

# ARDUINO

## Proyecto final 2º Trimestre

MEDIDOR DE BATERÍA CON ARDUINO

18/02/2019

---

Juan Carlos Durán Caballero

Jose Luis Rubio Alcalde

Jose Antonio del Cueto González

1º DAM FESAC

## Visión general:

En este proyecto, vamos a hacer un **medidor de carga de baterías y pilas**. Lo haremos a través del montaje de un circuito con **Arduino**.

Suele ser muy típico que tengamos por casa diferentes aparatos eléctricos que utilizan pilas o baterías. En ocasiones, dudamos si el dispositivo está roto o las pilas se han gastado. Este proyecto consiste en un aparato que nos ayudará a saber si las pilas o batería que queremos usar tienen o no vida útil.

## Objetivos:

Utilizaremos **Arduino** para leer a través de una **entrada analógica** el voltaje que suministra una pila. Dependiendo de este voltaje, encenderemos un LED de un color. Si la pila está **nueva**, se encenderá un **LED verde**. Si la pila **no es nueva** pero se ha consumido parte de su energía encenderemos un **LED amarillo**. Por último, si la pila está **gastada** o no suministra el suficiente voltaje, encenderemos un **LED rojo**.

## Componentes de arduino a utilizar:

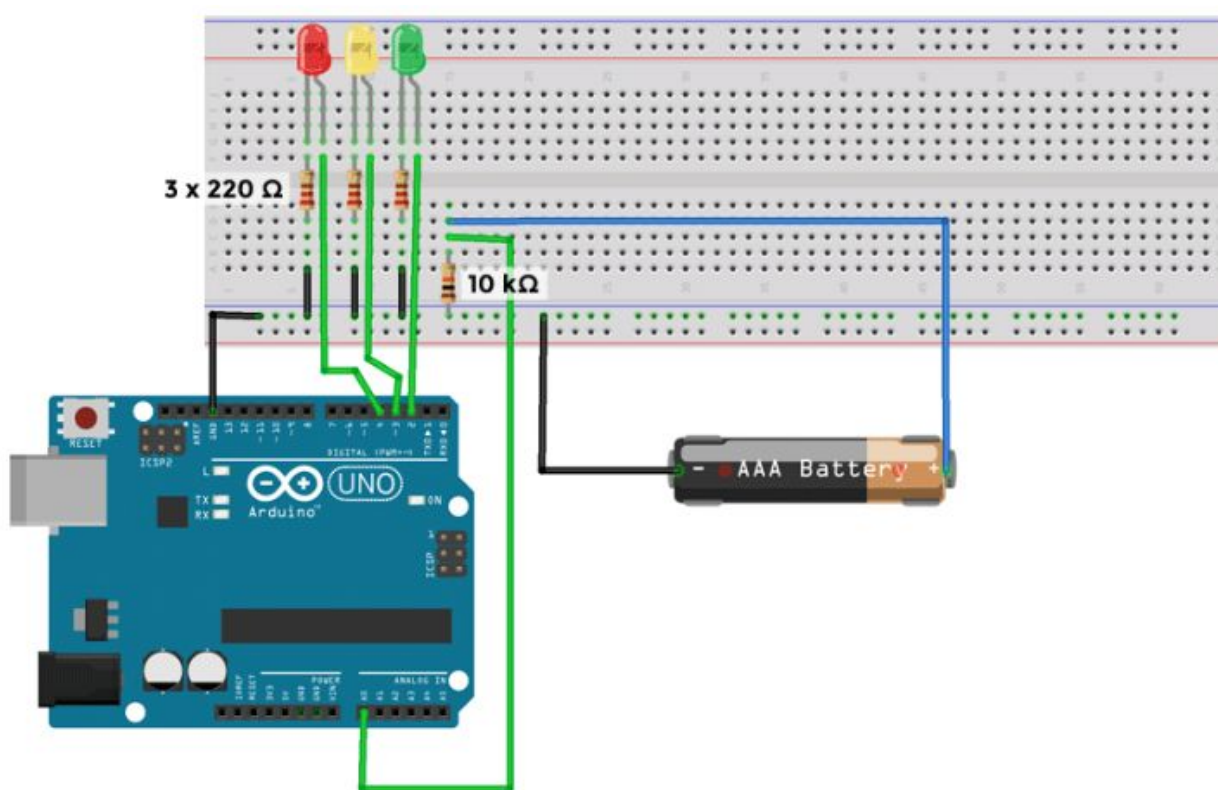
- Arduino UNO o cualquier placa de Arduino
- Protoboard donde conectaremos los componentes
- Cables para la conexión entre los componentes y la placa
- 3 resistencias de 220  $\Omega$
- 1 resistencia de 10 k $\Omega$
- 1 LED rojo de 5 mm
- 1 LED amarillo de 5 mm
- 1 LED verde de 5 mm



Hemos comprado un kit en el cual vienen todos los componente que necesitamos para este proyecto, lo puedes ver en este [enlace](#).

## Hitos

### I. Montaje del circuito con arduino



Lo primero son los LEDs, cada uno está conectado en serie con una resistencia de 220 Ω para alargar la vida útil de los mismos. El **LED verde** está conectado al **pin 2**, el **LED amarillo** está conectado al **pin 3** y **LED rojo** está conectado al **pin 4**.

Para medir la batería hemos colocado una **resistencia pull-down** (que sería la de 10 kΩ).

Este tipo de resistencia lo que hace es **mantener un estado lógico bajo es decir, a 0V**. Es importante utilizar este tipo de resistencias ya que, cuando no tengamos la batería o pila conectada para medir, tenemos un estado

indeterminado a la entrada del pin analógico, lo que hace que oscile y puede que hasta que se enciende algún LED.

El polo positivo de la batería lo conectamos a la resistencia de pull-down y a la entrada analógica A0. El otro extremo de la resistencia a tierra. Por último, el polo negativo de la batería debemos conectarlo a la tierra de Arduino.

## II. Programando el medidor de carga de baterías y pilas con Arduino

Aquí dejamos el **Algoritmo** en pseudocódigo, luego sacaremos el código con la IDE de Arduino:

1. Leer el pin analógico donde tenemos conectada la pila
2. Calculamos el voltaje para el valor que nos ha dado
3. Evaluamos el voltaje
  1. Si es mayor o igual que el umbral máximo
    1. Encendemos LED verde
  2. Si es menor que el umbral máximo y mayor que el umbral medio
    1. Encendemos LED amarillo
  3. Si es menor que el umbral medio y mayor que el umbral mínimo
    1. Encendemos LED rojo
  4. El resto de los casos
    1. No enciende ningún LED
4. Apagamos todos los LEDs

Analizando el algoritmo que vamos a implementar, sacamos como conclusión que vamos a utilizar **3 umbrales**:

- **Umbral máximo:** indicará que la pila está totalmente cargada.
- **Umbral medio:** de este umbral a el umbral máximo la pila se ha usado pero todavía tiene energía.
- **Umbral mínimo:** de este umbral a el umbral medio la pila no suministra suficiente energía. Por debajo de este umbral interpretamos que no hay una pila conectada.

## Aquí dejamos el código, escrito en Arduino:

```
// Variables
int VALOR = 0;
float voltaje = 0;
int ledDelay = 800;

// Umbrales
float maximo = 1.6;
float medio = 1.4;
float minimo = 0.3;
/*
 * 4--> rojo
 * 3--> amarillo
 * 2--> verde
 * 0--> pila
 */
void setup(){
    // Iniciamos el monitor serie
    Serial.begin(9600);

    // Los pines de LED en modo salida
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);

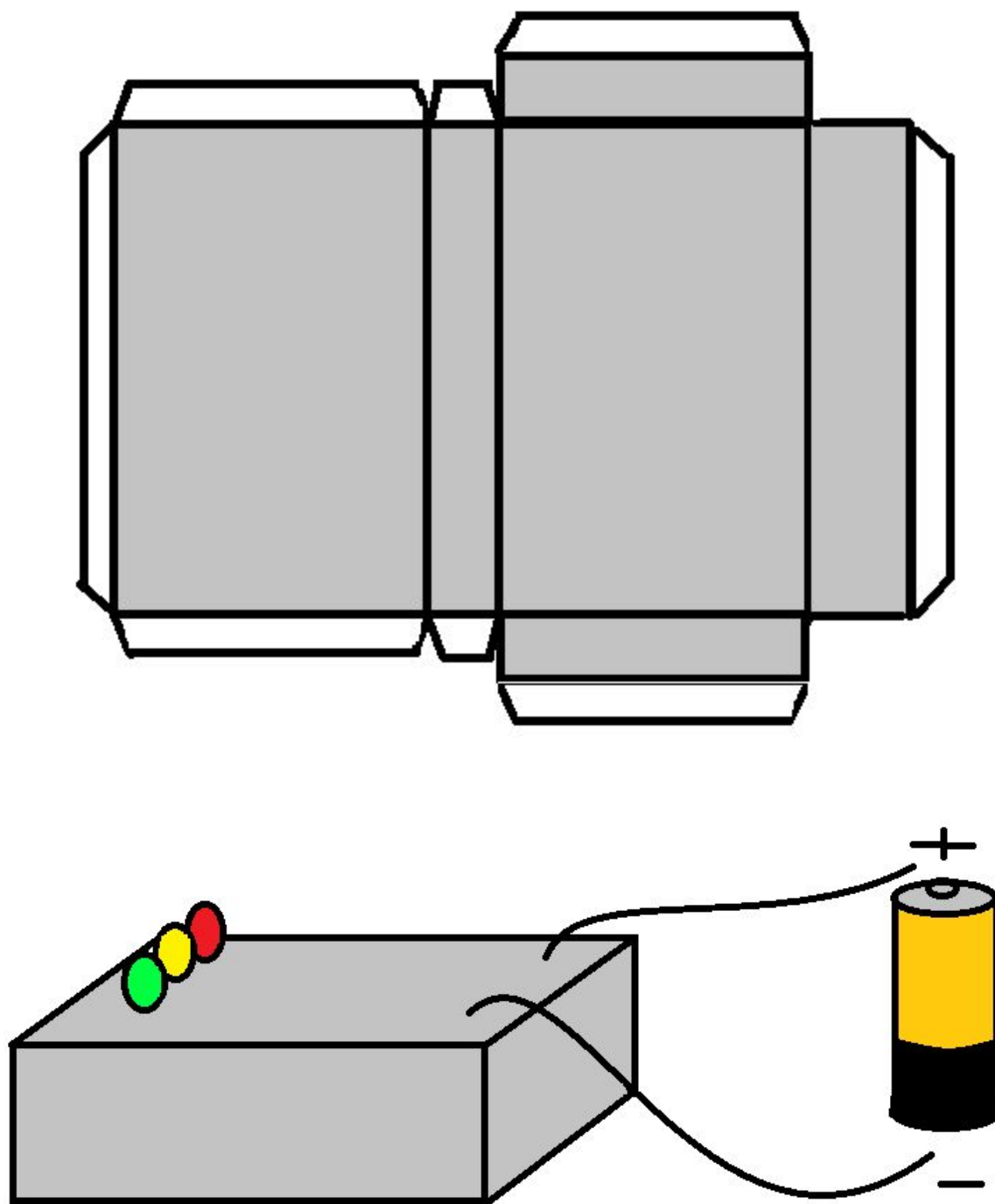
    // Dependiendo del voltaje mostramos un LED u otro
    if (voltaje >= maximo){
        digitalWrite(2, HIGH);
        delay(ledDelay);
        digitalWrite(2, LOW);
    }else if (voltaje < maximo && voltaje > medio){
        digitalWrite(3, HIGH);
        delay(ledDelay);
        digitalWrite(3, LOW);
    }else if (voltaje < medio && voltaje > minimo){
        digitalWrite(4, HIGH);
        delay(ledDelay);
        digitalWrite(4, LOW);
    }

    // Apagamos todos los LEDs
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
}

void loop(){
    // Leemos valor de la entrada analógica
    VALOR = analogRead(0);

    // Obtenemos el voltaje
    voltaje = 0.0048 * VALOR;
    Serial.print("Voltaje: ");
    Serial.println(voltaje);
}
```

III. Diseño del estuche 'contenedor' con el que cubriremos "esqueleto" de la máquina.



Y ESTE SERÍA EL RESULTADO FINAL:

