Curs d'Arduino CTM16-17 Mòdul 2

EXEMPLES I PRÀCTIQUES AMB ARDUINO

1. *Blink*
2. *Encesa de 5 LEDs simultàniament*
3. *Cotxe fantàstic*
4. *Encès/apagat gradual d'un LED*
5. *Encesa i apagat gradual de 3 LEDs simultàniament*
6. *Polsadors*
7. *Polsador II*
8. *Lectura d'una sortida digital*
9. *El potenciòmetre, entrades analògiques i comunicació sèrie*
10. *Regulació de la il·luminació d'un LED amb potenciòmetre*

EXEMPLES I PRÀCTIQUES AMB ARDUINO

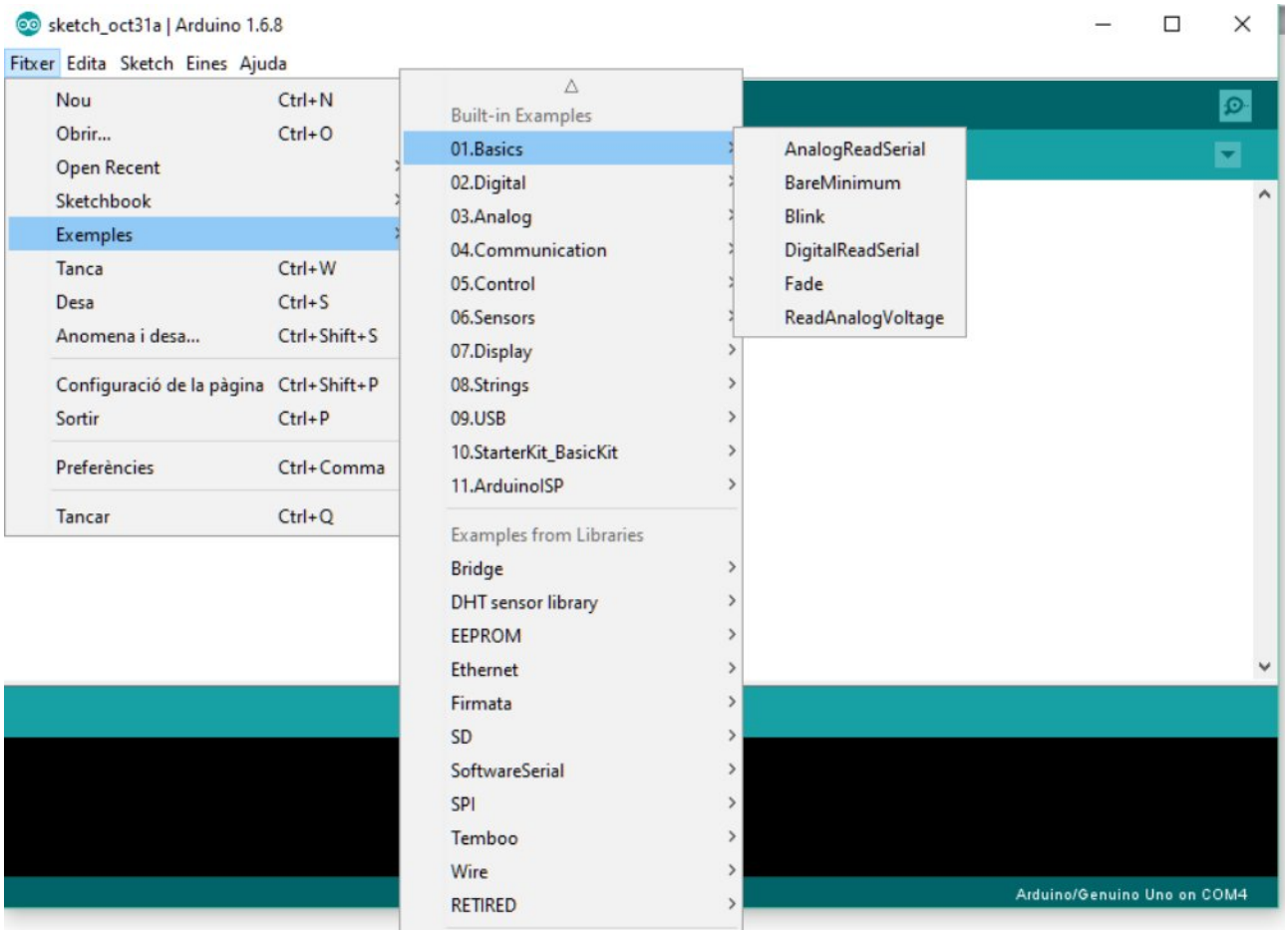
Buscant per Internet podem trobar una infinitat d'exemples i projectes molt ben tutoritzats.

També, podem trobar una colla d'exemples de programes amb Arduino des de la web Arduino.cc :

https://www.arduino.cc/en/Tutorial/BuiltInExamples

O des del mateix IDE d'Arduino:

[Menú] / Fitxer / Exemples / ...



1. Blink

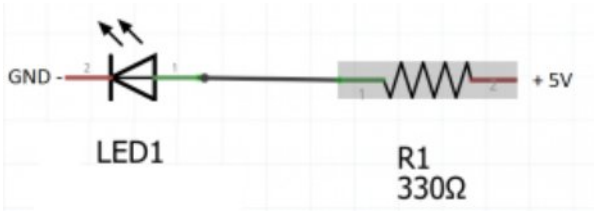
Objectiu: Parpalleig d'un LED

Components:

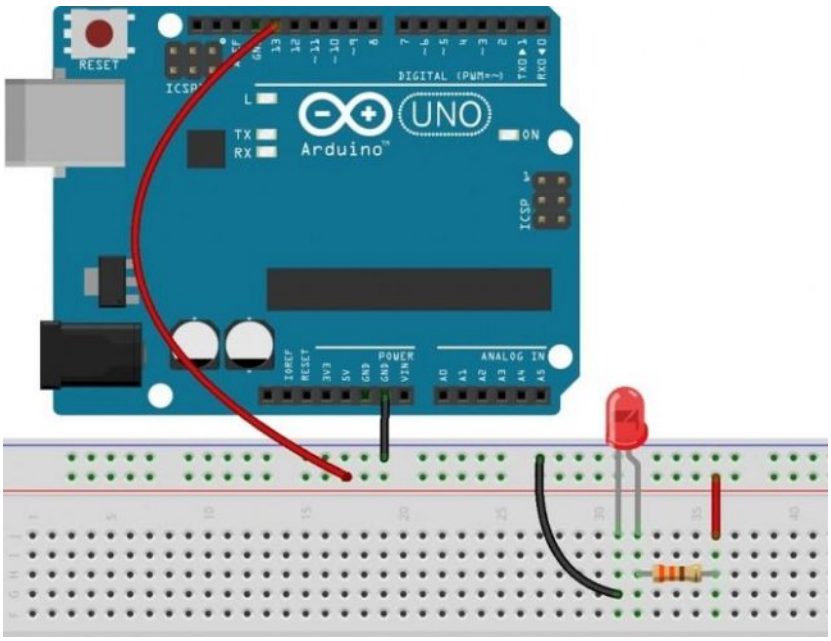
1	Arduino UNO o similar
1	protoboard
1	LED
1	R = 330 Ω (o també R = 220 Ω)
	cables

Circuit electrònic:

El LED es pot connectar directament al pin 13 d'Arduino. Per a qualsevol altre pin 0-12 ha de connectar-se una R = 330 Ω per no malmetre el LED.



Muntatge:



Codi:

```
/* BLINK

Intermitent d'un LED amb un temps d'espera d'1 segon
Si el LED es connecta al pin 13 no fa falta connectar cap resistència
-------------------------------------------------------------------------------------
*/

void setup()
   {
      pinMode(13, OUTPUT);       // Utilitzem el pin 13 com sortida
   }

void loop()
   {
     digitalWrite(13, HIGH);     // Encén el LED
     delay(1000);                // Espera un segon
     digitalWrite(13, LOW);      // Apaga el LED
     delay(1000);                // Espera un segon
  }
```

Referències:

- pinMode(pin, mode)
- digitalWrite(pin, value)
- delay (value)

2. Encesa de 5 LEDs simultàniament

Objectiu: Encendre una seqüència de 5 LEDs de manera simultània.

Introducció d'una variable

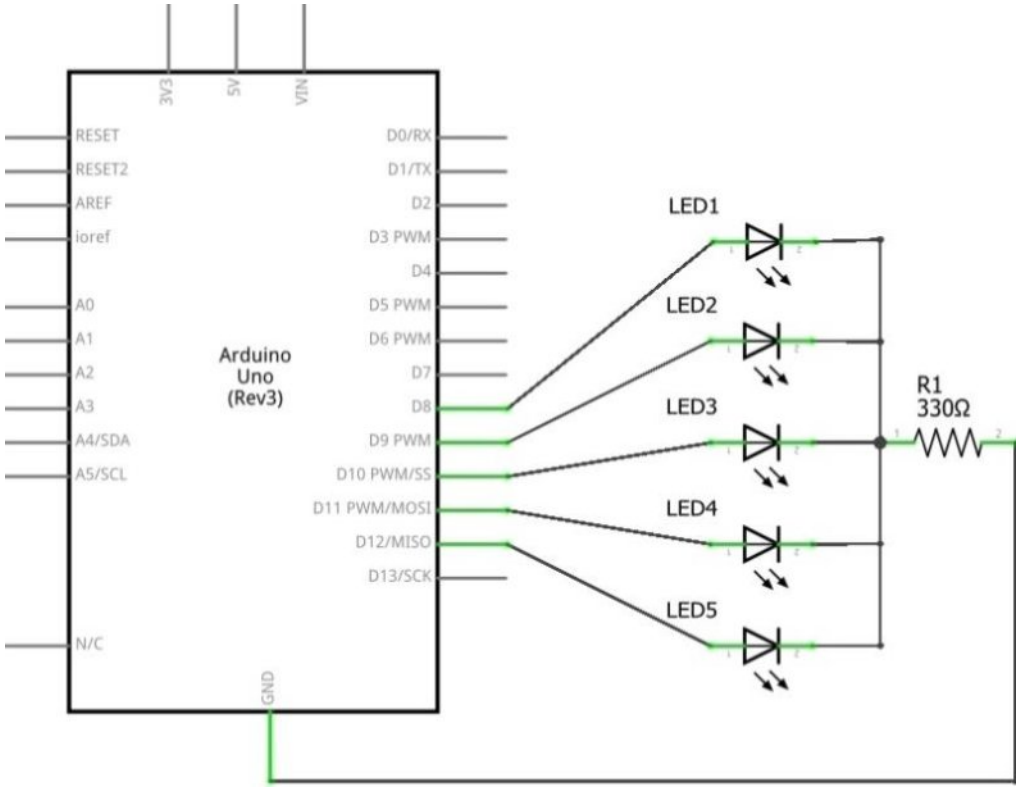
Instrucció "for { }"

Components:

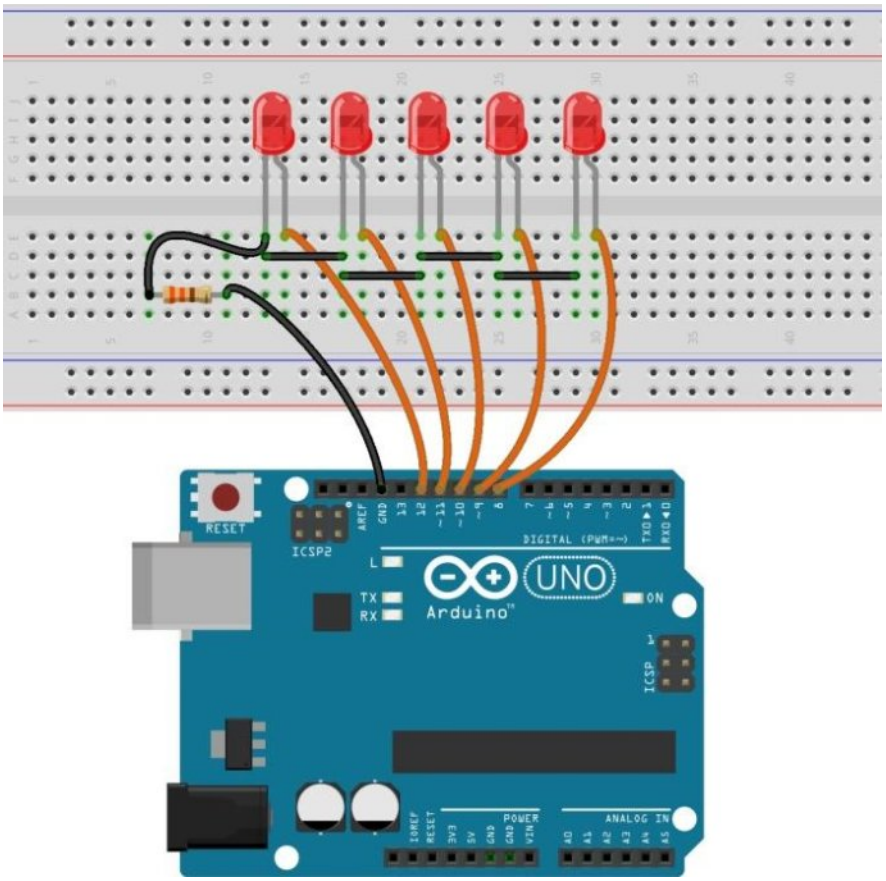
1	Arduino UNO o similar
1	protoboard
5	LEDs
1	R = 330 Ω (o també R = 220 Ω)
	cables

Circuit elèctric:

El LED es pot connectar directament al pin 13 d'Arduino. Per a qualsevol altre pin 0-12 ha de connectar-se una R = 330 Ω per no malmetre el LED.



Muntatge:



Codi:

```
/*  ENCESA DE 5 LEDs CICLICAMENT

   Encès i apagat de 5 LEDs simultàniament
-------------------------------------------------------------------
*/

int pin = 0 ;                       // Inicialitzem la variable “pin” amb un valor enter

void setup()
{
/* Suma 1 a la variable " pin ". Repeteix aquest bucle des de pin = 8 fins pin = 12 per identificar els 5 pins de sortida*/
 for (pin = 8 ;  pin < 13 ; pin++)
    pinMode(pin, OUTPUT);
}

void loop()                         // Encén i apaga els 5 LEDs i torna a repetir el cicle
{
   for (pin = 8 ; pin < 13 ; pin ++)    // Suma 1 a la variable " pin ". Repeteix aquest bucle des de pin = 8 fins pin = 12
    {
       digitalWrite(pin, HIGH) ;
       delay (500) ;                    // Espera mig segon
       digitalWrite(pin, LOW);
       delay (500) ;                    // Espera mig segon
    }

}
```

Referències:

- for (initialization; condition; increment)

3. Cotxe fantàstic

Objectiu: Encendre una seqüència de 5 LEDs simultàniament en dos sentits creant un deixant lluminós.

Introducció d’un array

Els components, circuit i muntatge és el mateix que l’exemple anterior

Codi:

```
/*  COTXE FANTÀSTIC

   Encès i apagat de 5 LEDs simultàniament en dos sentits creant
   un deixant lluminós.
   Encén el LED següent abans que s'apagui l'anterior per mantenir dos
   LEDs encesos consecutivament.
-------------------------------------------------------------------
```

```
int LEDArray[] = {8, 9, 10, 11, 12};              // Array de valors per identificar tots els pins amb els LEDs
int nPin = 0;                                     // Comptador del número de pin a 0. Indica la posició de l'array
int temps = 30;                                   // Temporitzador

void setup()
{
 for (nPin =0; nPin <5; nPin ++)                  // Configurem tots els pins de cop
 {
    pinMode(LEDArray [nPin], OUTPUT);
 }
}

void loop()
{
 for (nPin =0;  nPin <5;  nPin ++)                // Encén els LEDs en ordre creixent creant un deixant lluminós.
 {
    digitalWrite(LEDArray [nPin], HIGH);          // Encén el primer LED i espera un temps encès
    delay(temps);
    digitalWrite(LEDArray [nPin + 1], HIGH);      // Encén el segon LED i espera el mateix temps encès
    delay(temps);
    digitalWrite(LEDArray [nPin], LOW);           // Apaga el primer LED i espera el doble de temps apagat
    delay(temps*2);
 }

 for (nPin =5;  nPin >0;  nPin --)                // Encén els LEDs en ordre decreixent creant un deixant lluminós
 {
    digitalWrite(LEDArray [nPin], HIGH);
    delay(temps);
    digitalWrite(LEDArray [nPin - 1], HIGH);
    delay(temps);
    digitalWrite(LEDArray [nPin], LOW);
    delay(temps*2);
 }
}
```

Referències:

- Array

4. Encès/apagat gradual d'un LED

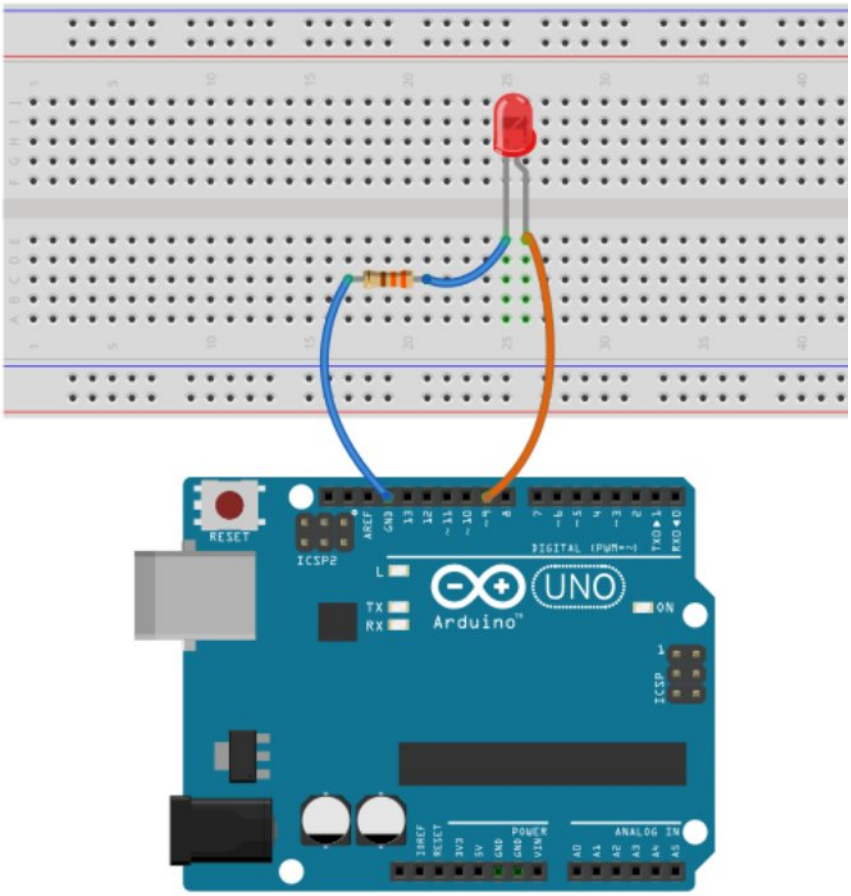
Objectiu: Encendre i apagar un LED de manera gradual utilitzant un pin amb sortida analògica tipus PWM (Pulse Width Modulation)

Components:

1	Arduino UNO o similar
1	protoboard
1	LED
1	R = 330 Ω (o també R = 220 Ω)
	cables

Muntatge:

Connectem el LED, amb la seva R = 330 Ω, a qualsevol pin (per exemple el pin 9) amb sortida analògica PWM indicat amb el símbol ~ .



Codi:

```
/*  ENCESA GRADUAL D'UN LED

   Encès i apagat de manera gradual d'un LED
   Regula la intensitat de llum del LED en funció d'un valor PWM
   El LED es connecta a un pin PWM
   El programa pot fer canviar la velocitat d'encès i apagat canviant el valor de la variable velocitat
   ------------------------------------------------------------------------------------------------
*/

int LED1 = 9;                          // Variable per guardar el nº del pin on va connectat el LED
int velocitat = 10;                    // Variable per canviar la velocitat d'encès/apagat
int pwm;                               // Variable que emmagatzema un valor entre 0 i 255 inclosos

void setup()
{
  pinMode(LED1, OUTPUT);               // Configura el pin LED1 com sortida
}

void loop()
{
  for(pwm=0;  pwm<256;  pwm++)         // Des de pwm = 0 fins pwm = 255 incrementa +1 a pwm
  {
     analogWrite(LED1, pwm);           // Envia a LED1 el valor de la variable pwm
```

```
    delay(velocitat);                    // Espera el valor de "velocitat", en milisegons abans de tornar a començar
                                         // el bucle "for"
 }

 for(pwm=255;  pwm>-1; pwm--)            // Aquest bucle fa el mateix però decrementant pwm
 {
   analogWrite(LED1, pwm);
   delay(velocitat);
 }
}
```

Referències:

- PWM
- analogWrite ()

5. Encesa i apagat gradual de 3 LEDs simultàniament

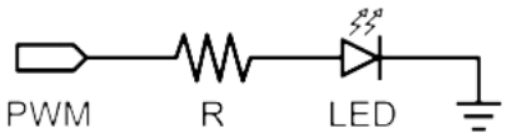
Objectiu: Encendre i apagar tres LEDs gradualment utilitzant pins amb sortida analògica tipus PWM
Introducció de l'ordre " if ".

Components:

1	Arduino UNO o similar
1	protoboard
3	LEDs
1	R = 330 Ω (o també R = 220 Ω)
	cables

Circuit elèctric

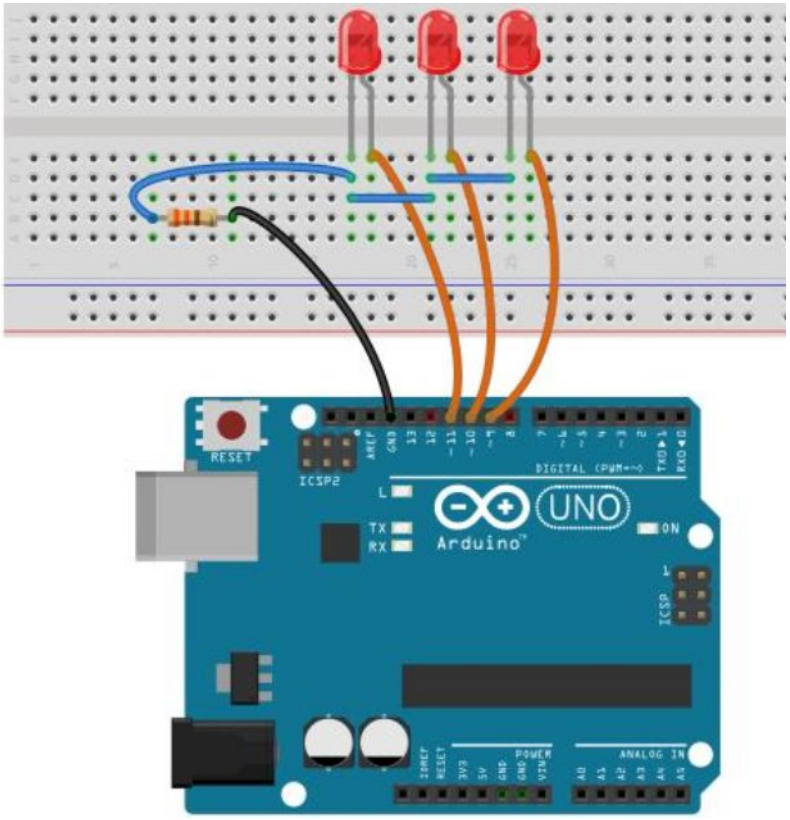
La Modulació d'Amplada d'Impulsos (PWM) és una manera d'aconseguir una sortida pròxima a una analògica a partir de 256 valors diferents.



Muntatge:

Així doncs, connectem els LEDs als pins 9, 10 i 11 a les sortides analògiques PWM.

Podem utilitzar només una resistència perquè hi haurà únicament un LED encès en tot moment.



Codi:

```
/*  ENCESA GRADUAL SUCCESSIVA DE 3 LEDS

   Encès i apagat de 3 LEDs simultàniament i gradualment en dos sentits.
   S'encén i apaga gradualment un LED i a continuació s'encén el següent
-----------------------------------------------------------------------
*/

int LEDArray [] = {9, 10, 11};                  // Array de valors per identificar tots els LEDs
int nPin = 0;                                   // Número de pin corresponent a cada LED que anirà de 0 a 2
int velocitat = 10;                             // Velocitat en que s'encendrà i apagarà un LED
int pwm;                                        // Valor que variarà entre 0 i 255

void setup()
{
 for (nPin =0; nPin <4; nPin ++)                // Configura el mode de 3 pins PWM com sortida
 {
    pinMode(LEDArray[nPin], OUTPUT);
 }
 nPin = 0;                                      // Reinicia la variable "nPin"
}

void loop()
{
 for(pwm=0;  pwm<256;  pwm ++)
 {
    analogWrite(LEDArray[nPin], pwm);
    delay(velocitat);
 }
```

```
 for(pwm=256;  pwm>-1;  pwm --)
 {
    analogWrite(LEDArray[nPin], pwm);
    delay(velocitat);
 }

 nPin ++;                                    // Incrementa el nº de LED

 if (nPin == 3) nPin = 0;                    // Condició: Quan nPin = 3 reinicia nPin
}
```

Referències:

- if (condicional) i ==

6. Polsadors

Objectiu: Encendre un LED mentre es prem un polsador

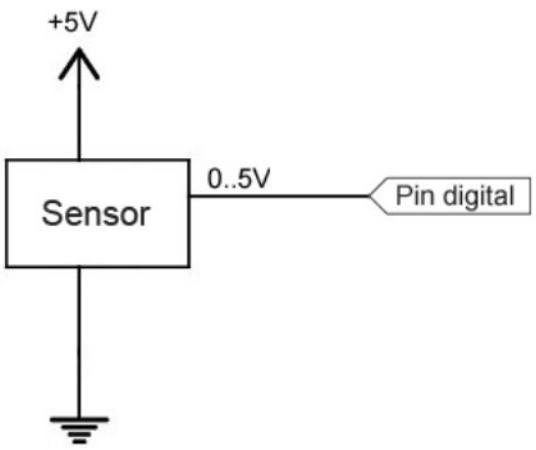
Utilitzar les entrades digitals

Components:

1	Arduino UNO o similar
1	protoboard
1	LED
1	R_L = 330 Ω (o també R = 220 Ω)
1	R_P = 10kΩ
	cables

Circuit elèctric

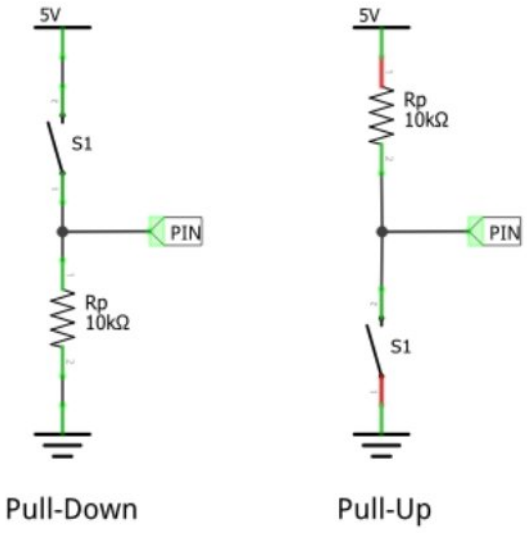
La lectura d'un sensor digital, amb lectures tot o res (0V o 5V), a un pin de la placa d'Arduino es fa de la següent manera:



Resistències Pull-Down i Pull-Up

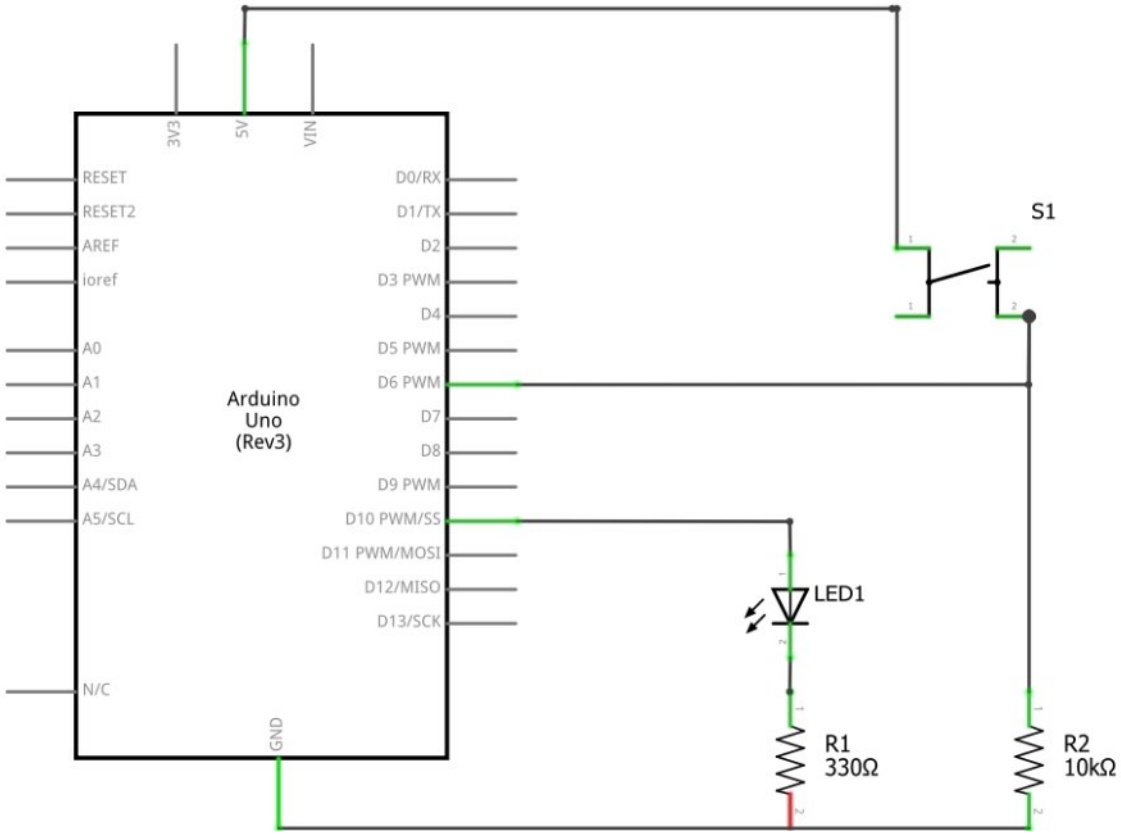
Connectar una pota d'un polsador a Vcc o a terra (GND) i l'altra directament al microcontrolador no és aconsellable ja que es generen sorolls elèctrics i pot donar errors a les lectures dels valor enviats. Així, per evitar aquests casos s'utilitzen les resistències Pull- Down i Pull-Up.

En el cas d'un polsador, la connexió necessària i més senzilla serà de la següent manera pels dos casos Pull-Down i Pull-Up:



Segons la configuració i la posició del polsador, el voltatge a l'entrada del pin (Vpin) d'Arduino serà:

	S1 Obert	S1 Tancat
Pull-Down	Vpin = 0V	Vpin = 5V
Pull-Up	Vpin = 5V	Vpin = 0V

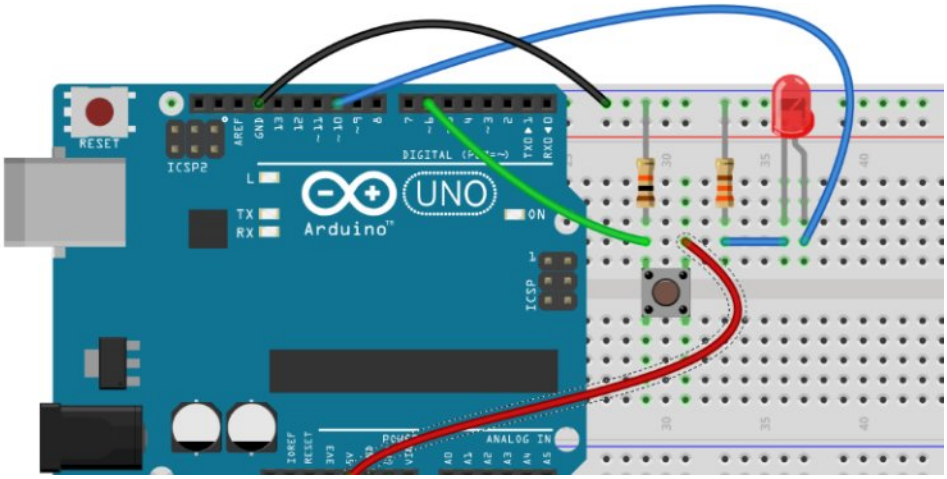


El polsador S1 té quatre pins. Això és perquè cada entrada de l'interruptor té dos pins connectats. Al nostre circuit ignorem els altres dos pins.

Muntatge:

Connectem el LED al pin 10 amb una R_L = 330 Ω i el polsador al pin 6 amb una R_P = 10kΩ.

El polsador, la resistència R_P, el LED i la resistència R_L els connectem de la següent manera:



Codi:

```
/*  POLSADOR

   Encès d'un LED al premer un polsador S1
-----------------------------------------------------------------
*/

int LED = 10 ;
int boto = 6;

void setup()
{
  pinMode(LED, OUTPUT) ;           // LED, pin configurat com sortida OUTPUT
  pinMode(boto, INPUT) ;           // botó, pin configurat com entrada INPUT
}

void loop()
{
  int valor = digitalRead(boto) ;   // Guarda la lectura del valor de botó a la variable "valor"
  digitalWrite(LED, valor) ;        // Canvia l'estat del LED en funció de la variable "valor"
}
```

Referències:

- digitalRead ()

Variant al codi

Si suposem que inicialment el LED està encès i s'apaga al polsar el botó, aleshores, el codi quedaria de la següent manera:

```
void loop()
{
  int valor = digitalRead(boto) ;    // Guarda la lectura del valor de botó a la variable "valor"
  digitalWrite(LED, !valor) ;        // Canvia al valor oposat de la variable "valor" amb el símbol " ! "
}
```

Referències:

- ! (not)

7. Polsador II

Objectiu: Encendre i apagar un LED cada cop que es prem un polsador.
El LED canviarà d'estat cada cop que es polsi el polsador
L'operador negació " ! ".

Els components, circuit i muntatge són els mateixos que l'exemple anterior. Només modificarem una part del codi.

Codi:

```
/*  POLSADOR II

   S'encén un LED al prémer un polsador S1 i s'apaga al prémer el polsador per segon cop
   El LED canvia d'estat cada cop que es polsa el polsador
-----------------------------------------------------------------------
*/

int LED = 10 ;
int boto = 6;
bool estatActual = 0;                    // Variable booleana. Només té dos possibles valors 0 i 1
bool estatAnterior = 0;
int pulsacions = 1;                      // La variable "pulsacions" la inicialitzem amb el valor 1 per no haver de
                                         // prémer dos cops per encendre el LED la primera vegada

void setup()
{
  pinMode(LED, OUTPUT) ;                 // LED, pin configurat com sortida OUTPUT
  pinMode(boto, INPUT) ;                 // botó, pin configurat com entrada INPUT
}


void loop()
{
  estatActual = digitalRead(boto);

  if (estatAnterior != estatActual)       // Si hi ha un canvi d'estat, aleshores ...
  {
    if (estatActual == HIGH) pulsacions++;  // ... si el polsador està polsat, incrementa +1 a la variable "pulsacions"
      estatAnterior = estatActual;          // ... i iguala les dues variables d'estat a l'últim valor
  }

   /* L'operador " % " dóna el residu de la divisió entre dos valors
     per tant, si és 0 significa que el dividend (en aquest cas "pulsacions" és parell.
     Interessa comparar-lo amb un nombre parell per indicar la quantitat de pulsacions necessàries per
     retornar a l'estat inicial ON-OFF --> ON-OFF*/
  if (pulsacions % 2 == 0){               // Si el residu entre "pulsacions" i 2 és 0 ...
     digitalWrite(LED, HIGH);             // ... encén el LED
  } else {                                // si no ...
     digitalWrite(LED, LOW);              // ... apaga el LED
  }
}
```

Referències:

- bool o boolean
- != (not equal to)
- % (modulo)
- if ()... else

8. Lectura d'una sortida digital

Objectiu: Llegir l'estat d'una sortida digital. Per exemple, per guardar l'estat d'un LED encès o apagat.

Utilitzant digitalRead(pin) i digitalWrite(pin, valor):

- Millorar el codi de la pràctica 1, BLINK,
- Millorar el codi de la pràctica 7, Polsador II

Els components, circuit i muntatge són els mateixos que els corresponents a les pràctiques BLINK i Polsador II. Només modificarem una part del codi.

Millora del codi BLINK

```
/* BLINK II

Intermitent d'un LED amb un temps d'espera d'1 segon
Millora del codi BLINK utilitzant la lectura d'una sortida digital
-------------------------------------------------------------------------------
*/

int LED = 13;                          // Definim el pin 13 amb la variable "LED"
bool valor = 0;                        // Definim la variable "valor" com booleana. Dos estats 0 o 1

void setup()
{
  pinMode(LED, OUTPUT);                // LED, pin configurat com sortida OUTPUT
}

void loop()
{
  valor = digitalRead (LED);           // Guardem l'estat (LOW, HIGH) del pin LED a la variable "valor"
  digitalWrite(LED, !valor);           // Envia al pin LED el valor oposat al llegit anteriorment
  delay (1000) ;                       // Espera 1 segon
}
```

Millora del codi Polsador II

```
/*  POLSADOR II

   S'encén un LED al prémer un polsador S1 i s'apaga al prémer el polsador per segon cop
   El LED canvia d'estat cada cop que es polsi el polsador
   Millora del codi Polsador II utilitzant la lectura d'una sortida digital
-----------------------------------------------------------------------------------------
*/

int LED = 10 ;
int boto = 6;
bool estatActual = 0;                  // Variable booleana. Només té dos possibles valors 0 i 1
bool estatAnterior = 0;

void setup()
{
  pinMode(LED, OUTPUT) ;               // LED, pin configurat com sortida OUTPUT
```

```
  pinMode(boto, INPUT) ;                    // botó, pin configurat com entrada INPUT
}

void loop()
{
  estatActual = digitalRead(boto);

  if (estatAnterior != estatActual)         // Si hi ha un canvi d'estat, aleshores ...
  {
    if (estatActual == HIGH)                // ... i si el polsador està polsat,...
    {
      digitalWrite(LED, !digitalRead(LED)); // ... llegeix l'estat del LED i envia al mateix LED el valor oposat
    }
    estatAnterior = estatActual;            // ... i iguala les dues variables d'estat amb l'últim valor
  }
  delay (100);                              // Breu espera per polsar el botó i evitar que canviï varis cops mentre
                                            // s'està polsant
}
```

Com que hi ha un condicional dins un altre es poden ajuntar utilitzant l'operador &&, que equival al compliment de dos condicional a l'hora. D'aquesta manera, el codi es podria simplificar encara més:

```
void loop()
{
  estatActual = digitalRead(boto);

  /*
  Si hi ha un canvi d'estat i el polsador està polsat aleshores llegeix l'estat del LED i envia al mateix LED el valor oposat.
  Com que només hi ha una instrucció dins el condicional es pot escriure a la mateixa línia i sense les claus { }
  */

  if (estatAnterior != estatActual  &&  estatActual == HIGH)   digitalWrite(LED, !digitalRead(LED));
  estatAnterior = estatActual;
  delay (100);
}
```

Referències:

- && (logical and) i operadors booleans

9. El potenciòmetre, entrades analògiques i comunicació sèrie

Objectiu: Llegir la posició d'un potenciòmetre o entrada analògica en general.

Convertir un valor analògic a digital.

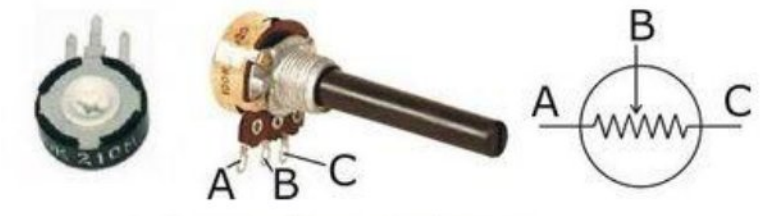
Llegir valors d'entrada i sortida mitjançant el monitor sèrie.

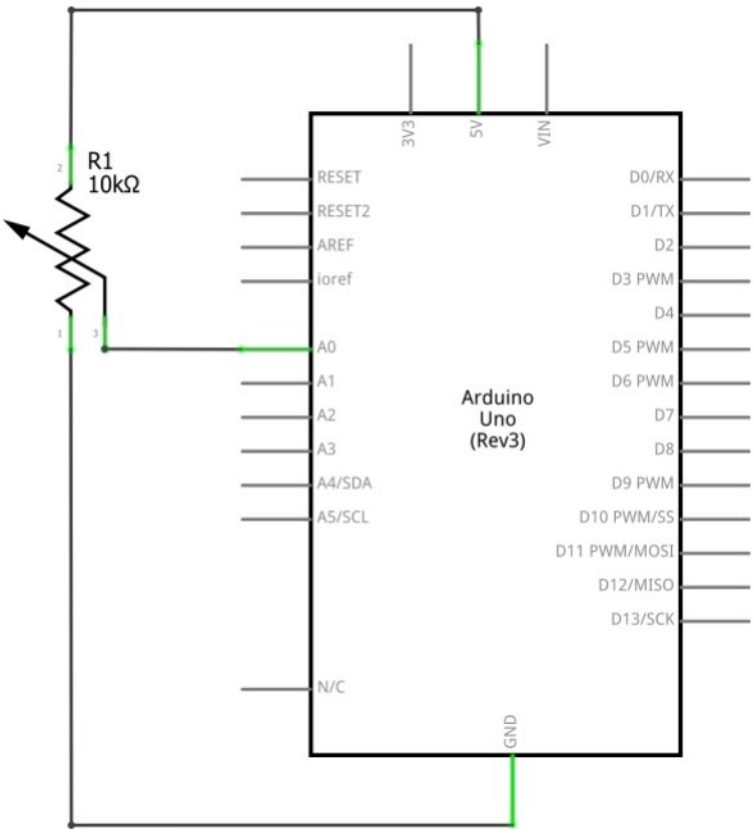
Components:

1	Arduino UNO o similar
1	protoboard
1	Potenciòmetre (o resistència variable) de 10k Ω
	cables

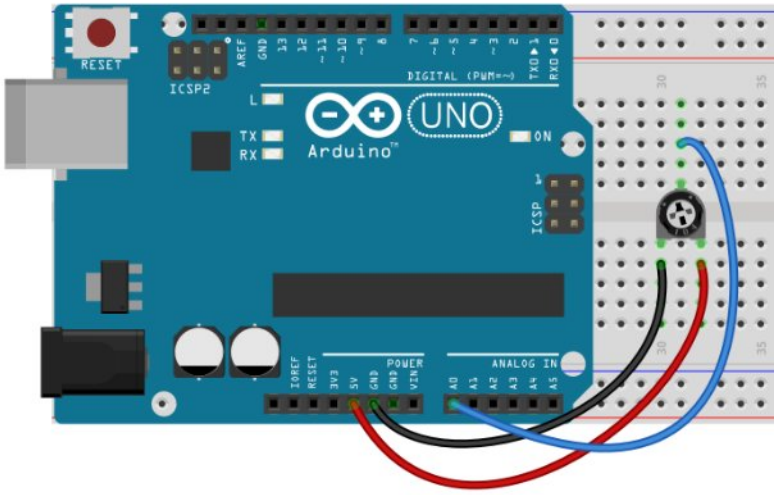
Circuit elèctric

El potenciòmetre o una resistència variable té 3 terminals. La connexió dels terminals extrems (A i C) fa que funcioni com una resistència fixa amb un valor igual a l'indicat i al màxim que pot arribar el potenciòmetre. El terminal del mig (B) juntament amb un de l'extrem es comporta com una resistència variable segons la posició de l'element giratori.





Muntatge:



Codi:

```
/*  POTENCIÒMETRE I MONITOR SÈRIE

   Lectura d'un potenciòmetre connectat a una entrada analògica
   Escriptura del valor llegit en l'entrada analògica al monitor sèrie
-------------------------------------------------------------------------------------
*/

void setup()
{
                                        // Al setup() no farà falta declarar les entrades analògiques
   Serial.begin(9600);                  // Inicialitzem el port sèrie a 9600 bauds
}

void loop()
{
   int Lectura = analogRead(A0);        // Definim la variable "Lectura" com el valor de l'entrada analògica A0
   Serial.println(Lectura);             // Escriu "Lectura" i canvia de línia al monitor sèrie
   delay(200) ;
}
```

Referències:

- analogRead ()
- Serial.begin ()
- Serial.print ()
- Serial.println ()

10. Regulació de la il·luminació d'un LED amb potenciòmetre

Objectiu: Llegir la posició d'un potenciòmetre en un rang de 0 a 1023 i assignar el resultat a un rang de 0 a255

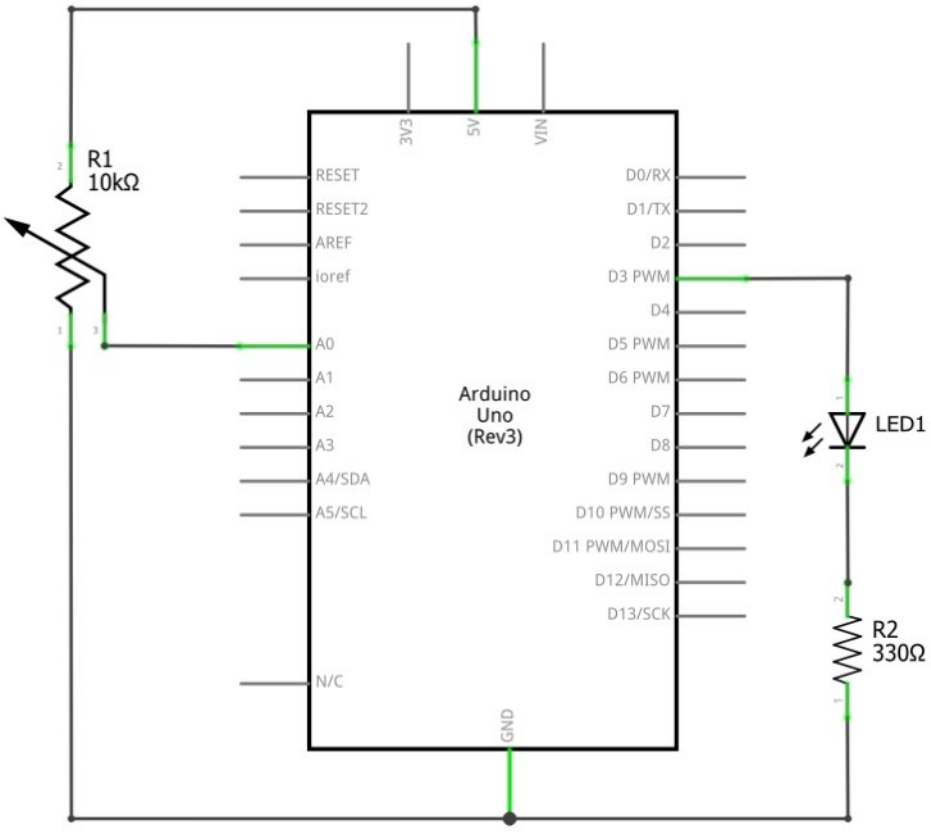
Utilitzar aquest resultat per atenuar o il·luminar un LED

Imprimir els resultats al monitor sèrie

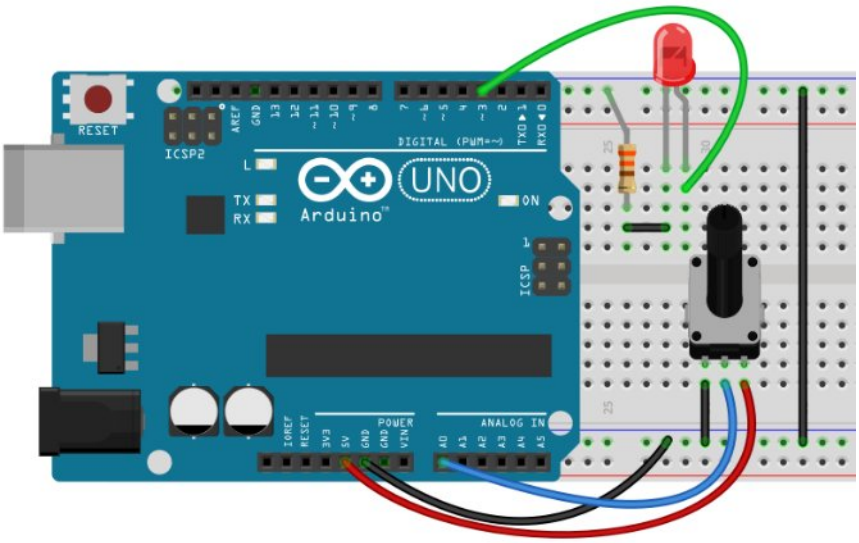
Components:

1	Arduino UNO o similar
1	protoboard
1	LED
1	R_L = 330 Ω (o també R = 220 Ω)
1	Potenciòmetre (o resistència variable) de 10k Ω
	cables

Circuit elèctric



Muntatge



Codi:

```
/*  REGULACIÓ DE LA IL.LUMINACIÓ D'UN LED AMB POTENCIÒMETRE

   Lectura d'un potenciòmetre connectat a una entrada analògica i assigna el resultat a una gamma de 0 a 255
   Escriptura del seu valor al monitor sèrie
----------------------------------------------------------------------------------------------------------------------
*/

const int LED = 3;                          // Pin de sortida analògica connectat al LED.
float valorPot = 0;                         // Valor llegit del potenciòmetre.
                                            // Tipus de dades "float" perquè s'espera el resultat amb decimals
int pwmLED = 0;                             // Valor de sortida PWM amb rang de 0 a 255 per il·luminació del LED

void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);                       // Inicialitzem la comunicació amb el port sèrie a 9600 bauds
}

void loop() {
  valorPot = analogRead(A0);                // Llegeix el valor analògic del potenciòmetre
  pwmLED = map(valorPot, 0, 1023, 0, 255);  // Converteix el rang 0-1023 del potenciòmetre al rang 0-255 pel LED
  analogWrite(LED, pwmLED);                 // Il·lumina el LED amb el valor PWM de la sortida analògica


/* Imprimeix els valors al monitor sèrie */

  Serial.print("Resistencia = ");           // "Resistència = "
  Serial.print(valorPot*10/1023, 3);        // A continuació el valor del potenciòmetre en KΩ i 3 decimals
  Serial.print(" Kohms");                   // "  Kohms"
  Serial.print("\t Sortida PWM = ");        // Un espai de tabulació i "Sortida PWM = "
  Serial.println(pwmLED);                   // El valor PWM i canvia de línia

  delay(20);
}
```

Referències:

- float
- map ()