

# AUXILIAR 2

Curso: Mecatrónica – ME4250

Profesor: Harold Valenzuela

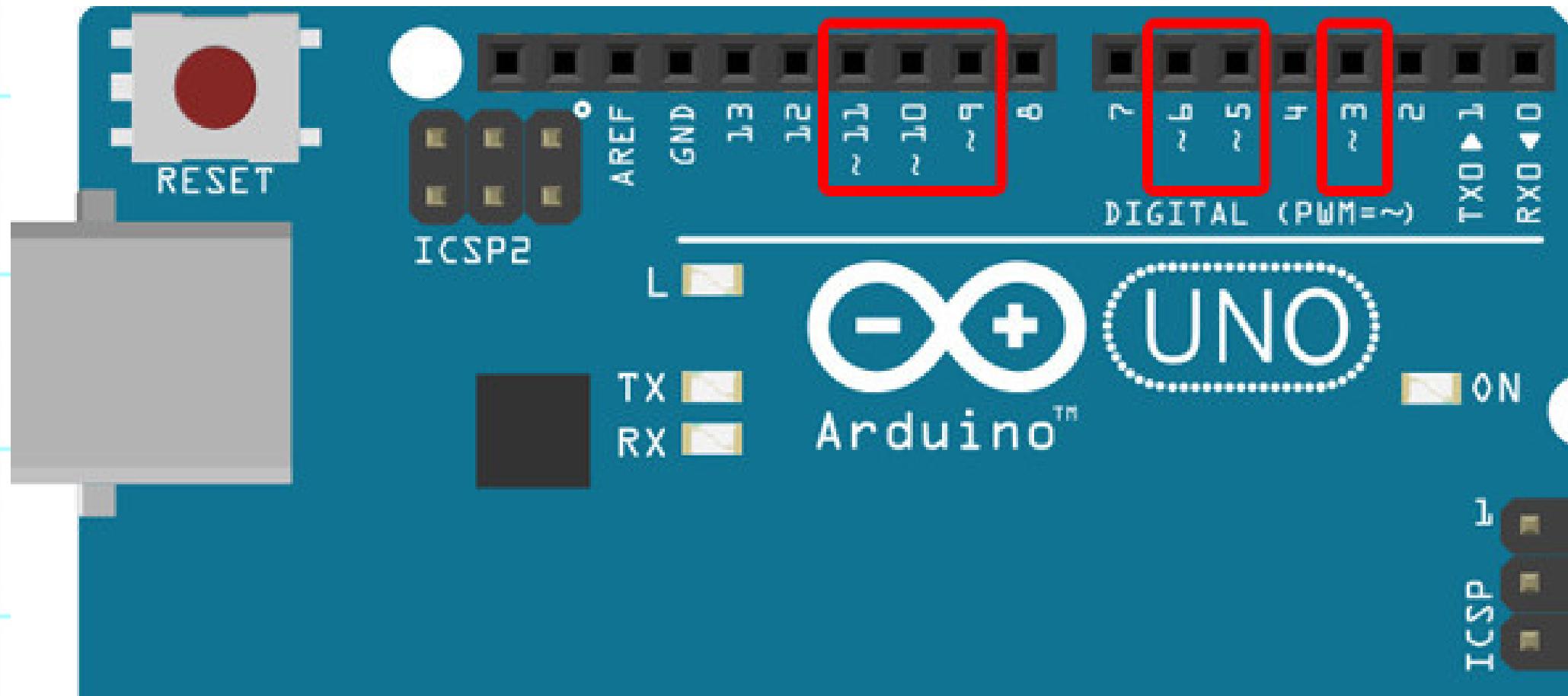
Auxiliares: Francisco Cáceres – Fernando Navarrete

# PULSE WIDTH MODULE (PWM)

PULSE WIDTH MODULATED



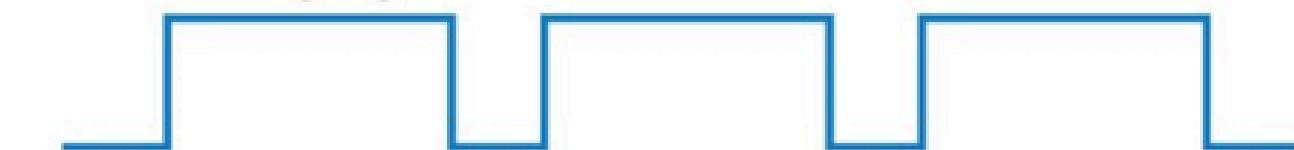
MODULACIÓN POR ANCHO DE PULSO



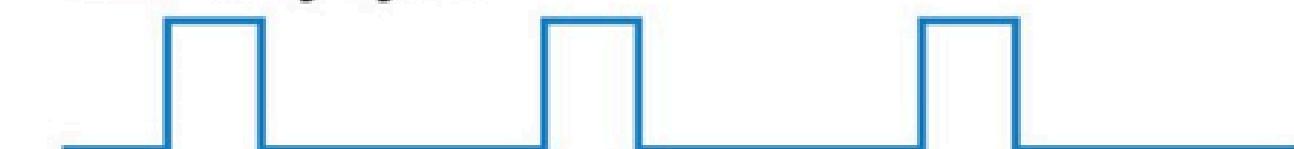
50% duty cycle



75% duty cycle



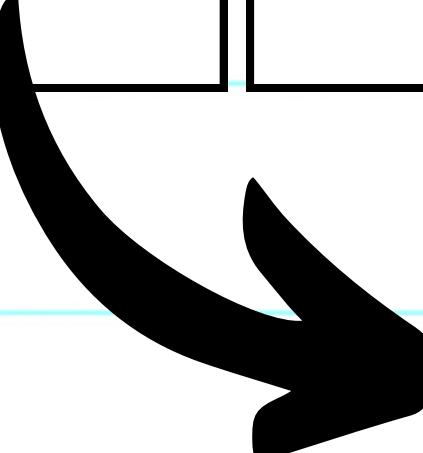
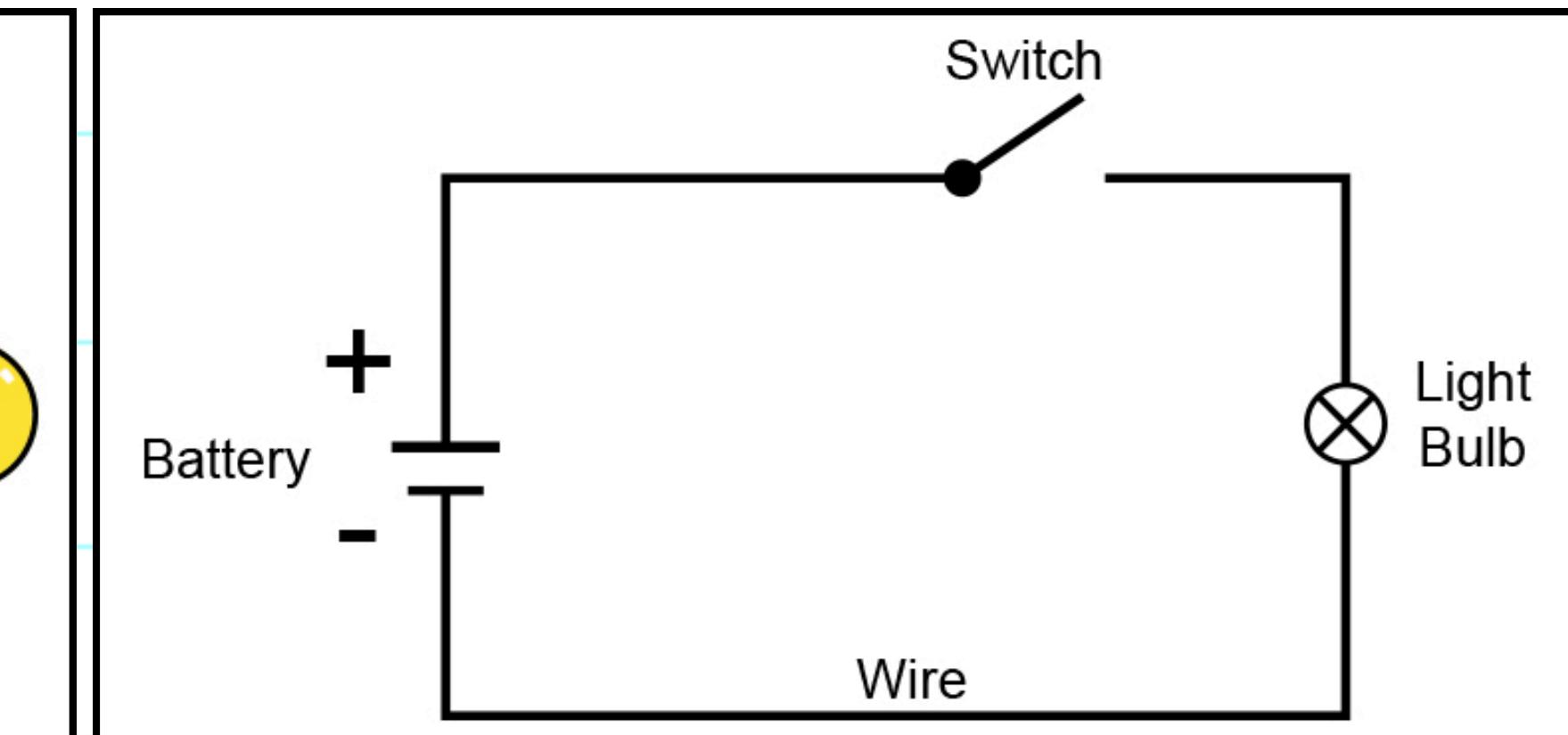
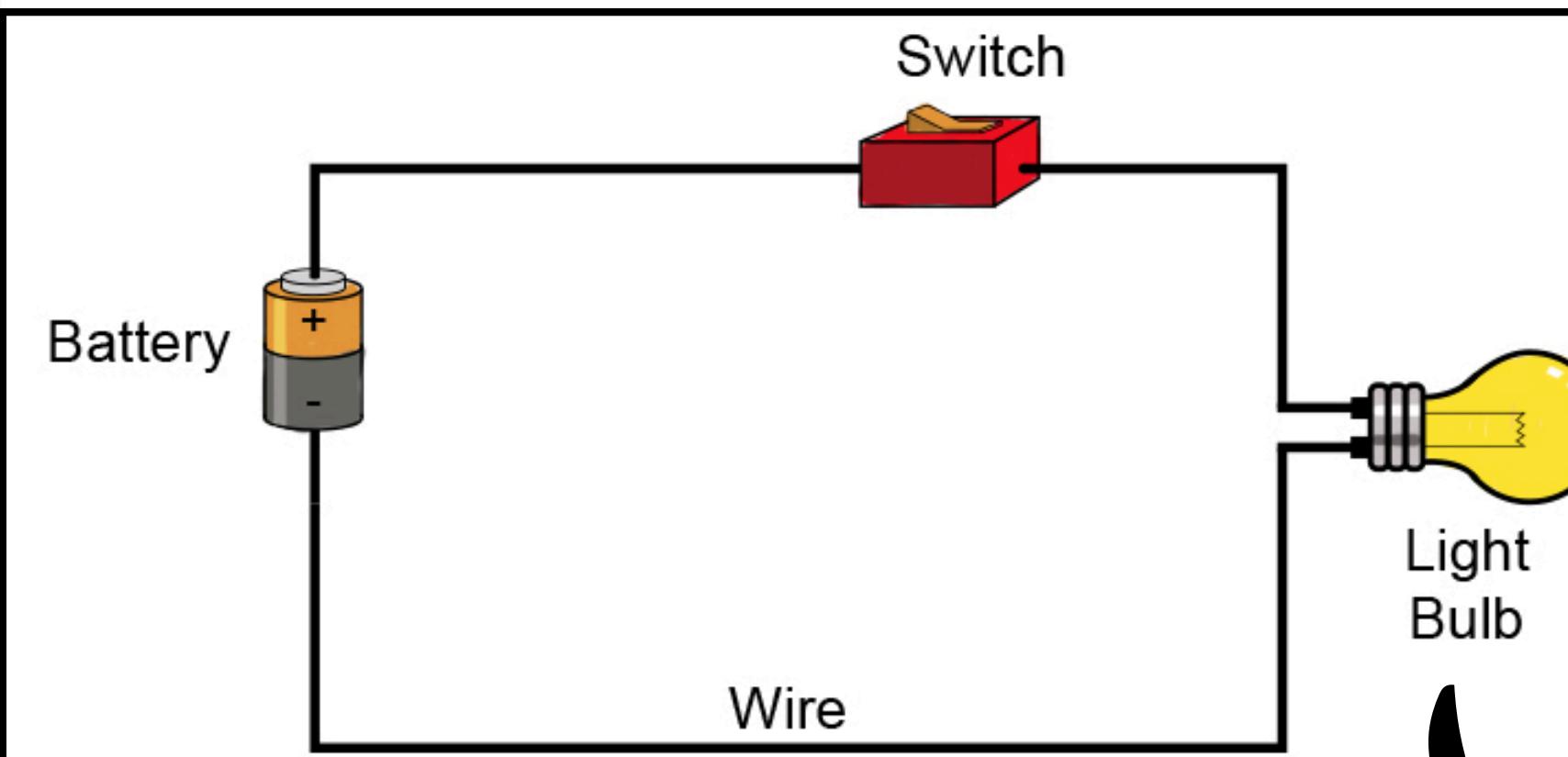
25% duty cycle



$$V_{promedio} = (V_{max} - V_{min}) \cdot \text{Duty Cycle}$$

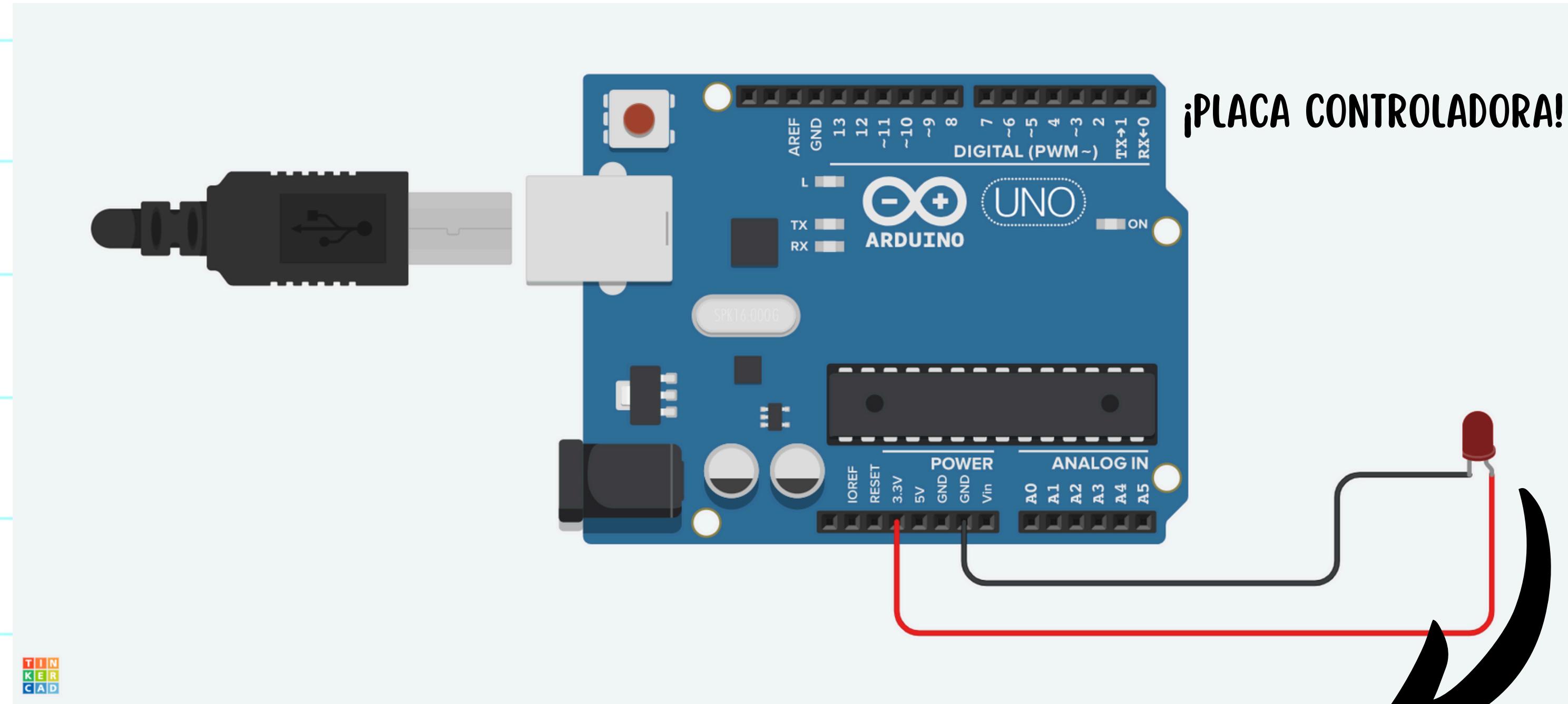
$$\text{Duty Cycle} = t/T$$

# CIRCUITOS CLÁSICOS



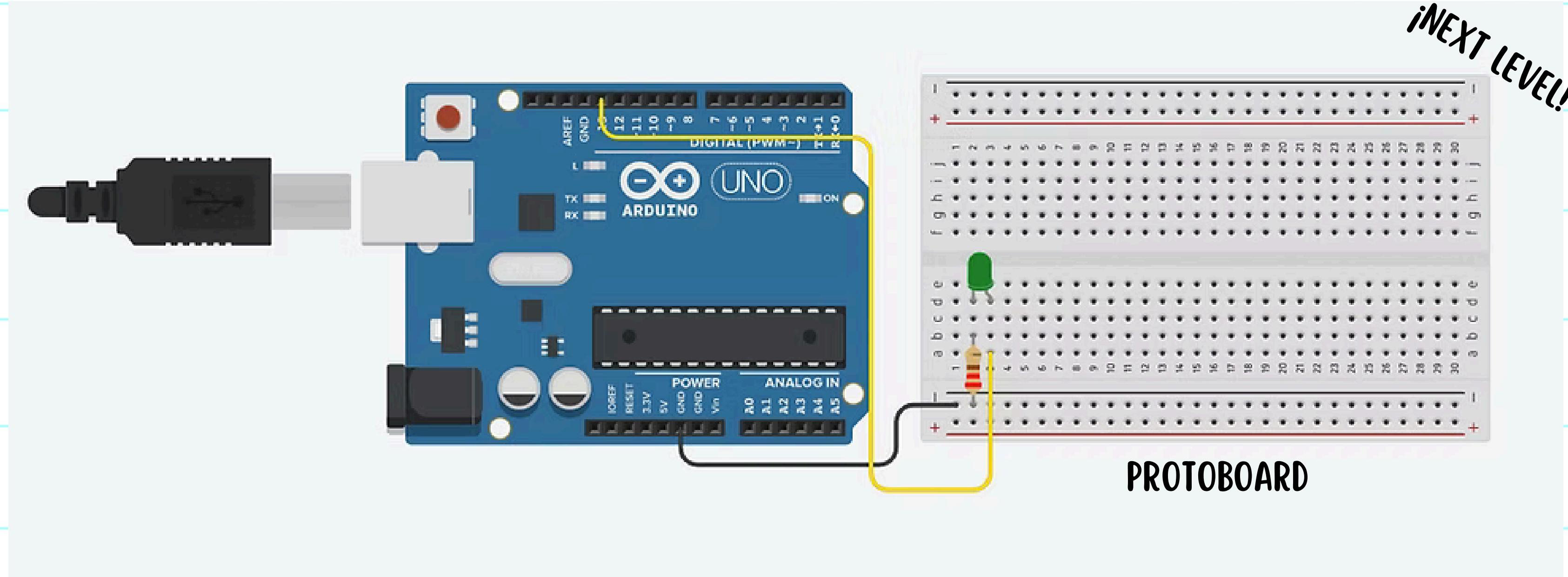
Ó CUALQ. OTRO COMPONENTE ELÉCTRICO

# CIRCUITOS+ARDUINO

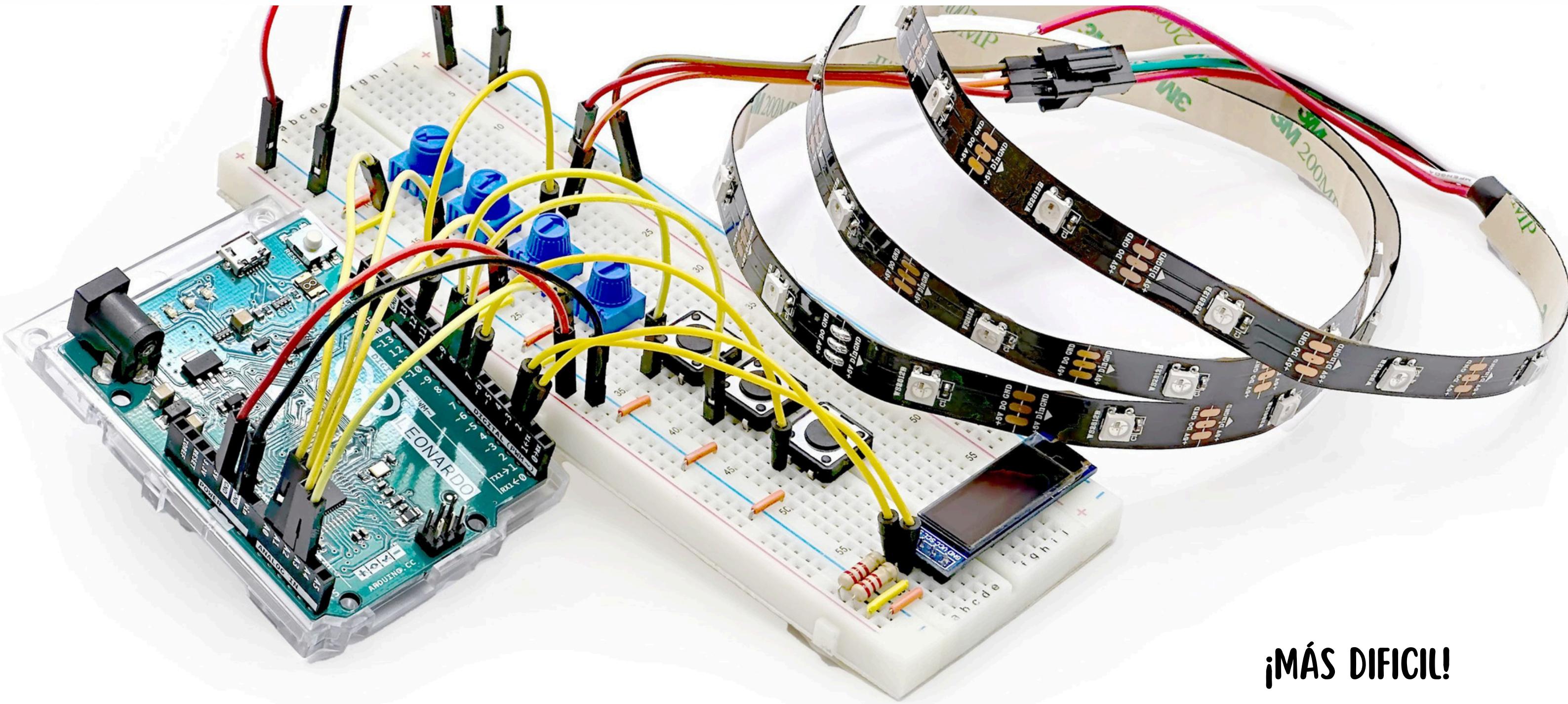


Ó CUALQ. OTRO COMPONENTE ELÉCTRICO

# CIRCUITOS+ARDUINO

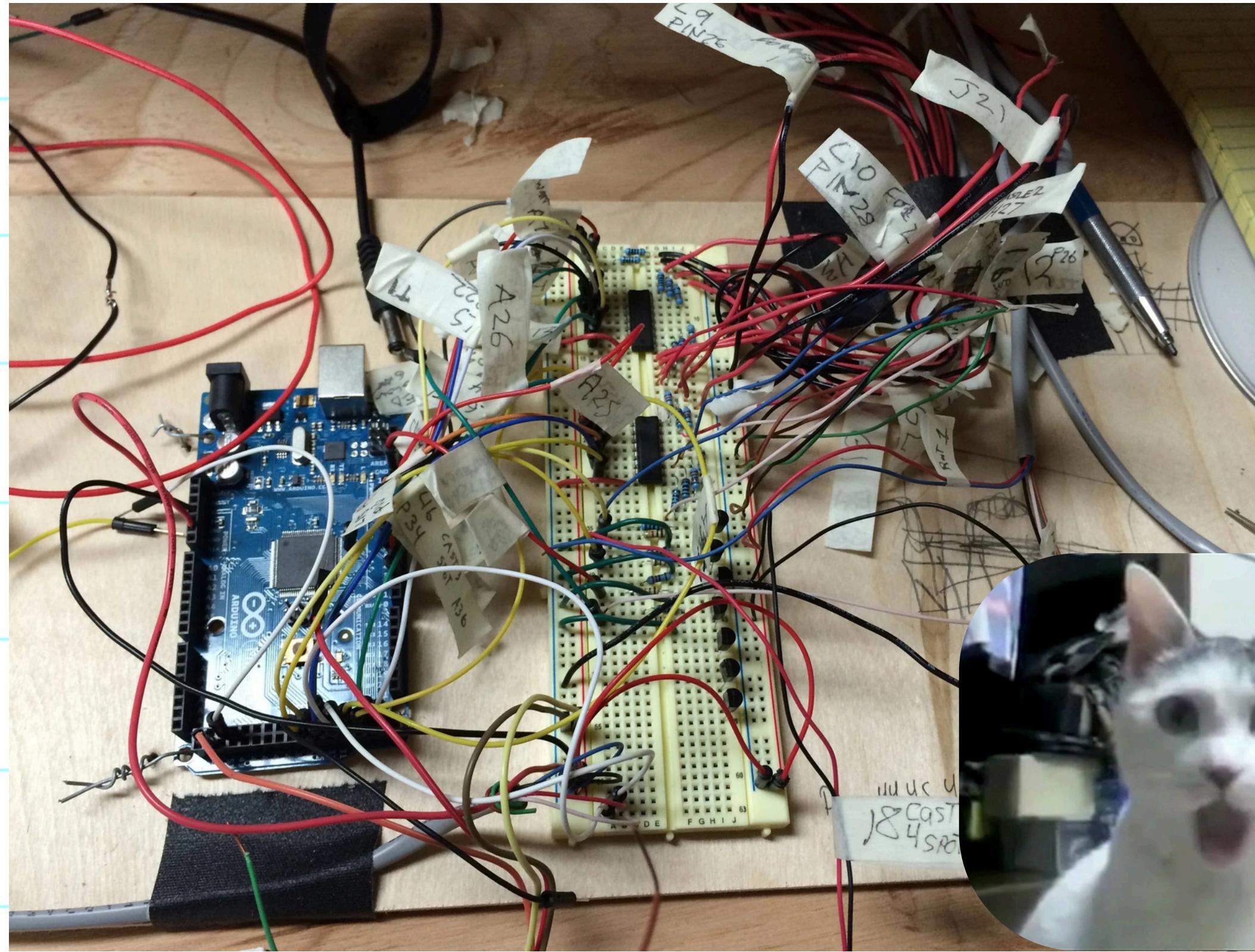


# CIRCUITOS+ARDUINO

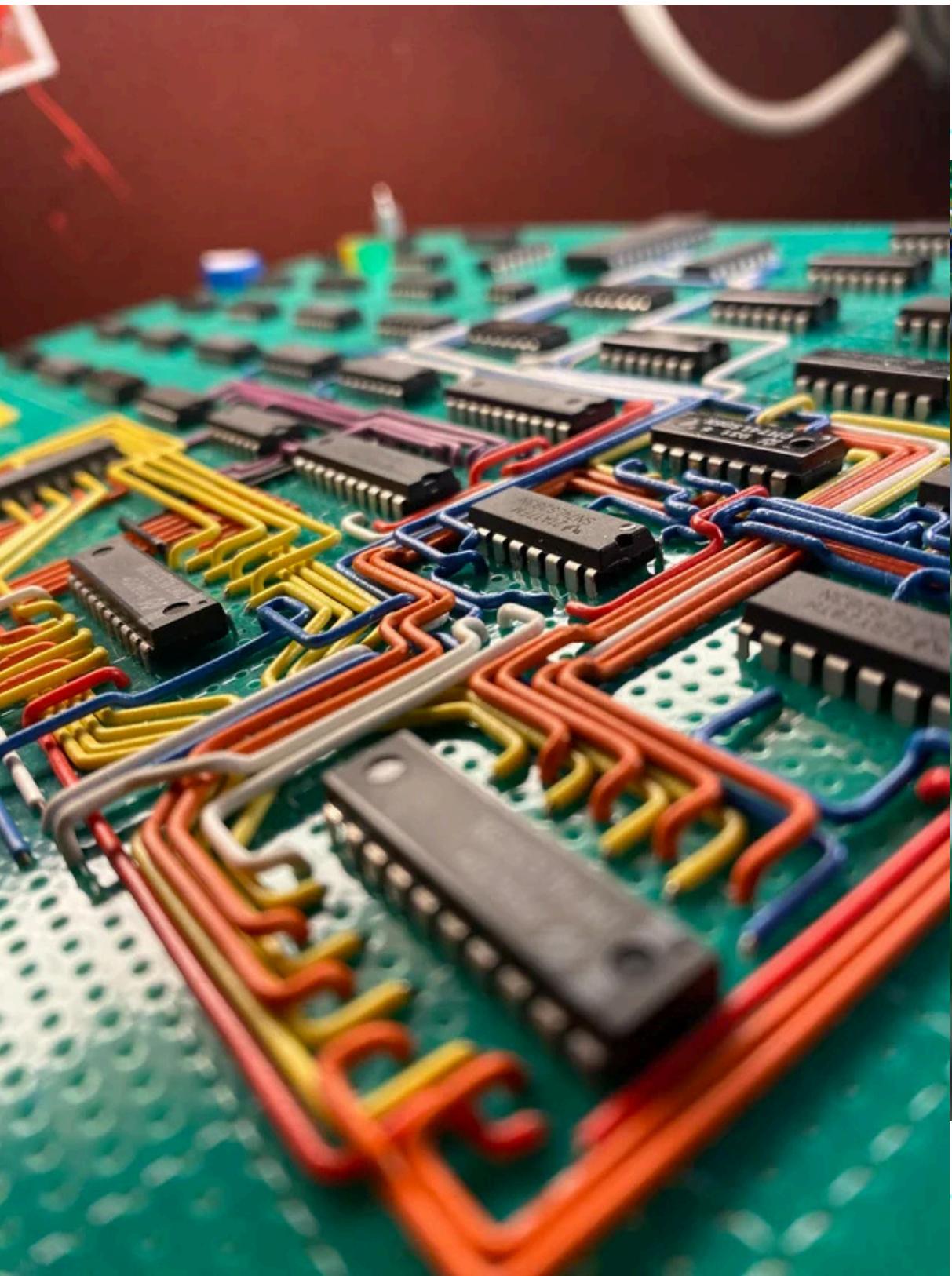


¡MÁS DIFÍCIL!

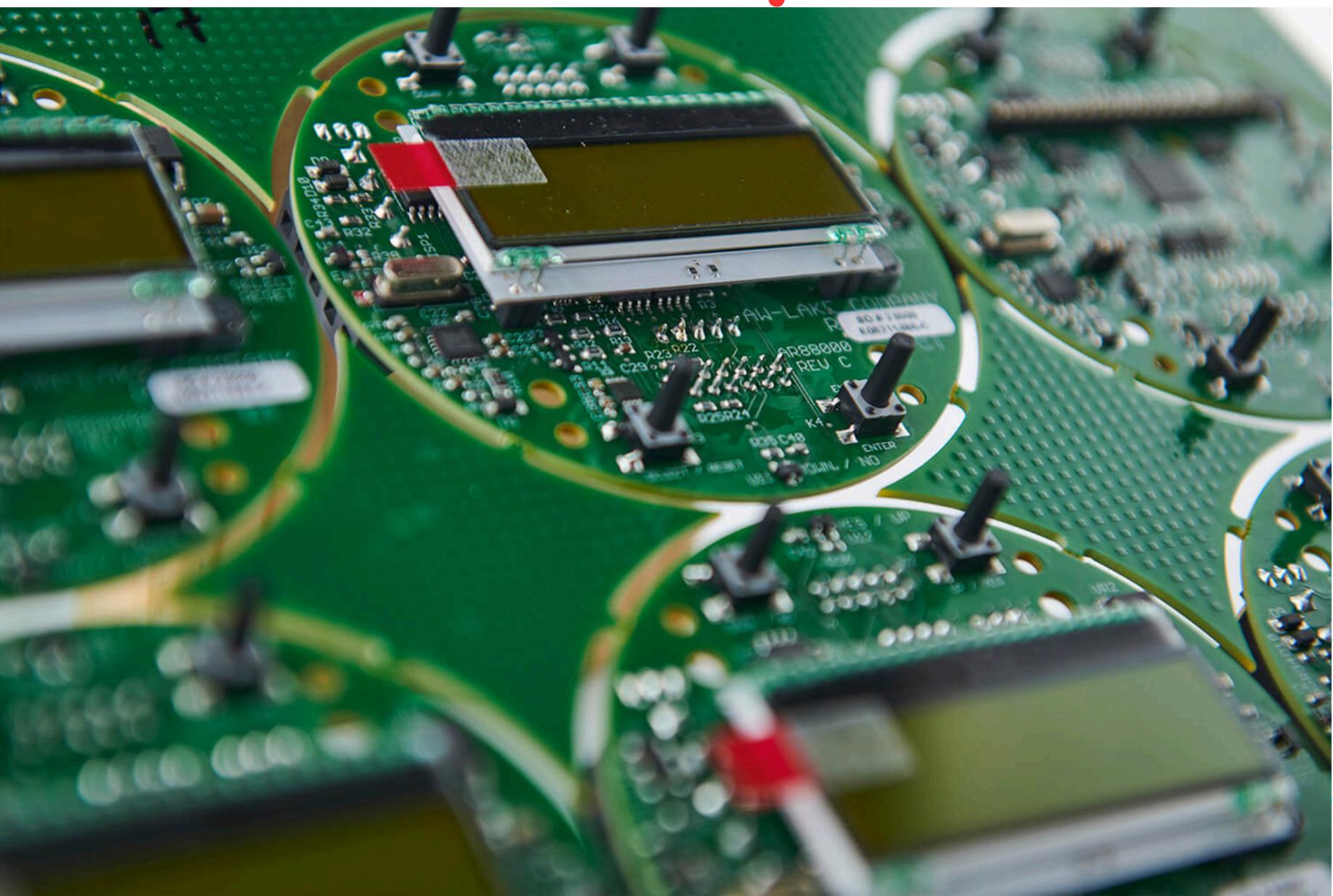
# CIRCUITOS+ARDUINO



# CIRCUITOS + ARDUINO



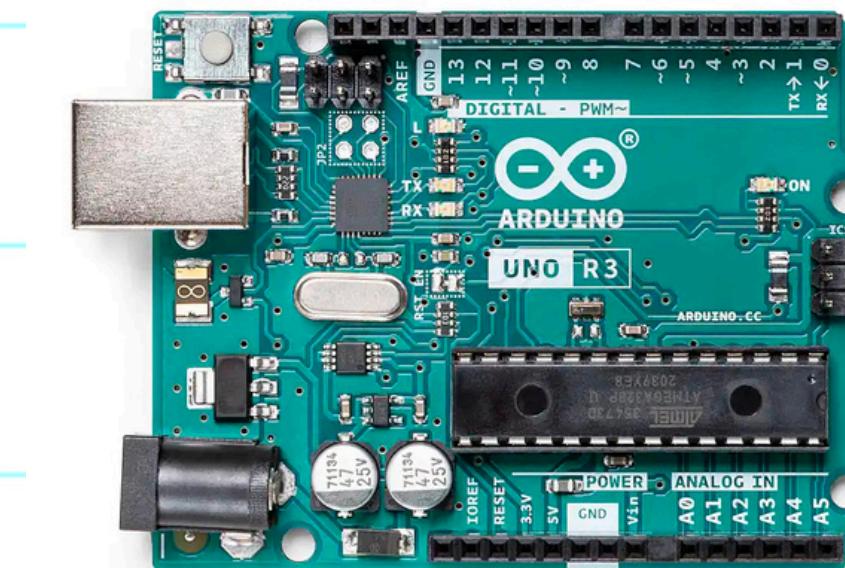
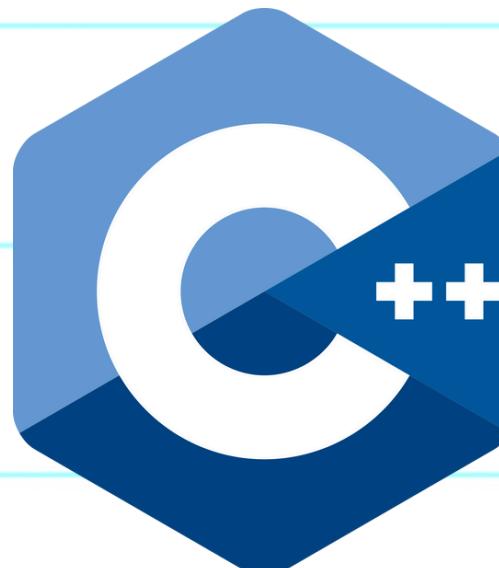
# ¡PLACAS PCB!



# ¡PERFECTO!

# PROGRAMACIÓN EN ARDUINO

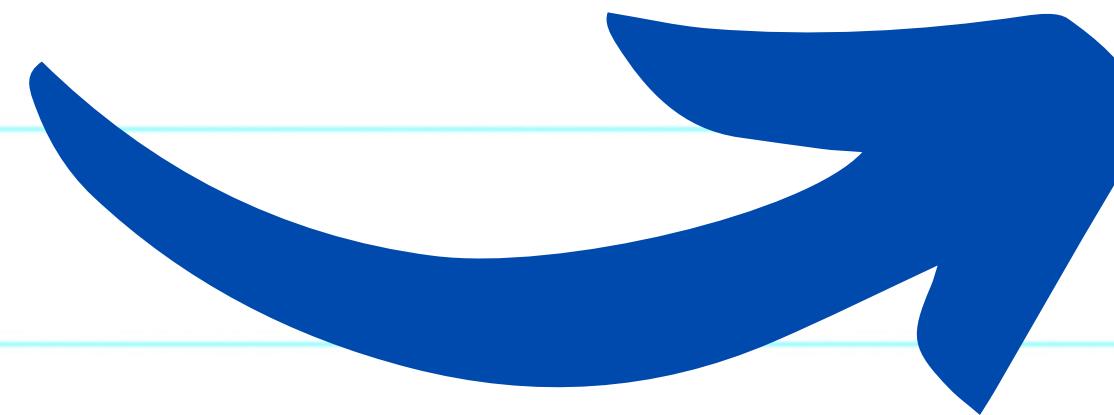
- TIPO DE LENGUAJE: C++
- 14 PUERTOS DIGITALES (6PWM)
- 6 PUERTOS ANALÓGICOS LECTURA
- FUENTES DE ALIMENTACIÓN: 5V Y 3.3V (O EXTERNAS)



# PROGRAMACIÓN: ESTRUCTURA

VOID SETUP{

}

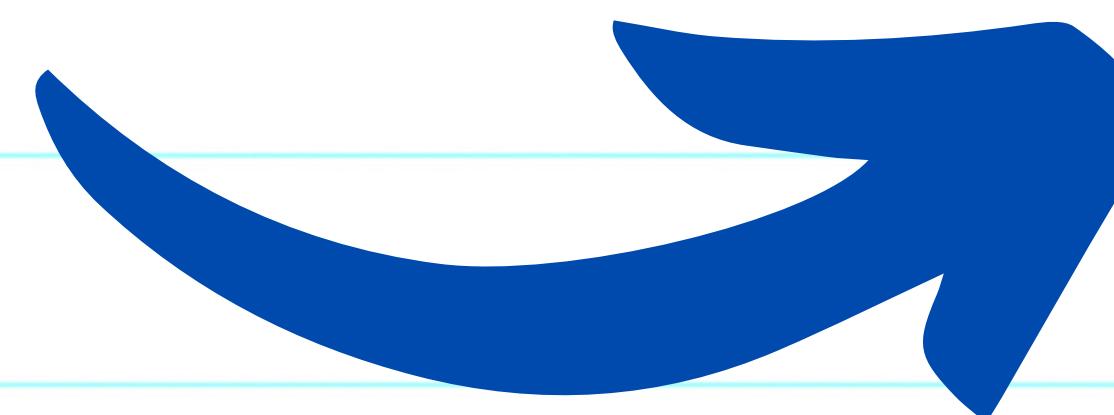


- Importar Librerias
- Definir Variables
- Establecer Estados Iniciales

¡1 VEZ!

VOID LOOP{

}



- Definir programa
- Cálculos y Operaciones Iterativas

∞

# PROGRAMACIÓN: BÁSICO

## TIPOS DE DATOS

BOOLEAN	TRUE/FALSE	VALOR BOOLEANO
BYTE	0-255	8 BIT DE DATOS
INT	16 BIT	NÚMERO ENTERO
LONG	32 BIT	NÚMERO GRANDE
FLOAT	32 BIT	NÚMERO GRANDE DECIMAL
ARRAY[]	{1,2,3,4,...,}	LISTA DE DATOS
STRING	'HOLA MUNDO'	TEXTO

# PROGRAMACIÓN: SINTAXIS

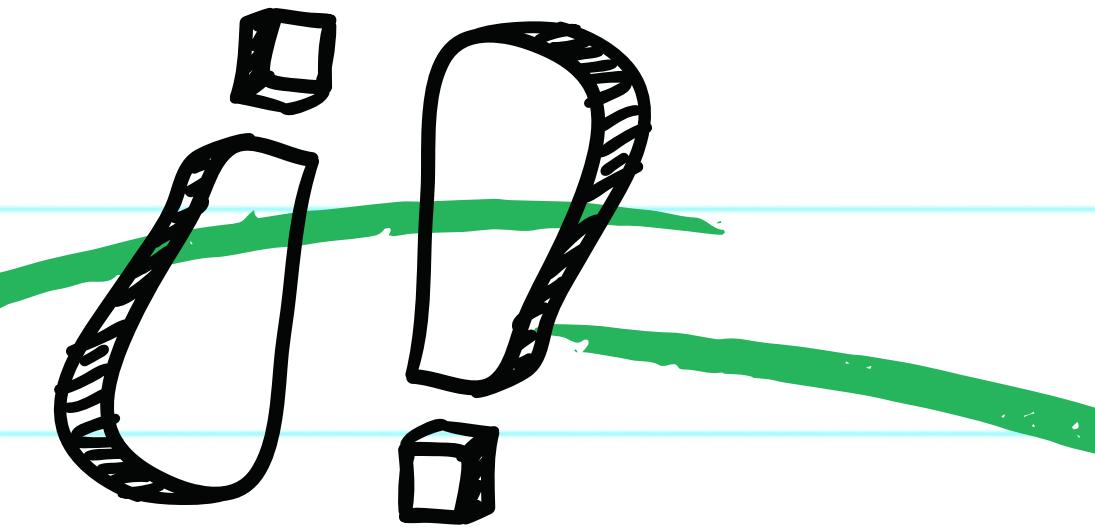
```
Casco_hermes_code.ino
1 // C++ code
2 //
3
4 #include <Servo.h>
5
6 //Variables
7 int angulo = 0;
8 int pulsador = 7;
9 int estado = 0;
10 int valor_actual = 0;
11 int valor_anterior;
12
13 //Definimos Servomotores
14 Servo motor1;
15 Servo motor2;
16 void setup()
17 {
18     motor1.attach(9);
19     motor2.attach(10);
20     pinMode(pulsador, INPUT);
21     Serial.begin(9600);
22 }
23
24 void loop()
25 {
26     delay(100);
27     valor_actual = digitalRead(pulsador);
28     //Serial.println(valor_actual); (verificar respuesta)
29     //Para que detecte bien el pulsador, es necesario mantener presionado
30     // hasta que se detengan los servomotores.
31     if (valor_actual == 1 && valor_anterior == 0){
32         estado = 1 - estado;
33         delay(1000);
34     }
35 }
```

- **//** : para comentar (línea simple)
- **/\*\*/** : para comentar (línea múltiple)
- **#Include <librerie>**: importar librerias
- **;** : separador de líneas de código
- **void** : estructura/función
- **{}** : abrir y cerrar funciones o estructuras
- **()** : para operaciones o ingresar parámetros a estructuras o funciones
- **delay()** : retardo entre líneas de código (milisegundos)
- **delayMicroseconds()** : retardo entre líneas de código (microsegundos)

[LINK DE REFERENCIA](#)

**TINKERCAD**

AUTODESK®  
TINKERCAD®





AUTODESK®  
TINKERCAD®

Join **Mecatrónica - ME4250** with a link or enter this Class Code:

**M7K KPB UYQ**

# ALGUNAS FUNCIONES

Función	Descripción
<code>pinMode(pin, mode)</code>	Configura un pin como entrada o salida.
<code>analogRead(pin)</code>	Lee valores analógicos de 0 a 1023.
<code>analogWrite(pin, value)</code>	Escribe valores PWM de 0 a 255 en un pin.
<code>map(value, fromLow, fromHigh, toLow, toHigh)</code>	Transforma un número de un rango a otro.
<code>Serial.begin(baudrate)</code>	Inicia la comunicación con el <b>Monitor Serial</b> .
<code>Serial.print(valor) / serial.println(valor)</code>	Envía datos al Monitor Serial.
<code>delay(ms)</code>	Detiene la ejecución del código por <b>X milisegundos</b> .

[LINK DE REFERENCIA](#)

*¡* GRACIAS !

