

Tutorial Arduino i IDE

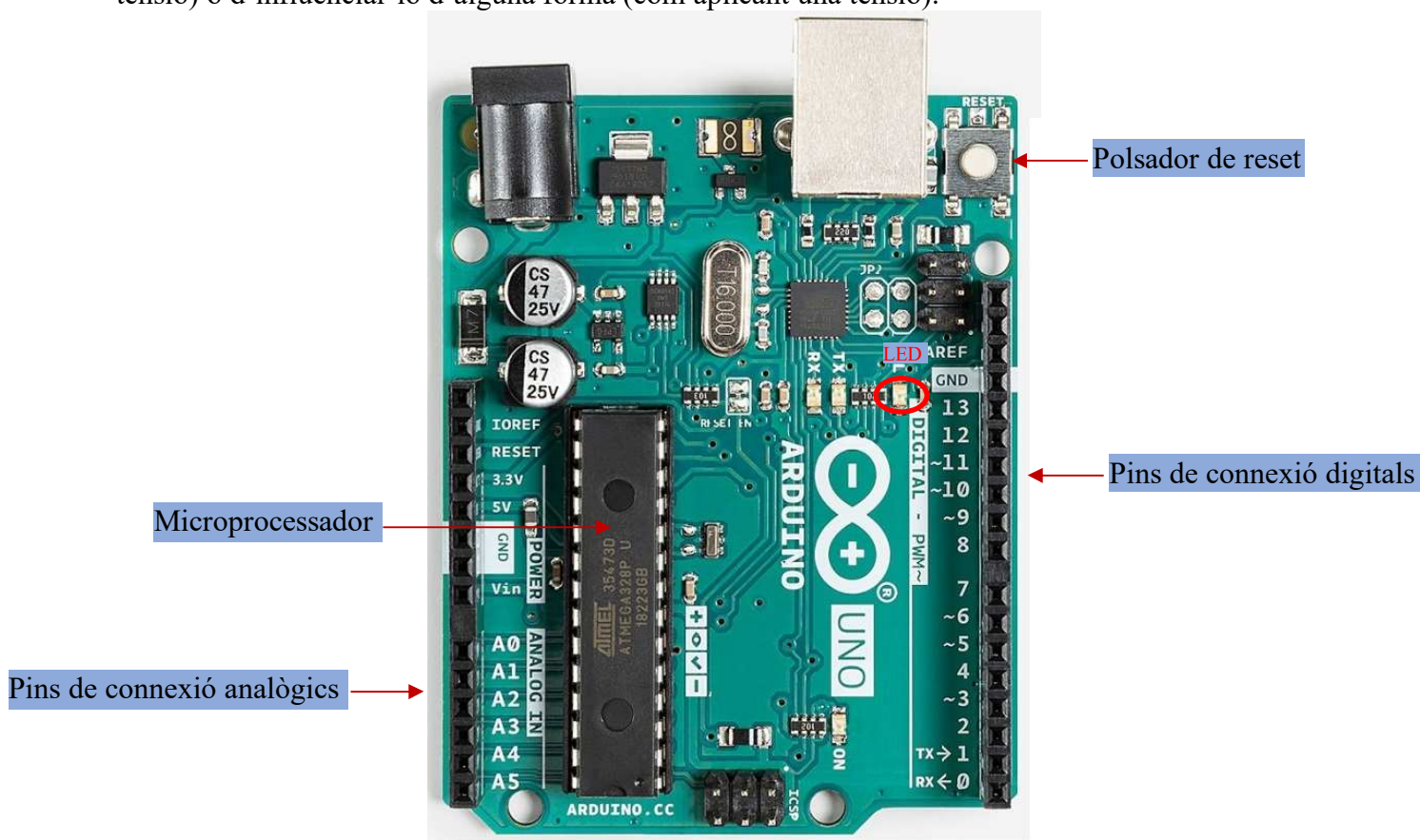
1. INTRODUCCIÓ

En aquest tutorial de pràctiques veurem informació bàsica sobre la placa Arduino Uno que utilitzarem en pràctiques per poder implementar diferents aplicacions basades en mesures electròniques. A més, veurem també el seu entorn de programació (IDE: Integrated Development Environment).

Per informació més detallada, us podeu adreçar als enllaços proporcionats en aquest guió.

2. PLACA D'ARDUINO UNO

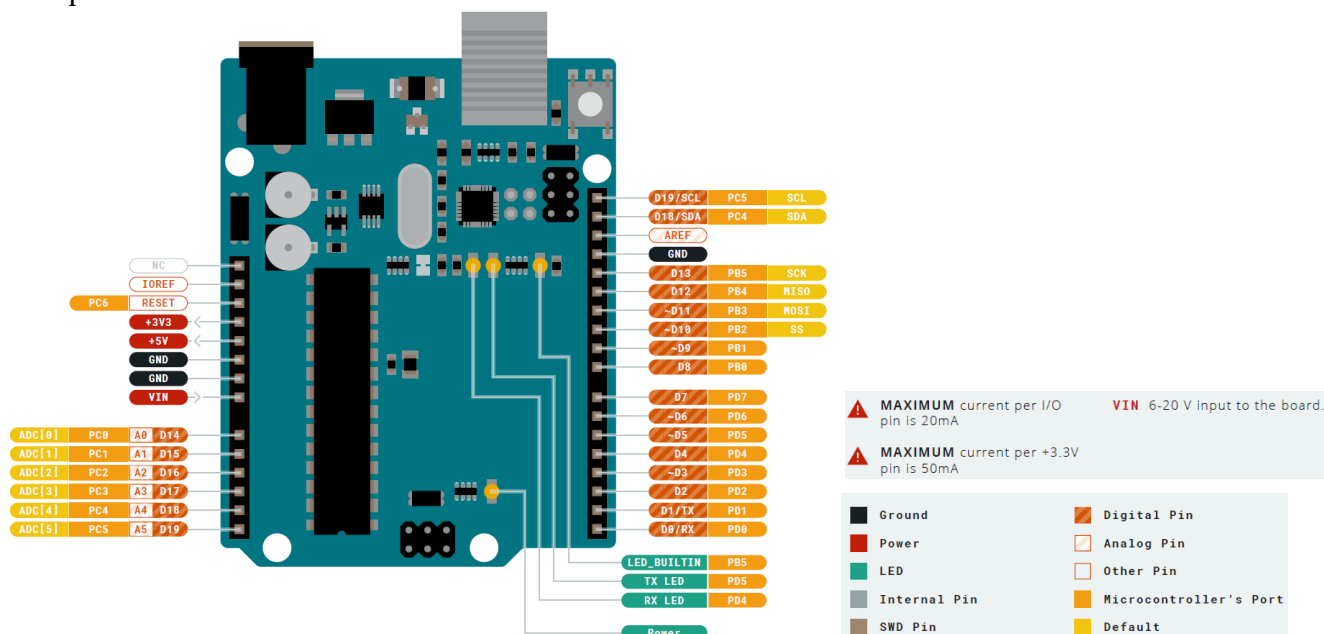
A les pràctiques utilitzarem la placa d'[Arduino Uno](#). Aquesta placa permet programar un microprocessador per poder fer coses com, entre d'altres, mesurar tensions en alguns punts de connexió (pins) (tant analògiques com digitals) i poder també posar una tensió en aquests pins (aquesta placa només permet aplicar tensions digitals, però no analògiques). En certa forma és com tenir un voltímetre (mesura de tensió) o una font de tensió (aplicar tensió). Normalment aquests pins aniran connectats a un circuit extern per tal d'obtenir informació d'aquest circuit (mesurant una tensió) o d'influenciar-lo d'alguna forma (com aplicant una tensió).



Podeu trobar informació detallada d'aquesta placa a la pàgina principal donada al principi d'aquest apartat.

Important: Quan manipuleu la placa, agafeu-la pels costats, procurant de no tocar els components.

Nosaltres utilitzarem sobretot els pins per poder connectar la nostra placa amb un circuit fet per nosaltres. A la següent figura es poden veure la relació dels pins laterals de la placa amb els pins del microprocessador:



Algunes de les coses més importants que necessitem saber d'aquest esquema per les pràctiques són:

- Pins digitals (Dx) són pins digitals d'entrada i/o sortida (es pot configurar programàticament segons les necessitats). Les sortides són com fonts de tensió (ja que apliquem tensió al pin i la podem aplicar al nostre circuit extern) i les entrades són com voltímetres (ja que podem mesurar la tensió a la que s'ha connectat el pin externament).
- Pins analògics d'entrada (Ax) són entrades analògiques, amb les quals podem mesurar una tensió. En aquesta placa no hi ha pins analògics de sortida.
- 'PWM out' (indicats amb ~) indica que en aquests pins digitals es pot generar un senyal PWM (Pulse Width Modulation).
- Haurem d'utilitzar en la programació el número de pin indicat al costat de cada pin de la figura.
- Hi ha un led connectat al pin 13.

3. IDE (entorn de programació)

Podeu trobar tota la informació sobre l'[entorn de programació](#) a l'enllaç proporcionat. Per la seva instal·lació, simplement aneu a 'Download' i descarregueu la versió que millor s'ajusti a vosaltres. Bàsicament, el llenguatge és molt semblant a 'C', amb una llibreria de [funcions](#) per controlar la placa. El codi sempre ha de tenir al menys dos funcions void: `setup()` i `loop()`. La primera funció (`setup()`) només s'executa una vegada al principi de l'execució, i s'utilitza normalment per configurar els recursos de la placa. La segona funció (`loop()`) s'executa indefinidament (quan arriba al final del codi, es torna a executar), i és a on s'escriu el programa en sí.

Algunes de les funcions que utilitzarem més habitualment són:

- *pinMode(pin,mode)*: Configura els pins, per exemple per especificar si és sortida o entrada (amb opció de [pull-up](#)) (*INPUT*, *OUTPUT*, *INPUT_PULLUP*). Exemple: *pinMode(19,OUTPUT)*.
- *analogRead(pin)*: Mesura la tensió analògica que hi ha al pin indicat. Exemple: *analogRead(23)*. Retorna un valor entre 0 i 1024, que es correspon a 0V i 3.3V respectivament.
- *digitalWrite(pin,valor)*: Posa la tensió digital (HIGH o LOW) indicada a valor al pin. Exemple: *digitalWrite(19,HIGH)*.
- *digitalRead(pin)*: Llegeix la tensió digital al pin indicat. Exemple: *digitalRead(18)*. Retorna HIGH o LOW.
- *delay(t)*: Atura l'execució del programa durant el temps *t* indicat (en mil·lisegons). Exemple: *delay(1000)*.
- *millis()*: Ens proporciona el temps en mil·lisegons quan l'executem des de que es va començar a executar el codi.

Es poden [mostrar gràfiques](#) durant l'execució usant l'opció “*Serial Plotter*” a l'apartat “*Eines*”. Per mostrar més d'una corba a la mateixa gràfica s'hauria de fer servir al programa *Print(val_n)* i *Print(", ")* per cada corba, i per l'última corba l'enviarem amb *Println(val_ult)*.

Exemple de programa que mostra a una gràfica dues corbes amb els valors mesurats a una entrada analògica i una constants, però només quan la tensió a una entrada digital té un '0' lògic (0V):

```
//Variables globals
unsigned char pinlect=1;
unsigned char pindig=5;

//Configuració
void setup() {
  Serial.begin(9600);
  pinMode(pindig, INPUT_PULLUP);
}

//Codi
void loop() {
  unsigned int control=0;
  rval=analogRead(pinlect); //Lectura tensió analògica al pin número 'pinlect'
  control=digitalRead(pindig); //Lectura tensió digital (LOW, HIGH) al pin 'pindig'
  //Enviem els valors a la connexió sèrie per graficar-los només si la lectura digital era LOW
  if (control==LOW) {
    Serial.print(rval);
    Serial.print(", ");
    Serial.println(500);
  }
  if (rval>500) Serial.end(); //Finalitzem la connexió sèrie
  delay(100);
}
```

Podeu trobar molt més exemples al mateix programa, anant a ‘*File → Examples*’.