

SISTEMAS DE CONTROL AUTOMÁTICOS PROGRAMADOS (ROBÓTICA)

4º ESO TECNOLOGÍA

1. Introducción.

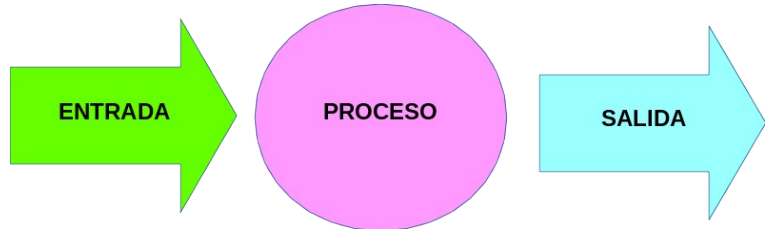
Un sistema de control automático o automatismo, es un conjunto de elementos técnicos que unidos son capaces de realizar una serie de acciones para resolver un problema sin intervención humana.

Los que vamos a usar en nuestro curso son programados, que significa que el automatismo funcionará dependiendo de un programa que le haremos a medida de la función a realizar.

2. Elementos de un sistema de control.

Nuestro sistema de control va a contener básicamente tres bloques de dispositivos:

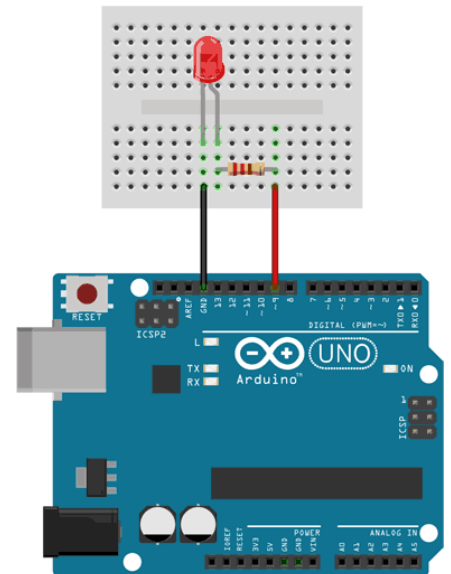
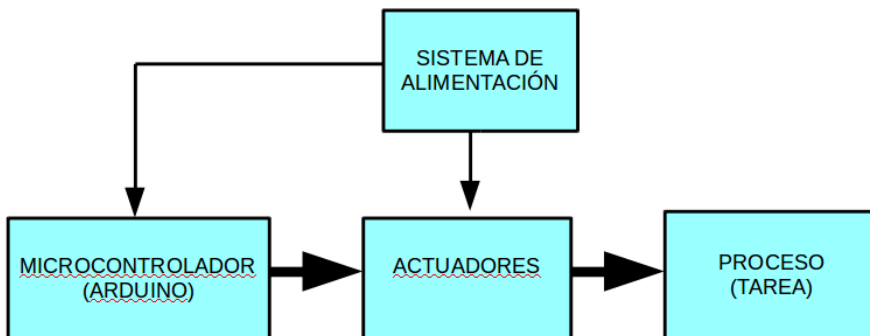
- Elementos de proceso o control: son los que se van a programar, recibirán datos de los elementos de entrada para decidir que realizar con los elementos de salida. En nuestro caso es *Arduino*.
- Elementos de entrada: serán sensores que se le pueden conectar a nuestro sistema de control, pueden ser: pulsadores, interruptores, finales de carrera, ntc, ldr, etc.
- Elementos de salida: van a ser los actuadores, los que va a decidir nuestro programa poner en funcionamiento o parar, como: diodos leds, zumbadores, motores, etc.



3. Tipos de sistemas de control

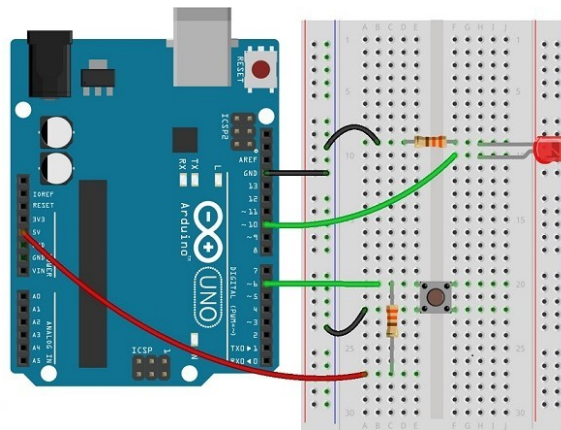
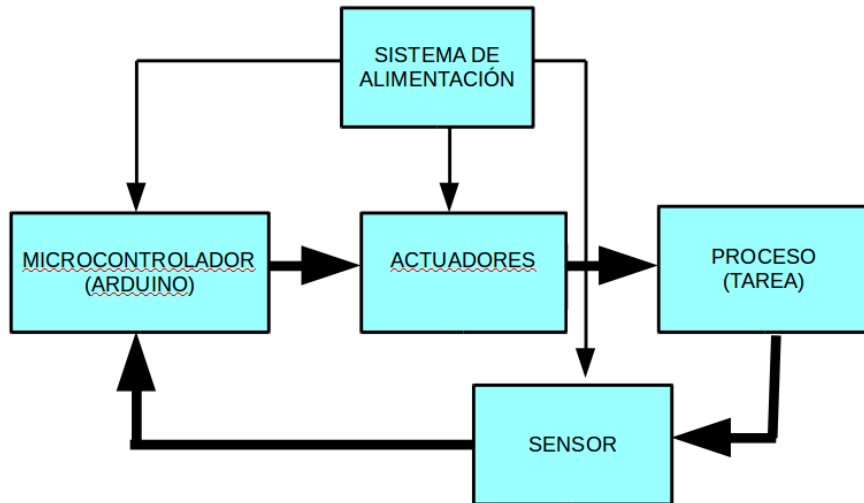
Lazo Abierto

No hay una realimentación, lo que significa el controlador da una orden de puesta en marcha a los actuadores sin que se vigile en ningún momento la señal de los sensores.



Lazo Cerrado

hay realimentación, lo que quiere decir que el sistema de control siempre está vigilando los sensores para realizar en función del estado de los mismos una acción u otra.



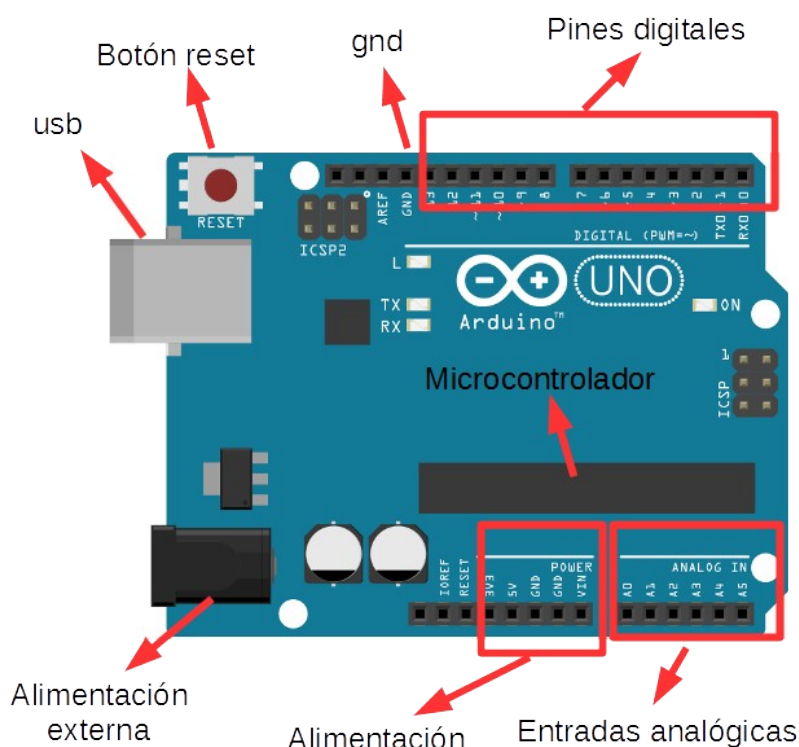
3. Elementos de control programado.

En nuestro caso vamos a utilizar una placa de control programable que se llama **Arduino**, la cual tiene un uso muy extendido por varias razones:

- Hardware libre: los diseñadores han puesto a disposición de todo el mundo los circuitos físicos de la placa para poder fabricarla.
- Software libre: el software que controla la placa también está puesto a disposición de todo el mundo para compartirlo e incluso modificarlo.
- Comunidad: hay una gran cantidad de personas compartiendo proyectos, código y soluciones por internet.
- Precio: muy económico.
- Simplicidad: se pueden realizar soluciones de una manera sencilla.



3.1. Partes de arduino.



Pines digitales (0 a 13, 5V 40mA) (6 Salidas PWM 8 bits): algo es digital cuando sólo puede tomar dos valores (1 o 0, encendido o apagado). Sirven para conectar tanto salidas (actuadores) digitales (diodos led, motores, zumbadores, etc), como entradas (sensores) digitales (interruptores, pulsadores, finales de carrera, etc). Estos pines cuando se usan como salidas podemos decirles que estén conectados o no, que tengan corriente (5V) o no (0V). Cuando se usan como entrada detectan cuando los elementos de entrada están cerrados o no. Cuando son de salida dan como máximo 5V y 40 mA.

Pines de alimentación o power: tenemos los pines gnd (ground tierra), sirven para cerrar el circuito, son el polo negativo de nuestra alimentación. Tenemos un gnd en la fila de los pines digitales y dos gnd más en la zona de alimentación (power). Además tenemos pines de alimentación de 5V, 3.3V, y Vin.

Entradas analógicas (A0 a A5, 0-5V 10 bits): sirven para introducir en el sistema datos que no toman los valores 1 o 0 (digitales), pueden tomar muchos valores, como temperaturas (con ntc), luz (con ldr), etc.

Microcontrolador (ATMEGA 328P, 8 bit, 16Mhz, 32KB flash, 1KB EEPROM, 2KB SRAM): es el dispositivo que recibe el programa y hace que se ejecute, así como controla todos los elementos de la placa. Sólo tiene almacenado un programa en memoria que se sobrescribe cuando se le sube otro.

Usb: es un conector usb que sirve para dar alimentación a la placa y además comunicarse con el ordenador.

Alimentación externa: conector que sirve para alimentar arduino con un elemento externo y que no se alimente por el PC (cargador, pila de 9v, etc). Soporta de 7 a 12V.

Botón de reset: cuando se pulsa reinicia el programa que tiene en memoria.

4. Estructura de programa

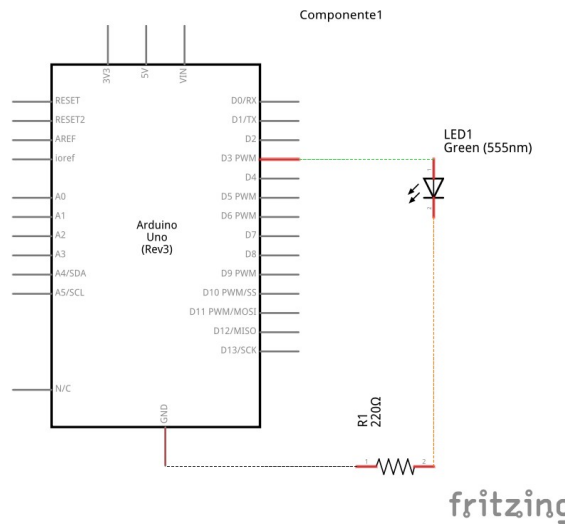
```
int variable;//zona de variables antes de setup
```

```
void setup() { //se pone el código de lo que quiero que se ejecute una vez por ejemplo pinMode (pin, MODO)
  instrucciones a ejecutar una sólo vez
}
```

```
void loop() { //se pone el código de lo que quiero que se ejecute indefinidamente
  instrucciones a ejecutar en bucle infinito
}
```

5. Trabajar con Salidas digitales

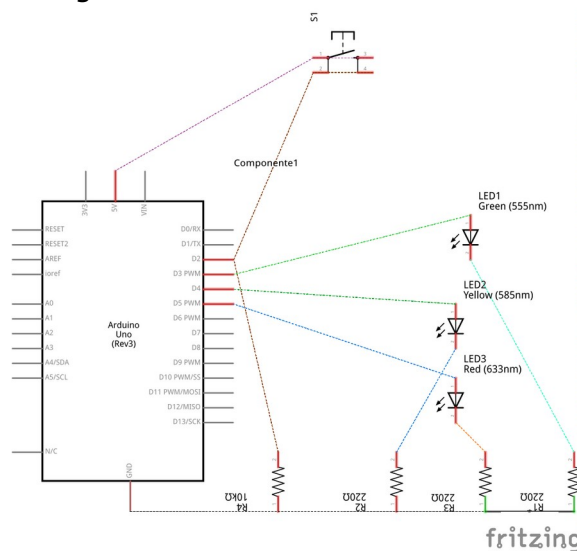
- Pines 0 a 13.
- Conectar LEDs a pines digitales



- Órdenes: pinMode (OUTPUT, pin), digitalWrite (pin,HIGH/LOW), delay (tiempo).

6. Trabajar con Entradas digitales

- Pines de 0 a 13.
- Conectar pulsador a entradas digitales



- Órdenes: pinMode (INPUT, pin), digitalRead (pin)

7. Órdenes de control de flujo programa

bucles:

for (valor inicial de condición;condición de salida;acción sobre condición) {instrucciones;}

while (condición a cumplir) {instrucciones;}

condicionales:

if (condición) {instrucciones;}

else {instrucciones;}

switch (expresion)

```
{
    case valor1:
        instrucciones;
        break;
    case valor2:
        instrucciones;
        break;
    default:
        instrucciones;
        break;
}
```

}

8. Prácticas realizadas

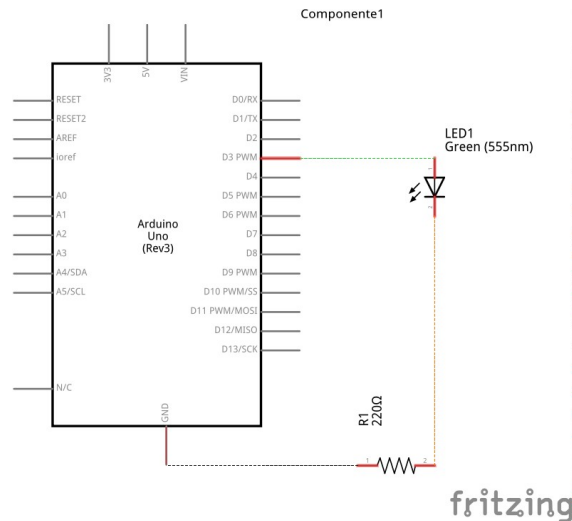
Reto 1: LED parpadeante

El reto consiste en iluminar de manera intermitente un led utilizando cualquier pin digital de Arduino, en nuestro caso vamos a usar el 3, que edubasica lo provee de un led verde. El led estará encendido durante 500ms y que se apague 100ms y así de forma ininterrumpida. Si se utiliza el pin 13 no hace falta resistencia en otros casos sí.

Objetivos:

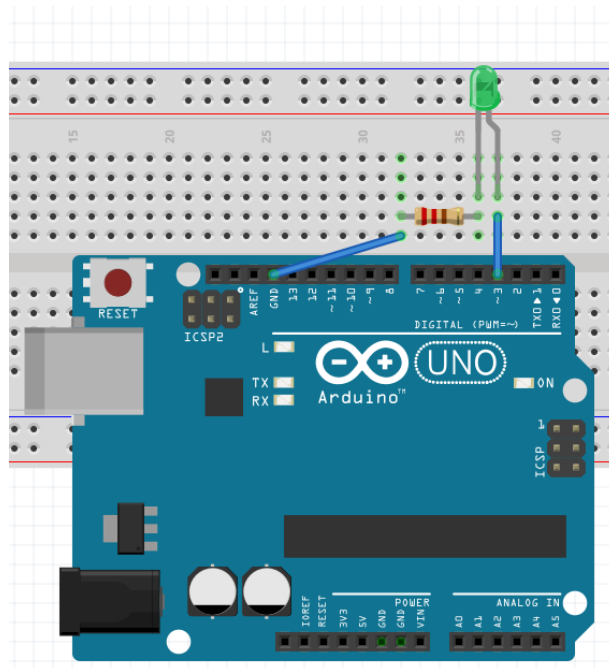
- Familiarizarse con el entorno de programación.
- Reconocer las partes de un programa de Arduino.
- Conocer órdenes como: pinMode, digitalWrite y delay.

Esquema



Conexionado

La forma de conexión será la que se muestra en la figura.



Código fuente

reto01 §

```
void setup() { //comienza la configuracion
  pinMode(3, OUTPUT); //configura el pin 3 o el que queramos como salida
} //termina la configuracion

void loop() { //comienza el bucle principal del programa
  digitalWrite(3, HIGH); //envia 5V al pin (salida) 3
  delay (500); //espera 500 ms pin 3 con 5V
  digitalWrite(3, LOW); //envia 0V al pin (salida) 3
  delay (100); //espera 100 ms pin 3 con 0V
}
```

Reto 2: Secuencia de leds

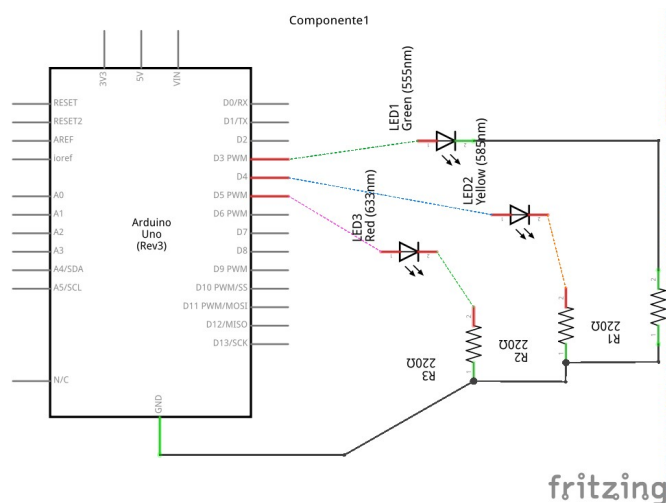
Se trata de encender y apagar 3 leds secuencialmente. Los leds deben estar conectados a los pines 3, 4 y 5 (los pines de los leds de edubasica). Tanto el tiempo de encendido y apagado será de 200 milisegundos.

Nota: en una segunda solución la secuencia principal del programa debe estar reproducida en una función a la que llamará el programa principal.

Objetivos:

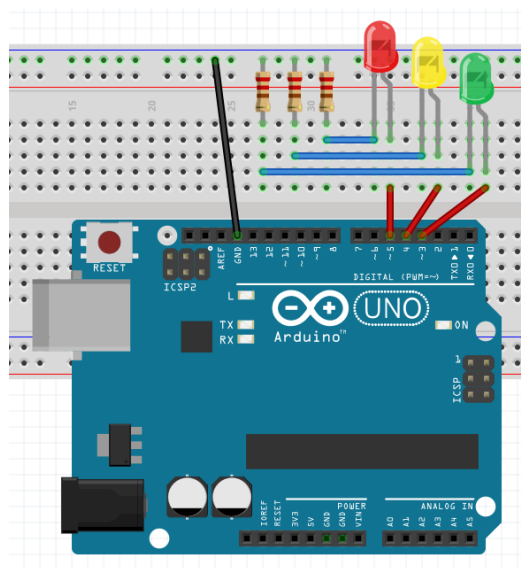
- Familiarizarse con el entorno de programación.
- Estructura de control for.
- Escribir funciones.

Esquema



Conexionado

La forma de conexión será la que se muestra en la figura:



Código fuente

```

reto02_bs
int tiempo = 200;

void setup() {
  for (n = 3; n < 6; n++) {

    pinMode (n, OUTPUT);
  }

}

void loop() {

  for (n = 3; n < 6; n++) {
    digitalWrite (n, HIGH);
    delay (tiempo);
    digitalWrite (n, LOW);
    delay (tiempo);
  }

}

```

Reto 3: Coche fantástico

Se trata de encender y apagar 3 leds secuencialmente. Los leds deben estar conectados a los pines 3, 4, y 5, que son los leds que provee edubasica.

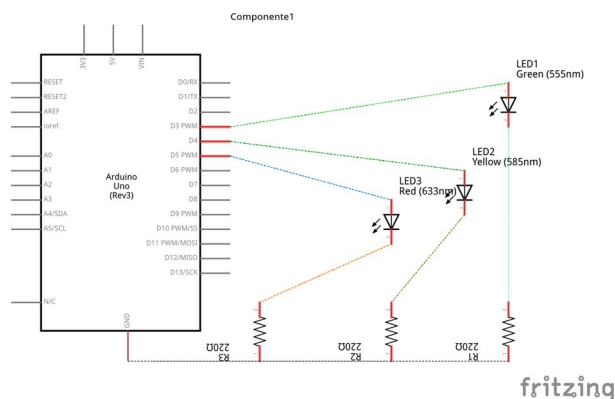
Se deben encender y apagar los leds desde el pin 3 al 5, con un tiempo de encendido y apagado de 50 ms, más tarde se deben encender y apagar los leds desde el pin 5 al 3, con un tiempo de encendido y apagado de 50 ms. La secuencia se debe repetir indefinidamente.

El efecto del programa es el de las luces delanteras de nuestro querido "Coche fantástico".

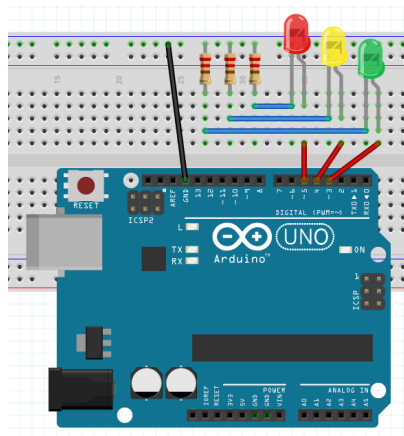
Objetivos:

- Familiarizarse con el entorno de programación.
- Repasar declaración de variables tipo lista de valores.
- Repasar órdenes de control de programa como: for.

Esquema



Conexionado



Código Fuente

Solución 1:

```
reto05_b
int n = 0;
int tiempo = 50;

void setup() { //comienza la configuración
  for (n = 3; n < 6; n++) {
    pinMode(n, OUTPUT);
  }
}

void loop() {
  for (n = 3; n < 6; n++) {
    digitalWrite (n, HIGH);
    delay(tiempo);
    digitalWrite (n, LOW);
    delay(tiempo);
  }
  for (n = 5; n >= 3; n--) {
    digitalWrite (n, HIGH);
    delay(tiempo);
    digitalWrite (n, LOW);
    delay(tiempo);
  }
}
```

Reto 4: SOS con zumbador

Se trata de un zumbador que en código morse (pitidos largos/cortos) especifica una palabra, en nuestro caso SOS. Para el que no lo sepa, la S son tres señales acústicas de corta duración y la O tres señales acústica de larga duración.

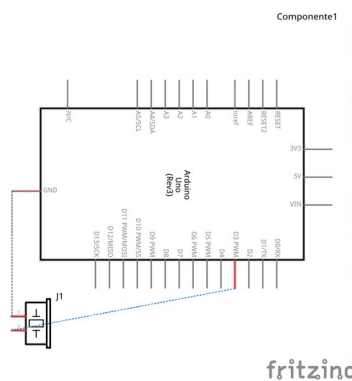
El zumbador debe estar conectado al pin 3, aunque para no ser ruidosos se puede hacer con el led asociado al pin 3 de edubasica (verde), los pitidos o destellos cortos tendrán una duración de 100 ms y los largos 300 ms. Entre letra y letra debe pasar un tiempo de 300 ms y entre SOS debe haber un tiempo de 1000 ms.

Nota: Debes usar variables para guardar los tiempos que vas a usar.

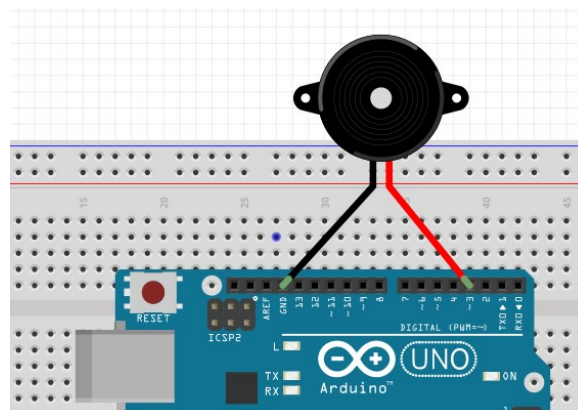
Objetivos:

- Repaso a bucle for.
- Repaso a funciones.
- Introducción del zumbador.
- Resolución y comprobación de reto.

Esquema



Conexionado



Código fuente

```

reto04_b
int tcorto = 100;
int tlargo = 300;
int pausa = 300;
int espera = 1000;
int n = 0;

void setup() { //comienza la configuracion
  pinMode(3, OUTPUT);
}

void s() { //comienza el bucle para la letra S
  for (n = 0; n < 3; n++) {
    digitalWrite (3, HIGH);
    delay (tcorto);
    digitalWrite (3, LOW);
    delay (tcorto);
  }
}

void o() { //comienza el bucle para la letra O
  for (n = 0; n < 3; n++) {
    digitalWrite (3, HIGH);
    delay (tlargo);
    digitalWrite (3, LOW);
    delay (tlargo);
  }
}

void loop() { //se ejecuta el bucle principal en el orden siguiente
  s();
  delay (pausa);
  o();
  delay (pausa);
  s();
  delay (espera);
}

```

Reto 5: Secuencia de leds con pulsador

Se trata de encender y apagar 3 leds secuencialmente al accionar un pulsador. El pulsador debe estar conectado al pin 2 (al que está conectado el pulsador de edubasica), y los leds a los pines 3, 4 y 5 (los de edubasica).

Se deben encender y posteriormente apagar los leds desde el pin 3 al 5, con un tiempo de duración de encendido y apagado de 200 milisegundos.

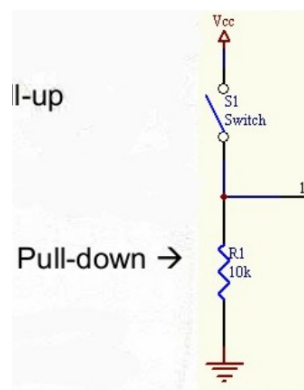
Nota: la secuencia principal del programa debe estar reproducida en una función a la que llamará el programa principal.

Objetivos:

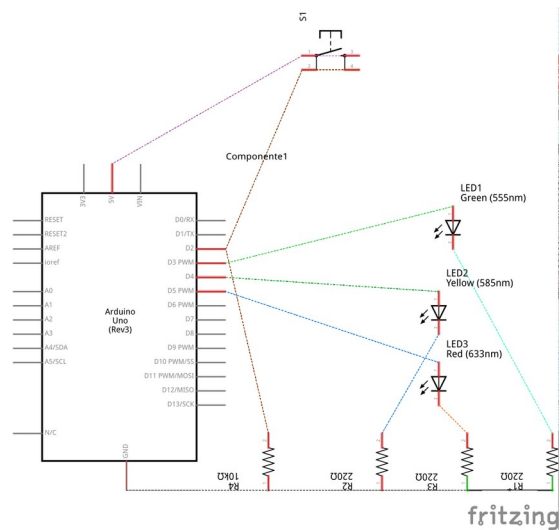
- Aprender a conectar una entrada digital a arduino (pulsador). Divisor de tensión
- Conexiones Pull-down y Pull-up.
- Conocer órdenes como: digitalWrite.
- Conocer órdenes de control de programa como: if.
- Aprender a enviar y visualizar datos por puerto serie.

Esquemas

El esquema eléctrico del pulsador será:

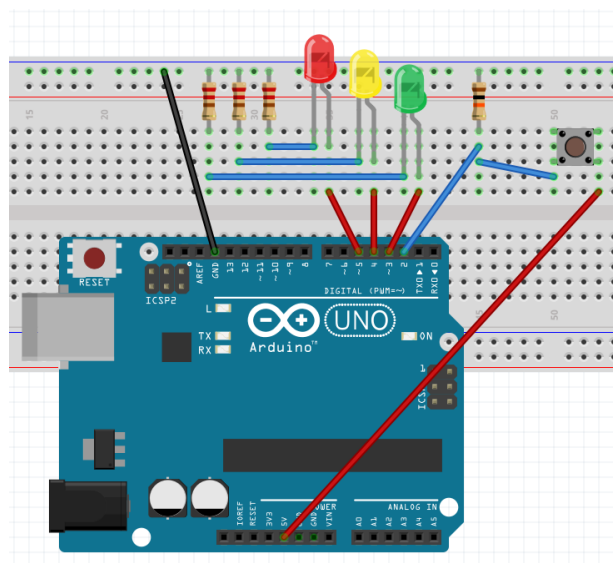


Siendo el esquema del conjunto el siguiente:



Conexionado

Conexionado real con pulsador:



Código fuente

Solución 1

```
reto06_a5
int pulsador = 2;
int tiempo = 200;
int n;

void setup() {
  for (n = 3; n < 6; n++) {
    pinMode (n, OUTPUT);
  }
  pinMode (pulsador, INPUT);
}

void flash() {
  for (n = 3; n < 6; n++) {
    digitalWrite (n, HIGH);
    delay(tiempo);
    digitalWrite (n, LOW);
    delay(tiempo);
  }
}

void loop() {
  if (digitalRead (pulsador) == HIGH) {
    flash();
  }
}
```

Reto 6: Ruleta de la fortuna

Se trata de tres leds que se van encendiendo y apagando formando una secuencia, el jugador debe dar al pulsador cuando el led intermedio se enciende, si acierta funciona un zumbador y la velocidad de la secuencia aumenta.

Los leds deben estar conectados de los pines 3 a 5 (los de edubasica), el zumbador al pin 7, el pulsador al pin 2 (el de edubasica).

El tiempo inicial entre encendido y encendido de leds debe ser 200 ms, si se acierta el tiempo disminuye en 20 ms, si el tiempo entre encendidos llegase a 10 ms, se devuelve el tiempo a 200 ms.

Objetivos:

- Conocer órdenes: if, &&, ||
- Repaso a uso de funciones
- Repaso a variables tipo lista
- Repaso a entradas digitales

Esquema y conexionado

Como el ejercicio anterior

Código Fuente

```
reto_06_d
int n = 0;
int tiempo = 200;
int zumbador = 8;//puede ser led azul
int pulsador = 2;

void setup () {
  for (n = 3; n < 8; n++) {
    pinMode(n, OUTPUT);
  }
  pinMode(zumbador, OUTPUT);
  pinMode(pulsador, INPUT);
}
void compruebaacierto() {
  if (digitalRead(pulsador) == HIGH && n == 5) {
    digitalWrite(zumbador, HIGH);
    delay (1000);
    digitalWrite(zumbador, LOW);
    tiempo = tiempo - 20;
    if (tiempo < 10) {
      tiempo = 200;
    }
  }
  if (digitalRead(pulsador) == HIGH && n != 5) {
    for (n = 3; n < 8; n++) {
      digitalWrite(n, HIGH);
    }
    delay(100);
    for (n = 3; n < 8; n++) {
      digitalWrite(n, LOW);
    }
    delay (100);
    tiempo = 200;
  }
}
void loop () {
  for (n = 3; n < 8; n++) {
    digitalWrite(n, HIGH);
    delay(tiempo);
    compruebaacierto();
    digitalWrite(n, LOW);
    delay(tiempo);
  }
}
```

Por: Pedro Ruiz Fernández

