

PRACTICAS ARDUINO – EJERCICIOS PRACTICOS

ÍNDICE

Practica 1. Hello world	2
Practica 2. Led intermitente. (Salida digital)	2
Practica 3. Activación de un led, mediante un pulsador. (Entrada digital)	3
Practica 4. El coche fantástico. (Bucles de control for y while)	4
Practica 5. Entrada analógica.	5
Practica 6. Coche fantástico más potenciómetro.	6
Practica 7. Salida analógica. PWM	7
Practica 8. Controlando la luminosidad de un led a través de un potenciómetro.	8
Practica 9. Luz de vela	9
Practica 10. Sensor de luz.....	9
Practica 11. Control de una bombilla de 240V, utilizando un relé.....	11
Practica 12. Control motor paso a paso.....	12
Practica 13. Sensor de temperaturas	14
Practica 14. Control servomotor.	15
Práctica 15. Práctica semáforo.....	16
Practica 16. Control mediante infrarrojos.....	17
Practica 17. Display de 7 segmentos.....	18

Practica 1. Hello world

Enunciado:

Esta práctica consiste en visualizar por la pantalla del ordenador, a través del monitor serial el mensaje "hello world".

Una vez compilado y cargado el programa, abrimos para mostrar el mensaje sobre el monitor serial. Si nos fijamos, hay unos baudios establecidos en el monitor serial, los cuales coinciden con los del código (Serial.begin).

Teoría necesaria:

Seral.begin() teoría página 31

Serial.println().....teoría página 33

Serial.print() teoría página 32

delay ().....teoría página 37

Practica 2. Led intermitente. (Salida digital)

Enunciado:

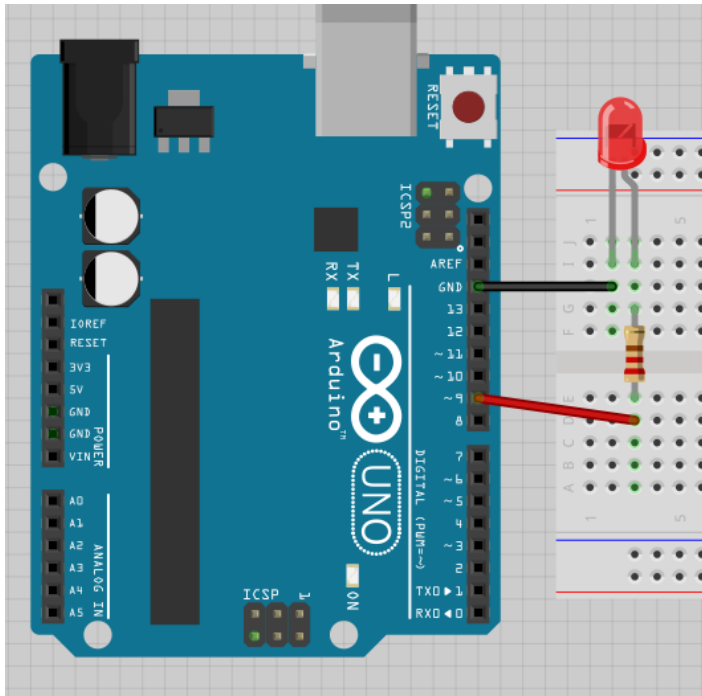
En esta práctica se trata de realizar un ejercicio básico que consiste en encender y apagar un led, de manera intermitente (intervalo de un segundo), utilizando para ello uno de los Pines digitales (PIN 9), configurado como salida.

Material necesario:

- Led
- Resistencia 220 ohm

Teoría necesaria:

- Declaración de constantes..... teoría página 30
 - Asignar pines digitales (como entradas o salidas) **pinMode()** teoría página 34
 - Uso de las salidas digitales. **digitalWrite()**.....teoría página 35
 - Uso de retardos. **delay()**.....teoría página 37
- delay(1 000); = retardo de 1 segundo



Practica 3. Activación de un led, mediante un pulsador. (Entrada digital)

Enunciado:

Se pretende controlar el encendido de un led, mediante un pulsador.

Material necesario:

- Led
- pulsador
- resistencias de 10k (pulsador) y 220 ohm (led).

Teoría necesaria:

- Teoría practicas anteriores.
- Declaración de variables..... teoría página 24
- Uso de entradas digitales. **digitalRead()**.....teoría página 35
- Bloque condicional (estructura de control) **if...else**..... teoría página 42

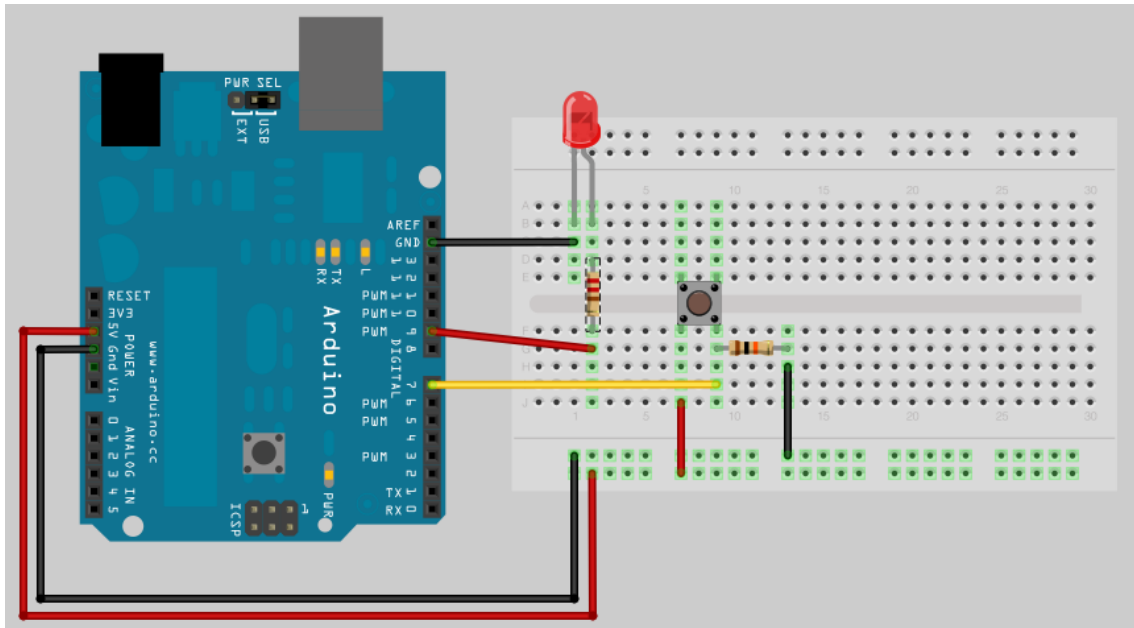
Aclaraciones:

La variable que habrá que declarar en dicha práctica será en la cual almacenaremos el valor actual del pulsador, la cual puede variar entre el valor 1 (5V) y 0 (0V).

Si nos fijamos en el montaje del pulsador, utiliza una resistencia de **pull-down**, para entender bien su funcionamiento, basta con desconectarla y ver que le sucede al led, el cual cambia de

luminosidad y parpadea, debido al ruido producido cuando la entrada digital esta activada para su lectura y esta no está conectada a ninguna parte.

Montaje:



Practica 4. El coche fantástico. (Bucles de control for y while)

Parte 1.

Enunciado:

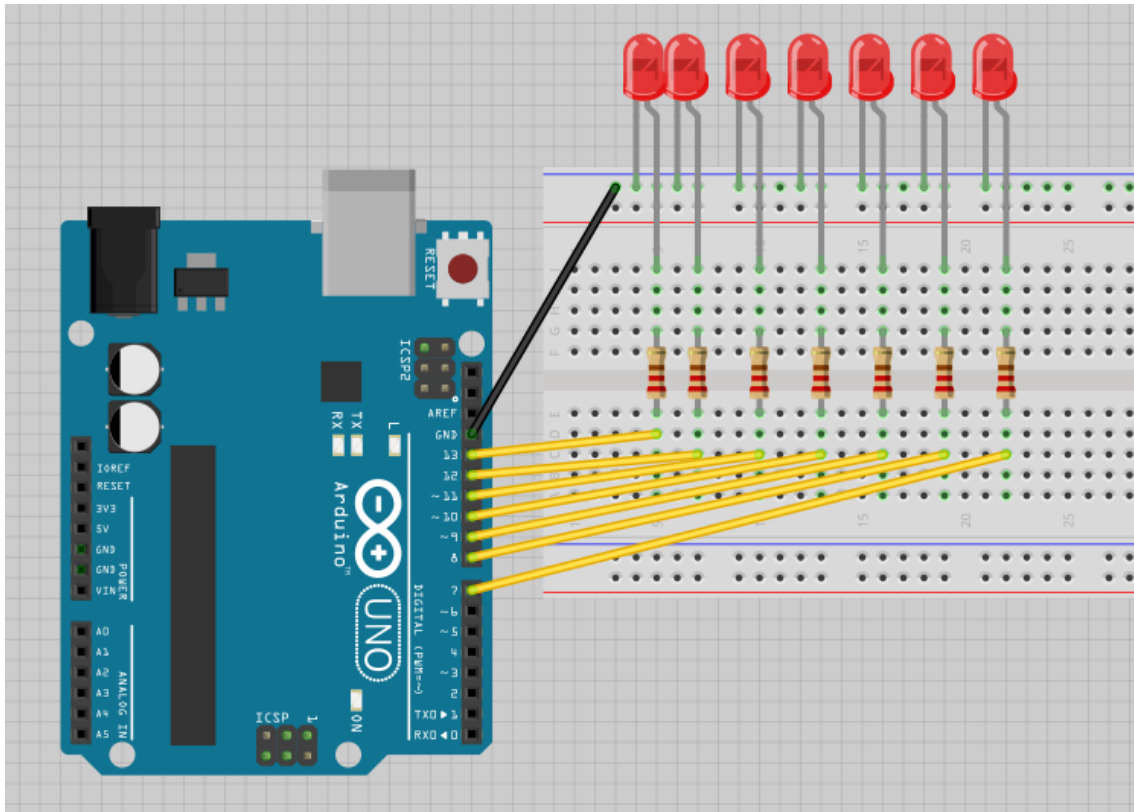
En la siguiente práctica se pretende construir la luz frontal del Coche Fantástico.

Esto se conseguirá activando y desactivando 7 leds de manera consecutiva en orden ascendente y después descendente.

Material necesario:

- 7 Leds
- 7 resistencia de 220 ohms

Montaje:



Una vez montado el circuito, abrimos y cargamos el programa coche fantástico 1.

Como podemos ver el código es bastante largo y repetitivo, para evitar esto se podrían utilizar o el bucle condicional **for** o el bucle **while**, además también podríamos simplificarlo, declarando las entradas (todas del mismo tipo), como un **array**.

Parte 2.

Enunciado:

Partiendo del programa anterior cargado, se pretende simplificar este, haciendo uso de los bucles de control for o while:

Teoría nueva necesaria:

- Uso de **arrays** teoría página 28
- Uso de **operadores compuestos**..... teoría página 46
- **Operadores de comparación**..... teoría página 43
- Bucles condicional **while**..... teoría página 45
- Bucle condicional **for**..... teoría página 46

Practica 5. Entrada analógica.

Enunciado:

En la siguiente práctica, se pretende visualizar a través del monitor serial, el valor leído por una entrada analógica. **analogRead()**, a través de un potenciómetro, el cual variara el valor de la entrada analógica entre 0 y 5V.

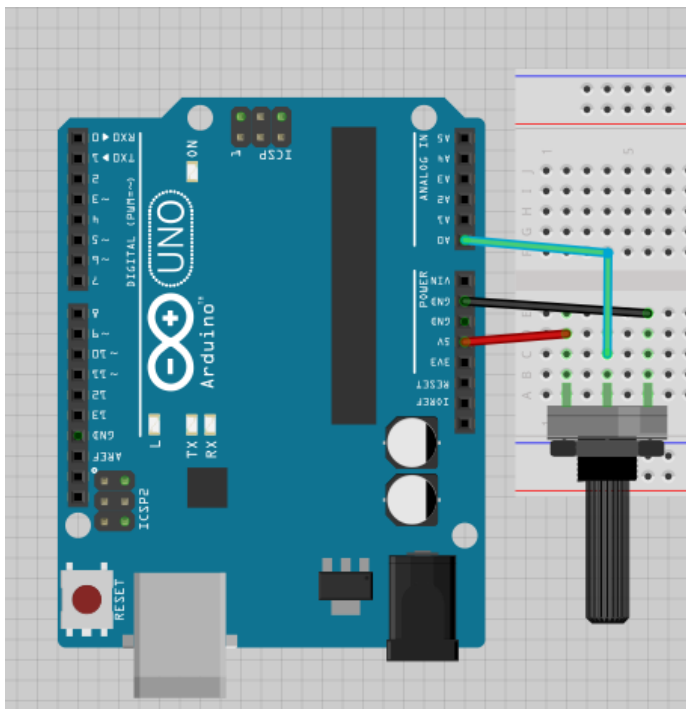
Material necesario:

- Potenciómetro

Teoría nueva necesaria:

- Teoría práctica 1.
- **analogRead()**.....teoría página 36

Montaje:



Practica 6. Coche fantástico más potenciómetro.

Enunciado:

En esta práctica, se pretende controlar el tiempo de delay de encendido entre leds, de la práctica del coche fantástico, a través de la lectura del potenciómetro tomada por la entrada analógica. Con esto podremos variar la velocidad de progresión de los leds de 0 a 1023 milisegundos.

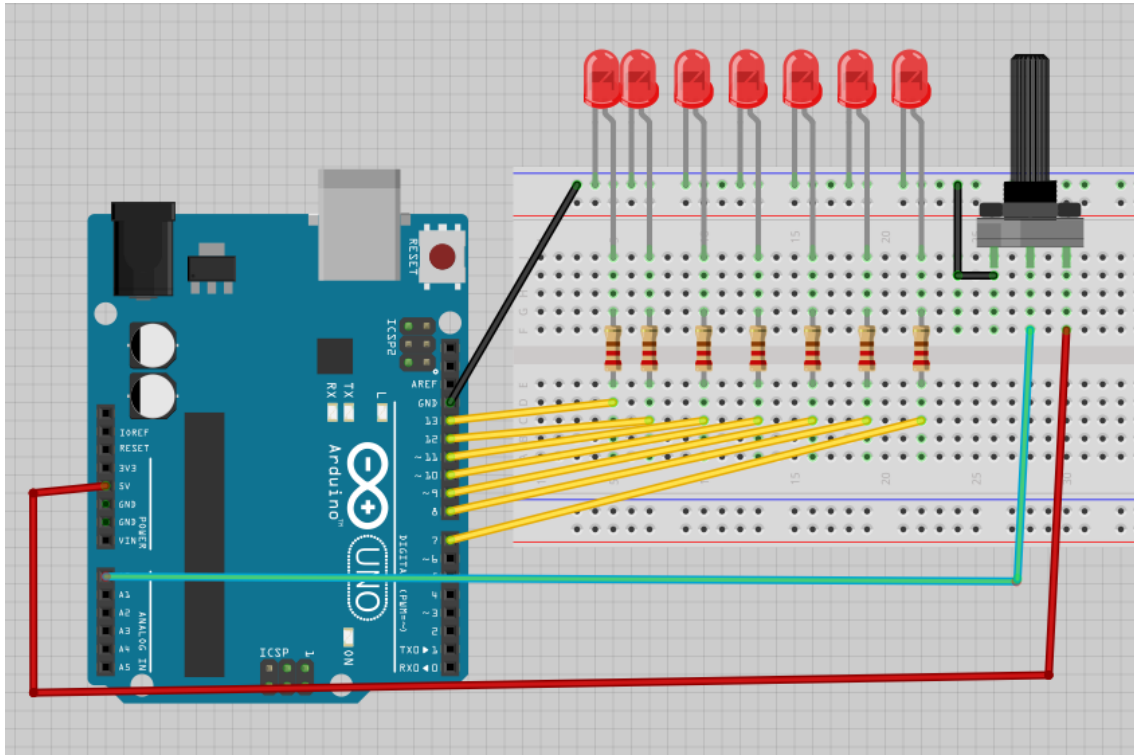
Material utilizado:

- 7 Leds
- 7 resistencia de 220 ohms

Teoría necesaria:

- Teoría practica 4
- Teoría practica 5

Montaje:



Practica 7. Salida analógica. PWM

Enunciado:

En la siguiente práctica, se pretende controlar la luminosidad de un led, de tal forma que aumente y disminuya su luminosidad de manera gradual. Para dicho fin, se utilizara la salida analógica (PWM) y será también necesaria el uso de un bloque de control de tal manera que la salida analógica recorra todos sus valores (0-255, 255-0).

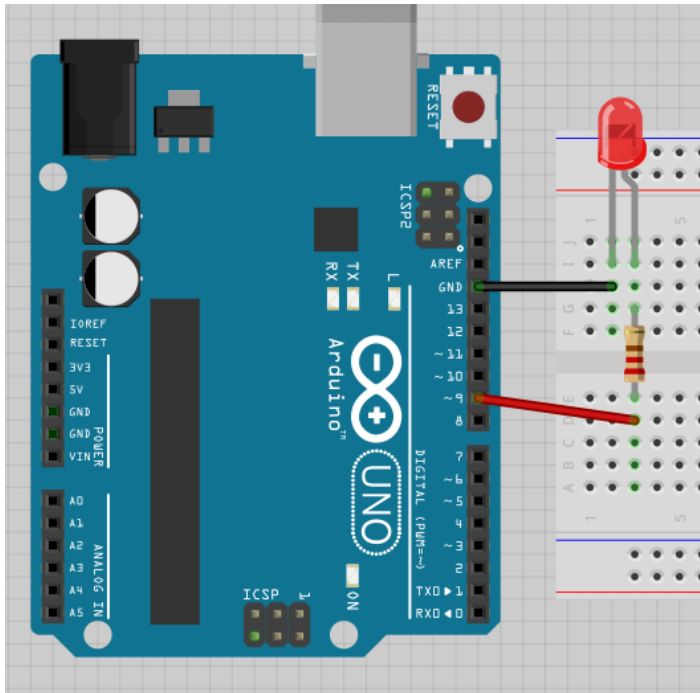
Material necesario:

- Led
- Resistencia de 220 ohms

Teoría necesaria:

- `analogwrite()`.....teoría página 36
- Bloque de control `for`..... teoría página 46

Montaje:



Practica 8. Controlando la luminosidad de un led a través de un potenciómetro.

Enunciado:

En esta práctica se pretende variar la luminosidad de un led (salida analógica), en función de la variación de un potenciómetro conectado a una de las entradas analógicas **analogRead()**, de tal manera que cuando el potenciómetro este al mínimo el led no lucirá, y cuando esté al máximo el led lucirá como si este estuviera conectado directamente a 5V.

Como podemos ver en la teoría de **analogRead()**, vemos que la lectura de la entrada analógica es de 10bits (0-1023) y que la salida analógica es de 8 bits (0-255), por eso para que el programa funcione correctamente, debemos escalar los valores de entrada, con respecto los de salida. Esto se conseguirá con la función **map()**.

Material a utilizar:

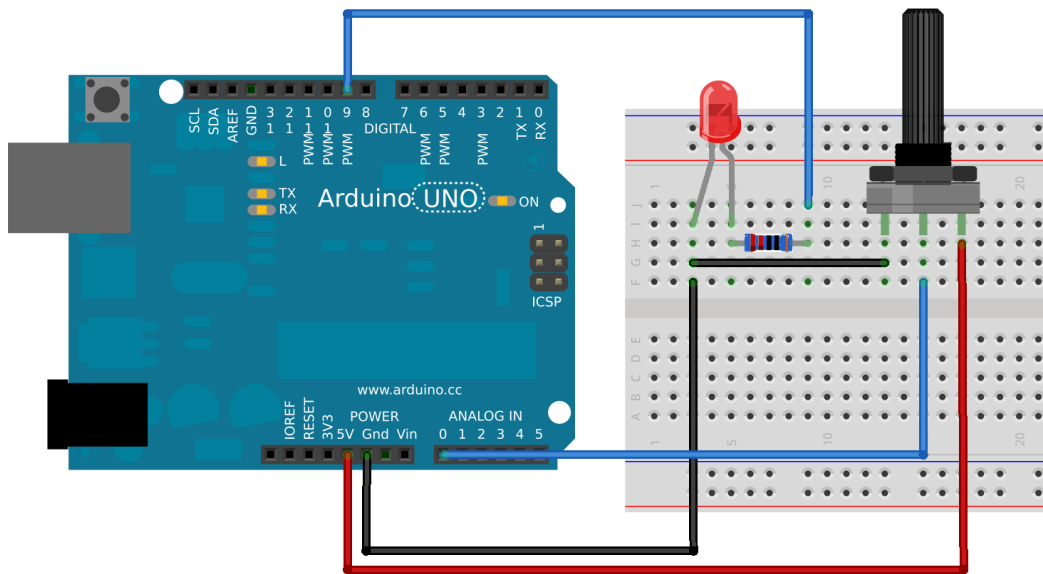
- Led
- Resistencia 220 ohms
- potenciómetro

Teoría necesaria:

- **#define** (declaración de constantes, otra forma)..... teoría página 30
- **analogWrite()**.....teoría página 36
- **analogRead()**.....teoría página 36

- `map()`.....teoría página 38

Montaje:



Practica 9. Luz de vela

Enunciado:

En la siguiente práctica se pretende simular la luz de una vela, utilizando para ello las funciones **random()** y **randomSeed()**, funciones de pseudoaleatoriedad. Con dichas funciones se conseguirá, que tanto la intensidad como el tiempo el cual iluminara con dicha intensidad, varíe de forma aleatoria, produciendo un efecto visual parecido al de una llama.

Material utilizado:

- Led
- Resistencia 220 ohms

Teoría necesaria nueva:

- `random()`.....teoría página 40
- `randomSeed()`.....teoría página 40

Montaje:

Mismo montaje que la practica 2

Practica 10. Sensor de luz

Enunciado:

En este proyecto, vamos a utilizar un LDR (Light Dependent Resistor o resistencia dependiente de la luz) para simular una hipotética compensación lumínica de 5 niveles, es decir, a través de una resistencia que varía su valor dependiendo de la luz recibida, aprovecharemos dicha

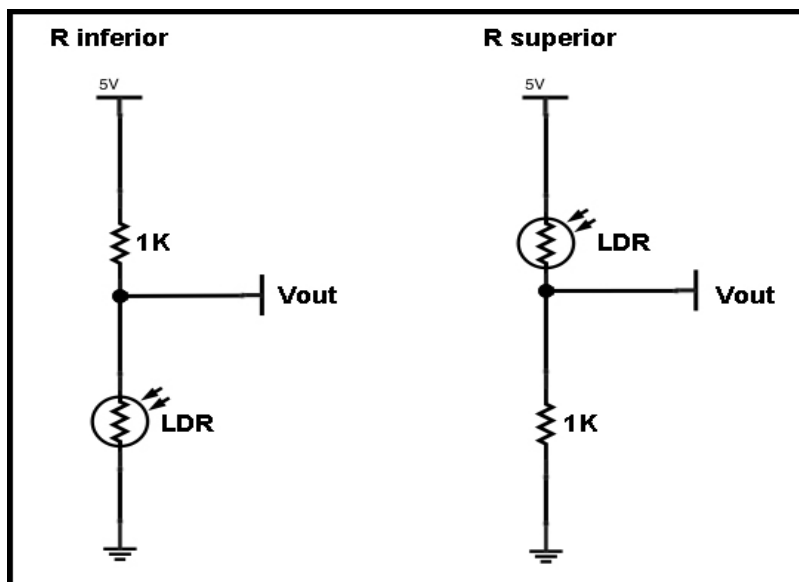
variación para hacer un programa que nos encienda o apague una serie de LED dependiendo de si hay más luz o menos luz, esto podríamos extrapolarlo a un sistema de regulación de luz de un jardín con cinco líneas de luz que se vayan encendiendo según va cayendo la noche compensando progresivamente la deficiencia de luz.

Además le hemos implementado un potenciómetro para ajustar el nivel crítico mínimo de luz que queremos soportar, a partir del cual se activará nuestro circuito y empezará a aumentar la luz del lugar progresivamente.

Que es una LDR

Es una resistencia que varía con la intensidad lumínica ejercida sobre ella, de tal forma que cuando más luz haya, menos será su resistencia interna y viceversa.

Como se conecta una LDR:



Si utilizamos el **LDR** como resistencia inferior del divisor de tensión, nos dará la tensión máxima cuando tengamos el LDR en plena oscuridad, ya que estará oponiendo el máximo de su resistencia al paso de la corriente derivándose esta por Vout al completo, si lo utilizamos como resistencia superior, el resultado será el inverso, tendremos la tensión máxima cuando esté completamente iluminado, ya que se comportará prácticamente como un cortocircuito, con una resistencia de 50Ω o 100Ω.

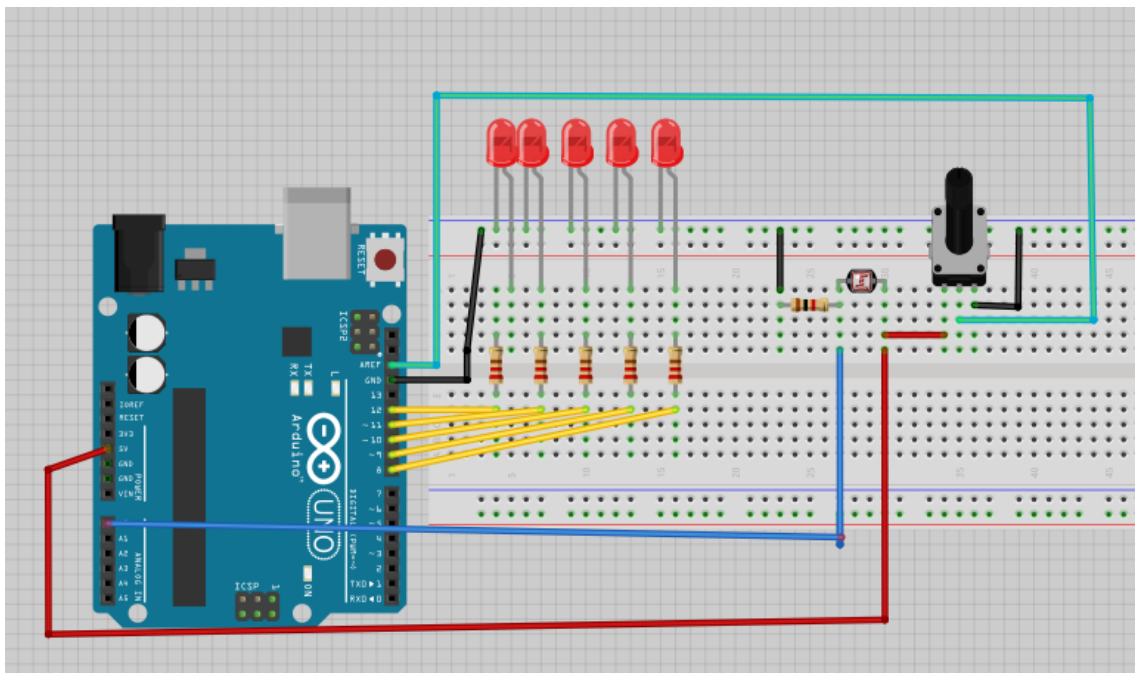
Para este caso utilizaremos el de R superior.

Para ajustar la sensibilidad lo mejor posible, utilizaremos la función **analogReference()**, con lo que podremos conseguir mayor sensibilidad, incluso con pequeñas variaciones de luz.

Para ayudarnos a la hora de programar los rangos a los cuales se activaran o desactivaran leds, podemos visualizar el dato leído por la entrada analógica por el monitor serial, además de ver el efecto que produce la variación de la tensión de referencia.

Conocimientos teóricos necesarios:

- teoría vista en prácticas anteriores (map(), analogWrite(), analogread(),...
- función **analogReference()**.....teoría página 36
- bloque condicional **switch case**..... teoría página 44



Practica 11. Control de una bombilla de 240V, utilizando un relé.

Enunciado:

En esta práctica se va a controlar el encendido de una bombilla de 240, utilizando un relé.

Esta práctica consiste en encender y apagar una bombilla de forma parecida a la que se hizo con un led, mediante un interruptor, sensor de luz, sensor de presencia , etc..

Para el caso del interruptor, la bombilla lucirá al pulsar el interruptor, luciendo hasta que vuelva a ser pulsado el interruptor. Para la variable del estado de la bombilla, podríamos utilizar la variable de tipo boolean, ya que solo tendremos dos estados: FALSE (bombilla apagada), TRUE (bombilla encendida).

La única diferencia con respecto a lo hecho en prácticas anteriores, es que hay que controlar una señal de corriente alterna de 240V. Para ello utilizaremos un relé, el cual al excitarse su bobina, cierra los contactos. El problema aquí es excitar la bobina y una vez conseguido esto, mitigar las corrientes transitorias generadas debido a la inductancia de dicha bobina.

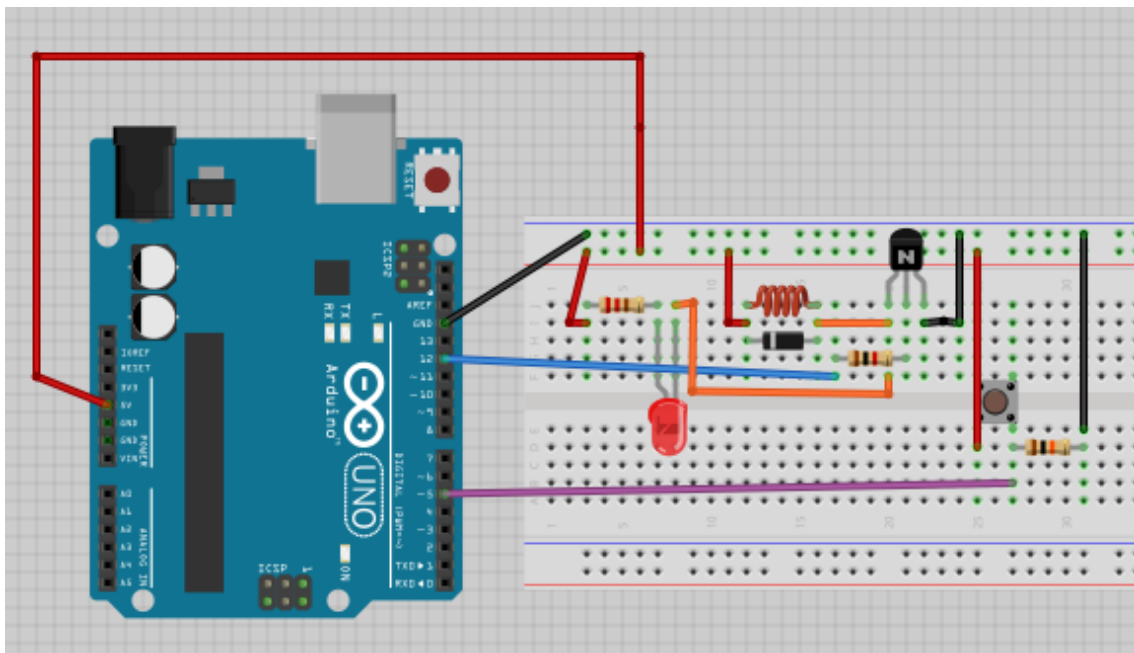
Tenemos que pensar que esta bobina necesita cierta cantidad de corriente para su activación, para el caso del relé utilizado para esta práctica, necesitamos proporcionar 200mA a dicha bobina, por lo que al ser superior a los 40mA que puede suministrar la salida digital, necesitaremos hacer uso de un driver, en este caso un transistor de tipo BJT que soporte una corriente de colector (I_c) mayor de 200mA.

Material necesario:

- Relé
- Transistor BJT 2N2222
- Resistencia de 220 ohmios
- Resistencia de 1k
- Led (para indicar cuando el relé está activado)
- Diodo rectificador 1N4041 (para suprimir las corrientes parasitas)

Teoría necesaria:

- Teoría sobre drivers..... **teoría página 61**

Montaje:

Practica 12. Control motor paso a paso.

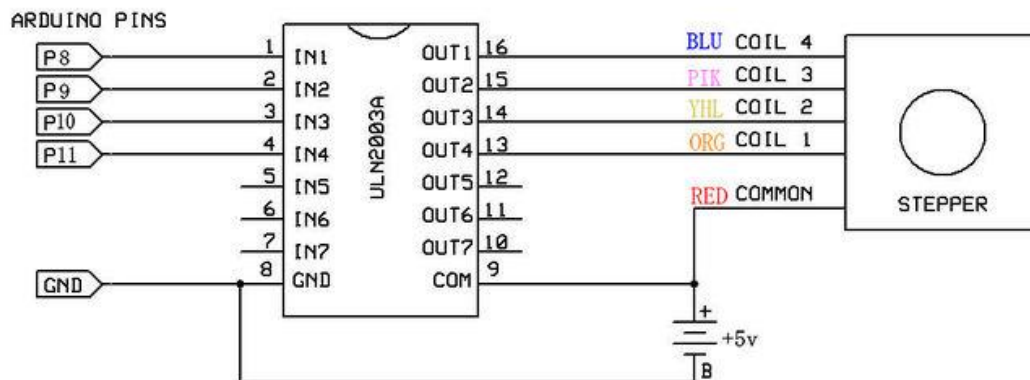
Enunciado:

En esta práctica se va a controlar un motor paso a paso, primeramente haciendo uso de una programación básica, basándonos en la secuencia de giro dada a continuación, para un motor paso a paso bipolar (motor paso a paso utilizado para dicha práctica).

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

Para controlar las bobinas de dicho motor, necesitaremos la utilización de un driver, ya que la potencia consumida por estas supera la máxima suministrada por las salidas digitales (40mA).

El conexionado seria el siguiente:



Dicho driver, junto con unos leds que indican la secuencia de activación de las bobinas y unos conectores, se encuentran ya integrados en la siguiente placa:



Más información sobre los motores paso a paso:

<http://www.iesleonardo.info/ele/pro/CURSO%202002-2003/Juan%20Carlos%20de%20Pedro%20Ramos/motores%20paso%20a%20paso.htm>

http://robots-argentina.com.ar/MotorPP_basico.htm

<http://www.ib.cnea.gov.ar/nmayer/monografias/pasoapaso.pdf>

Para la segunda parte de la práctica, se va a utilizar una librería que nos permite decirle al motor, cuantos pasos y la velocidad. (librería Stepper)

Además, el valor de los pasos tomara la diferencia entre la lectura inicial de potenciómetro y la siguiente.

Material necesario:

- Conjunto de driver ULN2003
- Motor paso a paso 28BYJ-48 (<http://robocraft.ru/files/datasheet/28BYJ-48.pdf>)
- Potenciómetro

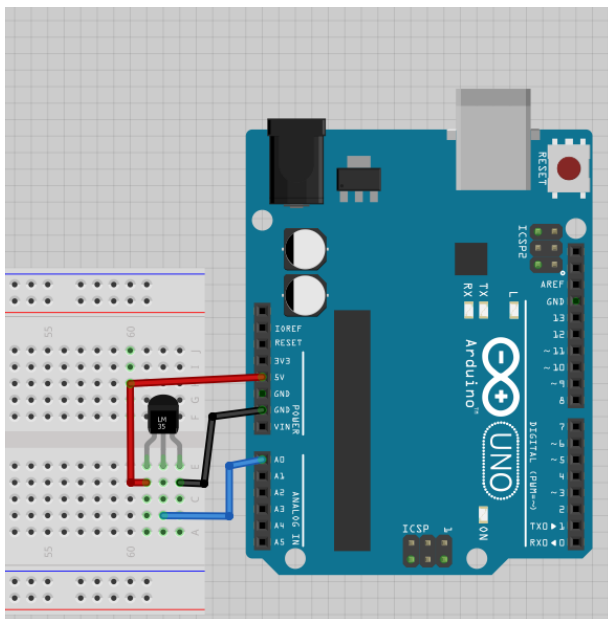
Practica 13. Sensor de temperaturas

Enunciado:

Un sensor de temperatura no es más que una resistencia que tiene la propiedad de variar su valor en función de la temperatura exterior. En la siguiente practica se va a utilizar el sensor de temperatura LM35, el cual produce una tensión es su patilla de salida de 10mV/°C.

Material a utilizar:

- Sensor de temperatura LM35



Practica 14. Control servomotor.

Primera parte:

Para controlar un servomotor, se utilizan las salidas PWM. El ancho del pulso de este, nos define el ángulo en el que se posicionara dicho servo. Para que su control sea más fácil y no haya que hacer conversiones, para saber que ancho dar en función del ángulo deseado, existe la librería **<servo>**, que realiza esto por nosotros.

Para utilizarla, deberemos incluirla al principio de nuestro sketch, en la zona global.

```
#include <Servo.h>
```

Debemos de tener cuidado de no enviarle ángulos fuera del rango: 0-180, ya que el servo SG90, no viene preparado para dar más ángulos.

En este primer programa con un servo, controlaremos dos posiciones fijas de este (0 y 170 grados), a través del monitor serial, mediante la función **Serial.Read()**.

Como vemos en la teoría, esta función solo lee caracteres (variables **tipo char**), por lo que asociaremos dichos caracteres a un valor deseado. Ej: a = 0 y b = 170.

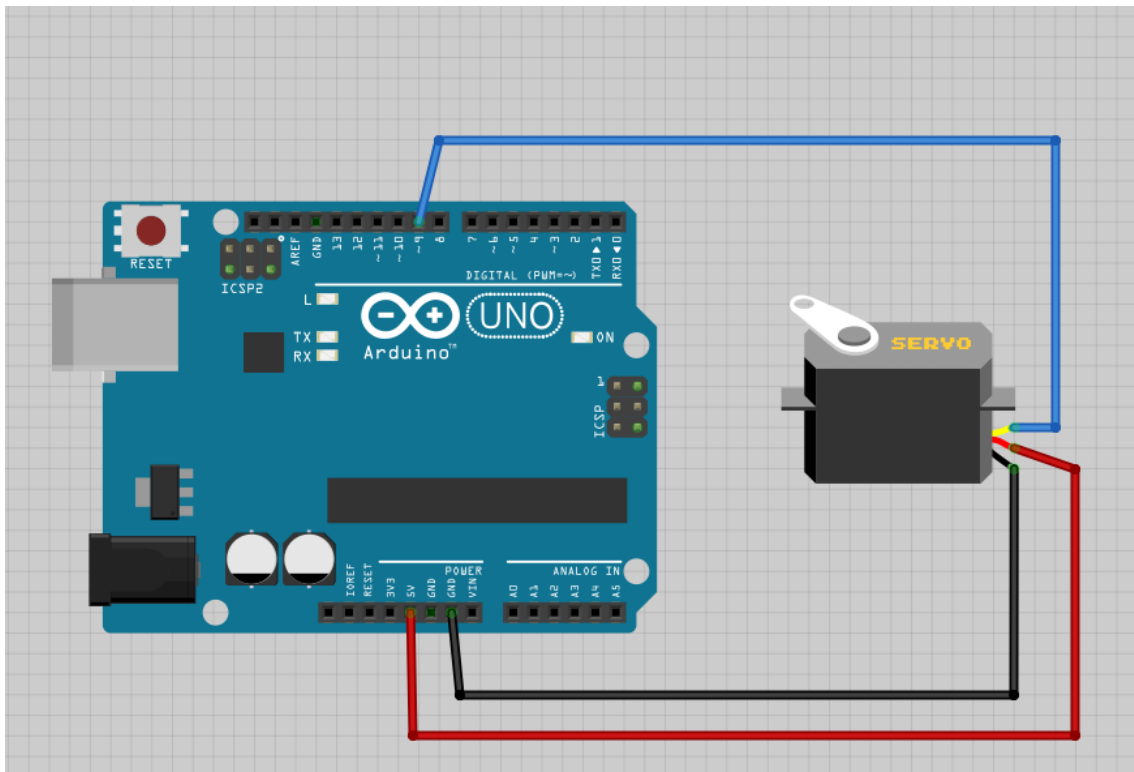
Teoría necesaria:

- **Serial.Read()**..... teoría página 33
- Variables tipo **char**..... teoría página 27

Material necesario:

- Mini servo.

Montaje:



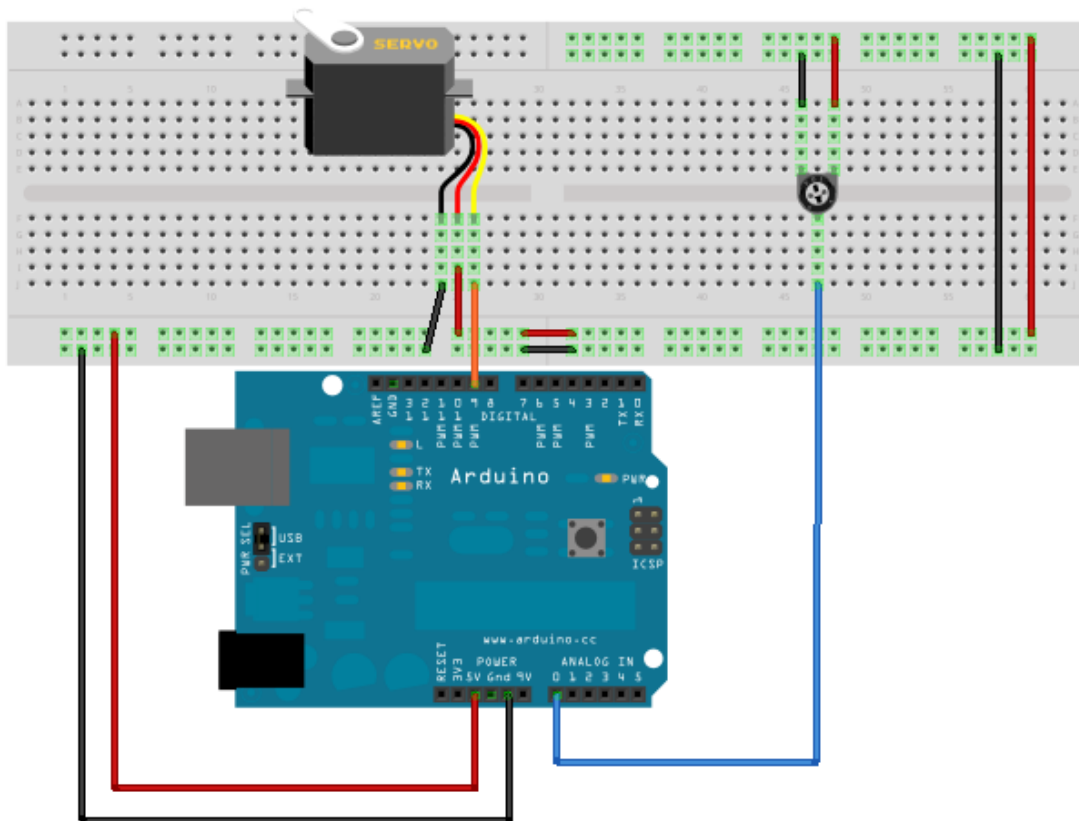
Segunda parte:**Enunciado:**

En esta práctica se pretende controlar el movimiento de un servo, con un potenciómetro.

Como hemos visto antes, este servo no debe de superar el rango 0-180, por lo que deberemos de referenciar (**map**) los valores tomados del potenciómetro, con el ángulo de trabajo del servo.

Material necesario:

- Mini servo.
- Potenciómetro

Montaje:

Práctica 15. Práctica semáforo

Enunciado:

La práctica consiste en un semáforo siempre en verde para los coches menos cuando se pulsa el botón de peatones, siempre y cuando el tiempo que lleva este en verde sea superior a un determinado tiempo.

La luz verde de peatón parpadeara cuando quede poco tiempo para cruzar.

Para lograr dicha práctica y poder contabilizar el tiempo, utilizaremos la función **millis()**, que contabiliza el tiempo desde que Arduino empezó a funcionar.

Teoría necesaria:

- Función **millis()**.....teoría página 37
- Crear funciones..... teoría página 41

Material necesario:

- 2 Leds verdes
- 2 leds rojos
- 1 led naranja
- Un pulsador

Practica 16. Control mediante infrarrojos.

Enunciado:

En la primera parte de esta práctica, se van a identificar los distintos botones de un mando, mediante el uso de un receptor IR y mostrando por pantalla, la información recibida.

Esta nos servirá para identificar los distintos botones de un mando y posteriormente, relacionar sus señales con actuadores o ciertas partes de código.

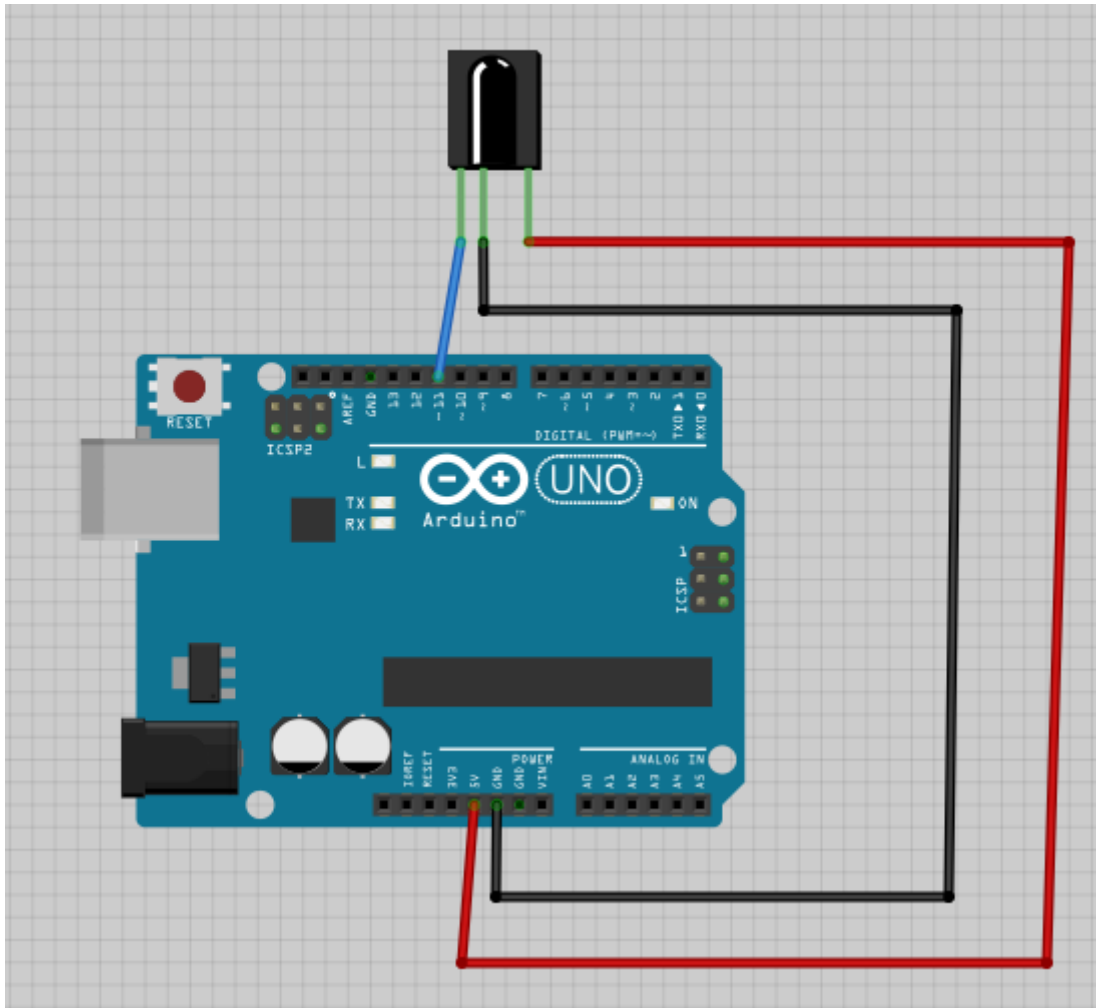
La segunda parte de la práctica consiste en utilizar el mando para controlar el encendido y apagado de unos leds.

Será necesario descargar la librería para el control de IR (<https://github.com/shirriff/Arduino-IRremote>) y guardarla en la carpeta correspondiente. Arduino / libraries

Material necesario:

- Receptor IR
- Mando a distancia de 36Khz de trabajo.

El conexionado del sensor de IR será:



Practica 17. Display de 7 segmentos.

Primera parte.

Enunciado:

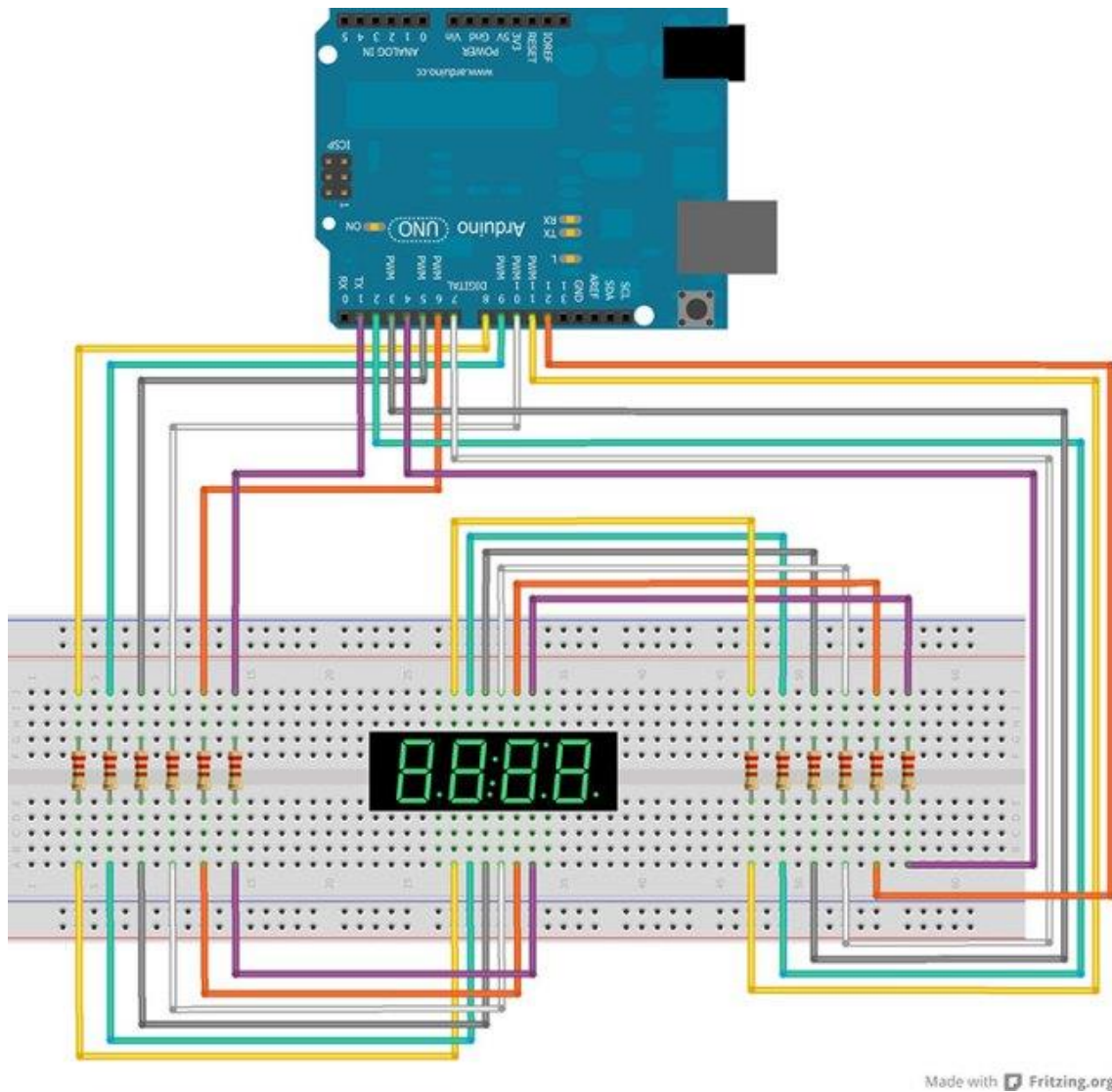
En esta primera parte de la práctica se va a montar el circuito y cargar un programa en nuestro Arduino que nos muestra el siguiente número: 23.45

Si nos fijamos en el código el número se trata como entero y se separa dígito a dígito, poniendo después del segundo dígito la coma decimal.

Material necesario:

- Display de 4 dígitos
- 12 resistencias de 220 ohmios.

Montaje:



Segunda parte.

Enunciado:

Para esta parte de la práctica y partiendo del código anterior, se va a mostrar por el display la temperatura percibida a través de un sensor de temperatura (LM35).

Material necesario:

- Display de 4 dígitos
- 12 resistencia de 220 ohms
- Sensor de temperatura LM35