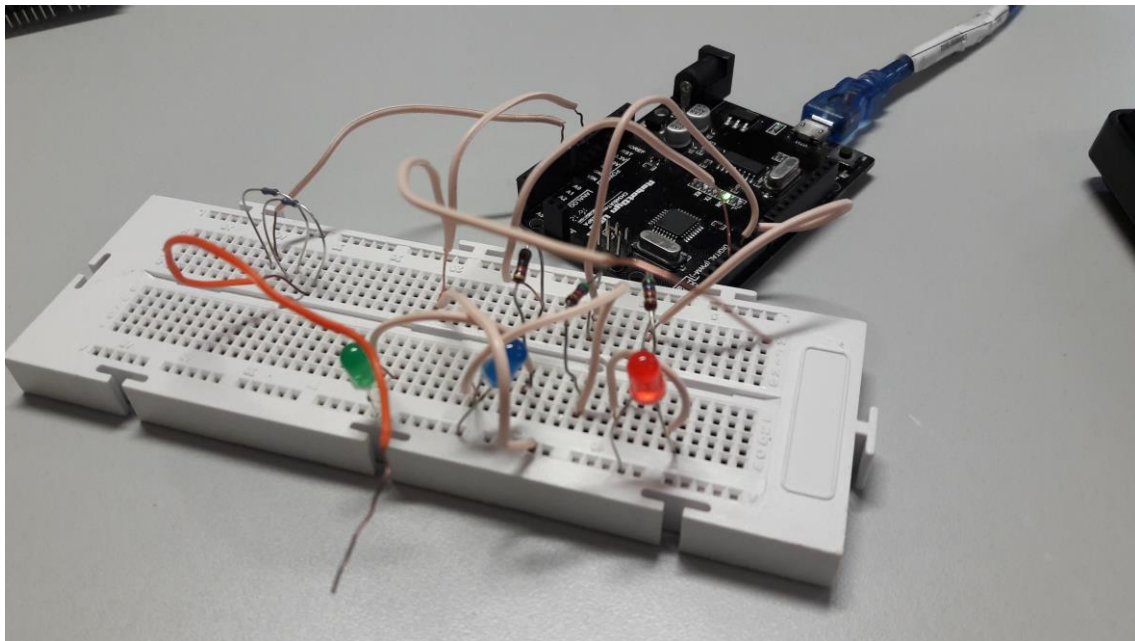


Comprobador de pilas



Alumno: Adrián Nodal Sandoya

Curso: 2ºBACHILLERATO C

Asignatura: TIC II

ÍNDICE

1.-Explicación del código.....	1
2.-Lista de componentes.....	1
3.-Caso 1.....	2
4.-Caso 2.....	3
5.-Caso 3.....	4

1.-Explicación del código

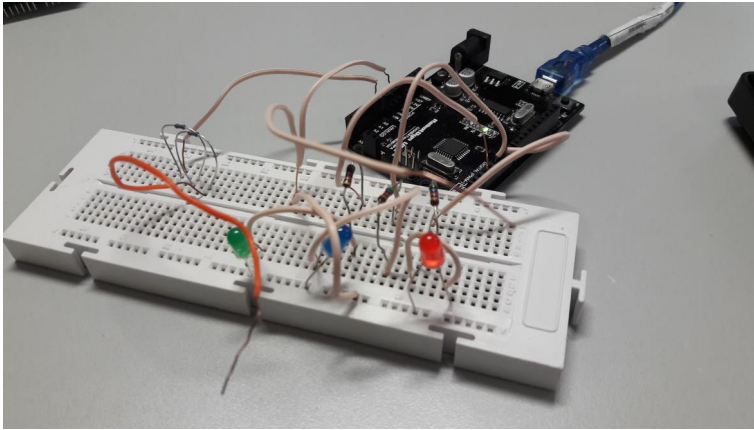
1. Leer desde el pin analógico cero.
2. Multiplique la lectura por 0.0048 para crear un valor de voltaje.
3. Si el voltaje es mayor o igual a 5 V, entonces se encenderá brevemente un LED verde.
4. Si el voltaje es superior a 4 V y menos de 5 V se encenderá uno azul.
5. Si la tensión es inferior a 4 V, se encenderá brevemente un LED rojo.
6. Repetir indefinidamente.

Para poder comprobar el funcionamiento de todos los LEDs variaremos los valores del voltaje.

2.-Lista de componentes:

- Placa Arduino
- Tres Resistores (R1 a R3) de $560\ \Omega$
- Una resistencia (R4) de $2,2\ k\ \Omega$
- Un LED verde (LED1)
- Un LED amarillo (LED2)
- Un LED rojo (LED3)
- Una BreadboardVarios
- Cables de conexión
- Un cable Arduino y USB cable
- Una pila (en este caso de 9v)

3.-Caso 1: se enciende el verde



```
#define newLED 2 //verde LED 'new'

#define okLED 4 // azul LED 'ok'

#define oldLED 6 // rojo LED 'old'

int analogValue = 0;

float voltage = 0;

int ledDelay = 2000;

void setup()
{
  pinMode(newLED, OUTPUT);
  pinMode(okLED, OUTPUT);
  pinMode(oldLED, OUTPUT);
}

void loop()
{
  analogValue = analogRead(0);
  voltage = 0.0048*analogValue;

  if ( voltage >= 5 )
  {
```

```

digitalWrite(newLED, HIGH);

delay(ledDelay);

digitalWrite(newLED, LOW);

}

else if ( voltage < 5 && voltage > 4 )

{

digitalWrite(okLED, HIGH);

delay(ledDelay);

digitalWrite(okLED, LOW);

}

else if ( voltage <= 4 )

{

digitalWrite(oldLED, HIGH);

delay(ledDelay);

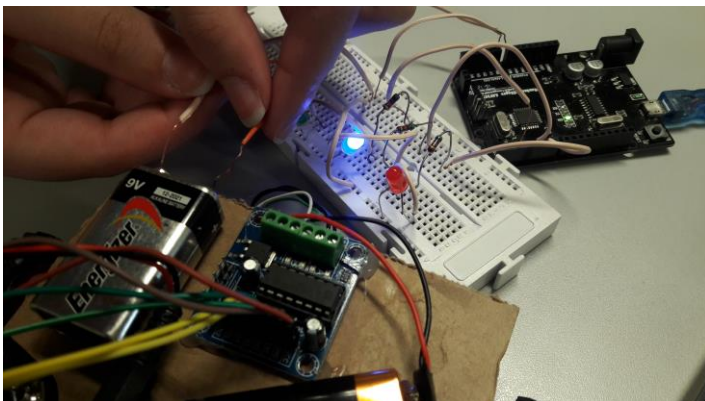
digitalWrite(oldLED, LOW);

}

}

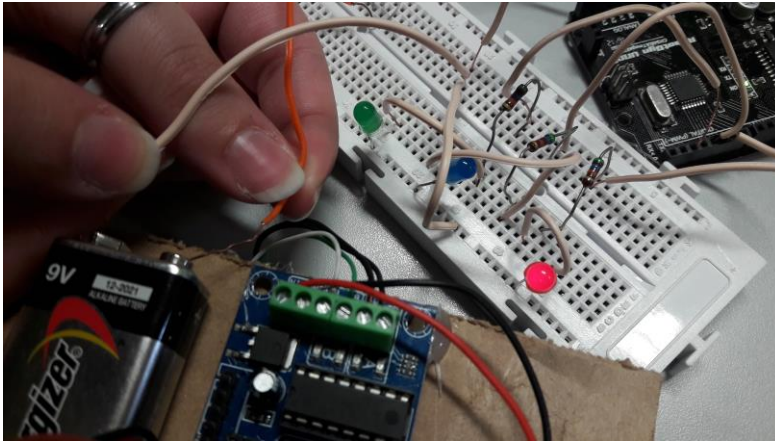
```

4.-Caso 2: se enciende el azul



Donde hay un **5** en el código se cambia por un **11** y el **4** se intercambia por un **7**

5.-Caso 3: se enciende el rojo



Donde hay un **5** en el código se cambia por un **14** y el **4** se intercambia por un **10**