

Grupo de Trabajo

Programación Informática I

IES TRINIDAD ARROYO

Departamento de Tecnología

Gloria Tobar Conde

Valentín Carpintero Rodríguez

Amaya Santamaría Gallego

Índice

1. SECA-MANOS AUTOMÁTICO	4
1. Introducción.....	4
2. Lista de materiales	4
3. Esquema eléctrico.....	5
4. Esquema del programa.....	5
5. Programa con Scratch	6
6. Montaje en placa board	9
7. Resultado del montaje.....	10
8. Video del funcionamiento.....	11
2. BAÑO TERMOSTÁTICO	12
1. Introducción.....	12
2. Lista de materiales	12
3. Diagramas de bloques.....	13
4. Esquema eléctrico.....	14
5. Montaje en la placa board.....	15
6. Programa con Scratch	15
7. Resultado del montaje.....	22
8. Video del funcionamiento.....	23
3. LUCES DE COLORES VARIADOS	24
1. Introducción.....	24
2. Lista de materiales	24
3. Esquema eléctrico.....	25
4. Esquema del programa.....	25
5. Programa con Scratch	27

6. Montaje en placa board	31
7. Resultado del montaje	32
8. Videos del funcionamiento	35

1. SECA-MANOS AUTOMÁTICO

1. Introducción

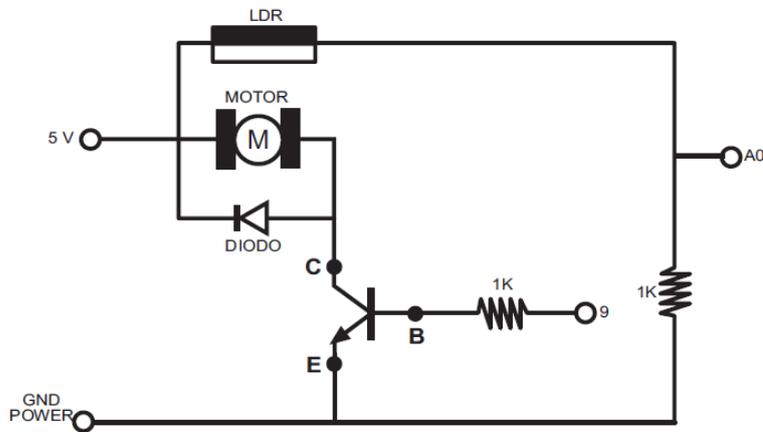
El siguiente proyecto consiste básicamente en un motor que se activa ante la falta de luz. Mediante la utilización de una LDR (fotorresistencia), controlaremos el funcionamiento del motor para que se active solo en el caso de detectar un nivel bajo de luz. La LDR actuará, por tanto, como un detector de presencia.

Como aplicación práctica del proyecto tenemos, por ejemplo, la implementación de un seca manos automático.

2. Lista de materiales

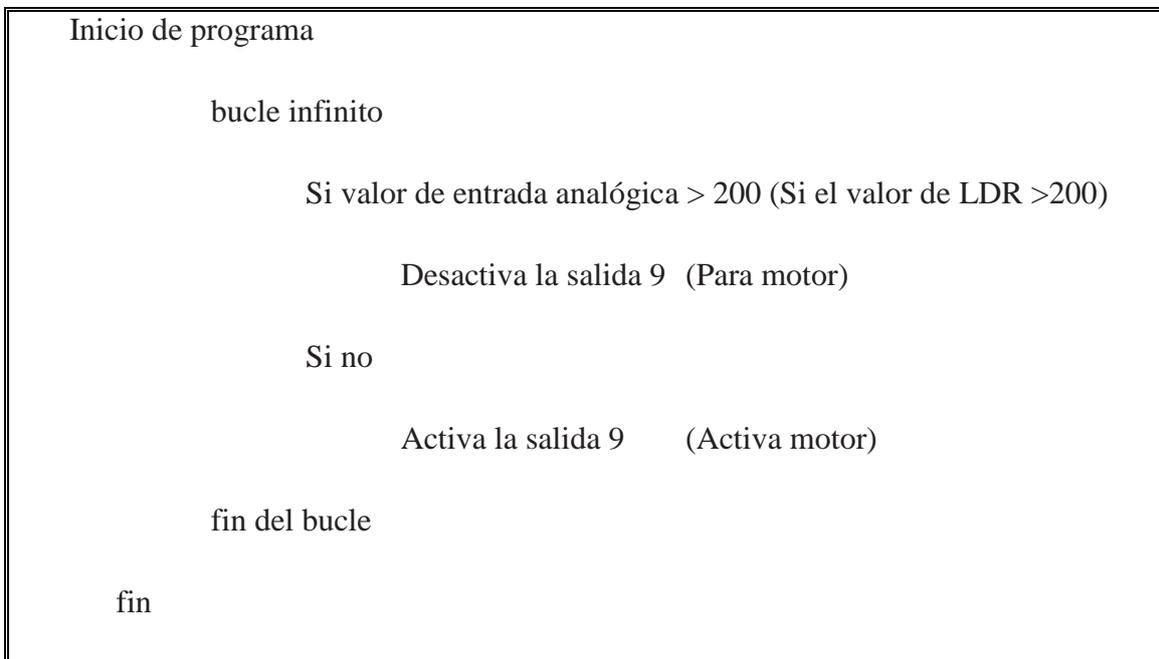
- 1 Tarjeta Arduino
- 1 Cable USB
- 1 placa board
- 1 LDR
- 2 Resistencia de 1K
- 1 Diodo
- 1 Transistor
- 1 Motor
- 1 Hélice
- 5 Latiguillos board macho-macho

3. Esquema eléctrico



4. Esquema del programa

El pseudocódigo del programa quedaría de la siguiente forma:

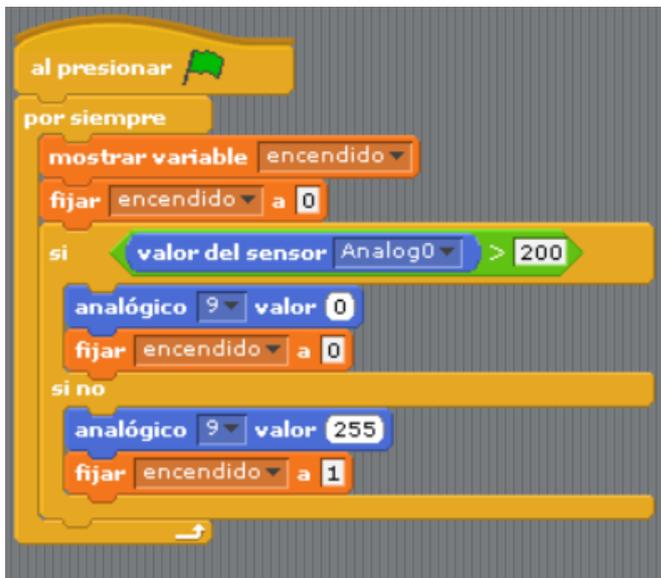


5. Programa con Scratch

Hemos dividido el programa en dos partes, por un lado la programación de la placa Arduino para que se active el motor en el caso de que la LDR detecte presencia, y por otro un pequeño programa para simular el movimiento de las aspas del ventilador en S4A.

➔ Programación de la placa Arduino

El programa en S4A quedaría como se muestra en la figura:

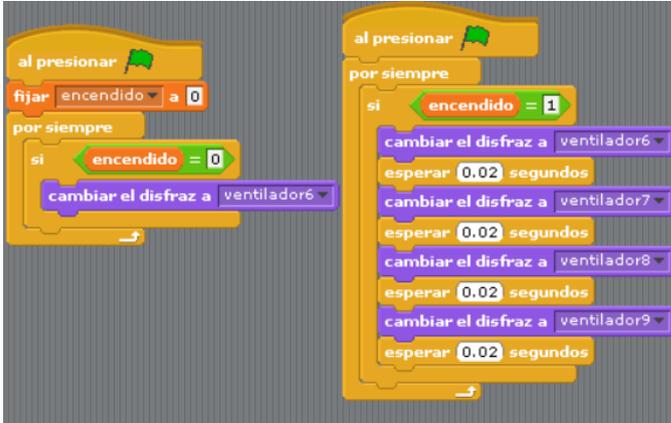


- Cuando la LDR detecta luz (sensor *Analog 0* > 200) el motor permanece apagado (sensor *Analog 9* = 0)
- Cuando la LDR detecta ausencia de luz (sensor *Analog 0* < 200) el motor comienza a funcionar (sensor *Analog 9* = 255)

➔ Simulación del movimiento en S4A

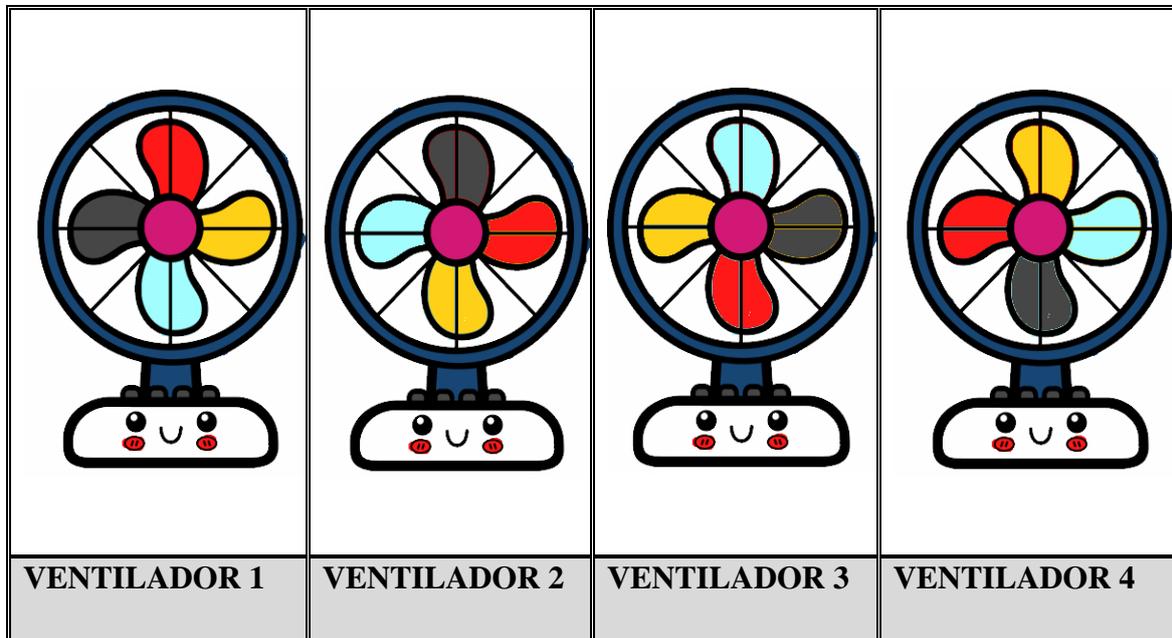
Mediante la utilización de diferentes disfraces simularemos el movimiento de las aspas del ventilador cuando éste esté en movimiento. Para ello haremos uso de una variable global (*encendido*) cuyo valor se modificará en función de los valores detectados por la LDR.

El programa en S4A quedaría como se muestra en la figura:

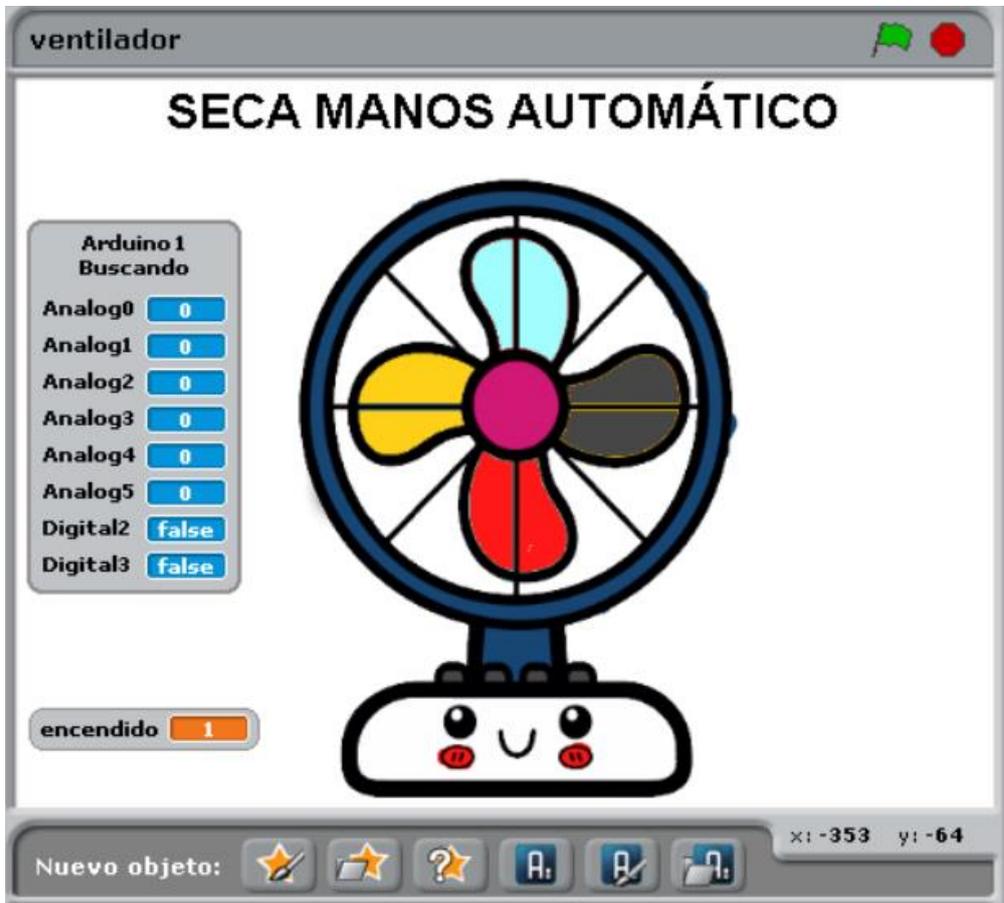


- Cuando la variable *encendido* es 0 el disfraz se mantiene.
- Cuando la variable *encendido* es 1 vamos variando el disfraz entre cuatro posibles para simular el movimiento.

A continuación se muestra la secuencia de disfraces:

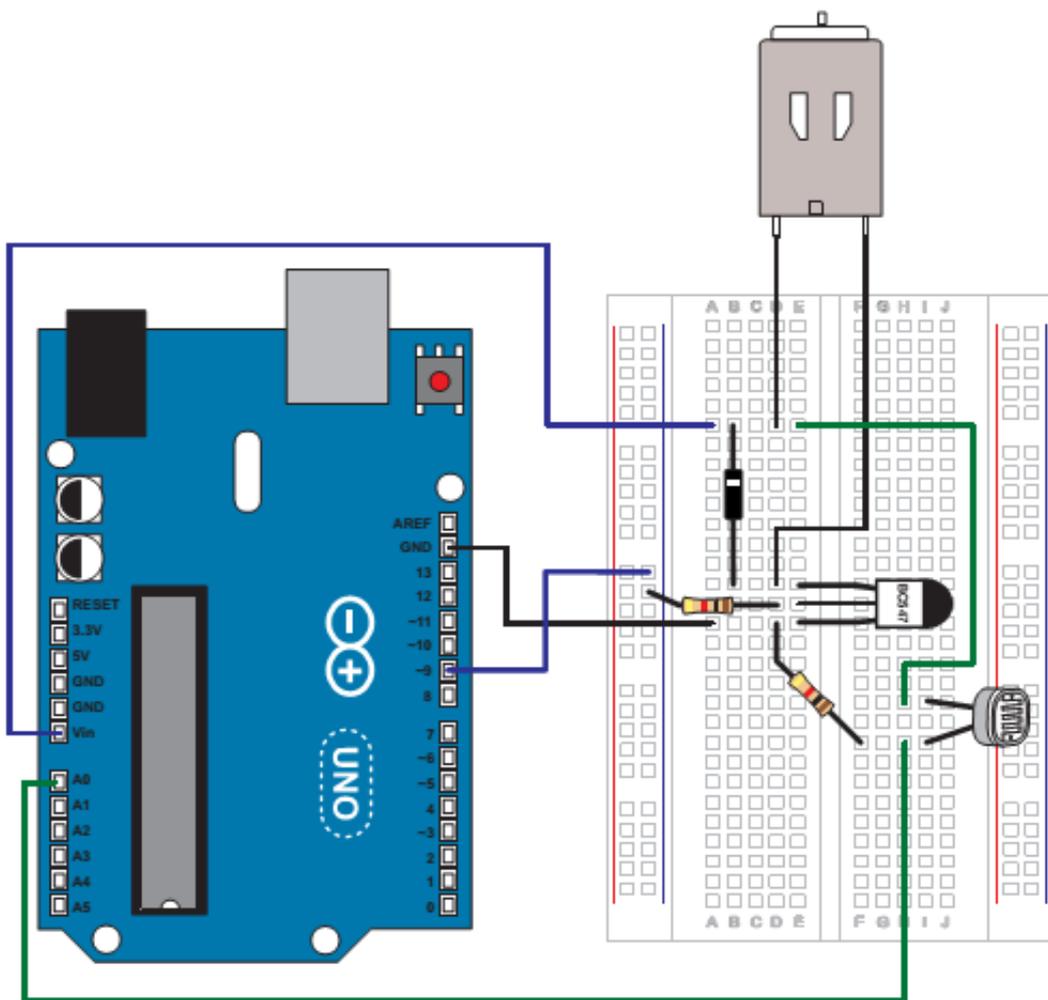


Con todo esto, la ventana de ejecución en S4A quedará como se muestra en la imagen:



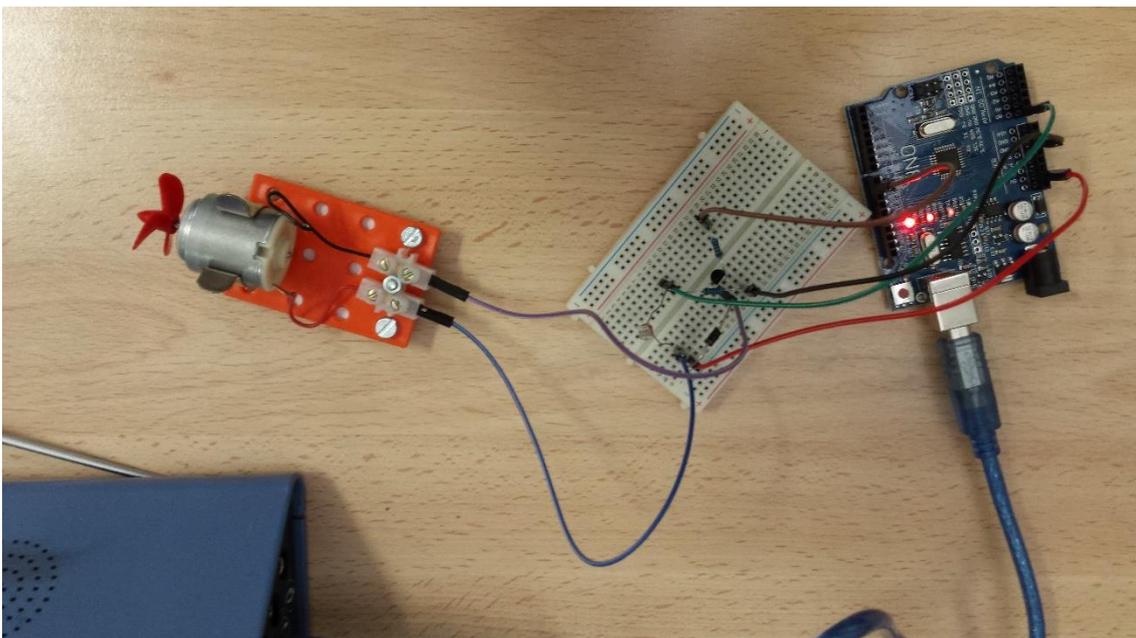
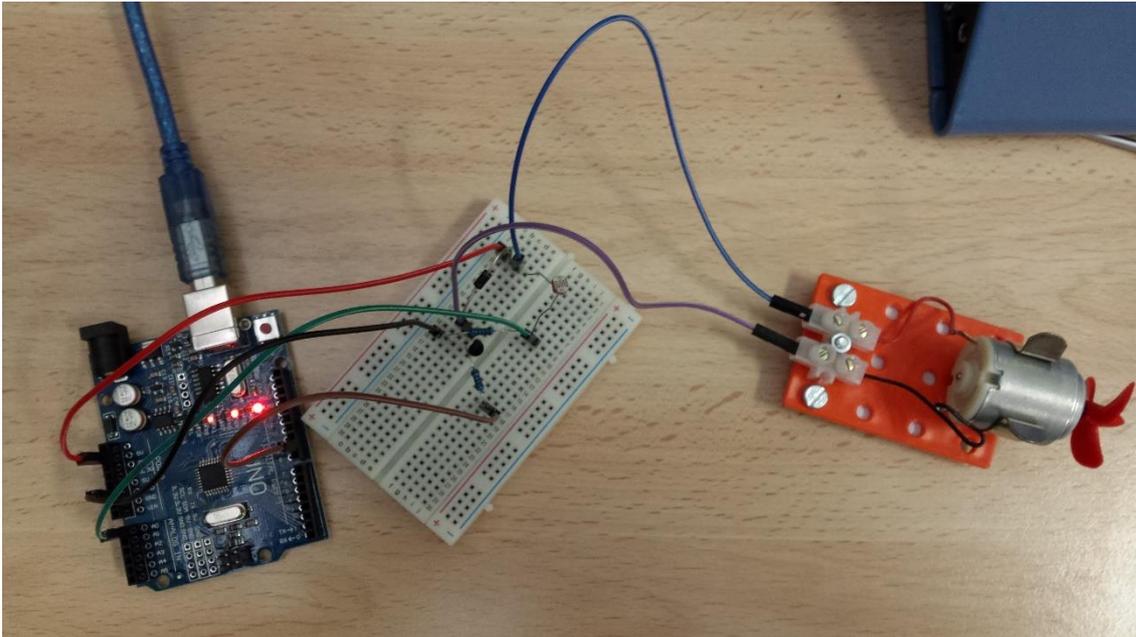
6. Montaje en placa board

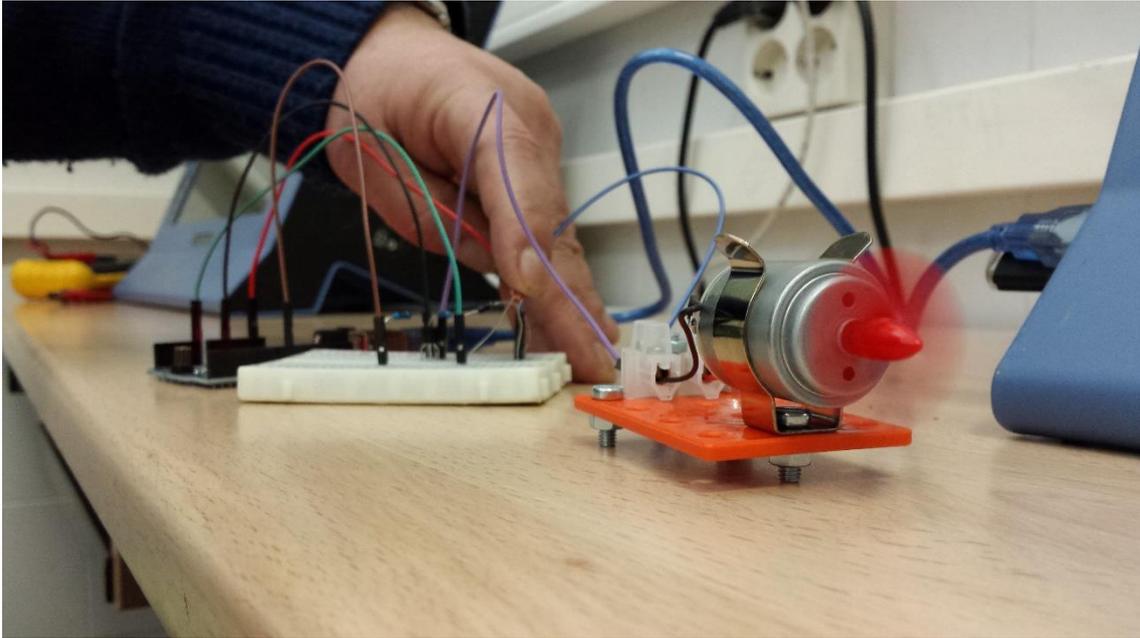
En la siguiente imagen se especifican las conexiones que realizaremos en la placa board y la tarjeta Arduino:



7. Resultado del montaje

Para ilustrar el resultado de nuestro montaje mostramos las siguientes fotografías:





8. Video del funcionamiento

Y como punto final incluimos en el documento un video ilustrativo en el que se puede observar cómo funciona nuestro seca-manos automático.

[multimedia\20160218_105521.mp4](#)

2. BAÑO TERMOSTÁTICO

1. Introducción

El siguiente proyecto consiste en realizar un baño termostático para la cocción de alimentos controlada en temperatura, es decir, mediante una señal de consigna (temperatura elegida), se efectúa el calentamiento del agua por unas resistencias eléctricas que están gobernadas por una tarjeta Arduino, la temperatura que va tomando el agua se mide mediante un sensor y Arduino lo compara con la temperatura de consigna. Cuando se igualan las temperaturas, Arduino apaga las resistencias y las vuelve a encender si la temperatura del agua baja debido a pérdidas de calor al exterior

2. Lista de materiales

- 1 Tarjeta Arduino
- 1 Cable USB
- 1 placa board
- 1 NTC
- 2 Potenciómetros de 10K
- 1 Diodo Led
- 1 Resistencia de 220 Ω
- 9 Latiguillos board macho-macho
- 1 destornillador de ranura pequeño

3. Diagramas de bloques

Diagrama de bloques del sistema realizado físicamente

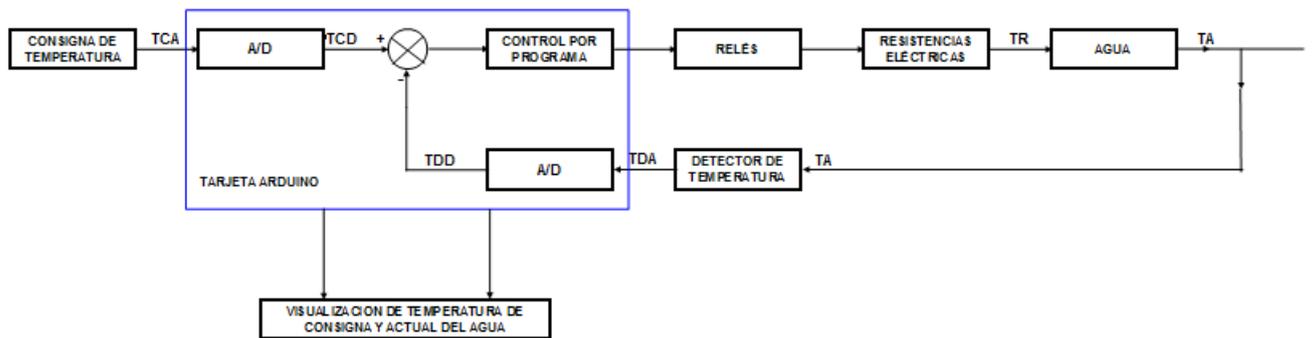
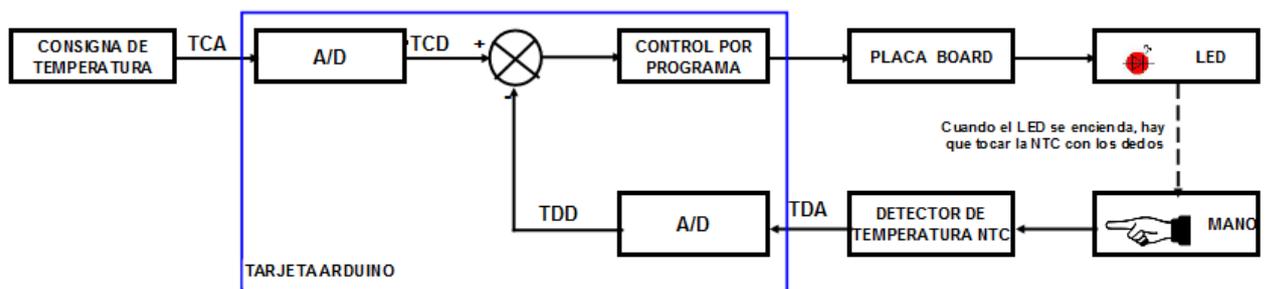
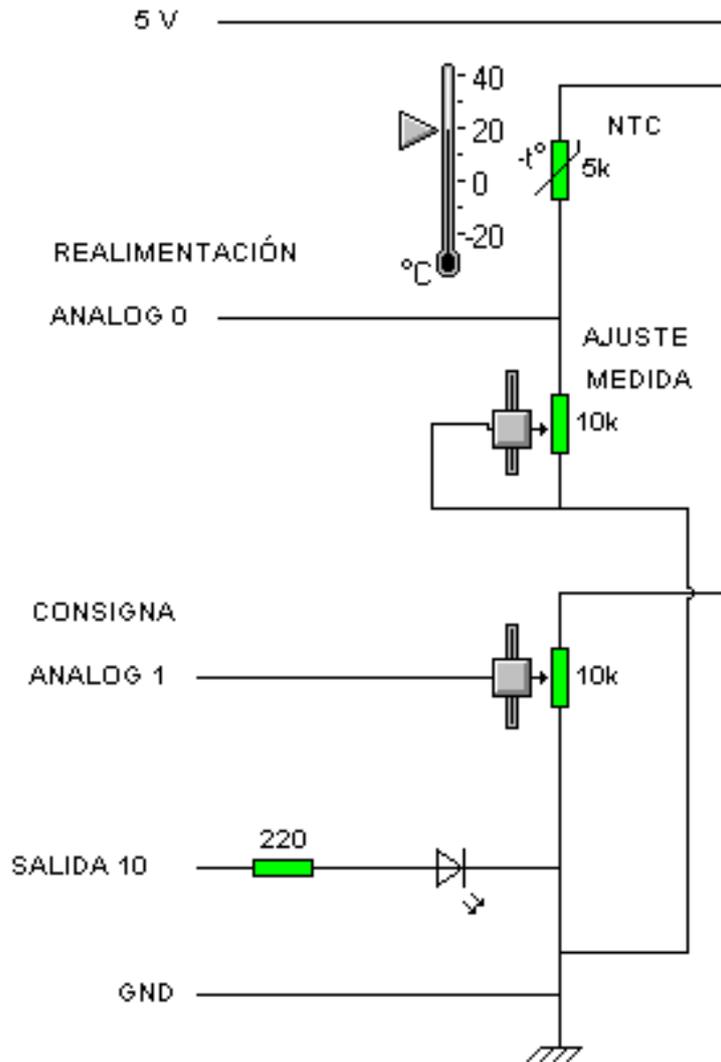


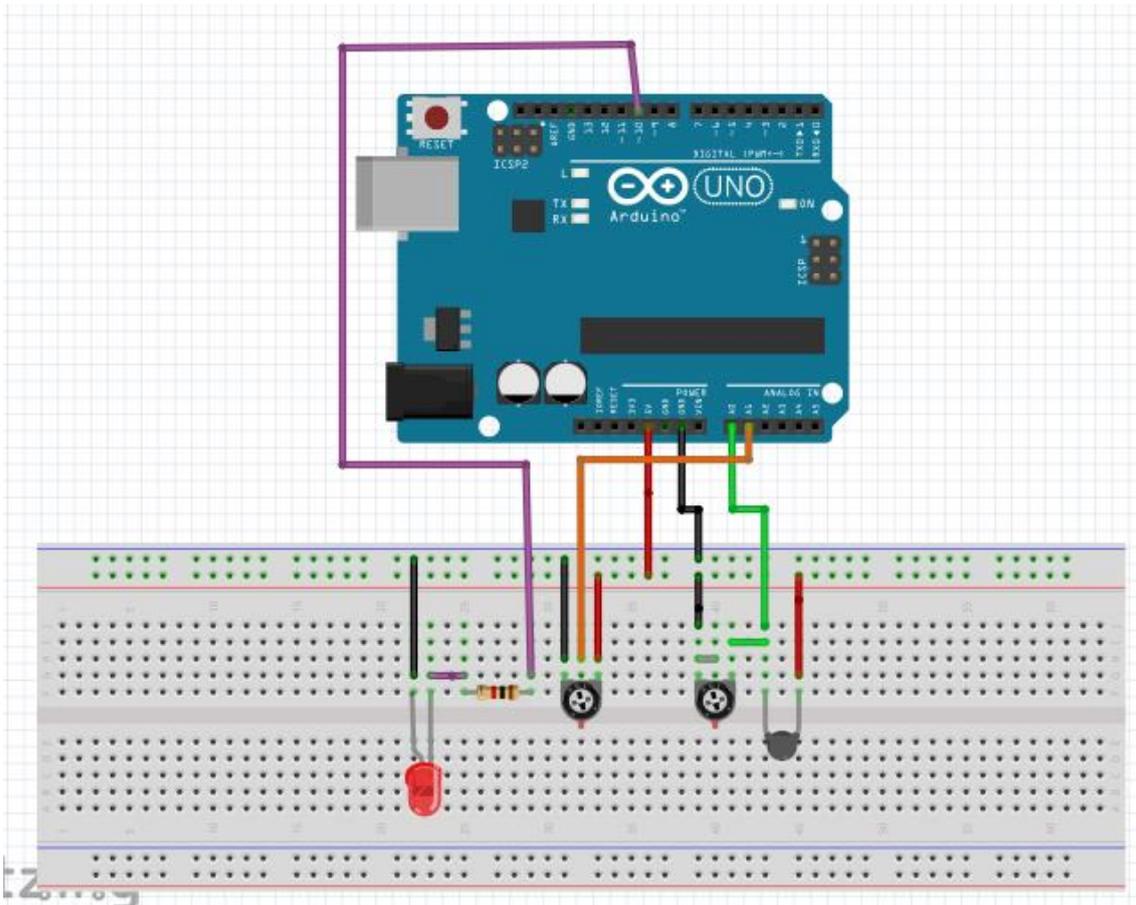
Diagrama de bloques del sistema de prueba



4. Esquema eléctrico

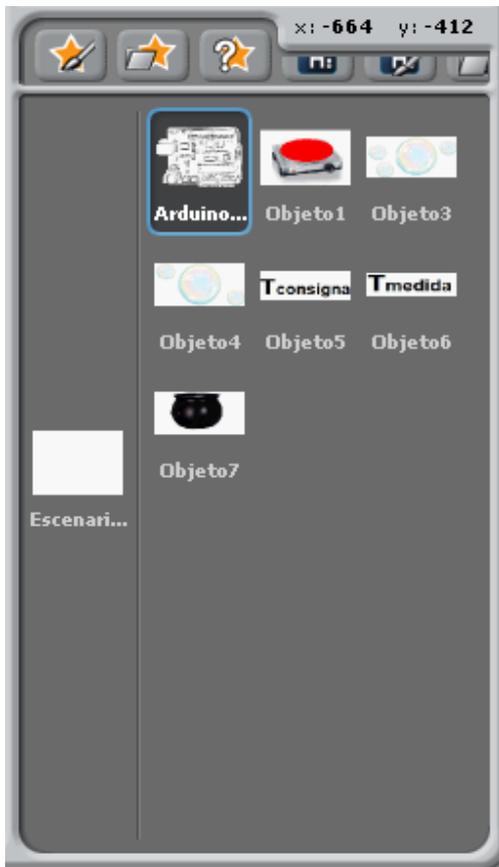


5. Montaje en la placa board



6. Programa con Scratch

Hemos dividido el programa en dos partes, por un lado la programación de la placa Arduino y por otro un pequeño programa para simular el funcionamiento del baño termostático en S4A. Para ello hemos creado varios objetos:



- En el objeto *Arduino* programaremos lo relativo a las entradas y salidas de la placa.
- Para la programación de la simulación utilizaremos siete objetos, a los que asociaremos diferentes disfraces:
 - *Objeto 1*: Hornillo (encendido y apagado)
 - *Objeto 3*: Burbuja I
 - *Objeto 4*: Burbuja II
 - *Objeto 5*: Título I
 - *Objeto 6*: Título II
 - *Objeto 7*: Cacerola

⇒ Programación de la placa Arduino

El programa en S4A quedaría como se muestra en la figura:



- Cuando el valor de *Analog 1* supera el de *Analog 0* la variable *encendido* tomará valor 1. En la placa se encenderá el LED conectado al pin 10.
- En caso contrario la variable *encendido* valdrá 0 y el LED estará apagado.

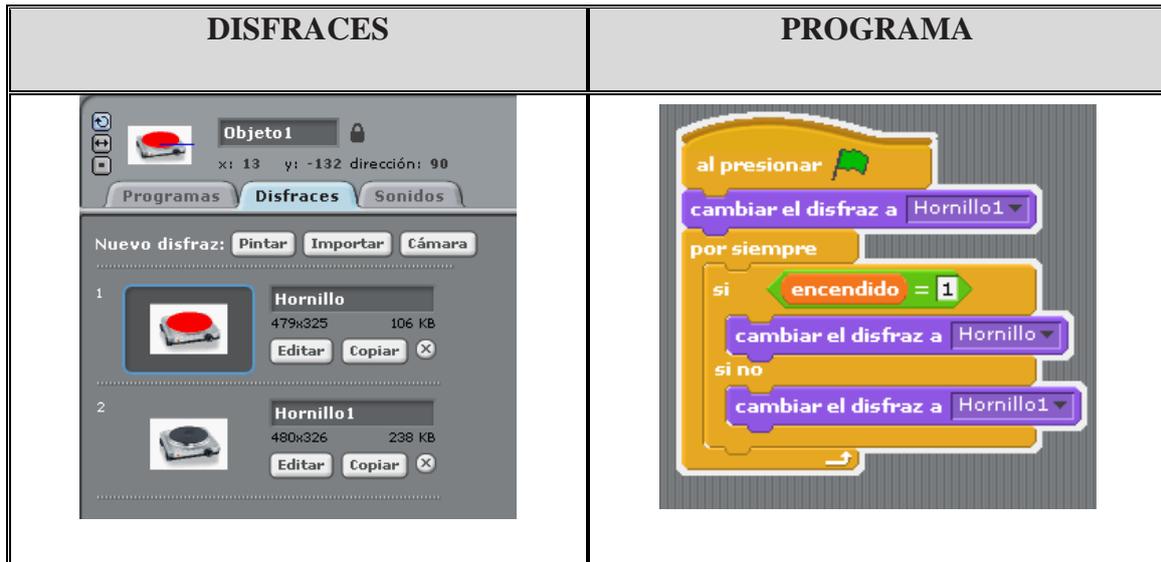
➤ Simulación del baño termostático en S4A

Mediante la utilización de diferentes disfraces simularemos el funcionamiento del sistema, en el que nos podemos encontrar las siguientes situaciones:

- Si no hemos superado la temperatura umbral, el hornillo estará encendido. Para simular el funcionamiento saldrán burbujas de la cacerola y se emitirá el sonido de la cocción. El efecto de movimiento lo conseguiremos trabajando con dos objetos, Burbujas I y Burbujas II.
- Cuando superamos la temperatura umbral, el hornillo se apagará. En este caso las burbujas y el sonido desaparecerán.
- Una vez que vuelva a bajar la temperatura por debajo del umbral volveremos a la primera situación.

Los diferentes programas en S4A quedarían así:

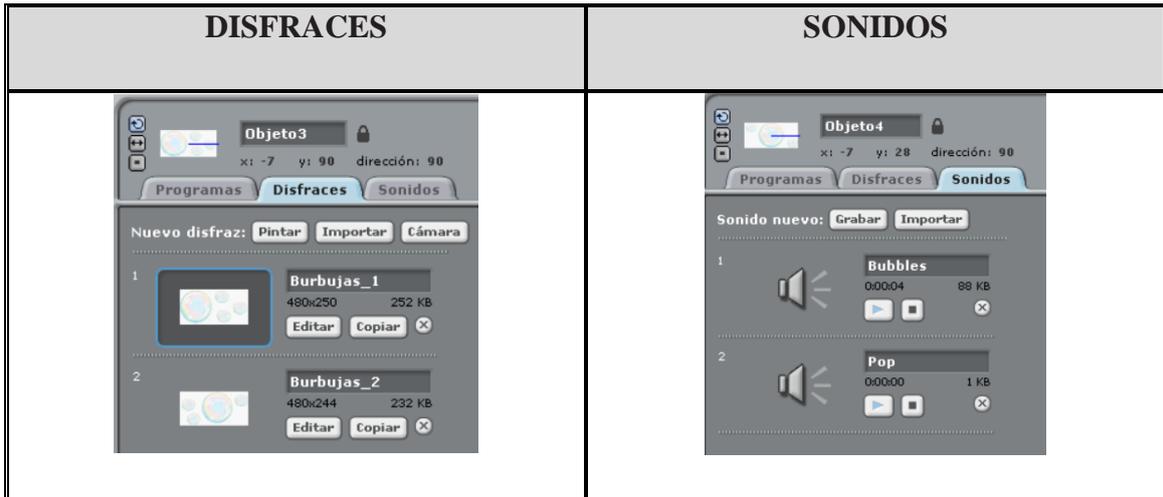
□ **Objeto 1 – Hornillo**



- Cuando la variable *encendido*=1 activamos el disfraz del hornillo encendido.
- Cuando la variable *encendido*=0 activamos el disfraz del hornillo apagado.

□ **Objetos 2 y 3 – Burbujas I y II**

Estos dos objetos comparten los disfraces y los sonidos. La diferencia está en la programación. En uno de ellos cambiaremos el disfraz de las burbujas y activaremos los sonidos cada segundo y en el otro cada dos segundos, para conseguir así el efecto de movimiento.



□ **Objetos 5, 6 y 7 – Títulos I y II y Cacerola**

Estos objetos no tienen programación. Tan solo se muestran los disfraces que nos faltan para implementar el escenario.

DISFRAZ TÍTULO I	DISFRAZ TÍTULO II	DISFRAZ CACEROLA
<p>Objeto5 x: -98 y: 128 dirección: 90 Programas Disfraces Sonidos Nuevo disfraz: Pintar Importar Cámara 1 Tcons 132x52 17 KB Editar Copiar X</p>	<p>Objeto6 x: -102 y: 161 dirección: 90 Programas Disfraces Sonidos Nuevo disfraz: Pintar Importar Cámara 1 Tmedida 118x39 13 KB Editar Copiar X</p>	<p>Objeto7 x: 10 y: -52 dirección: 90 Programas Disfraces Sonidos Nuevo disfraz: Pintar Importar Cámara 1 Caldero 450x237 169 KB Editar Copiar X</p>

□ **Imágenes asociadas a cada uno de los disfraces**

En la siguiente tabla podemos ver las imágenes que hemos utilizado para los diferentes disfraces:

<p>multimedia/Hornillo_apagado.jpg</p>	<p>multimedia/Hornillo_encendido.jpg</p>	<p>Tmedida</p> <p>multimedia/Tmed.jpg</p>	<p>Tconsigna</p> <p>multimedia/Tcons.jpg</p>
Hornillo	Hornillo 1	Tmed	Tcons
<p>multimedia/Burbujas_11.jpg</p>	<p>multimedia/Burbujas_12.jpg</p>	<p>multimedia/Burbujas_21.jpg</p>	<p>multimedia/Burbujas_22.jpg</p>
Burbujas 11	Burbujas 12	Burbujas 21	Burbujas 22



□ **Resultado final de la ejecución**

Con todo esto, la ventana de ejecución en S4A quedará como se muestra en la imagen:

➤ Cuando el hornillo está encendido:

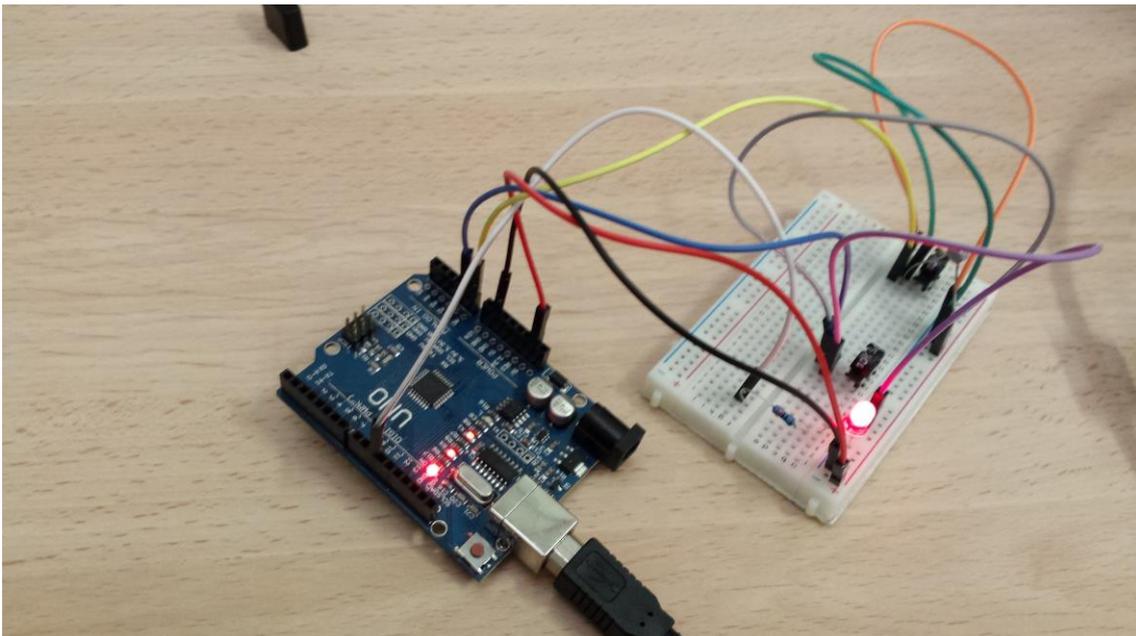


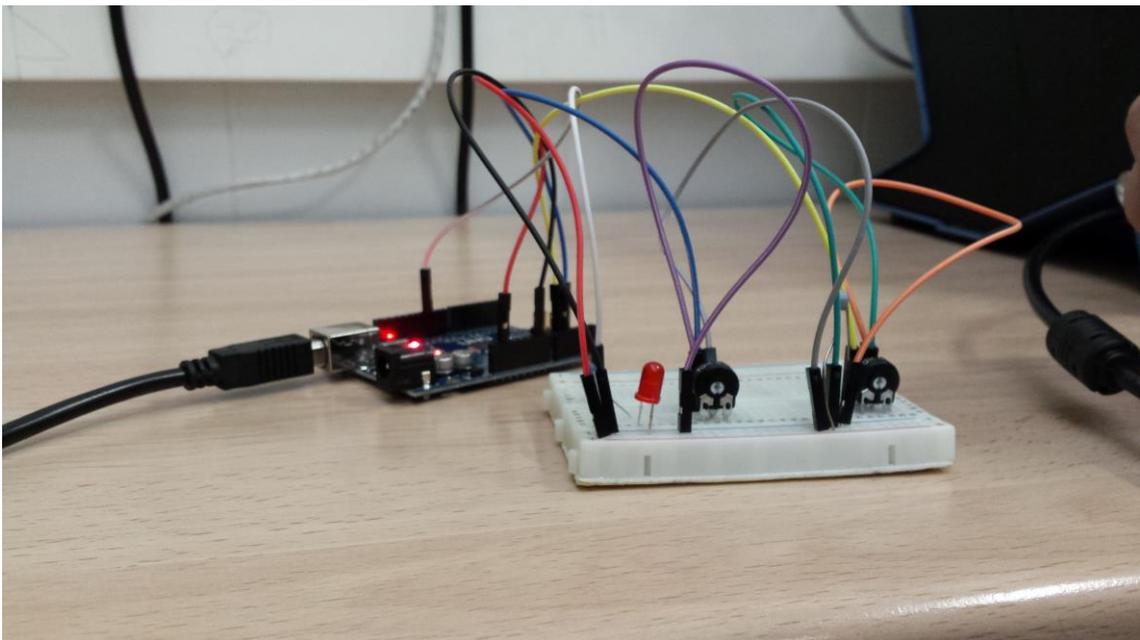
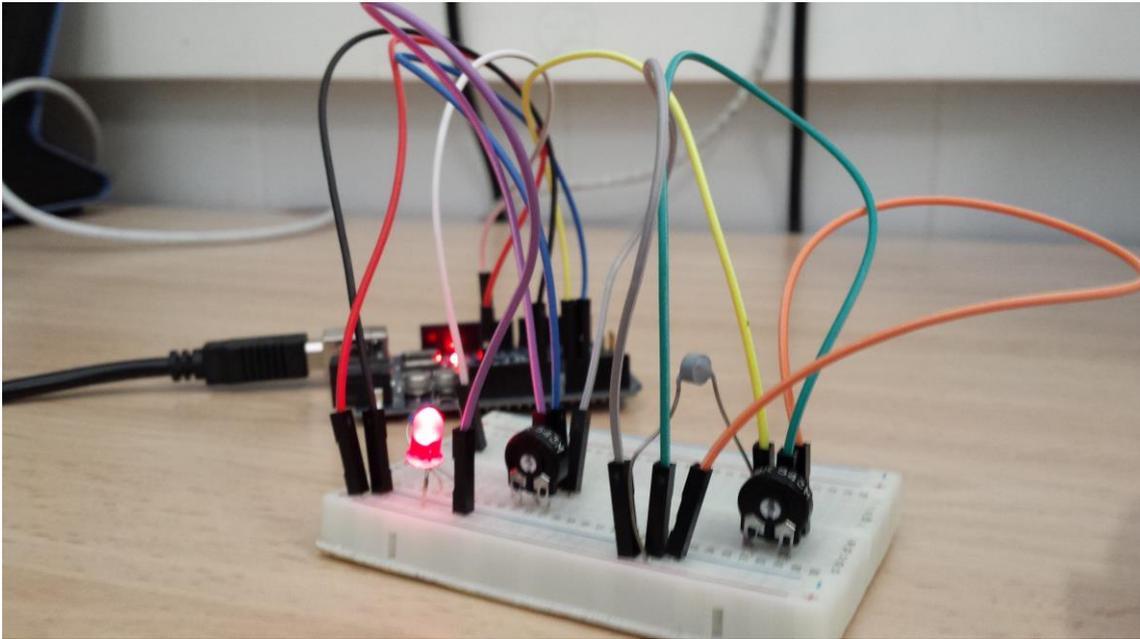
- Cuando el hornillo está apagado:



7. Resultado del montaje

Para ilustrar el resultado de nuestro montaje mostramos las siguientes fotografías:





8. Video del funcionamiento

Y como punto final incluimos en el documento un video ilustrativo en el que se puede observar cómo funciona nuestro sistema

[multimedia\20160303_110858.mp4](#)

3. LUCES DE COLORES VARIADOS

1. Introducción

En el siguiente proyecto utilizaremos un LED RGB para realizar dos prácticas diferentes:

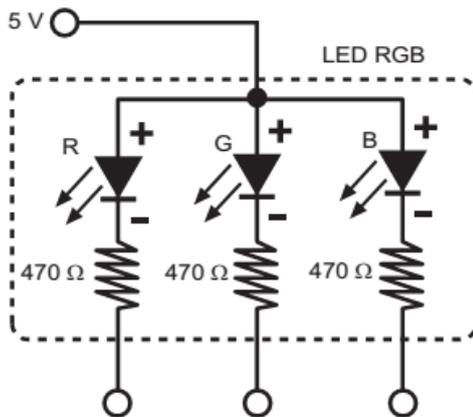
- En la primera práctica se mostrarán los colores rojo, verde y azul de forma alternativa.
- En la segunda el led nos mostrara toda la gama de colores que sus características le permitan.

Como aplicación práctica del proyecto tenemos, por ejemplo, la implementación de luces de colores para un árbol de navidad.

2. Lista de materiales

- 1 Tarjeta Arduino
- 1 Cable USB
- 1 placa board
- 3 resistencias de 470 Ohm
- 1 Led RGB
- 4 Latiguillos board macho-macho

3. Esquema eléctrico



4. Esquema del programa

El pseudocódigo de los programas quedará de la siguiente forma:

a) Luz roja, verde y azul

```

Inicio de programa
    Bucle infinito
        Activa la salida 6 a 255
        Activa la salida 9 a 255
        Activa la salida 5 a 0
        Esperar 1 s
        Activa la salida 6 a 0
        Activa la salida 9 a 255
        Activa la salida 5 a 255
        Esperar 1 s
        Activa la salida 6 a 255
        Activa la salida 9 a 0
        Activa la salida 10 a 255
        Esperar 1 s
    Fin del bucle
Fin
    
```

b) Gama de colores

Inicio del programa

Bucle infinito

Salida 5 varía de 0 a 255 cada 0.002 segundos

Salida 9 varía de 255 a 0 cada 0.002 segundos

Salida 9 varía de 0 a 255 cada 0.002 segundos

Salida 6 varía de 255 a 0 cada 0.002 segundos

Salida 9 varía de 255 a 0 cada 0.002 segundos

Salida 5 varía de 255 a 0 cada 0.002 segundos

Salida 5 varía de 0 a 255 cada 0.002 segundos

Salida 9 varía de 0 a 255 cada 0.002 segundos

Salida 5 varía de 255 a 0 cada 0.002 segundos

Salida 6 varía de 0 a 255 cada 0.002 segundos

Salida 9 varía de 255 a 0 cada 0.002 segundos

Salida 9 varía de 0 a 255 cada 0.002 segundos

Fin del bucle

Fin

5. Programa con Scratch

a) Luz roja, verde y azul

Hemos dividido el programa en dos partes, por un lado la programación de la placa Arduino para que implemente el cambio de color en el LED RGB, y por otro un pequeño programa para simular este cambio de color en S4A.

➤ Programación de la placa Arduino

El programa en S4A quedaría como se muestra en la figura:

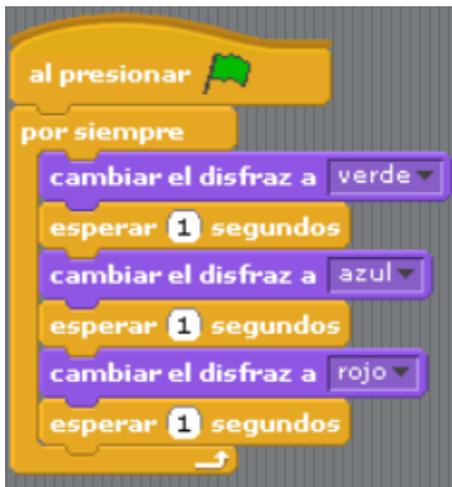


- Un LED RGB es en realidad la unión de tres LEDs (rojo, verde y azul).
- Vamos variando el valor de las salidas *Analog 6, 9 y 5* para modificar la tensión que proporcionamos a cada pin del LED.
- Dependiendo de esta tensión el LED se iluminará con un color u otro. Con ello conseguimos que la luz emitida varíe entre el verde, el azul y el rojo.

➤ **Simulación del cambio de color en S4A**

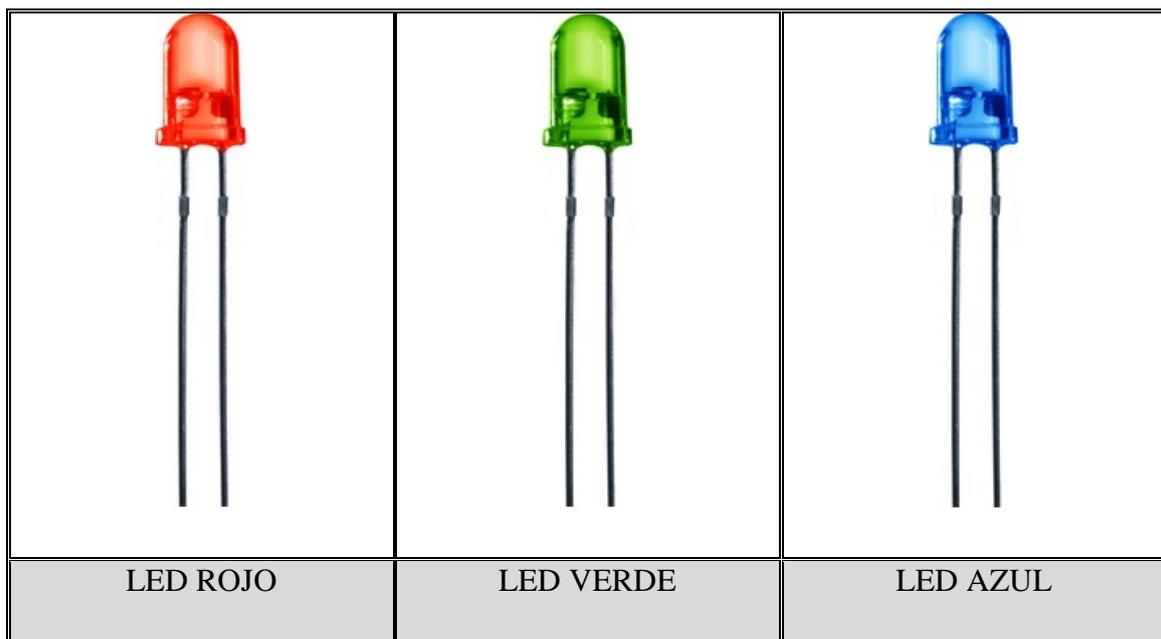
Mediante la utilización de diferentes disfraces simularemos el cambio de color del LED RGB. Para ello modificaremos el disfraz en función del color del LED activo en cada momento, esperando el mismo tiempo para el cambio.

El programa en S4A quedaría como se muestra en la figura:

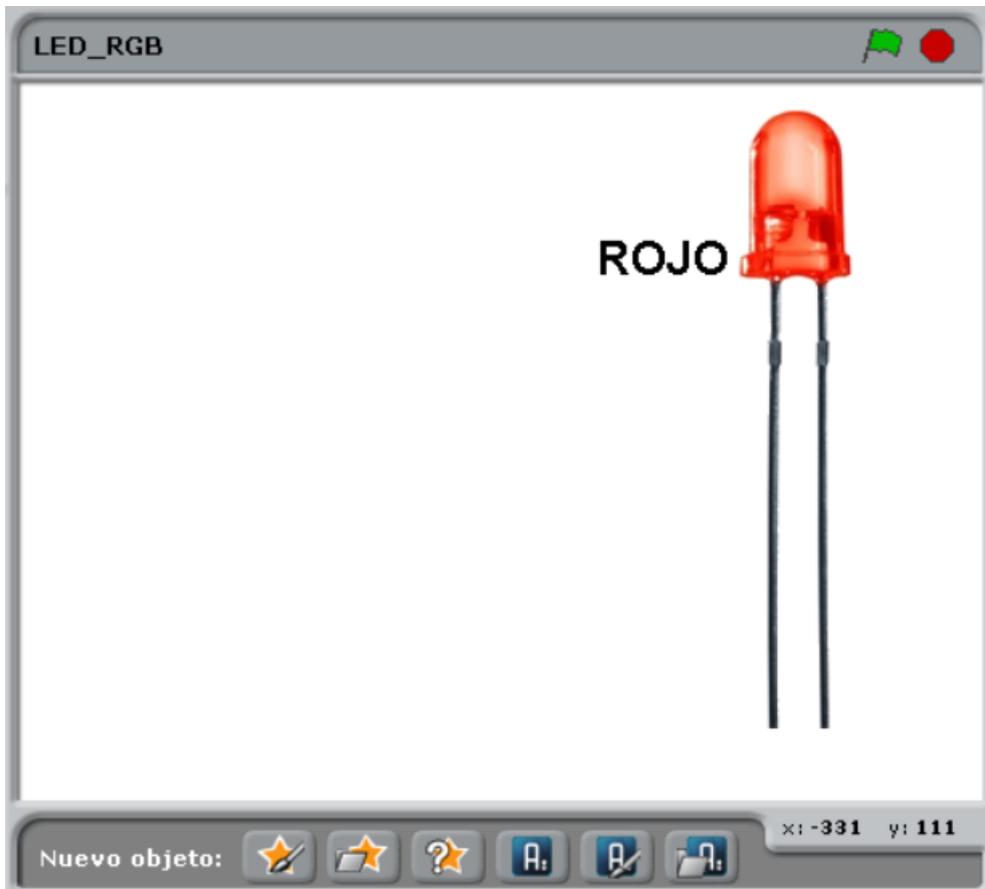


- Vamos variando el disfraz para que nos muestre un LED verde, uno azul y uno rojo cada segundo.

A continuación se muestra la secuencia de disfraces:



Con todo esto, la ventana de ejecución en S4A quedará como se muestra en la imagen, apareciendo en cada momento el LED que esté activo:



b) Gama de colores

En este caso solo programaremos la placa Arduino para que el LED emita la gama de colores, ya que si quisiésemos simularlo en S4A necesitaríamos implementar numerosos disfraces.

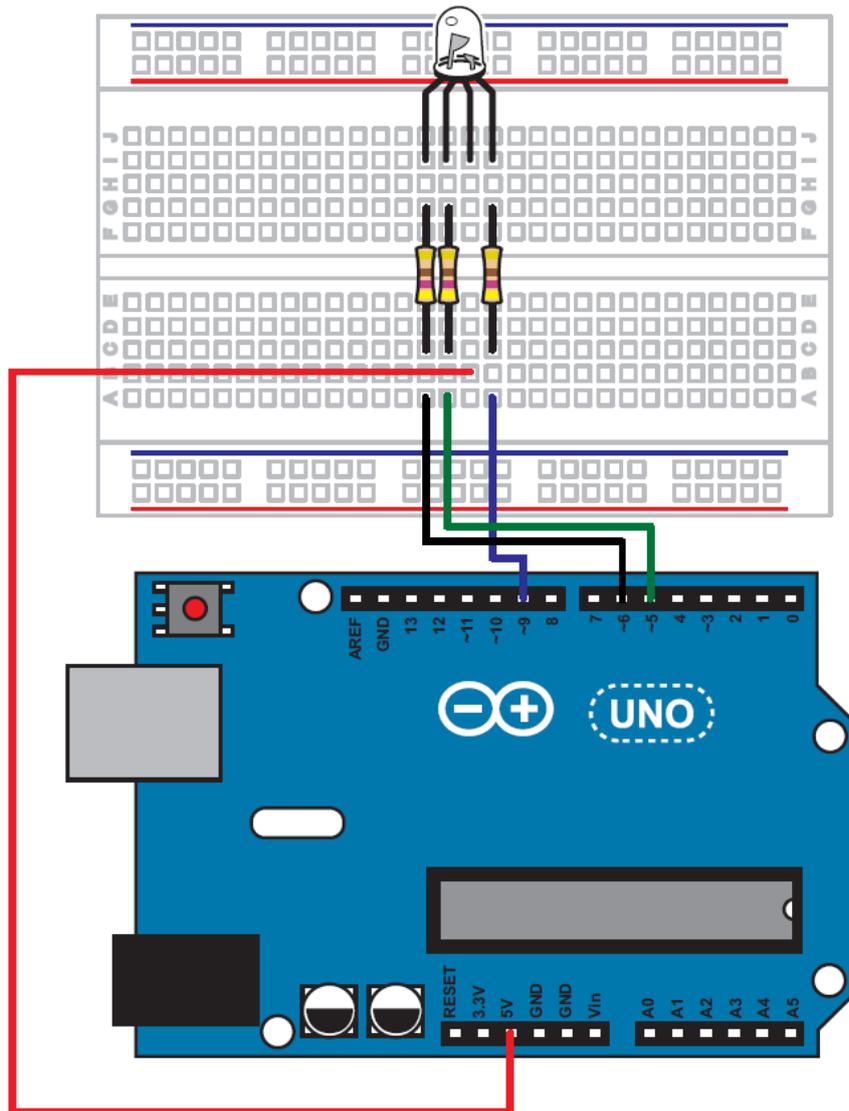
El programa en S4A quedaría como se muestra en la figura:



- Mediante el uso de una variable (I) vamos dando valor a las diferentes salidas analógicas en un rango de 0 a 255.
- Con ello conseguimos que el LED RGB emita una amplia gama de colores.

6. Montaje en placa board

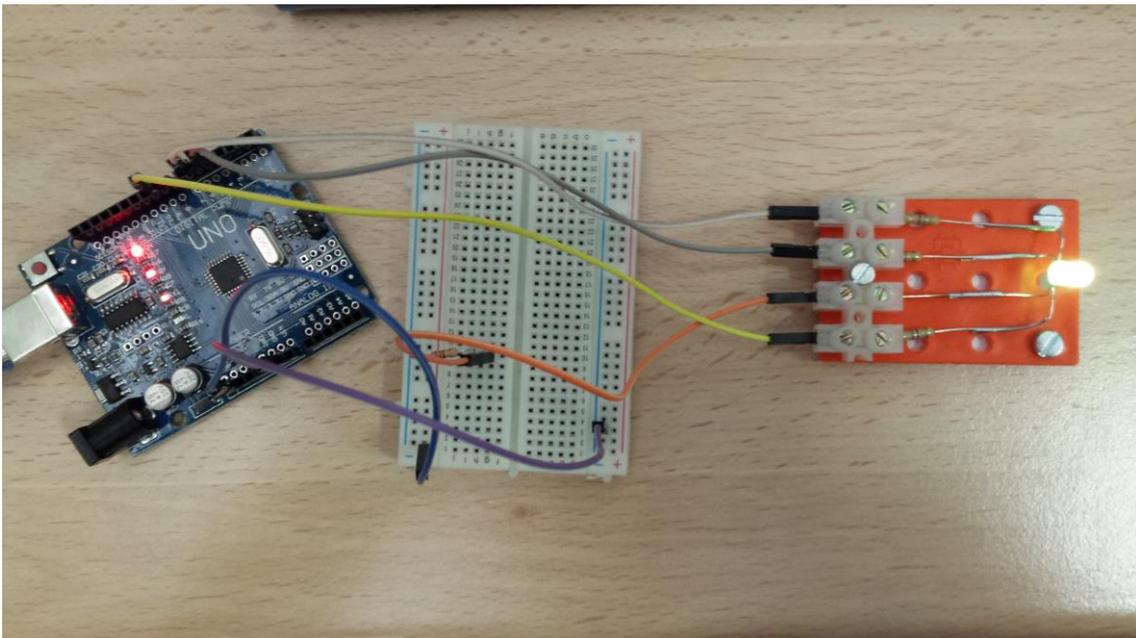
En la siguiente imagen se especifican las conexiones que realizaremos en la placa board y la tarjeta Arduino. El montaje será el mismo para los dos proyectos.



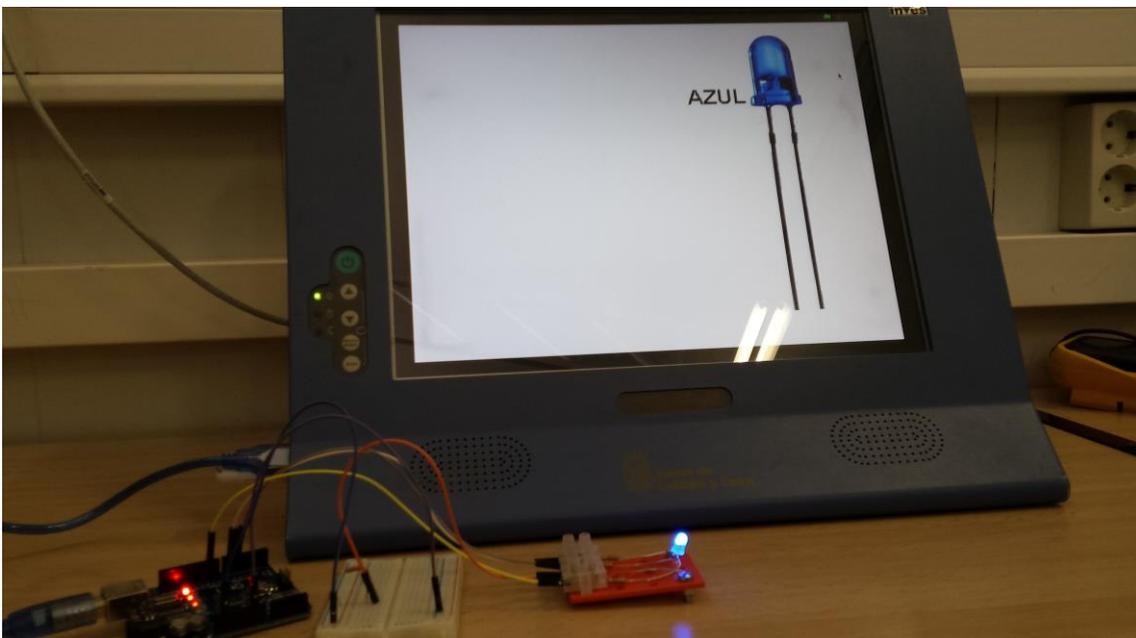
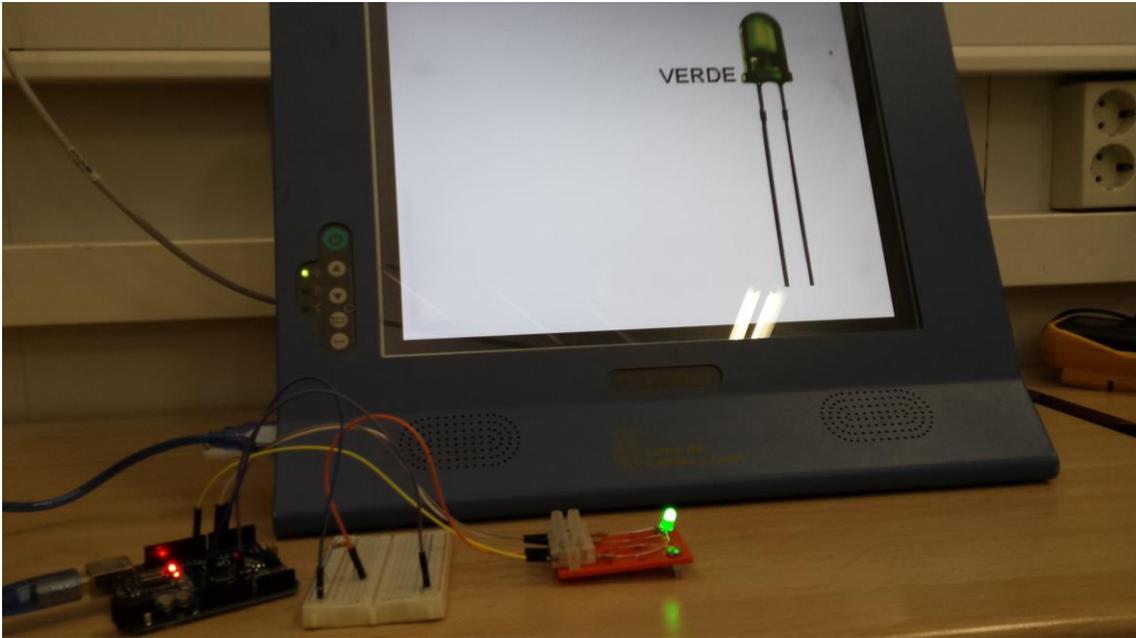
7. Resultado del montaje

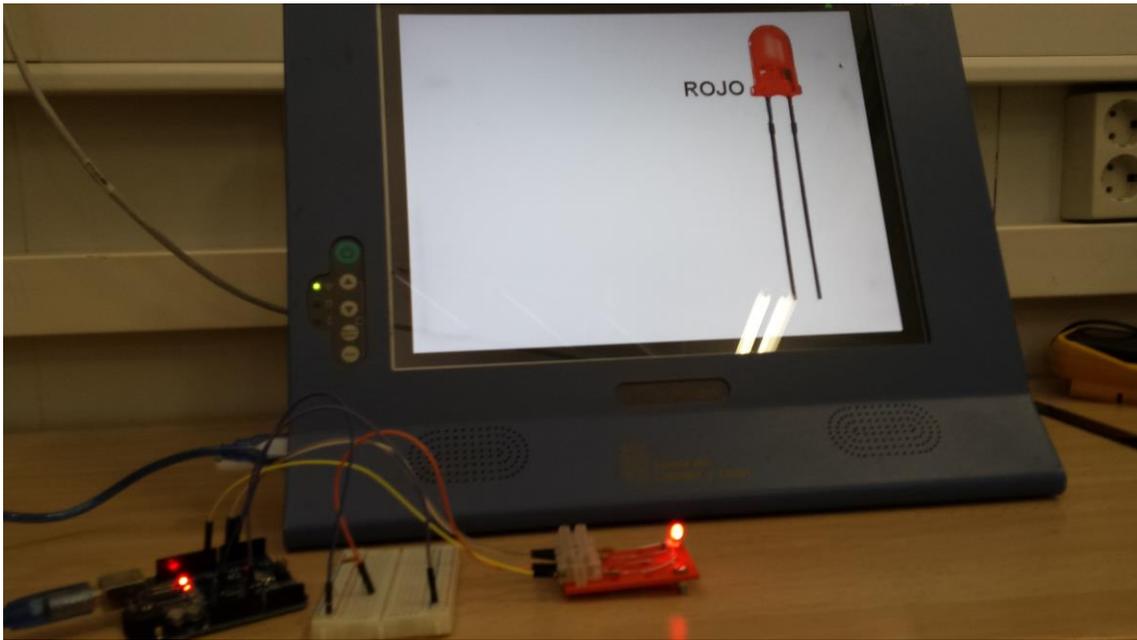
Para ilustrar el resultado de nuestro montaje mostramos las siguientes fotografías:

a) Montaje general

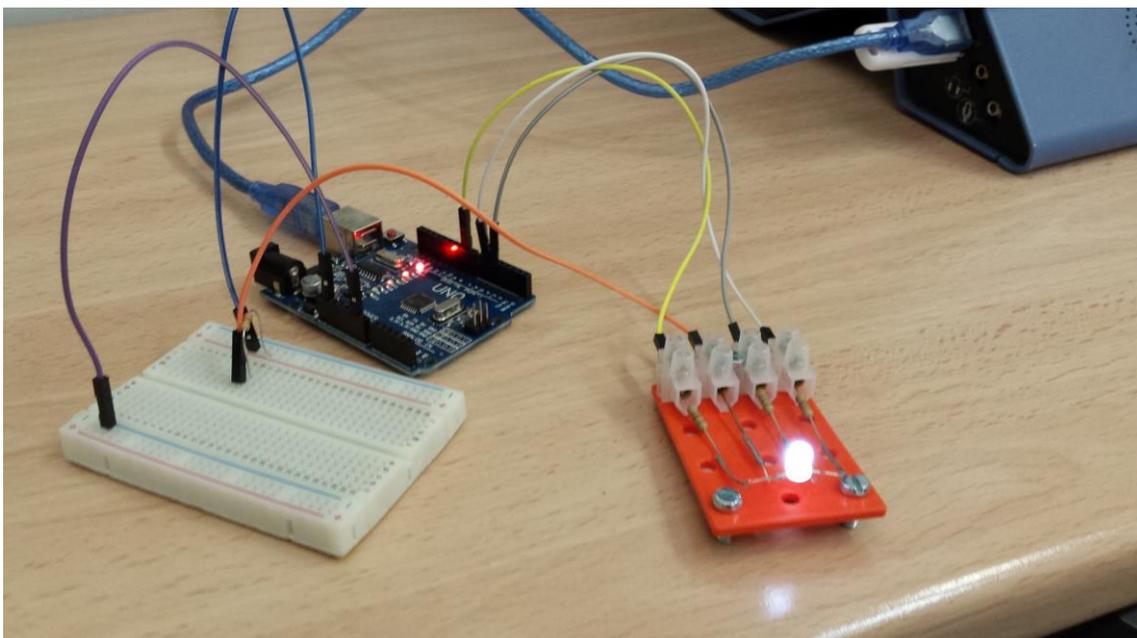


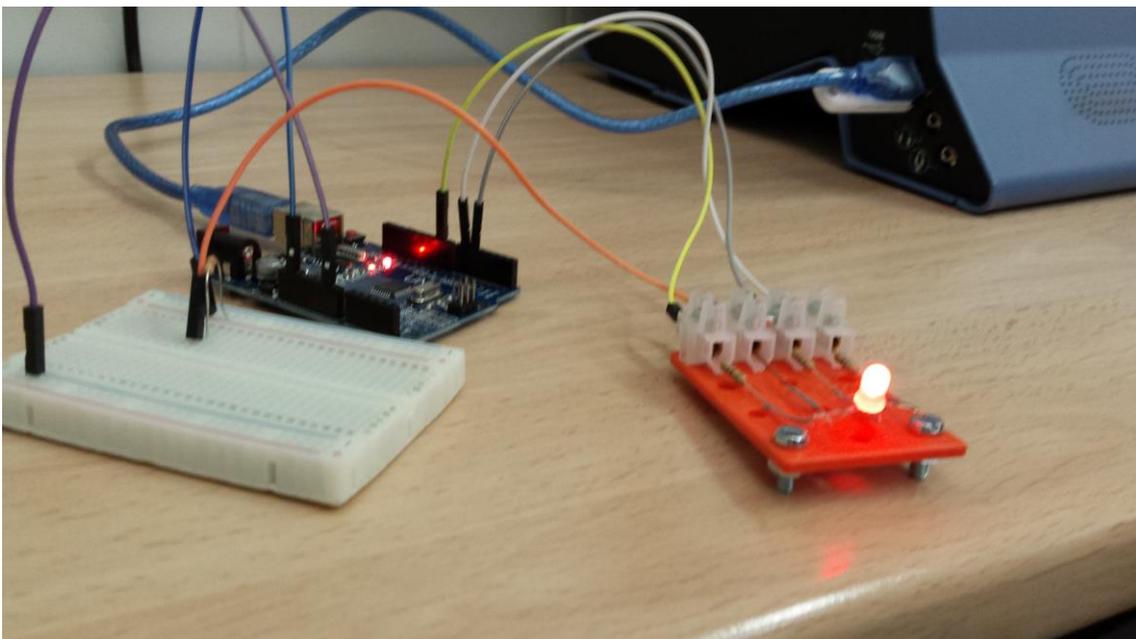
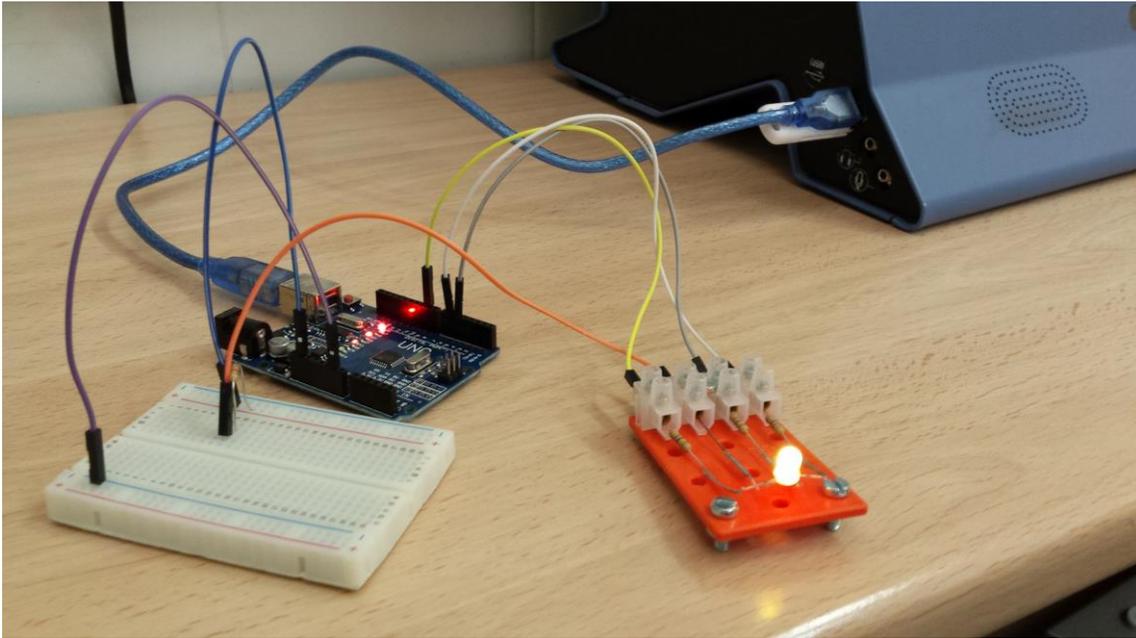
b) Luz roja, verde y azul





c) Gama de colores





8. Videos del funcionamiento

Y como punto final incluimos en el documento dos videos ilustrativos en los que se puede observar cómo funcionan nuestras luces de colores.

[multimedia/video.mp4](#)

[multimedia/gama.mp4](#)