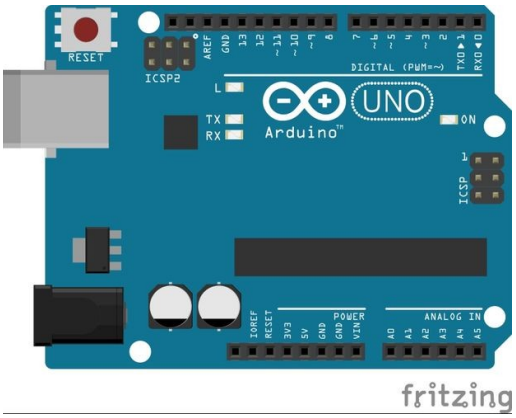
Preguntes i respostes bàsiques sobre Arduino i Raspberry PI

Què és i que no és Arduino?

Arduino **NO** és un ordinador.

Arduino és una petita placa que conté un microcontrolador programable i uns pins d'entrada i sortida. Les plaques Arduino permeten la introducció pel port USB d'un petit programa en el microcontrolador, i aquest microcontrolador executarà el programa cíclicament.

En no ser un ordinador, no té sistema operatiu.



La placa que et mostra la imatge anterior es Arduino Uno, la més emprada, que pots adquirir a partir de 10 euros, però hi ha d'altres plaques, més petites i senzilles, i també hi ha de més grans i potents.

Per què és famós Arduino?

Per que porta molts i molts anys al mercat.

Però sobretot perquè ha estat un disseny «lliure». Les especificacions per construir plaques Arduino estan a l'abast de tothom. La seva llicència oberta permet a qualsevol produir i vendre plaques Arduino.

Això ha fet que sigui econòmic, i molt emprat en projectes de domòtica i art. Alhora, això ha fet que s'escriguin molts llibres i manuals sobre ell. Alhora de nou això fa que sigui molt usat. És un cercle que es retroalimenta.

Com es programa Arduino?

Arduino es programa en llenguatge C.

Els programes s'escriuen a un IDE que pots descarregar a la pàgina

https://www.arduino.cc/en/Main/Software

Aquest IDE compilarà el programa en C i l'enviarà per port USB a executar-se a l'Arduino.

També hi ha un simulador d'Arduino a les pàgines:

https://www.tinkercad.com/#/?type=circuits

https://wokwi.com/

Les tres diferències que té quelcom programat en C per un ordinador de quelcom programat en C per Arduino són:

1) **Els programes escrits per Arduino no poden ser molt llargs**. Un cop compilats el codi màquina no pot ocupar més de 32 kB, que és la capacitat de memòria del microcontrolador en la placa més normal. És a dir, que no podem tenir programes complexos de desenes de milers de línies.

2) **Els programes escrits per Arduino no tenen programa principal *main()*** sinó que el codi està a una funció *loop()* que s'executa repetidament, i una funció *setup()* que s'executa un únic cop abans d'iniciar la funció *loop()*.

És a dir, és com si el programa principal estigués amagat però contingués quelcom semblant a:

```
int main(void) {
  setup();
  while (1) {
    loop();
  }
}
```

3) **Els programes escrits per Arduino no tenen entrada i sortida a consola, sinó que s'envien uns voltatges per un pin determinat o es llegeixen uns voltatges d'un pin determinat**.

Això vol dir que substituïm *scanf()* per:

- La funció *digitalRead(número_pin)*, que llegeix un voltatge LOW (0V) o HIGH (5V o 3.3V, depenent de la placa) al pin especificat, i el retorna com a valor de la funció.

- La funció *analogRead(número_pin)*, llegeix un voltatge representat per un número enter entre 0 (0V) i 1023 (5V o 3.3V, depenent de la placa, però es pot canviar amb la funció *analogReference()*) al pin especificat, i el retorna com a valor de la funció. Tarda 100 microsegons en llegir.

I que substituïm *printf()* per:

- La funció *digitalWrite(número_pin, voltatge)*, que envia un voltatge LOW (0V) o HIGH (5V o 3.3V, depenent de la placa) al pin especificat.

- La funció analog*Write(número_pin, voltatge)*, que envia un voltatge representat per un número enter entre 0 (0V) i 255 (5V o 3.3V, depenent de la placa) al pin especificat.

També es pot fer servir un port sèrie per a E/S, però a aquesta introducció no ho veurem.

Mira aquesta pàgina per aprendre les funcions d'E/S a Arduino:

https://www.arduino.cc/en/Reference/HomePage

Vols saber més d'Arduino?

Visita aquestes pàgines:

- https://www.arduino.cc/

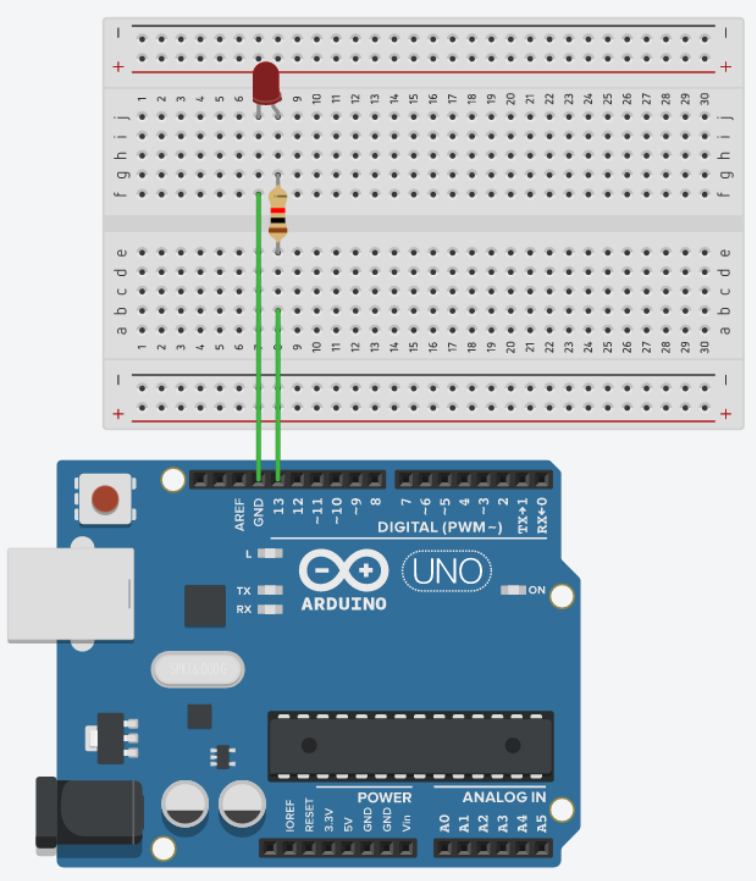
- https://en.wikipedia.org/wiki/Arduino

Exemple d'un programa bàsic a Arduino

A un ordinador , el següent programa C compta d'un a cinquanta, emprant una iteració:

```
int main(void) {
  int i;
  for (i = 1; i <= 50; i++) {
    printf("%d\n", i);
  }
}
```

Podem fer quelcom semblant a Arduino, emprant una iteració per variar la corrent que passa a través d'una llum LED i així incrementar la lluminositat. El muntatge seria quelcom semblant a:



Incís: Perquè no connectem directament el LED als pins 13 i GND de la placa Arduino? Es podria fer però podria arribar a cremar la placa. Pensa que $V = R \times I$, i que el LED presenta resistència quasi nul·la, el que vol dir que aplicat un voltatge la intensitat que circularà serà infinita. Per això emprem aquesta placa blanca per muntar un circuit més extens en que posem una resistència de 220 ohms davant el LED.

Si el volem adaptar a un programa Arduino que progressivament encengui una llum LED, mitjançant sortida analògica, tindríem (-però depenent de la placa, millor fer servir el pin 9-):

```
#define LED_PIN 13                    // Número de Pin connectat al LED

void setup() {
  pinMode(LED_PIN, OUTPUT);           // Configura el pin per sortida
}

void loop() {
  int i;
  for(i = 0; i <= 255; i += 5) {      // Mínim sortida es 0V y máxim 5V
    analogWrite(LED_PIN, i);          // però 5V és analogWrite de 255
    delay(100);                       // Espera cent mil.lisegons
  }
}
```

Sobre aquest exercici pots veure més a https://www.arduino.cc/en/Tutorial/Fade i https://www.arduino.cc/en/Tutorial/Fading

Si el volem adaptar a un programa Arduino que alternativament encengui i apagui una llum LED, mitjançant sortida digital, tindríem:

```
#define LED_PIN 13                    // Número de Pin connectat al LED

void setup() {
  pinMode(LED_PIN, OUTPUT);           // Configura el pin per sortida
}

void loop() {
  digitalWrite(LED_PIN, HIGH);        // Encén el LED (HIGH és 5V)
  delay(1000);                        // Espera 1 segon
  digitalWrite(LED_PIN, LOW);         // Apaga el LED (LOW és 0V)
  delay(1000);                        // Espera 1 segon
}
```

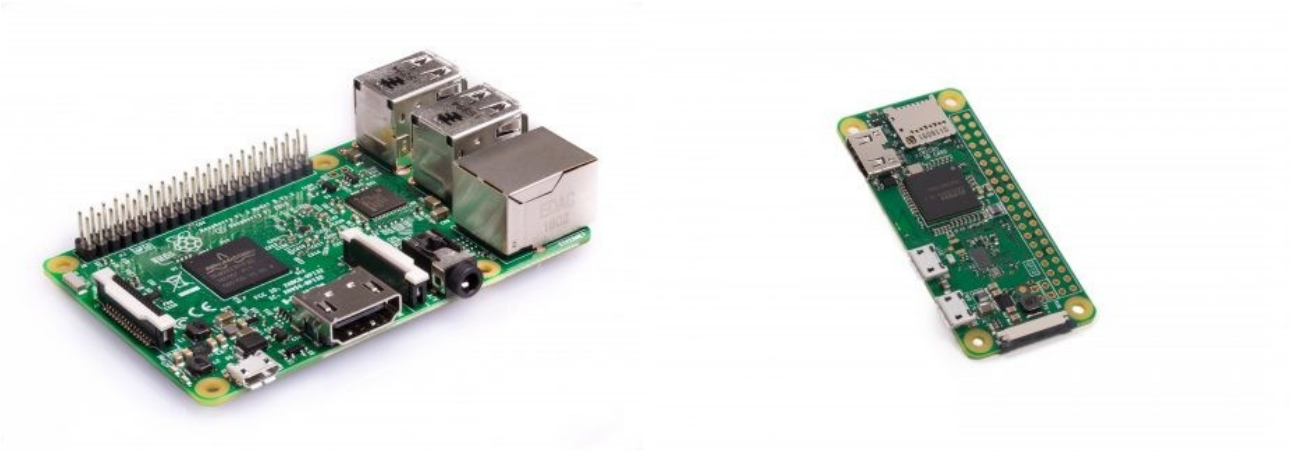
Sobre aquest exercici pots veure més a https://www.arduino.cc/en/Tutorial/Blink

Hi ha una extensa sèrie d'exemples que s'instal·len amb l'IDE d'Arduino i que es poden consultar dins l'IDE, però que també es poden consultar via web a https://www.arduino.cc/en/Tutorial/BuiltInExamples

Què és i que no és Raspberry PI?

Raspberry Pi **SI** és un ordinador. Per uns 45 euros aproximadament té CPU ARM, USBs, xarxa, sortida de vídeo HDMI, etc.

Això vol dir que necessita un sistema operatiu per funcionar, com qualsevol altre ordinador. En no venir amb disc dur, el sistema operatiu s'instal·la en una targeta MicroSD. El més emprat és Raspbian, però hi ha d'altres. Els oficials els pots trobar i descarregar a la pàgina https://www.raspberrypi.org/downloads/ , però hi ha molts més, com pots veure a http://www.makeuseof.com/tag/7-operating-systems-you-can-run-with-raspberry-pi/



La placa que et mostra la imatge a l'esquerra es Raspberry PI 3, i la de la dreta Raspberry PI Zero, a un cost molt inferior (5 euros sense wireless i 10 euros amb wireless i bluetooth).

Hi ha plaques similars d'altres fabricants, però amb potència superior a millor preu. Per exemple, a mi m'agraden les ODROID (http://www.hardkernel.com/). Llavors, per què comprar una Raspberry PI? Per què té una comunitat enorme, revistes, dotzenes de llibres, i els sistemes operatius miren de tenir disponibles controladors que reconeguin i facilitin l'ús de tot el seu hardware. Pensa que el que pagues no és el hardware, sinó tot aquest suport, que no té preu.

Per saber més de Raspberry PI llegeix https://en.wikipedia.org/wiki/Raspberry_Pi

Llavors, per què fer servir Raspberry PI?

Si Raspberry PI és com qualsevol altre ordinador, per què fer-la servir? Per què no fer servir un ordinador més potent?

A banda del baix preu, hi ha una diferencia entre Raspbery PI i un altre ordinador, i és que Raspberry PI té uns PINS d'entrada i sortida anomenats GPIO, similars als de Arduino, malgrat no tant versàtils, que la fan adient per projectes de domòtica i de «Internet of Things».

https://en.wikipedia.org/wiki/Raspberry_Pi#General_purpose_input-output_.28GPIO.29_connector

https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/

https://www.raspberrypi.org/documentation/usage/gpio/

Dels 40 pins de la GPIO, 26 poden interactuar amb aplicacions i controlar dispositius, mentre que la resta són pins de voltatge i terra. El voltatge màxim admès és de 3.3V. Els pins no estan protegits contra sobrevoltatge. Treballar amb un voltatge més gran pot cremar la teva Raspberri Pi.

Com es programa Raspberry PI?

Com qualsevol altre ordinador. En qualsevol llenguatge de programació. Raspberry PI és un ordinador amb un sistema operatiu, i qualsevol programa escrit per un PC hauria de funcionar igual a una Raspberry PI. El llenguatge de programació és indiferent: Java, C, Python, etc.

La única diferència és que, a més de poder compilar i executar qualsevol programa com a qualsevol altre ordinador, també podem escriure programes que facin us del GPIO. Per això haurem d'escollir un llenguatge de programació, i a continuació descarregar les corresponents llibreries d'E/S GPIO per al llenguatge escollit.

Al nostre institut iniciem a la programació sobretot amb llenguatge C a primer i amb Java a segon, malgrat també toquem una mica de PHP, Python o Bash scripting, per lo que em centraré a parlar dels dos primers llenguatges. Com que ja hem explicat com programar Arduino en C, per la Raspberry PI els exemples que utilitzaré aquí seran en Java. Tot i així, comento breument alguna coseta pels que programen en C puguin cercar informació:

Trobem vàries maneres de treballar amb el GPIO de la Raspberry emprant llenguatge C:

- La primera és intentar accedir directament als pins d'E/S a traves de registres. És la manera més directa però també la que porta més feina. Pots trobar una mica d'informació a http://www.raspberrypi.org/forum/educational-applications/gertboard/page-4/#p31555

- La segona és intentar accedir directament als pins d'E/S a traves de la interfície sysfs. https://www.kernel.org/doc/Documentation/hwmon/sysfs-interface i https://en.wikipedia.org/wiki/Sysfs

- La tercera i més senzilla és emprar alguna llibreria que faciliti l'accés. Per exemple http://wiringpi.com/ o també http://abyz.me.uk/rpi/pigpio/

Trobareu un exemple de codi de cadascuna a https://elinux.org/RPi_GPIO_Code_Samples#C

Programar en Java el GPIO de Raspberry PI

Encara per fer ...

https://elinux.org/RPi_GPIO_Code_Samples#Java

Llibreries a mirar:

http://pi4j.com/

https://github.com/jkransen/framboos

https://github.com/nkolban/jpigpio

https://github.com/mattjlewis/diozero

Programar en Python el GPIO de Raspberry PI

Encara per fer ...

https://realpython.com/python-raspberry-pi/

https://www.raspberrypi.org/magpi-issues/Essentials_GPIOZero_v1.pdf

https://gpiozero.readthedocs.io/en/stable/recipes.html