometro rotatorio dual

Desarrollo

Arduino

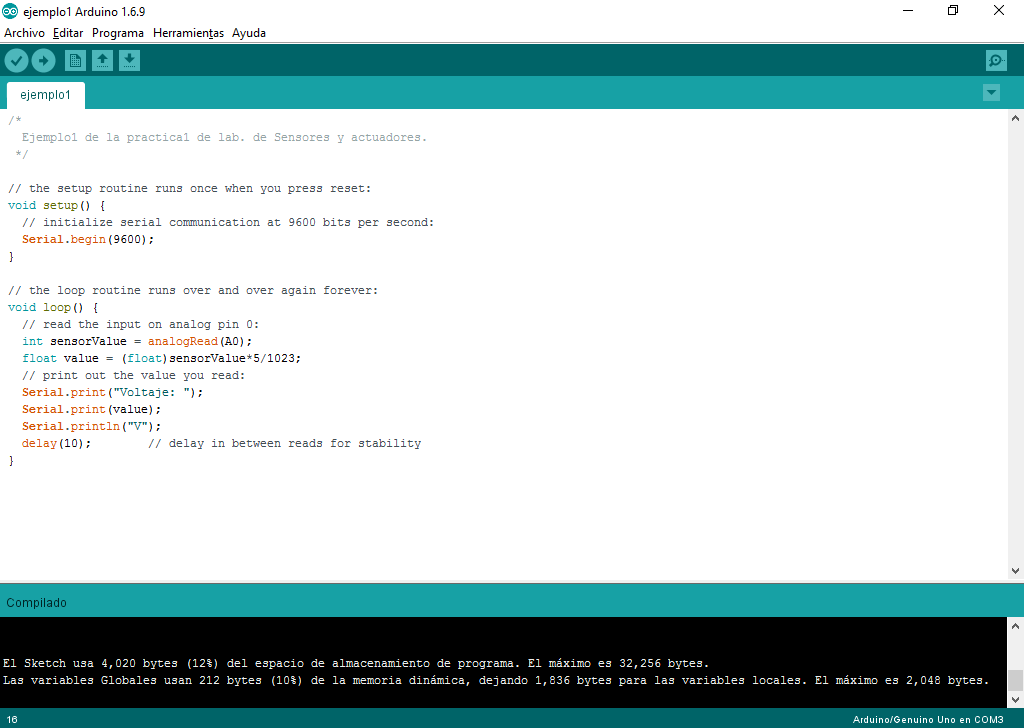
Para esta y las demás practicas usaremos un Arduino Uno por ser el mas comercial, sin embargo todas las practicas pueden ser realizadas con cualquier otra versión de arduino.

Para descargar el IDE de Arduino nos dirigimos a su pagina oficial y nos vamos a Download. Una vez ahí seleccionamos Windows Installer para instalar la versión mas nueva(1.6.9 cuando se realizo esta practica).



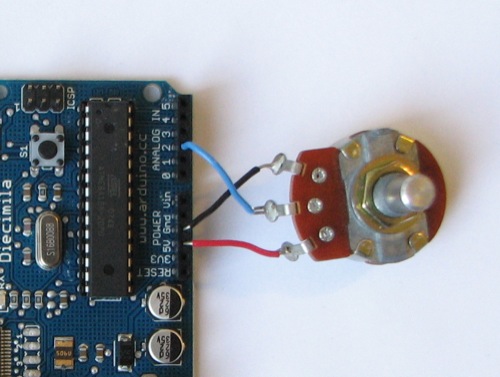
Ejemplo 1.

Una vez instalado arduino abrimos el ejemplo 1(Se encuentra en la carpeta arduino > ejemplo 1 > ejemplo1.ino).



En este ejemplo vamos a leer los valores arrojados por el potenciómetro en el arduino y los imprimiremos en pantalla para ver la variación en tiempo real de la diferencia de potencial generada por el potenciómetro.

Conectamos las 3 terminales del potenciómetro directamente al arduino. La primera va a tierra(GND), la segunda terminal va conectada a cualquier entrada analógica del arduino para este caso utilizaremos el pin 2 de entradas analógicas(En código se define como A2), y la tercer terminal del potenciómetro va 5V.



El código del ejemplo es el siguiente:



Arduino no es un lenguaje, si no que usa lenguaje C con librerías especificas para la plataforma arduino.

Los programas en arduino se llaman sketch y al momento compilarlos la computadora carga ese sketch al arduino, por lo que si cargas un sketch en el arduino y lo conectas con un adaptador directamente a la corriente eléctrica o con cualquier otra fuenta de CD de entre 5V y 9V seguirá funcionando.

Un sketch en arduino debe tener estas dos funciones para poder compilar correctamente:

-void setup: Se ejecuta solo una vez cada que se enciende o reinica el arduino. En esta función se deben de configurar e inicilizar todas las variables que vayamos a usar



-void loop: Es el análogo al método main de un programa en C, es un ciclo(loop) infinito que se ejecuta una vez que termine de ejecutarse la función setup, en esta función escribiremos toda la lógica del programa.



Primero definimos una variable de tipo int para guardar los valores leidos por el sensor.

En esta línea de código definimos una variable de nombre sensorValue de tipo int(numero entero) y le asignamos lo que el arduino este leyendo en la entrada analógica 2 que es donde esta conectada la segunda terminal del potenciómetro utilizando la función analogRead, básicamente lo que le estampos diciendo al arduino es *“Crea una variable de tipo entero que se llame sensorValue, lee el pin analogico 2 del arduino y guarda este valor en la variable sensorValue”.*

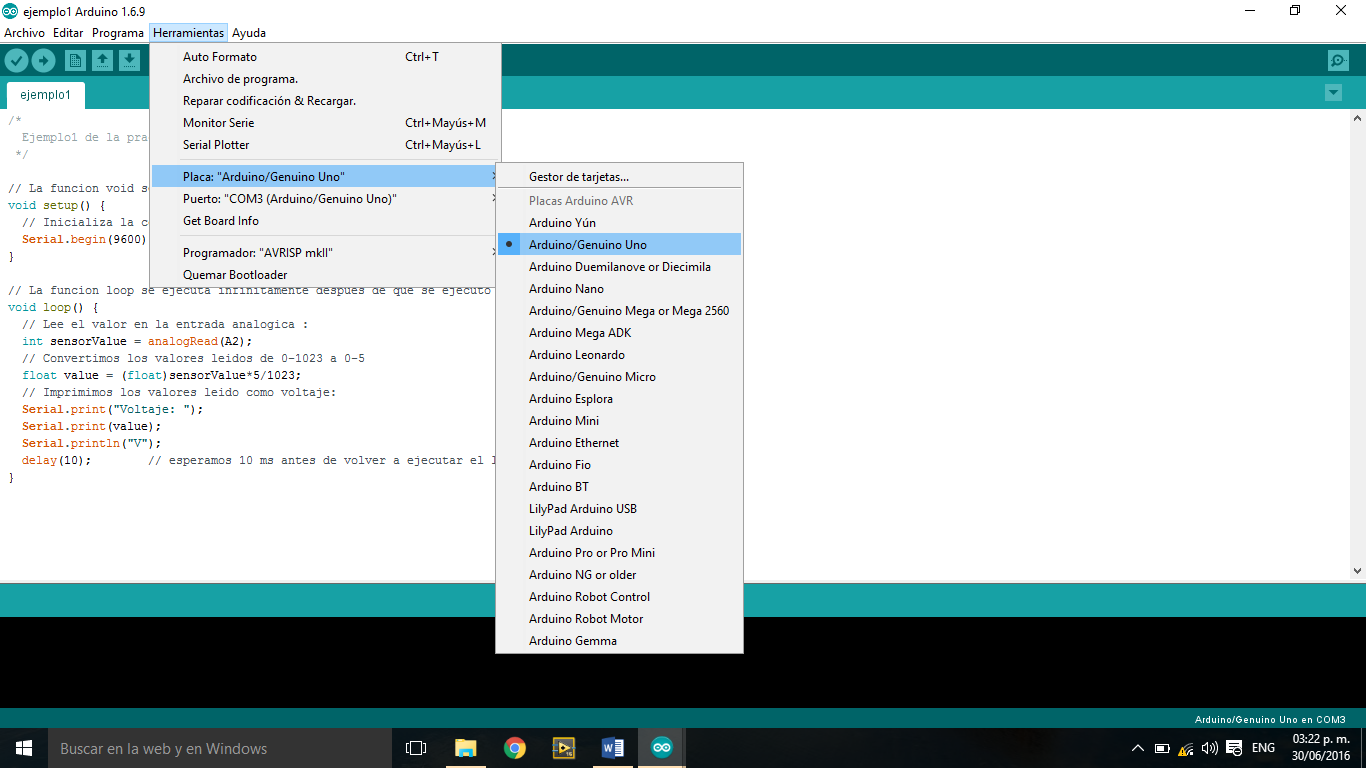
Los valores leidos varian de 0 a 1023, para una mejor lectura se convierten a valores entre 0 y 5, para esto el valor leído por el sensor se divide entre 1023 y se multiplica por 5 y se guarda en una variable de tipo float(de punto flotante o decimal) a la que llamamos value, como los valores que arroja el pot son de tipo int se usa un cast que es una conversión de un tipo a otro “(float)sensorValue\*5/1023”.

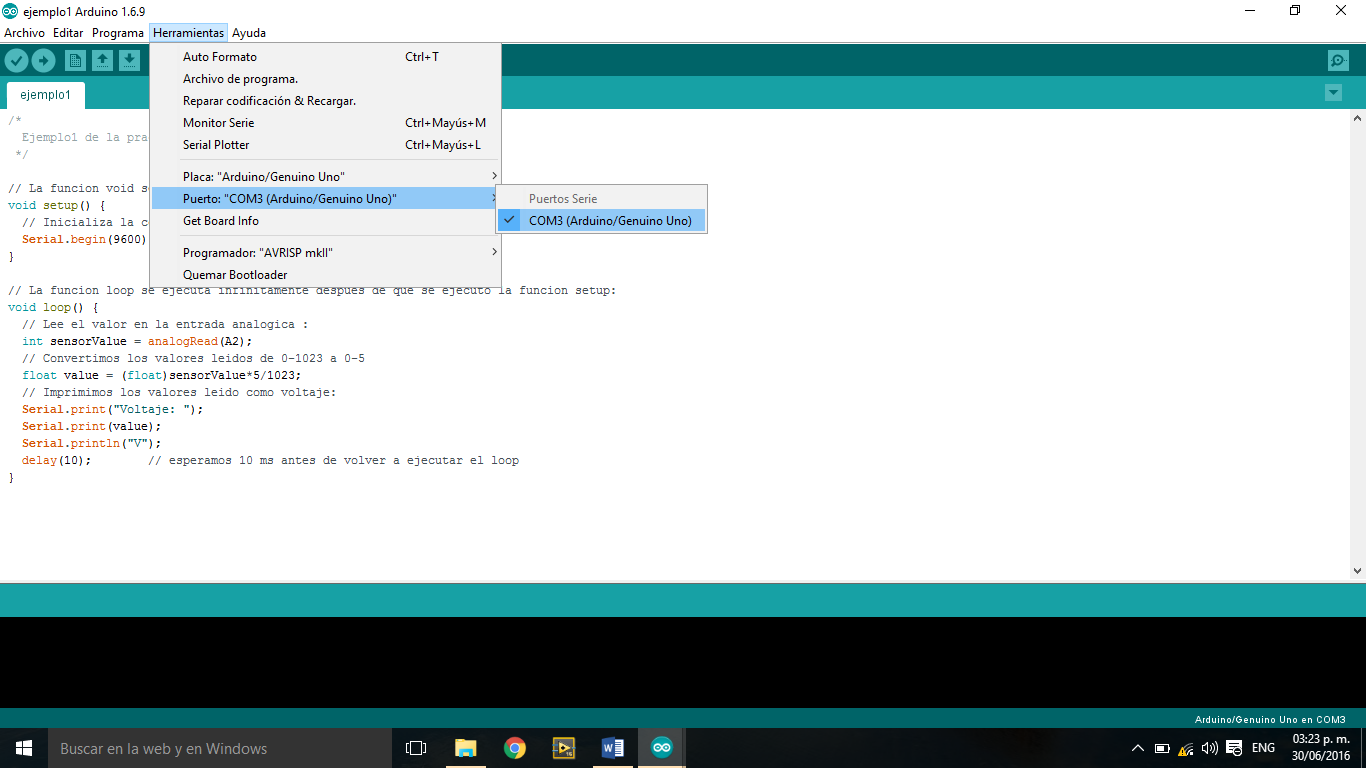


Ahora todo lo que falta es imprimir los valores leidos por el potenciómetro, el comando Serial.print se encarga de imprimir en el monitor serial lo que le pasemos como parámetro, el comando Serial.println hace lo mismo pero termina con un salto de línea.

Al final se encuentra la línea de código “delay(10);”, el comando delay hace una pausa durante el tiempo en milisegundos que le pasemos como parámetro para este caso 10, se recomienda siempre dejar un pequeño delay al final del código para no saturar al arduino.

Para cargar el sketch al arduino primero hay que asegurarnos de que esta seleccionado el arduino que estamos usando en este caso el UNO y el puerto correspondiente.





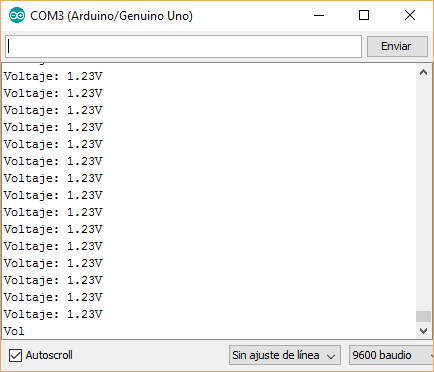
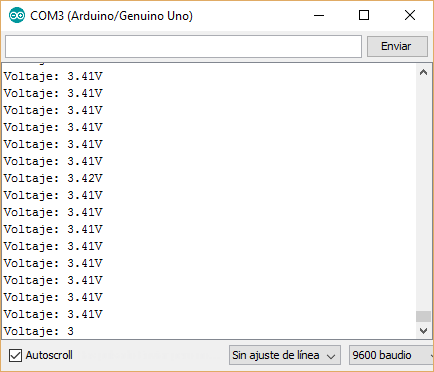
Despues de seleccionar el arduino y el puerto, nos aseguramos que el código no tenga errores con el botón de verificar y procedemos a cargar el sketch al arduino con el botón de subir.

Una vez cargado el código abrimos el monitor serie para poder ver lo que el arduino esta leyendo del potenciómetro.



Como podemos ver en el monitor serie se esta imprimiendo los valores leidos por el potenciómetro y lo representa como voltaje.

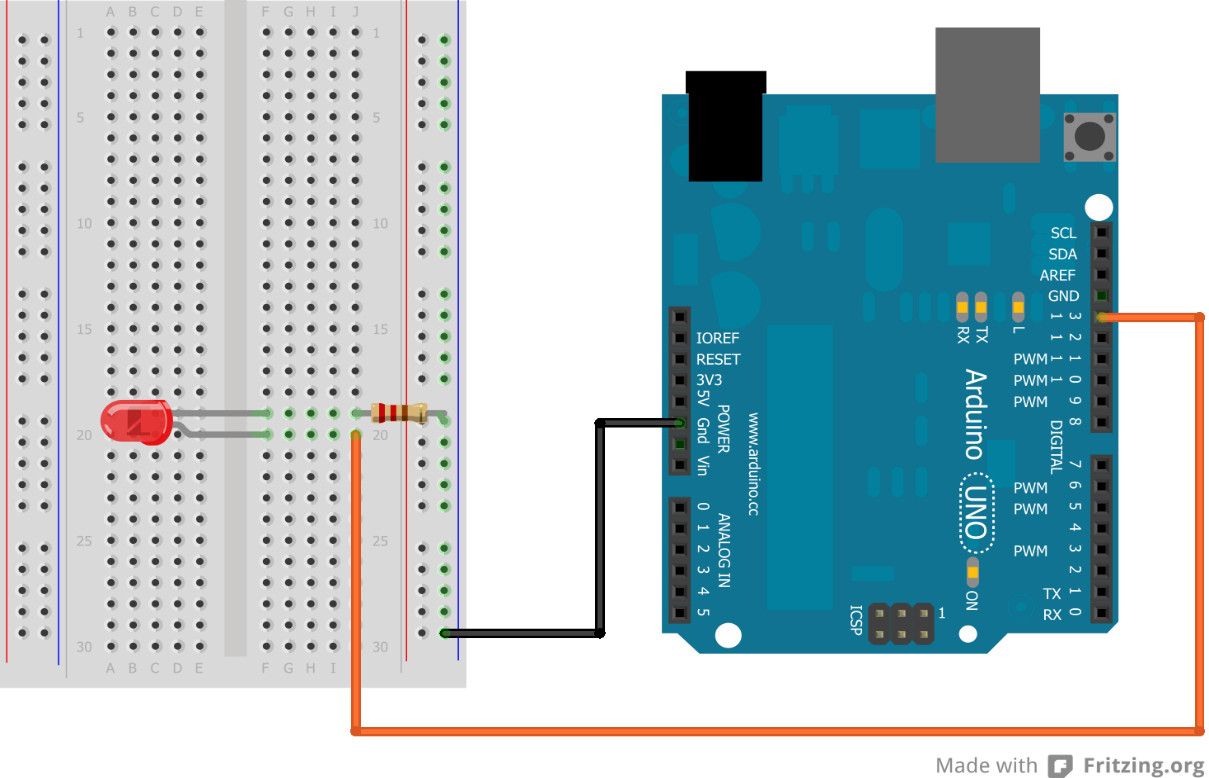
Ejemplo 2.

Abrimos el ejemplo 2, se encuentra en la carpeta de arduino de la practica 2.



En este ejemplo usamos dos leds uno rojo y uno verde y tomamos decisiones a partir de los datos leidos por el potenciómetro. El pin analogico de lectura tiene una resolución de 10 bits por lo que sus posibles valores varian de 0 a 1023, si el valor del pot es menor o igual a 512 encendemos el led rojo y si su valor es mayor a 512 encendemos el led verde.

La conexión de para un led en arduino es la siguiente.



El negativo del led se conecta a una resistencia de 220 ohms y la resistencia va a tierra y el positivo del led va al pin digital del arduino en este ejemplo el led rojo va al 13 y el verde al 12. El potenciómetro se queda conectado igual que en el ejemplo anterior.

Primero definimos los pines digitales a los que estarán conectados los leds como enteros.



Y los definimos como salida dentro de la función setup().

La función pinMode() define pines digitales ya sea como entrada o como salida, necesita dos parámetros, el primero es el pin al que se quiere definir y el segundo es la forma en la que trabajara el pin ya sea como entrada o salida(OUTPUT o INPUT).

Despues dentro de la función loop es donde tomamos decisiones a partir de los valores leidos



Primero obtenemos el valor leído por el pin A0 del arduino y lo guardamos en una variable de tipo int que llamamos pot.

Después con una sentencia if preguntamos si el valor leído es menor o igual a 512. Si la sentencia es verdadera entonces encendemos el led rojo y apagamos el verde en caso de que este encendido, para esto usamos la función digitalWrite() que pide dos parámetros, el primero es el pin al que quieres hacer referencia y el segundo es el nivel lógico al que quieres cambiar el pin(HIGH para encender el pin y LOW para apagar el pin). Los pines digitales trabajan con 5v asi que mientras estén encendidos tendrán una diferencia de potencial de 5v con respecto de la tierra del arduino. En caso de que el valor del pot sea mayor a 512 la sentencia pasa a ser falsa por lo que entra al bloque “else” ignorando las líneas de código del bloque if y usando otravez la función digitalWrite encendemos el led verde y apagamos el rojo.

LabView

