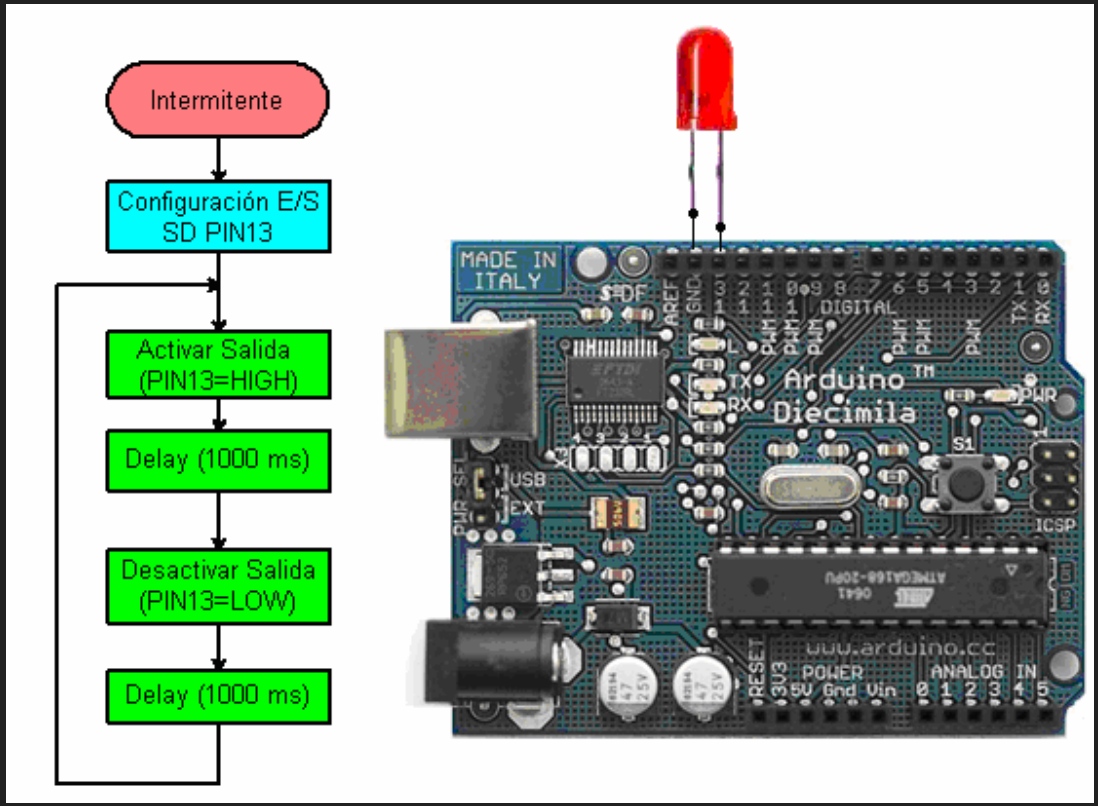
Ejercicios Arduino

Facultad de Estadística e Informática

03 / Julio / 2017

Ejercicio 1: Prender LED

Se trata de realizar un ejercicio básico que consiste en encender y a pagar un led que conectamos en el PIN 13 de Arduino que lo configuramos como salida.



Ejercicio: Encender LED

```
int ledPin = 13;                 // Definición de la salida en el PIN 13
void setup()                     //Configuración
{
        pinMode(ledPin, OUTPUT);      // designa la salida digital al PIN 13
}

void loop()                      // bucle de funcionamiento
{
        digitalWrite(ledPin, HIGH);    // activa el LED
         delay(1000);                  // espera 1 seg. (tiempo encendido)
}
```

Obsérvese que se ha colocado el diodo LED, sin resistencia en serie dado que el PIN13 de Arduino ya lleva incorporada una resistencia interior, en el caso de colocar el diodo LED en otra salida deberíamos colocar una resistencia de al entre 220 y 500 ohmios dependiendo del consumo de corriente del diodo

Ejercicio: Apagar LED

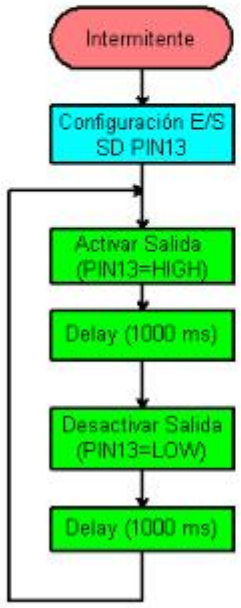
```
int ledPin = 13;                 // Definición de la salida en el PIN 13
void setup()                     //Configuración
{
  pinMode(ledPin, OUTPUT);       // designa la salida digital al PIN 13
}

void loop()                      // bucle de funcionamiento
{
 digitalWrite(ledPin, LOW);      // desactiva el LED
  delay(1000);                   // espera 1 seg. (tiempo apagado)
}
```

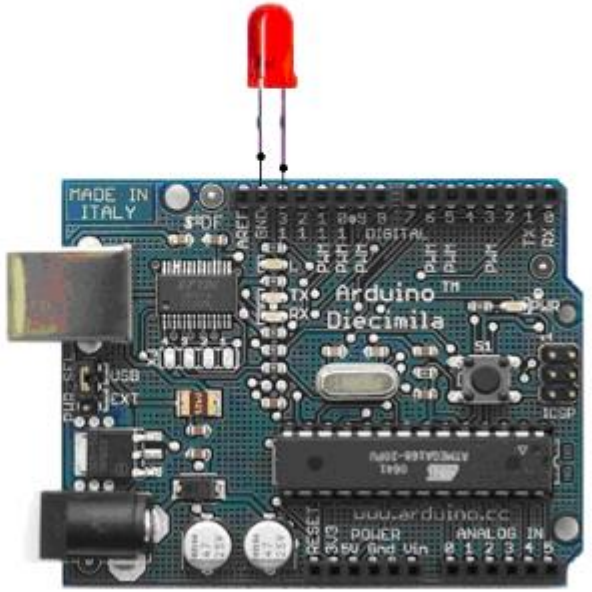
Obsérvese que se ha colocado el diodo LED, sin resistencia en serie dado que el PIN13 de Arduino ya lleva incorporada una resistencia interior, en el caso de colocar el diodo LED en otra salida deberíamos colocar una resistencia de al entre 220 y 500 ohmios dependiendo del consumo de corriente del diodo

Ejercicio: Intermitente

Se trata de realizar un ejercicio básico que consiste en encender y a pagar un led que conectamos en el PIN 13 de Arduino que lo configuramos como salida. El tiempo de encendido y apagado es de 1 segundo.

Organigrama y Esquema



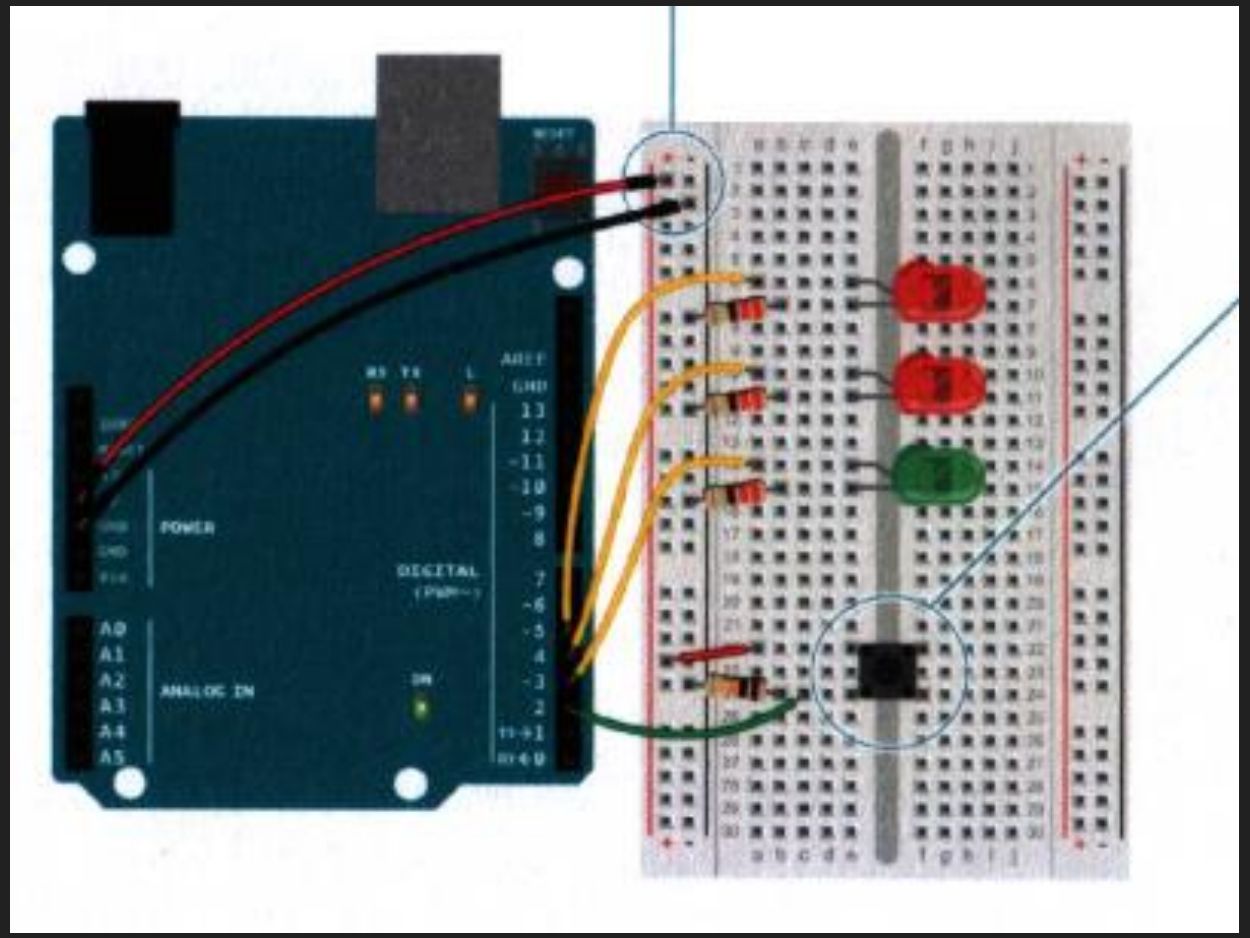


Ejercicio: Intermitente

Programa

```
/*
 * Intermitente
 *
 * Ejemplo básico con Arduino. Encendido y apagado de un led
 * con una cadencia de 1 sg. usando el PIN 13 como salida
 * no es necesario usar una resistencia para el led
 * la salida 13 de Arduino la lleva incorporada.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */
int ledPin = 13;                // Definición de la salida en el PIN 13
void setup()                    //Configuración
{
  pinMode(ledPin, OUTPUT);      // designa la salida digital al PIN 13
}

void loop()                      // bucle de funcionamiento
{
  digitalWrite(ledPin, HIGH);   // activa el LED
  delay(1000);                  // espera 1 seg. (tiempo encendido)
  digitalWrite(ledPin, LOW);    // desactiva el LED
  delay(1000);                  // espera 1 seg. (tiempo apagado)
}
```

Ejercicios: Pulsador



Ejercicios: Pulsador

```
1  int switchState = 0;

2  void setup(){
3    pinMode(3,OUTPUT);
4    pinMode(4,OUTPUT);
5    pinMode(5,OUTPUT);
6    pinMode(2,INPUT);
7  }

8  void loop(){
9    switchState = digitalRead(2);
10   // this is a comment
```

Ejercicios: Pulsador

```
11    if (switchState == LOW) {
12    // the button is not pressed
13      digitalWrite(3, HIGH); // green LED
14      digitalWrite(4, LOW);  // red LED
15      digitalWrite(5, LOW);  // red LED
16    }

17    else {  // the button is pressed
18      digitalWrite(3, LOW);
19      digitalWrite(4, LOW);
20      digitalWrite(5, HIGH);
```

Ejercicios: Pulsador

```
21      delay(250);  // wait for a quarter second
22      // toggle the LEDs
23      digitalWrite(4, HIGH);
24      digitalWrite(5, LOW);
25      delay(250); // wait for a quarter second

26    }
27  } // go back to the beginning of the loop
```

¿= o ==?

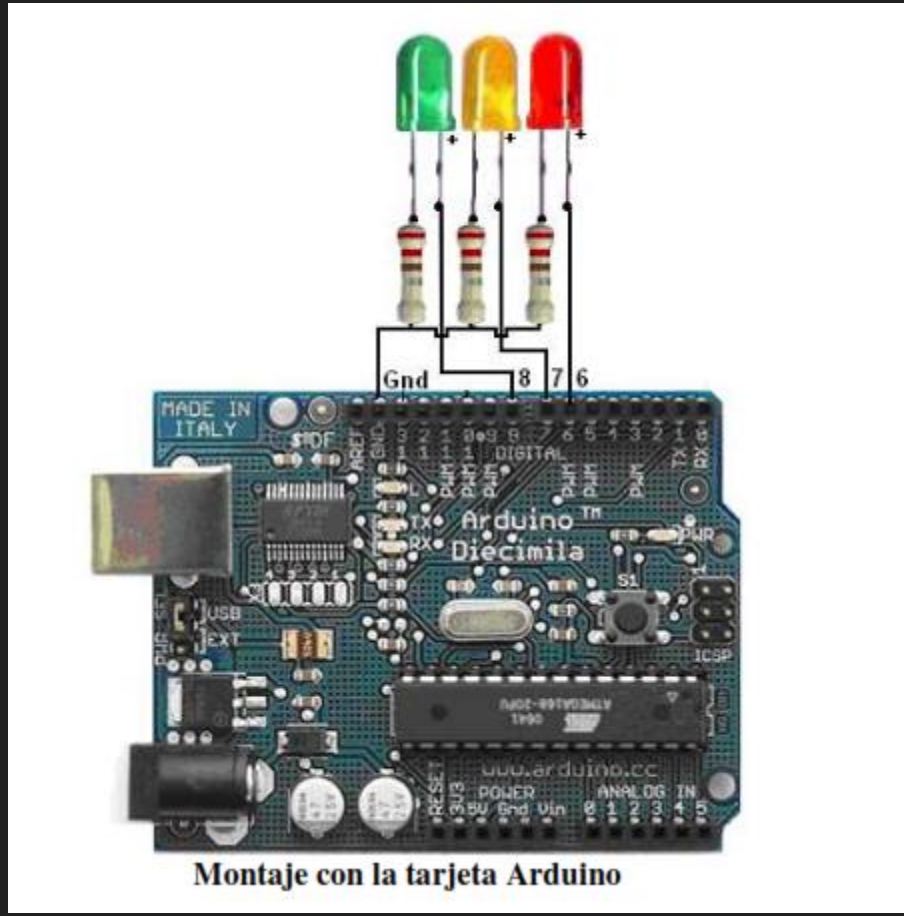
Variable = 1; → Asignación

Se utiliza para darle el valor de la derecha del = a la variable.

Variable == 1 → Comparación

Se utiliza para verificar si el valor de la variable es igual a el valor del lado derecho de los ==.

Ejercicio: Semaforo



Montaje con la tarjeta Arduino

Ejercicio: Semaforo

Programa

```
// Encendido y apagado de 3 LEDs

int ledPin1 = 6; // Define las salidas de los LED´s
int ledPin2 = 7;
int ledPin3 = 8;

void setup() { // Configura las SALIDAS
pinMode(ledPin1, OUTPUT); // declarar LEDs como SALIDAS
pinMode(ledPin2, OUTPUT);
```

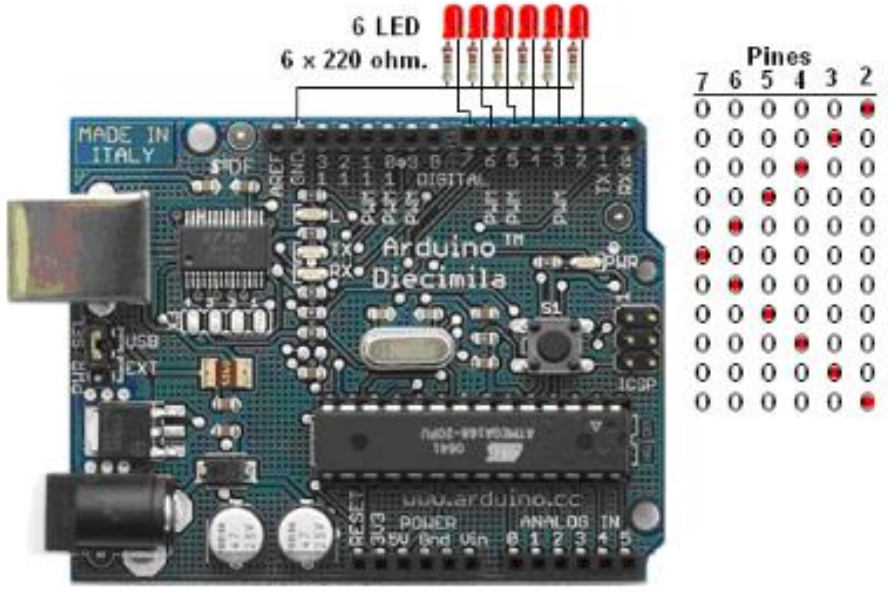
Ejercicio: Semaforo

```
pinMode(ledPin3, OUTPUT);
digitalWrite(ledPin1, LOW); // Apaga los LEDs
digitalWrite(ledPin2, LOW);
digitalWrite(ledPin3, LOW);
}

void loop(){ //Bucle de Funcionamiento
digitalWrite(ledPin1, HIGH); // Apaga y enciende los leds cada 200 ms
delay(200);
digitalWrite(ledPin1, LOW);
digitalWrite(ledPin2, HIGH);
delay(200);
digitalWrite(ledPin2, LOW);
digitalWrite(ledPin3, HIGH);
delay(200);
digitalWrite(ledPin3, LOW);
}
```

Ejercicio: Luces Auto Increíble

Elementos necesarios:

- 6 LED-s.
- 6 resistencias de 220 Ohmios.
- Una placa protoboard.
- Cables para realizar las conexiones



Esquema.

Ejercicio: Luces Auto Increíble

```
int pin2 = 2;          // PIN-es de los LED
int pin3 = 3;
int pin4 = 4;
int pin5 = 5;
int pin6 = 6;
int pin7 = 7;
int timer = 100;       // Temporizador

void setup(){

  pinMode(pin2, OUTPUT);  // Configuración de los PIN-es como salida
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  pinMode(pin5, OUTPUT);
  pinMode(pin6, OUTPUT);
  pinMode(pin7, OUTPUT);

}
```

Ejercicio: Luces Auto Increíble

```
void loop() {
  digitalWrite(pin2, HIGH); // Enciende y apaga secuencialmente LED-s
  delay(timer);
  digitalWrite(pin2, LOW);
  delay(timer);
  digitalWrite(pin3, HIGH);
  delay(timer);
  digitalWrite(pin3, LOW);
  delay(timer);
  digitalWrite(pin4, HIGH);
  delay(timer);
  digitalWrite(pin4, LOW);
  delay(timer);
  digitalWrite(pin5, HIGH);
  delay(timer);
  digitalWrite(pin5, LOW);
  delay(timer);
  digitalWrite(pin6, HIGH);
  delay(timer);
  digitalWrite(pin6, LOW);
  delay(timer);
  digitalWrite(pin7, HIGH);
  delay(timer);
  digitalWrite(pin7, LOW);
  delay(timer);
  digitalWrite(pin6, HIGH);
```

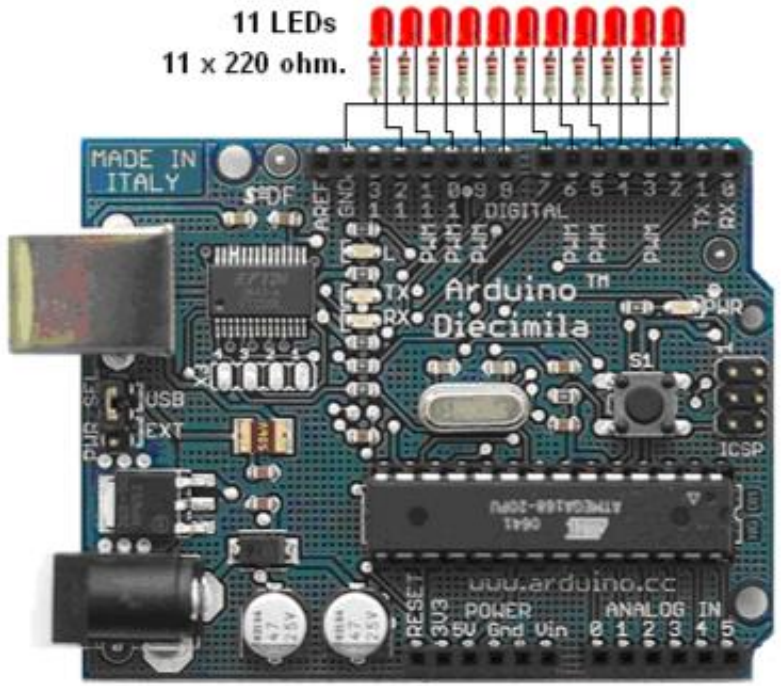
Ejercicio: Luces Auto Increíble

```
  delay(timer);
  digitalWrite(pin6, LOW);
  delay(timer);
  digitalWrite(pin5, HIGH);
  delay(timer);
  digitalWrite(pin5, LOW);
  delay(timer);
  digitalWrite(pin4, HIGH);
  delay(timer);
  digitalWrite(pin4, LOW);
  delay(timer);
  digitalWrite(pin3, HIGH);
  delay(timer);
  digitalWrite(pin3, LOW);
  delay(timer);
}
```

Ejercicio: Estrella Fugaz

Elementos necesarios:

- 11 LED-s.
- 11 resistencias de 220 Ohmios.
- Una placa protoboard.
- Cables para realizar las conexiones.



Esquema

Ejercicio: Estrella Fugaz

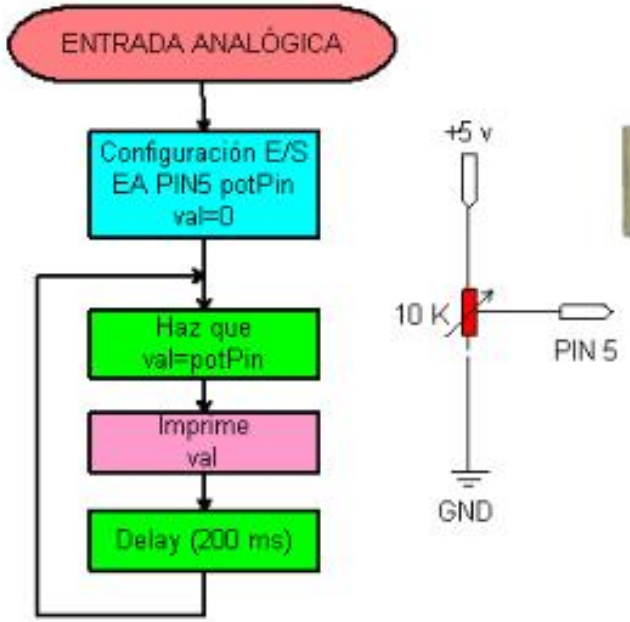
```
int pinArray [] = { 2,3,4,5,6,7,8,9,10,11,12 };
int controlLed = 13;          // LED de control
int waitNextLed = 100;        // Tiempo antes de encender el siguiente LED
// Número de LED-s que permanecen encendidos antes de empezar a apagarlos para
//formar la cola
int tailLength = 4;
 // Número de LED-s conectados (que es también el tamaño del array)
int lineSize = 11;

void setup()          // Configuración de los PIN-es como salida digital
{
  int i;
  pinMode (controlLed, OUTPUT);
  for (i=0; i< lineSize; i++)
  {
    pinMode(pinArray[i], OUTPUT);
  }
}
```

Ejercicio: Estrella Fugaz

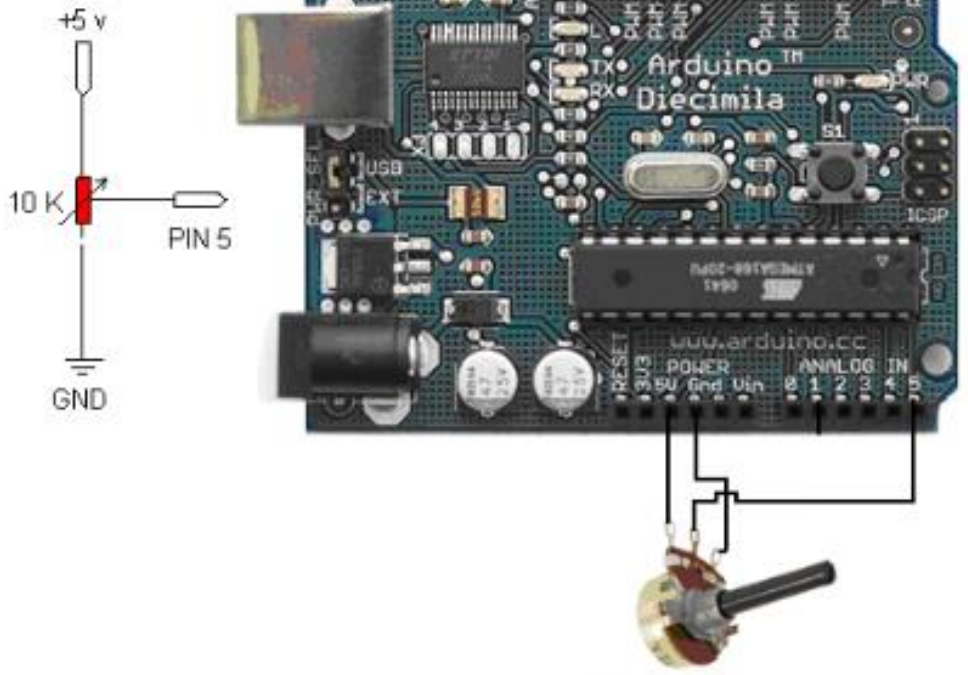
```
void loop()
{
  int i;
// Se establece la longitud de la cola en un contador
  int tailCounter = tailLength;
// Se enciende el LED de control para indicar el inicio del loop
  digitalWrite(controlLed, HIGH);
   for (i=0; i<lineSize; i++)
 {
   digitalWrite(pinArray[i],HIGH);    // Se encienden consecutivamente los L
 // Esta variable de tiempo controla la velocidad a la que se mueve la estrella
   delay(waitNextLed);
   if (tailCounter == 0)
    {
 // Se apagan los LED-s en función de la longitud de la cola.
 digitalWrite(pinArray[i-tailLength],LOW);
    }
   else
     if (tailCounter > 0)
       tailCounter--;
 }
   for (i=(lineSize-tailLength); i<lineSize; i++)
 {
   digitalWrite(pinArray[i],LOW);              // Se apagan los LED
// Esta variable de tiempo controla la velocidad a la que se mueve la estrella
   delay(waitNextLed);
 }
}
```

Ejercicio: Entrada Analógica

Se trata de configurar un canal de entrad analógico pin 5 y enviar el valor leído al PC para visualizarlo



Organigrama



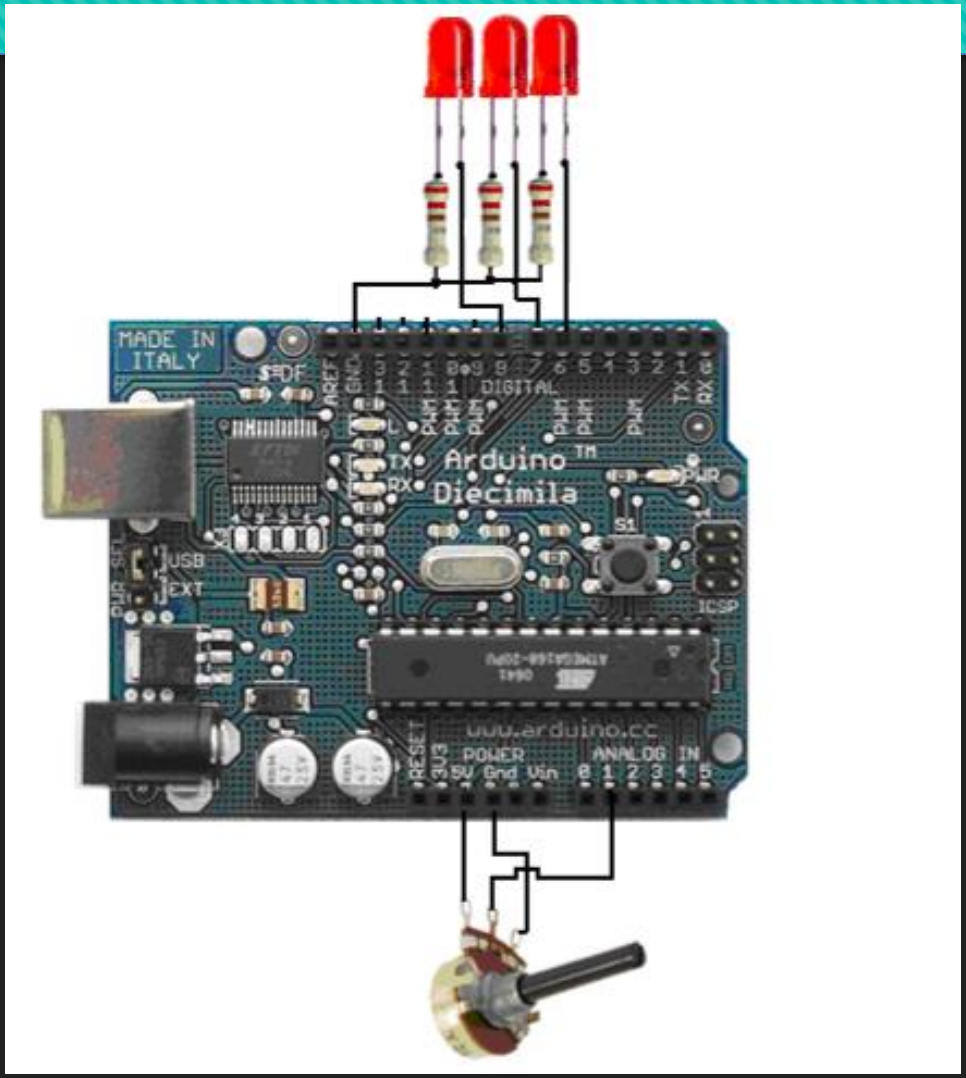
Esquema

Ejercicio: Entrada Analógica

Programa

```
/* Entrada Analógica  */

int potPin = 5;    // selecciona el pin de entrada para colocar el potenciómetro
int val = 0;       // variable para almacenar el valor leído por la entrada analógica

void setup() {
  beginSerial(9600);
}

void loop() {

  val = analogRead(potPin);    // lee el valor del canal de ENTRADA analógica
  printInteger(val); // Envía al PC el valor analógico leído y lo muestra en pantalla
  serialWrite(10);
  delay(100);
}
```

Ejercicio: Entrada Analógica



Ejercicio: Entrada Analógica

```
int ledPin1 = 8; // Selección de PIN para cada LED
int ledPin2 = 7;
int ledPin3 = 6;
int inPin=A1; // selecciona la entrada analógica 1 (potenciómetro)
```

```
void turn_off() { //Apaga los 3 LEDS
digitalWrite(ledPin1, LOW);
digitalWrite(ledPin2, LOW);
digitalWrite(ledPin3, LOW);
}
void setup() {
pinMode(ledPin1, OUTPUT); // declara LEDs como salidas
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
turn_off();  //
}
```

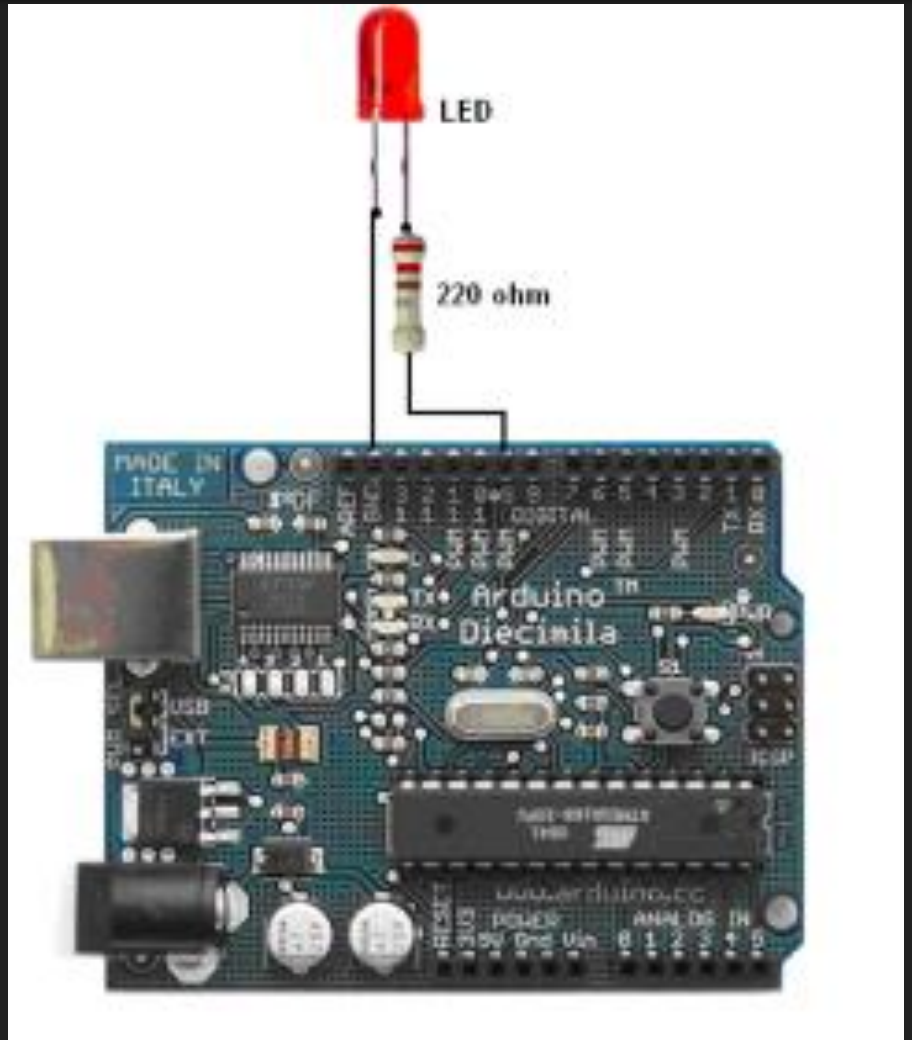
Ejercicio: Entrada Analógica

```
void loop(){
int val;
val= analogRead(inPin); // lee el valor de la señal analógica
turn_off();apaga los tres LED
// Si el valor de la señal medida es > 256 enciende LED del PIN8
if (val>= 256) digitalWrite(ledPin1, HIGH);
// Si el valor de la señal medida es > 512 enciende LED del PIN7
if (val>= 512) digitalWrite(ledPin2, HIGH);
// Si el valor de la señal medida es > 758 enciende LED del PIN6
if (val>= 768) digitalWrite(ledPin3, HIGH);
}
```

Ejercicio: Monitor Serial

- Sirve para observar datos que pasan a través de Arduino.
- NO todo lo que pasa en el Arduino se ve en el monitor serial, a menos que se indique.
- Serial.begin(frecuencia)
- 9600 baudios

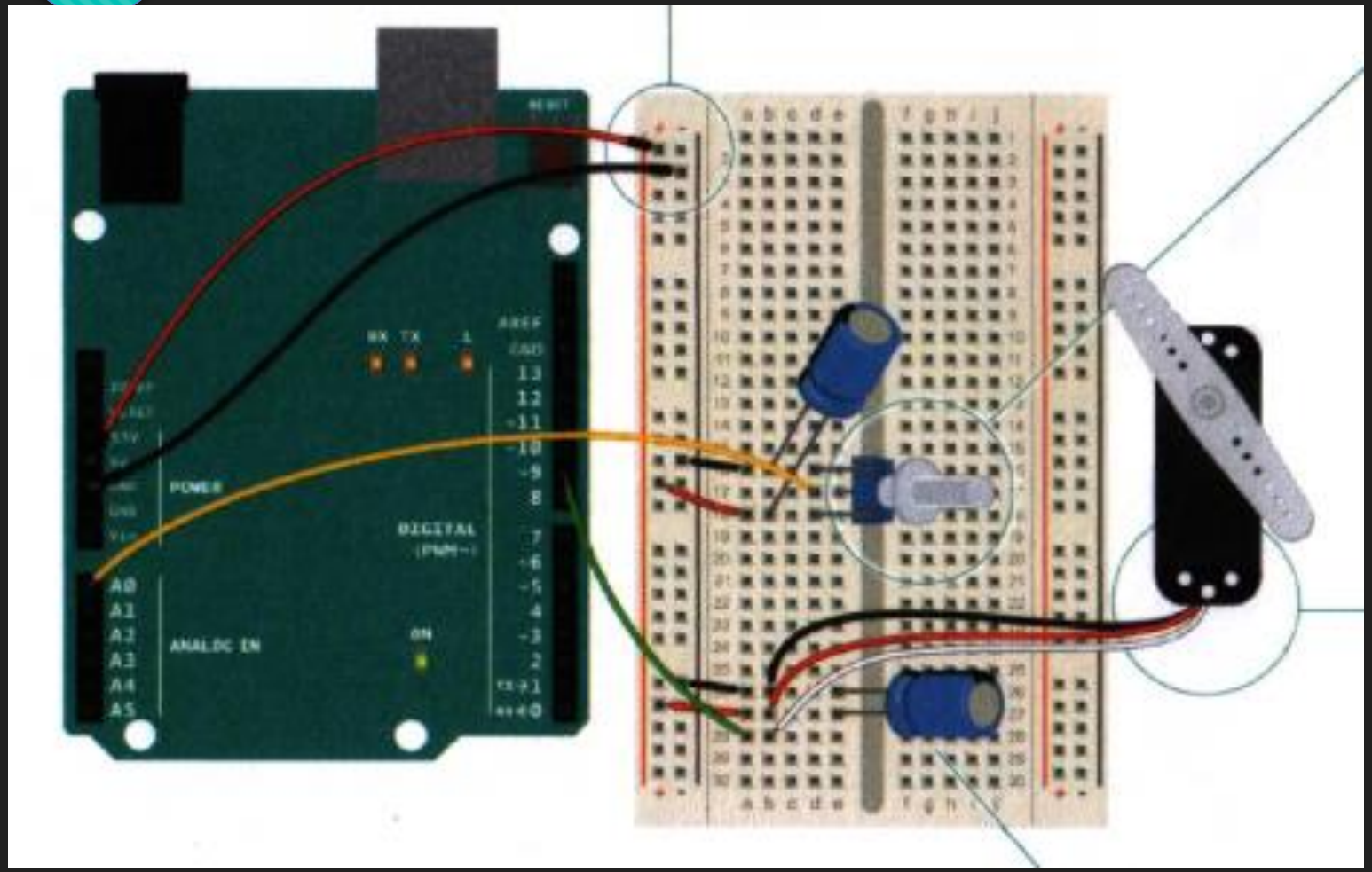
Ejercicio: Salida Analógica



Ejercicio: Salida Analógica

```
int value = 0;                  // Valor a sacar por la salida analógica PIN 9
int ledpin = 9;                 // Salida analógicas PIN 9

void setup()
{
  // nothing for setup
}

void loop()
{
        for(value = 0 ; value <= 255; value+=5) // Variación de la variable se salida
ente el MIN yMAX
  {
        analogWrite(ledpin, value);         // Enviar valor a la salida (entre 0 y 255)
        delay(30);                          // Esperar 30 ms para ver el efecto de variación
  }
        for(value = 255; value >=0; value-=5)   // Variación de la variable de salida
entre MAX y MIN
  {
    analogWrite(ledpin, value);
    delay(30);
  }
}
```

Ejercicio: Movimiento de Cervo



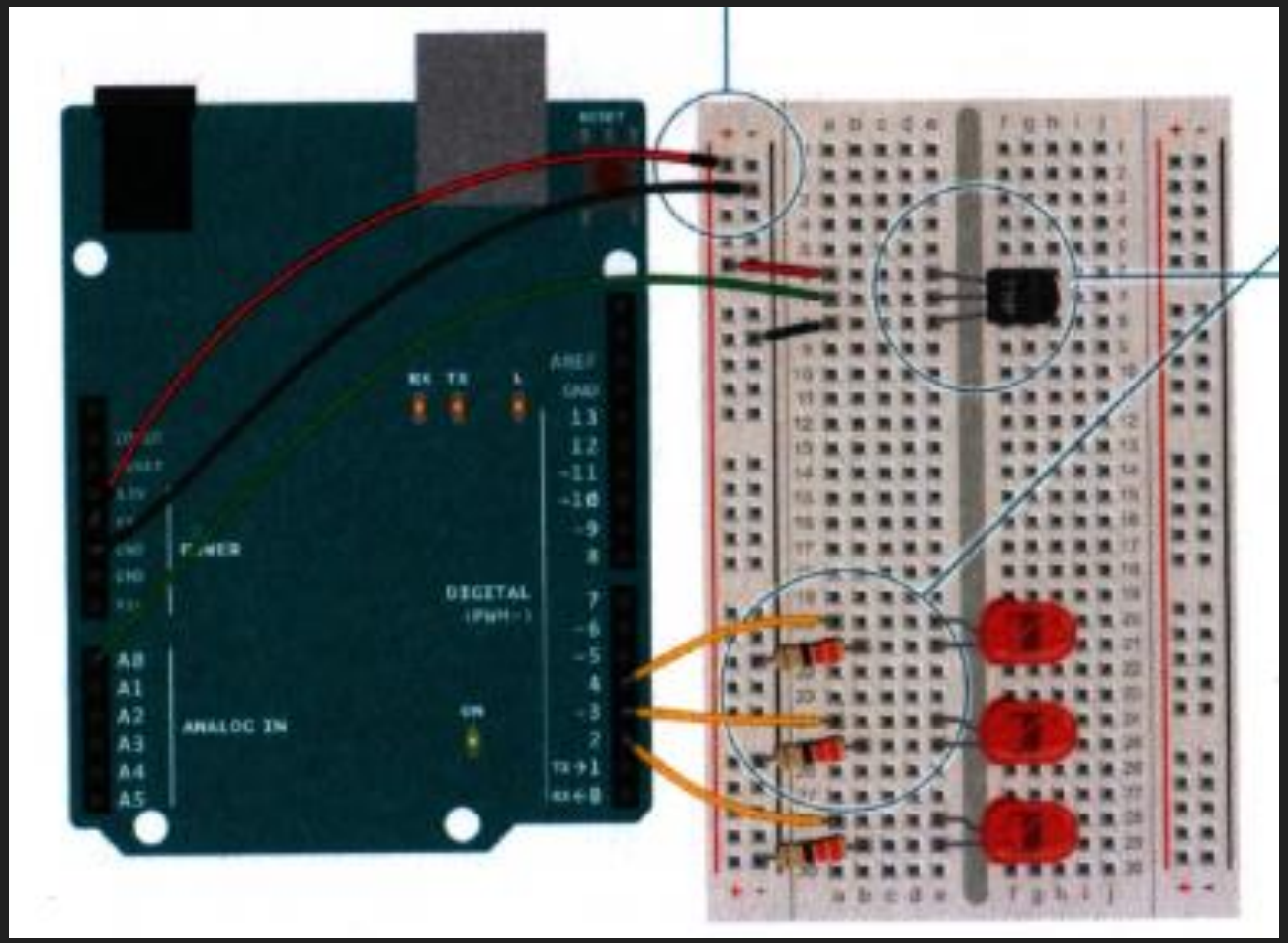
Ejercicio: Movimiento de Cervo

```
1 #include <Servo.h>

2 Servo myServo;

3 int const potPin = A0;
4 int potVal;
5 int angle;

6 void setup() {
7   myServo.attach(9);

8   Serial.begin(9600);
9 }
```

Ejercicio: Movimiento de Cervo

```
10 void loop() {
11   potVal = analogRead(potPin);
12   Serial.print("potVal: ");
13   Serial.print(potVal);

14   angle = map(potVal, 0, 1023, 0, 179);
15   Serial.print(", angle: ");
16   Serial.println(angle);

17   myServo.write(angle);
18   delay(15);
19 }
```

Ejercicio: Medidor de Calor



Ejercicio: Medidor de Calor

```
1 const int sensorPin = A0;
2 const float baselineTemp = 20.0;

3 void setup(){
4   Serial.begin(9600); // open a serial port
```

Ejercicio: Medidor de Calor

```
5    for(int pinNumber = 2; pinNumber<5; pinNumber++
6      pinMode(pinNumber,OUTPUT);
7      digitalWrite(pinNumber, LOW);
8    }
9  }
10 void loop(){
11   int sensorVal = analogRead(sensorPin);
12   Serial.print("Sensor Value: ");
13   Serial.print(sensorVal);
```

Ejercicio: Medidor de Calor

```
14    // convert the ADC reading to voltage
15    float voltage = (sensorVal/1024.0) * 5.0;

16    Serial.print(", Volts: ");
17    Serial.print(voltage);
18    Serial.print(", degrees C: ");
19    // convert the voltage to temperature in degrees
20    float temperature = (voltage - .5) * 100;
21    Serial.println(temperature);
```

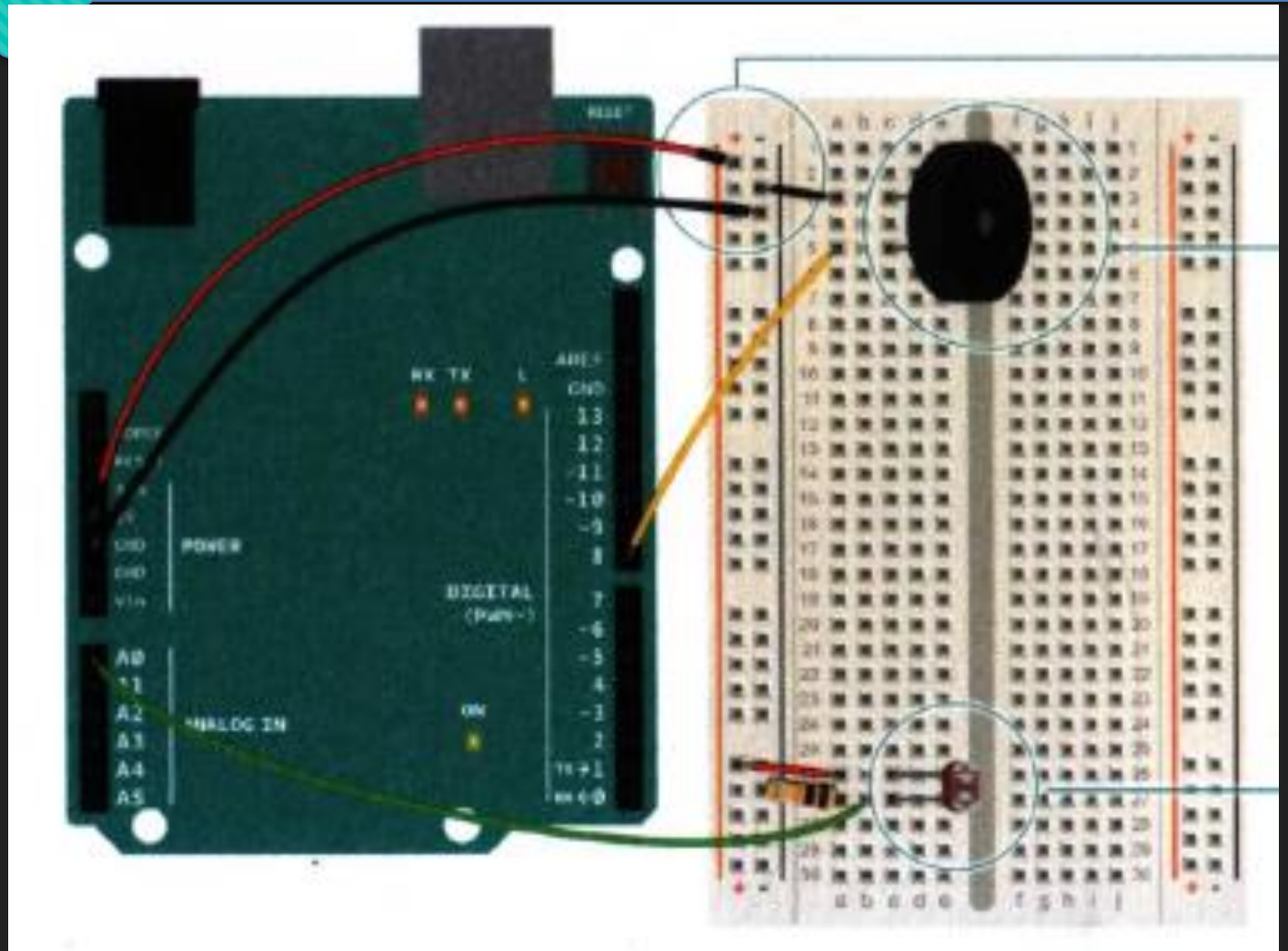
Ejercicio: Medidor de Calor

```
22    if(temperature < baselineTemp){
23      digitalWrite(2, LOW);
24      digitalWrite(3, LOW);
25      digitalWrite(4, LOW);

26    }else if(temperature >= baselineTemp+2 &&
        temperature < baselineTemp+4){
27      digitalWrite(2, HIGH);
28      digitalWrite(3, LOW);
29      digitalWrite(4, LOW);
30    }else if(temperature >= baselineTemp+4 &&
        temperature < baselineTemp+6){
31      digitalWrite(2, HIGH);
32      digitalWrite(3, HIGH);
33      digitalWrite(4, LOW);
```

Ejercicio: Medidor de Calor

```
34    }else if(temperature >= baselineTemp+6){
35      digitalWrite(2, HIGH);
36      digitalWrite(3, HIGH);
37      digitalWrite(4, HIGH);

38    }
39    delay(1);
40  }
```

Ejercicio: Theremin basado en luz



Ejercicio: Theremin basado en luz

```
1 int sensorValue;
2 int sensorLow = 1023;
3 int sensorHigh = 0;
4 const int ledPin = 13;
5 void setup() {

6   pinMode(ledPin, OUTPUT);
7   digitalWrite(ledPin, HIGH);
```

Ejercicio: Theremin basado en luz

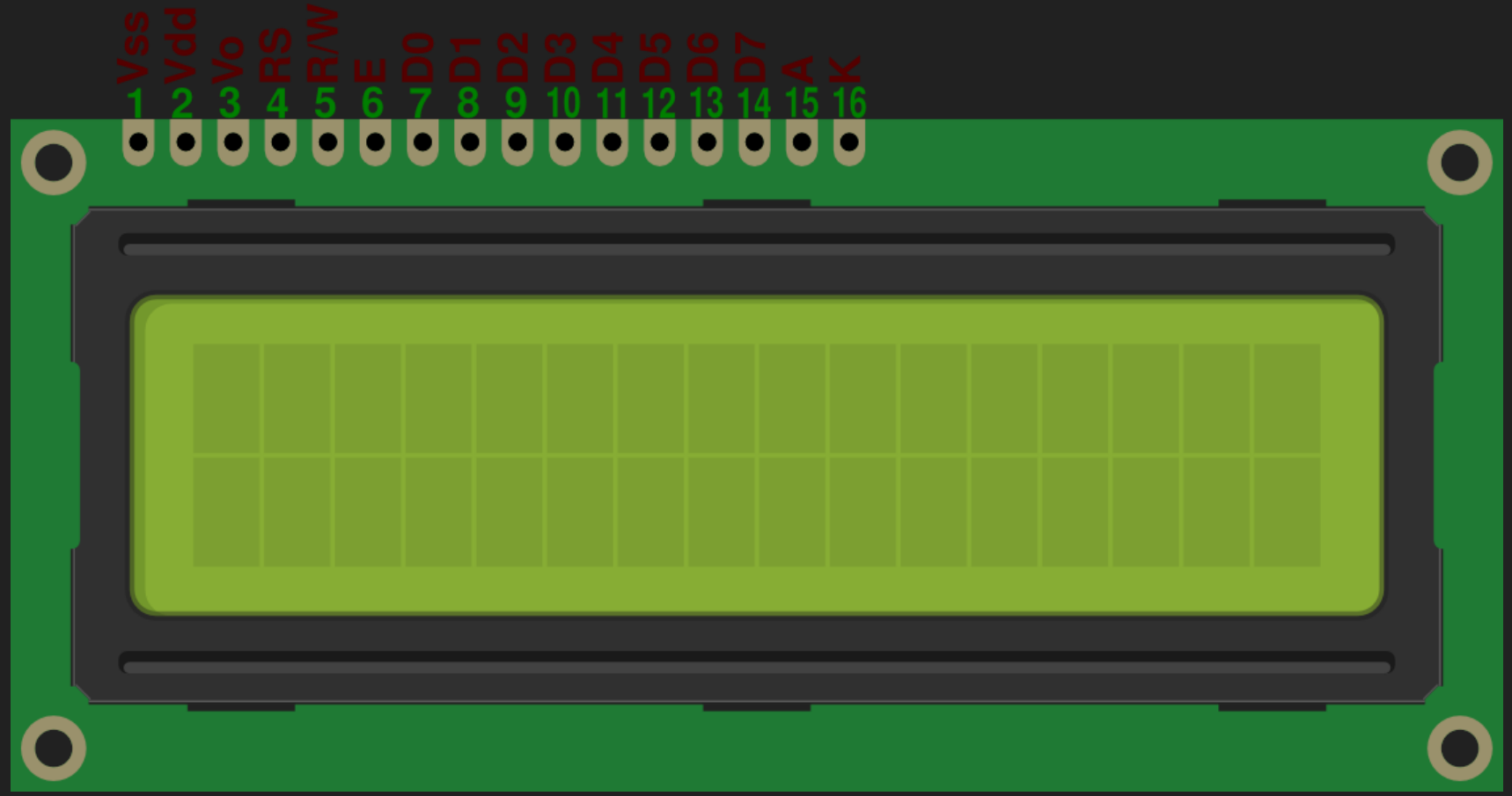
```
  while (millis() < 5000) {
    sensorValue = analogRead(A0);
    if (sensorValue > sensorHigh) {
      sensorHigh = sensorValue;
    }
    if (sensorValue < sensorLow) {
      sensorLow = sensorValue;
    }
  }
  digitalWrite(ledPin, LOW);
}
```

Ejercicio: Theremin basado en luz

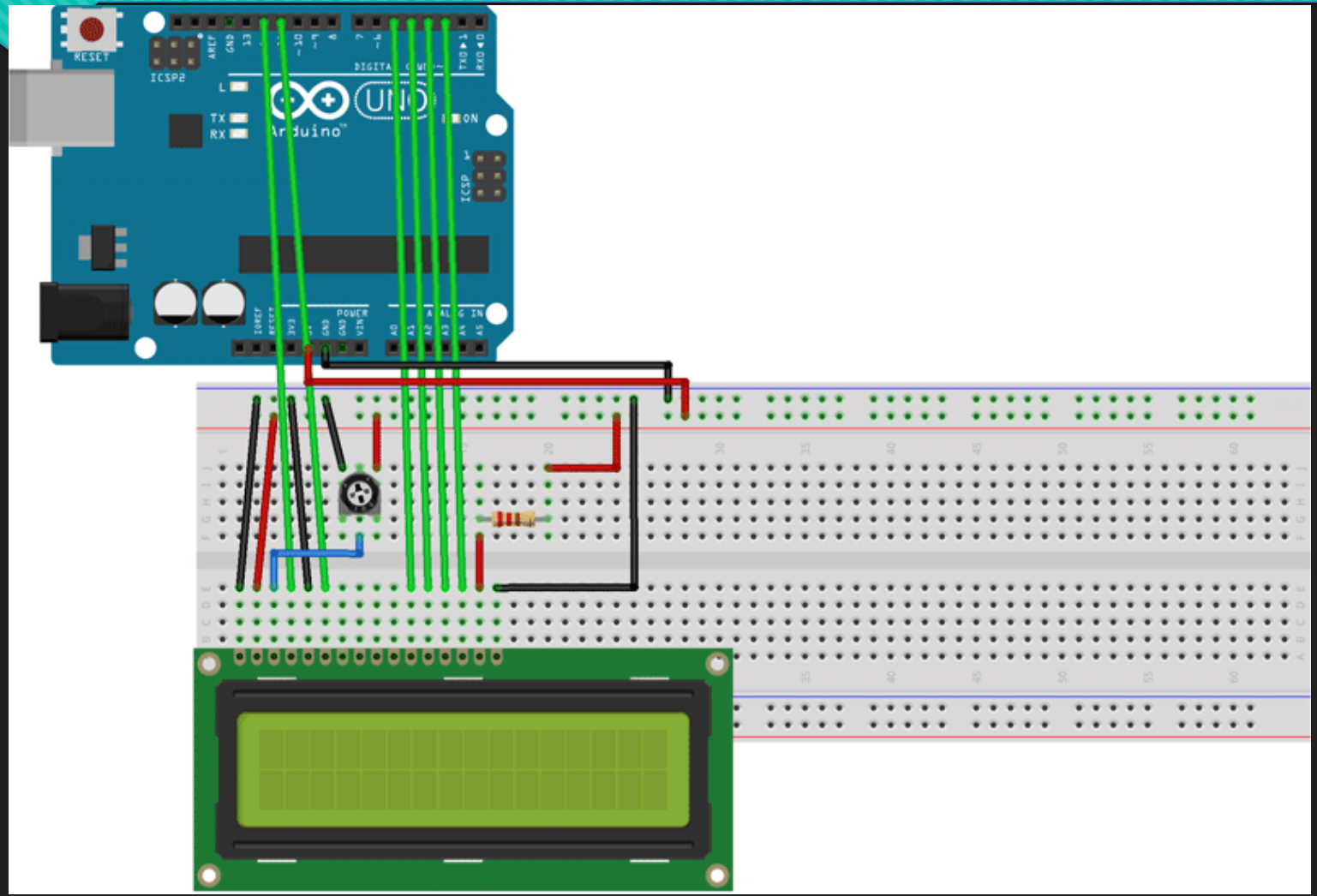
```
19 void loop() {
20   sensorValue = analogRead(A0);
21   int pitch =
       map(sensorValue,sensorLow,sensorHigh, 50, 4000);
22   tone(8,pitch,20);
23   delay(10);
24 }
```

Ejercicio: Pantalla LCD

- LCD, Liquid Crystal Display.
- Es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora.
- Utiliza cantidades pequeñas de energía eléctrica.



Ejercicio: Pantalla LCD



Ejercicio: Pantalla LCD

```
// Incluímos la biblioteca externa para poder utilizarla
#include <LiquidCrystal.h>

// Definimos las constantes
#define COLS 16 // Columnas del LCD
#define ROWS 2 // Filas del LCD

// Lo primero is inicializar la biblioteca indicando los pins de la interfaz
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//Donde 12 es el pin de Arduino donde hemos conectado el pin RW
//el pin 11 es el de sincronización
//y el 5, 4, 3 y 2 son los pines correspondientes a los datos.

void setup() {
    Serial.begin(9600);
    // Configuramos las filas y las columnas del LCD en este caso 16 columnas y 2 filas
    lcd.begin(COLS, ROWS);
}
```

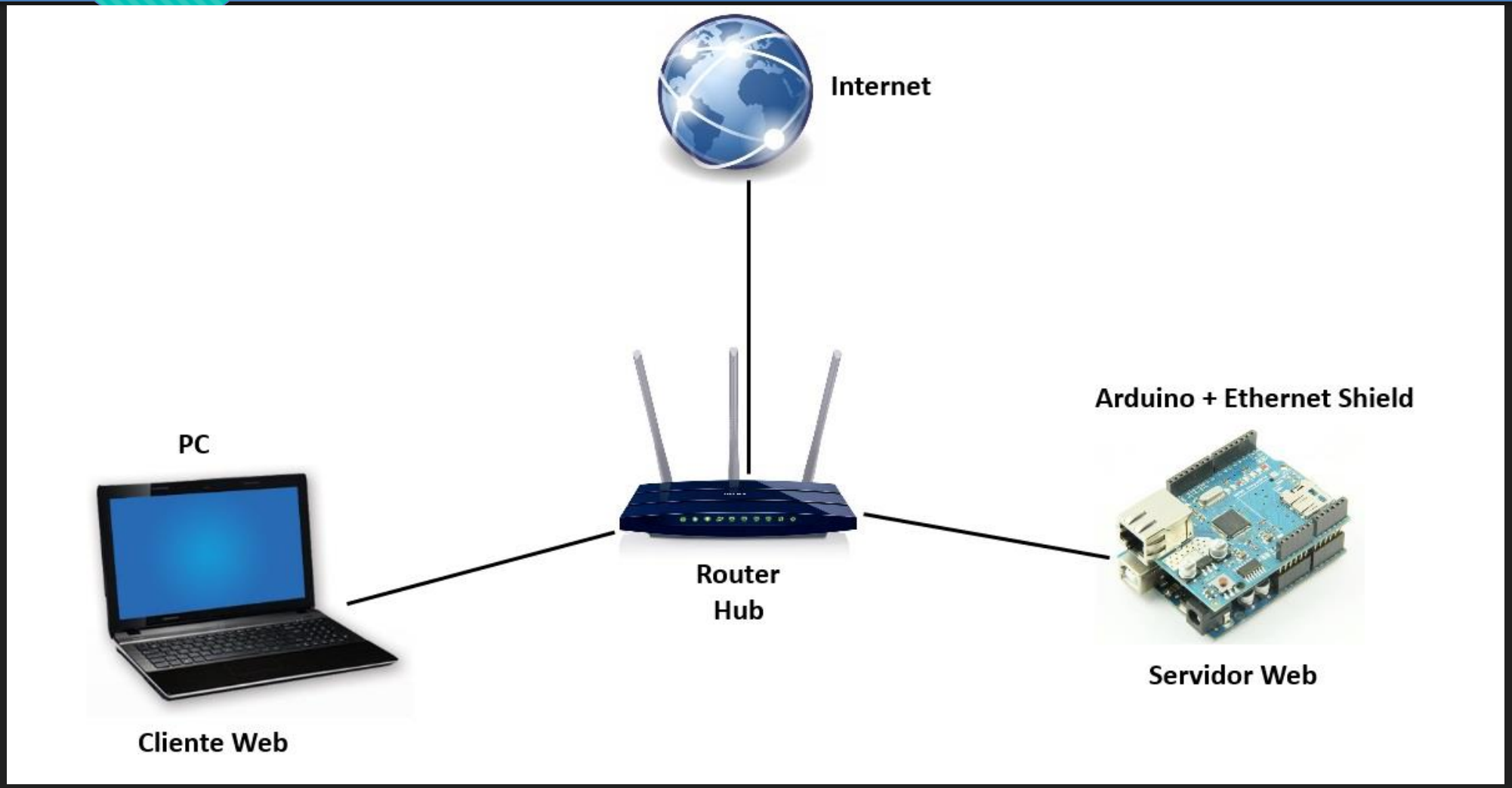
Ejercicio: Pantalla LCD

```
void loop() {
    lcd.clear(); // Se limpia la pantalla
    lcd.setCursor(0,0); // Situamos el cursor en la columna 0 fila 0
    lcd.print("Saludos Humanos."); // Escribimos los datos en la fila 0
    lcd.setCursor(0,1); // Situamos el cursor en la columna 0 fila 1
    lcd.print("Probando el LCD.");// Escribimos en la siguiente fila
    delay(2000);// Esperamos antes de que se repita la función loop
}
```

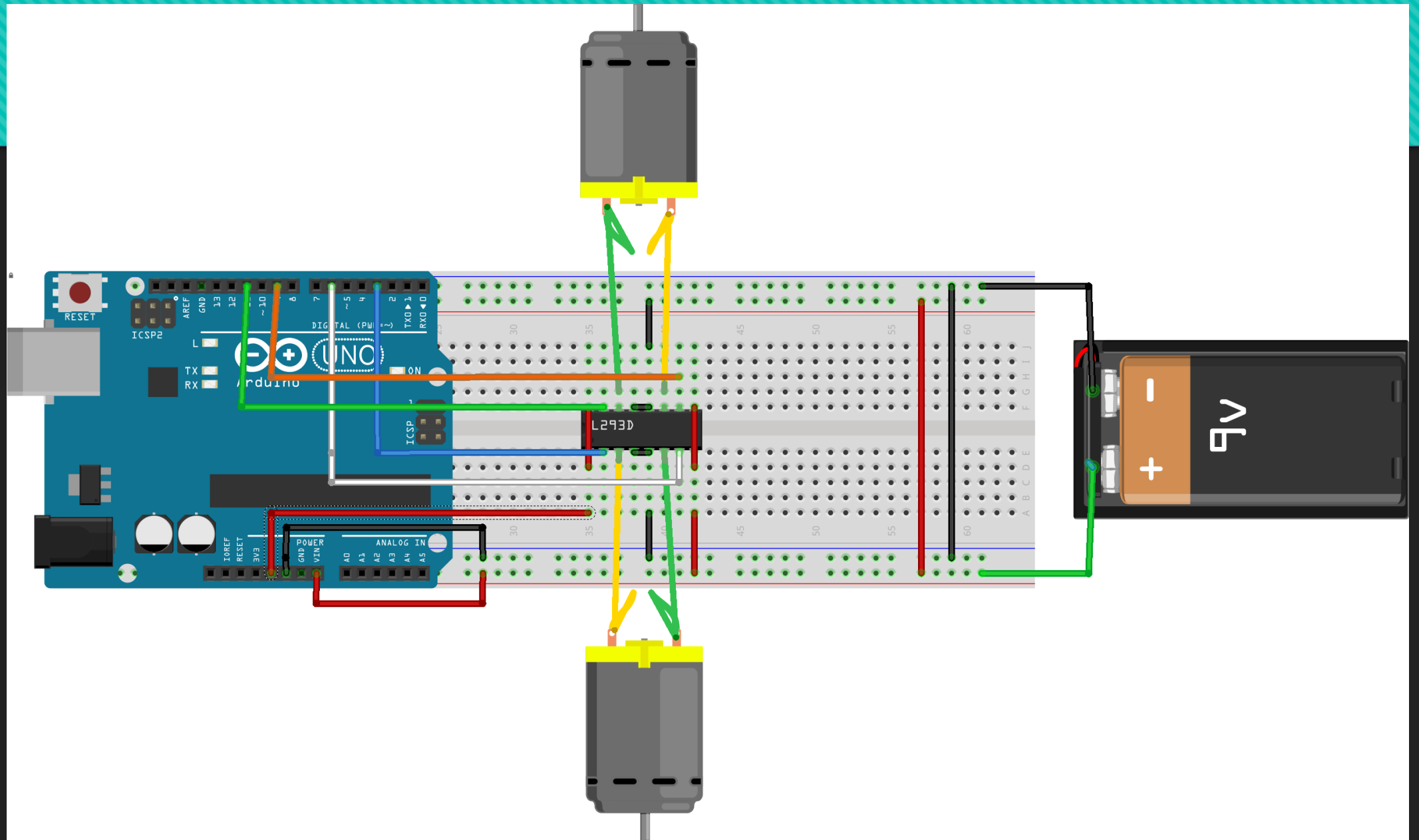
Ejercicio: Comunicación por Red

- Dirección IP.
- Dirección MAC.
- Red privada o pública.
- Telnet.

Ejercicio: Comunicación por Red



Ejercicio: Control de motores para carrito.



Ejercicio: Control de motores.

Ejercicios Arduino

Facultad de Estadística e Informática

2017