



MEDIDOR DE CARGA DE BATERÍA Y PILAS

Proyecto realizado por: Juan Carlos Durán Caballero
José Luis Rubio Alcalde
Jose Antonio del Cueto González



Introducción:

Suele ser muy típico que tengamos por casa diferentes aparatos electrónicos 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.



Introducción:

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**.

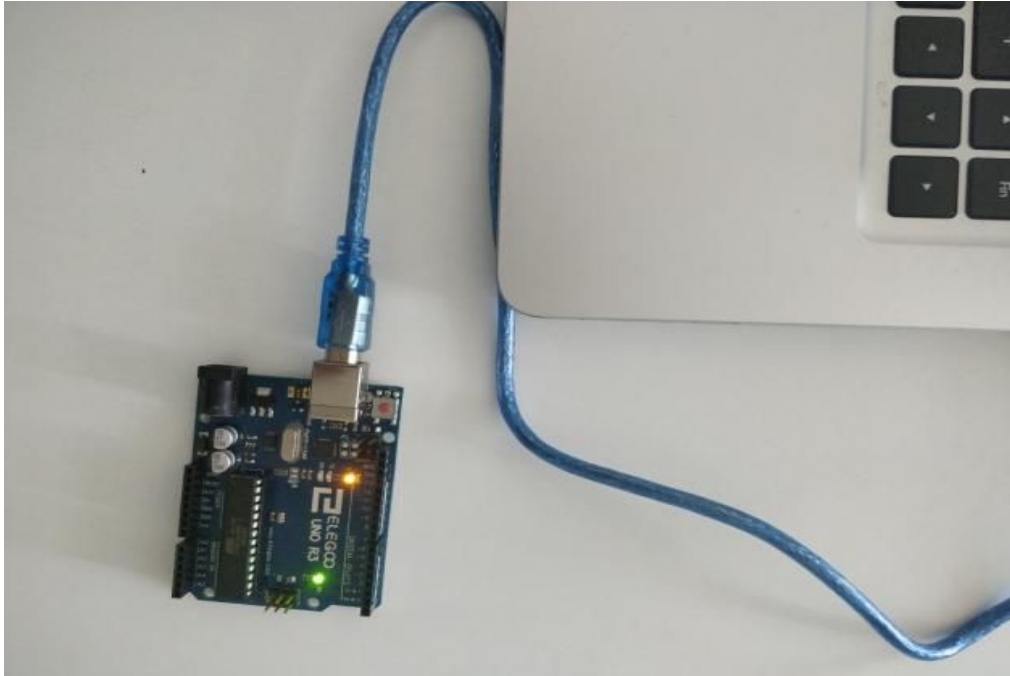


Componentes necesarios a utilizar:



- 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 $10\ \text{k}\Omega$
- 1 LED rojo de 5 mm
- 1 LED amarillo de 5 mm
- 1 LED verde de 5 mm

Iniciamos el proceso



Conectando la placa al PC.

Después descargamos el IDE de Arduino, para programar el código necesario para nuestro proyecto.

MedidorBateriaTest01 Arduino 1.8.8

Archivo Editar Programa Herramientas Ayuda

MedidorBateriaTest01.g

```
// 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);
}

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

  // Calculamos los umbrales
```

```
// 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
 */
```

Aquí declaramos las variables necesarias para nuestro código.

Declaramos los umbrales, el máximo, medio y mínimo.

Aquí señalamos en qué pines estarán colocados nuestros LEDs.

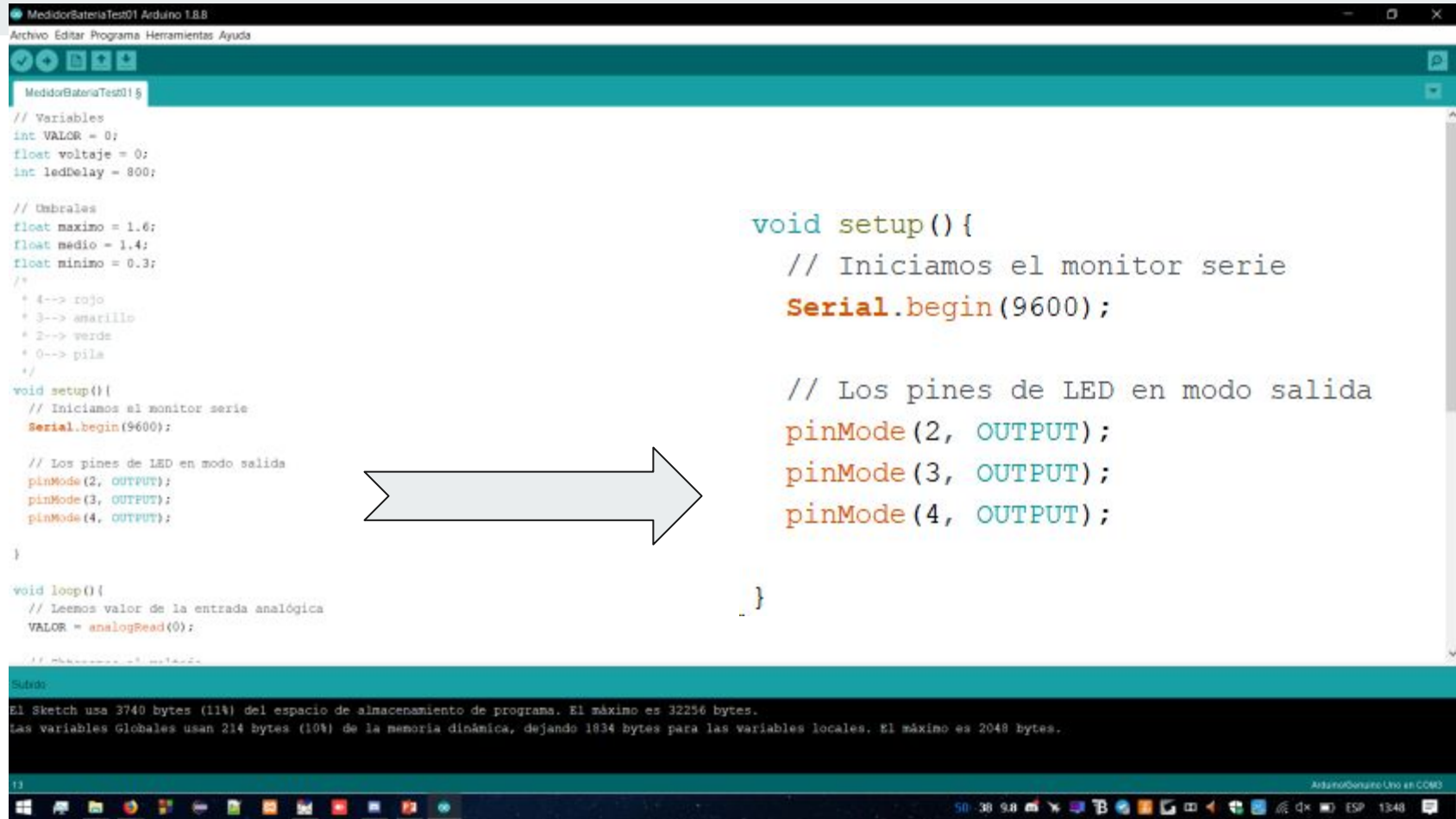
Salida:

El Sketch usa 3740 bytes (11%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 214 bytes (10%) de la memoria dinámica, dejando 1834 bytes para las variables locales. El máximo es 2048 bytes.

13

Arduino/Genuino Uno en COM3

50 30 9.8 ESP 1348





MedidorBateriaTest01\$

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

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

  // 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);
}
```

Salida:

Lorem fistrum está la cosa muy malar hasta luego Lucas sexual! Te voy a borrar el cerito fistro. Fistro qué dise usteer ese hombree condemor me cago en tus muelas condemor se calle usteé la caidita te va a hasé pupitaa caballo blanco caballo negroor! sexual! Ese hombree quietoor hasta luego Lucas ese hombre de la pradera te voy a borrar el cerito pupita caballo blanco caballo negroor! sexual! pupita ese hombre.

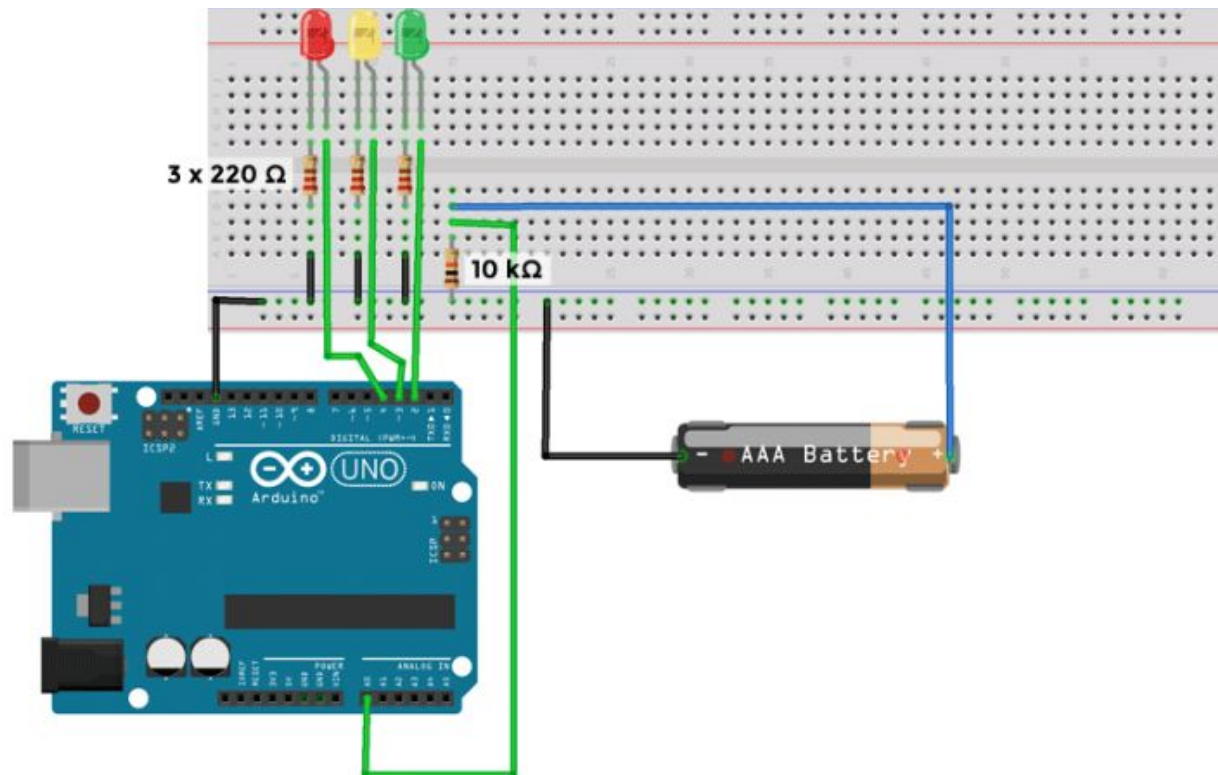
```
void loop(){
  // Leemos valor de la entrada analógica
  VALOR = analogRead(0);

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

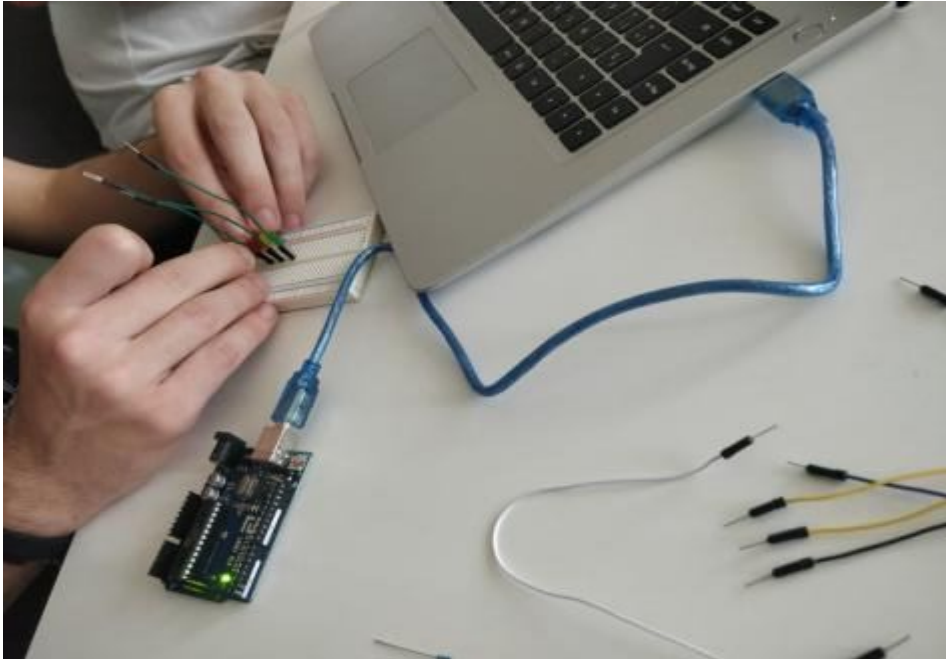
  // Dependiendo del voltaje mostramos un LED u otro
  if (voltaje >= maximo){
    digitalWrite(2, HIGH);
    delay(ledDelay);
    digitalWrite(2, LOW);
  }else if (voltaje < maximo && voltaje > medio){
    digitalWrite(3, HIGH);
    delay(ledDelay);
    digitalWrite(3, LOW);
  }else if (voltaje < medio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

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

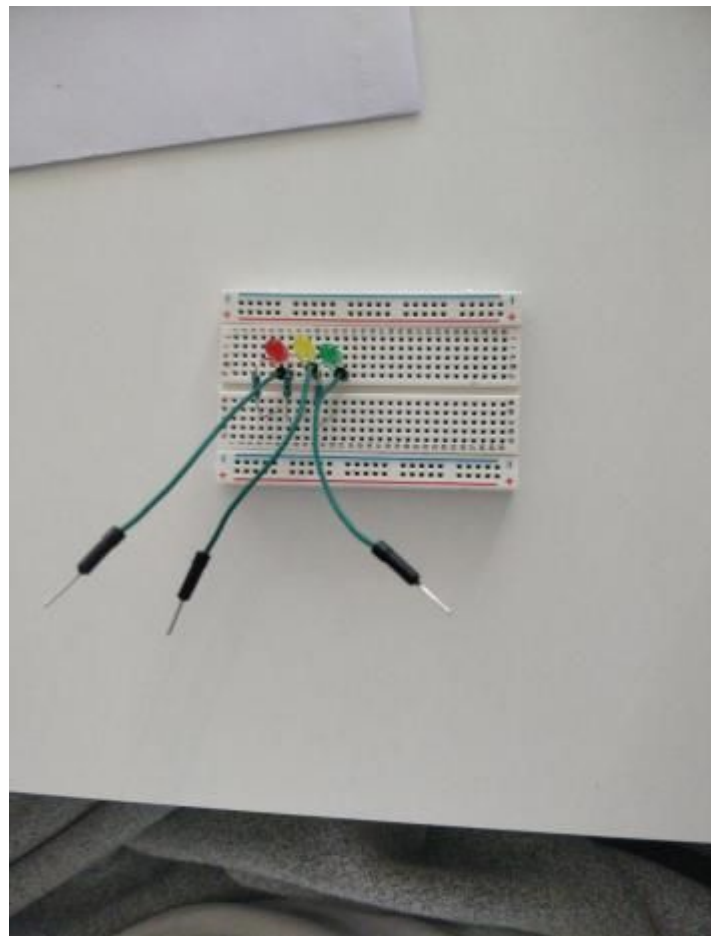
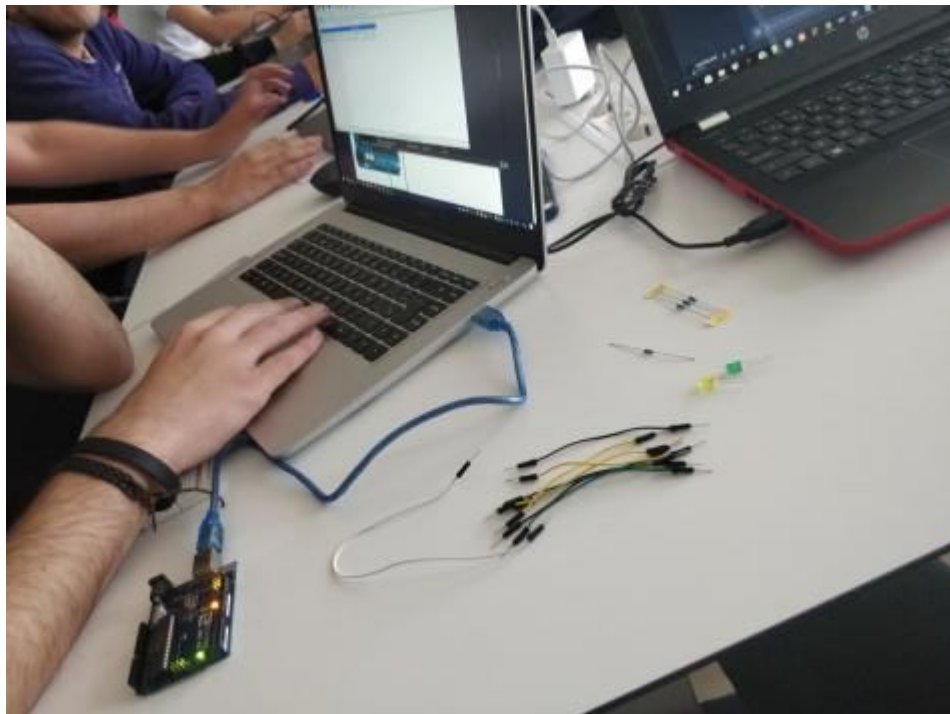
COMO



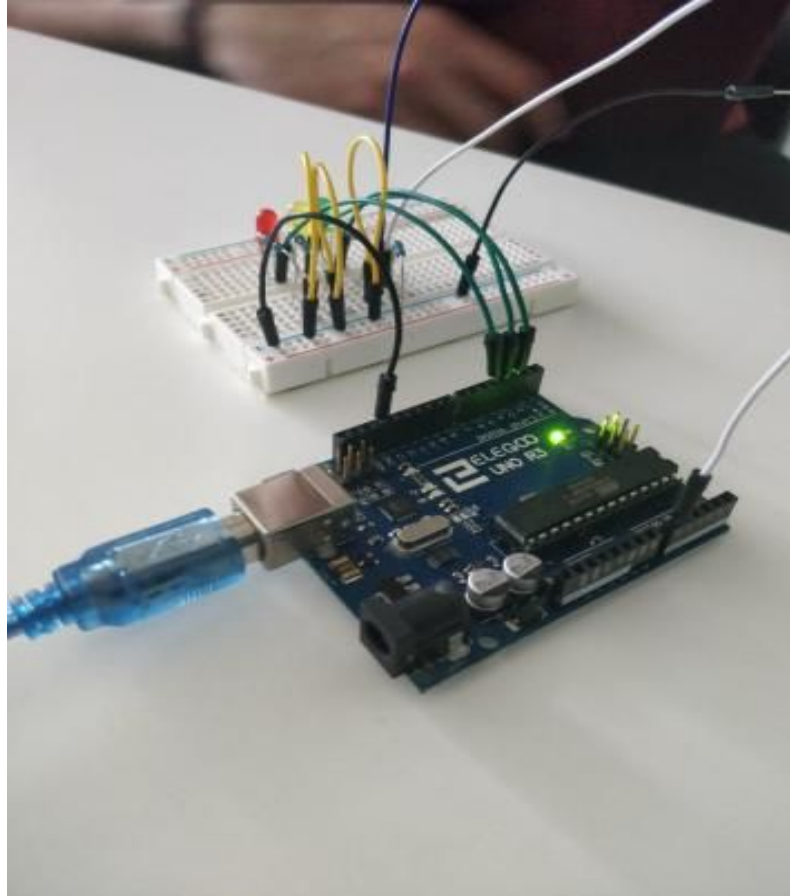
Empezamos a conectar los elementos



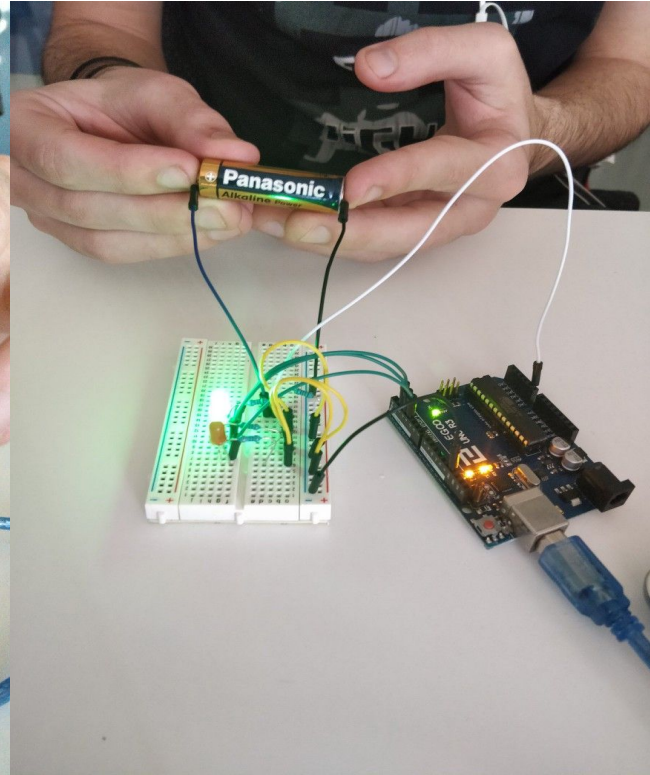
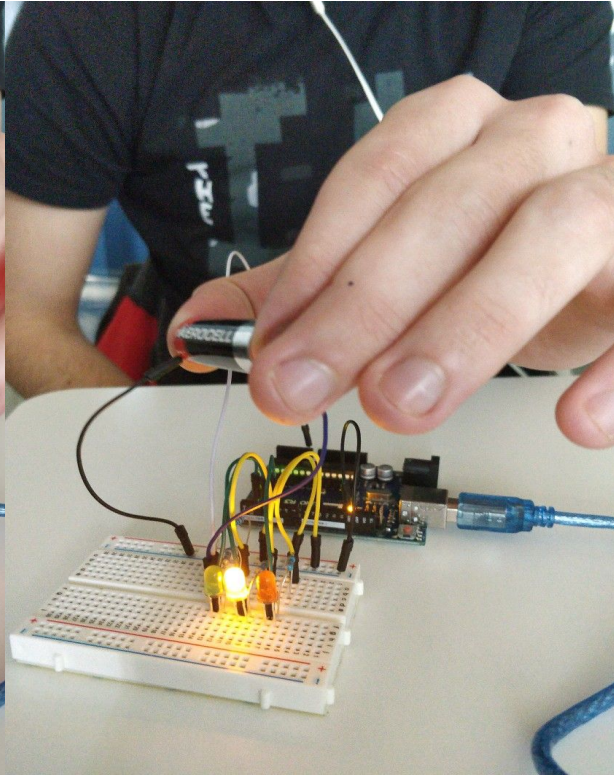
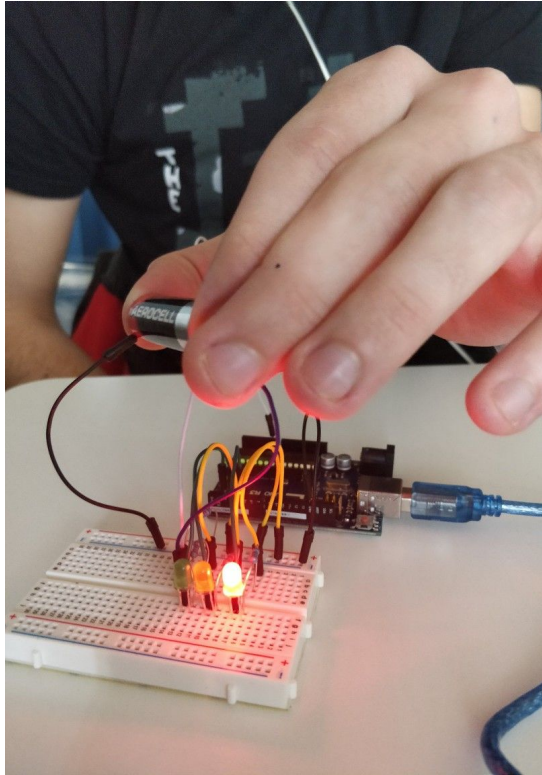
Empezamos a conectar en el protoboard los cables, resistencias y demás componentes.




**Conectamos
el protoboard
a la placa
arduino**



Hacemos las pruebas con diferentes pilas



MedidorBateriaTest01 Arduino 1.8.8

Archivo Editar Programa Herramientas Ayuda

MedidorBateriaTest01.g

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

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

  // 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);
}
```

Podemos observar el voltaje de cada pila

COM3 (Arduino/Genuino Uno)

Enviar

Voltaje: 2.80
Voltaje: 1.61
Voltaje: 1.84
Voltaje: 1.99
Voltaje: 1.96
Voltaje: 2.02
Voltaje: 0.01
Voltaje: 0.00
Voltaje: 0.00
Voltaje: 0.00
Voltaje: 0.00
Voltaje: 4.53
Voltaje: 1.98
Voltaje: 2.02
Voltaje: 4.80

☒ Autoscroll ☐ Mostrar marca temporal

Nueva línea 9600 baudo Limpiar salida

El Sketch usa 3740 bytes (11%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 214 bytes (10%) de la memoria dinámica, dejando 1834 bytes para las variables locales. El máximo es 2048 bytes.

13

Arduino/Genuino Uno en COM3

49:30 8.7 ESP 13:49

RESULTADO FINAL





FIN

PROYECTO FINAL DEL 2º TRIMESTRE