EPITECH. TECHNOLOGY

# SMART TRASH CANS

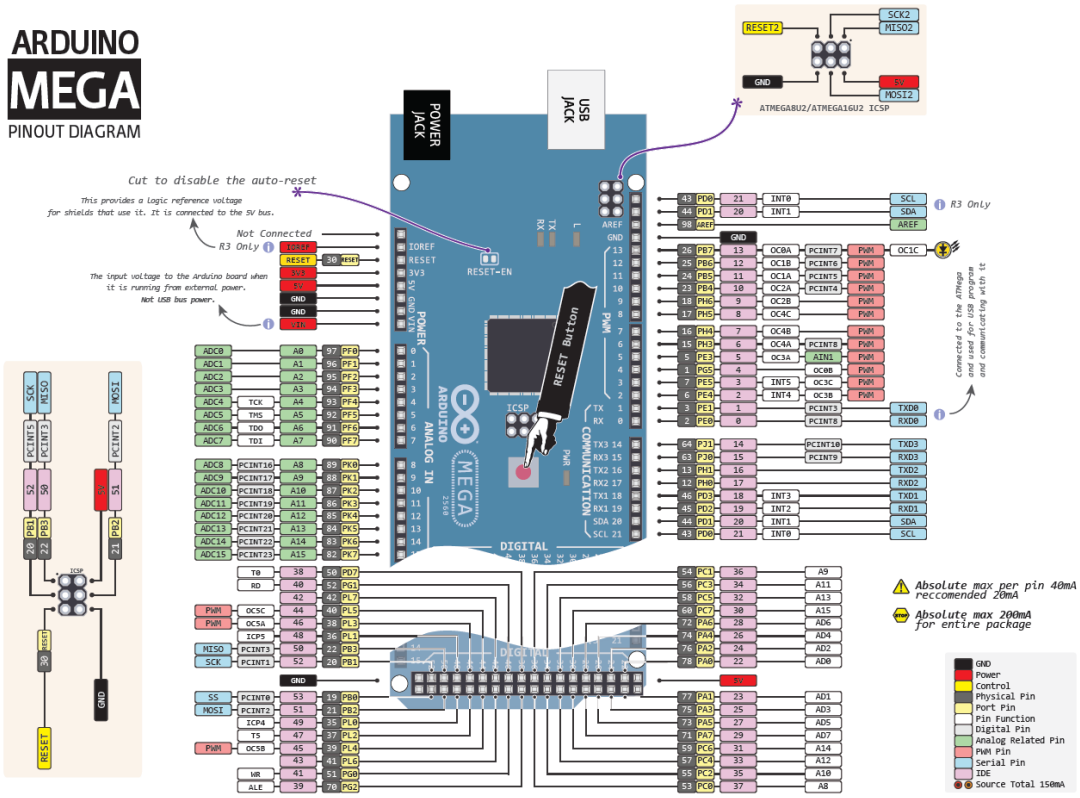
## BOOSTRAP 1

# SMART TRASH CANS

## IoT

> It's not an experiment if you know it's going to work.
>
> **— Jeff Bezos —**

The objective here is to discover your new *Arduino* card, of which here is the pinout:




Visiting the official page of the electronic products encountered is always useful and necessary. Find out what you're going to be dealing with and the vocabulary of the electronic world.

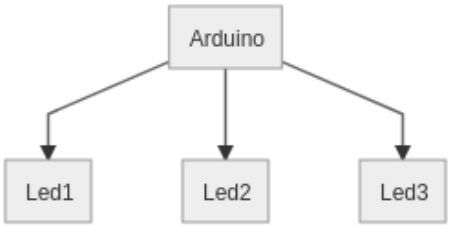
Knowing a lot of documentation can be both time-consuming and unnecessary.

## Game 1

In this first exercise, refer to the page *Getting started*. Find out how to upload a compiled file to your *Arduino*. Search and find useful documents for the rest of the exercises.

Check the pin connections to avoid short circuits.

## Game 2

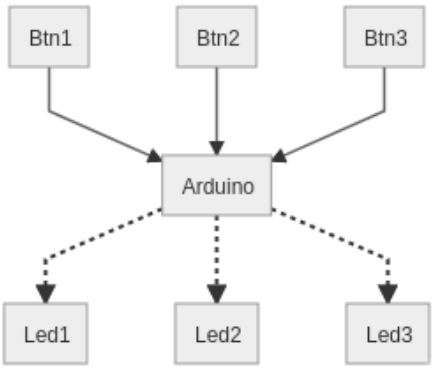


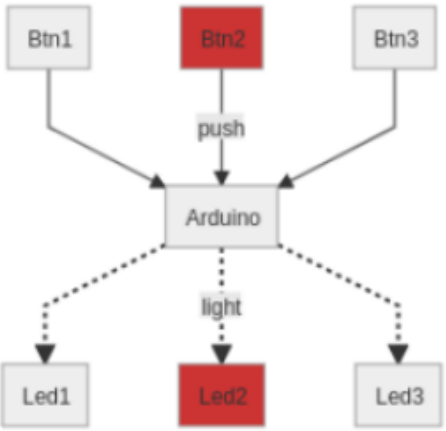

Play with the LEDs: turn them on:

- ✓ one after the other (timer) ;
- ✓ depending on the “chenillard” effects (light intensity).

## Game 3

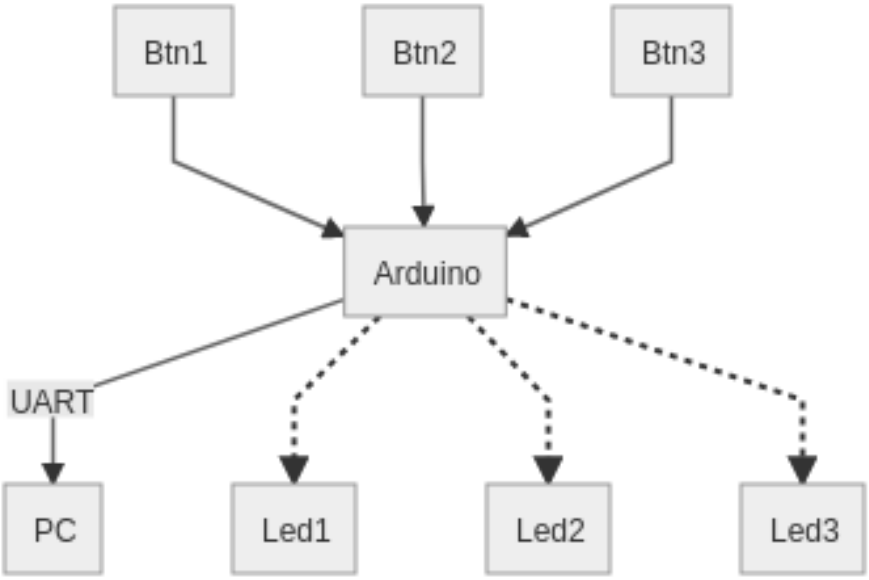
Connect buttons to the LEDs. Animate the latter depending on the state of the former.



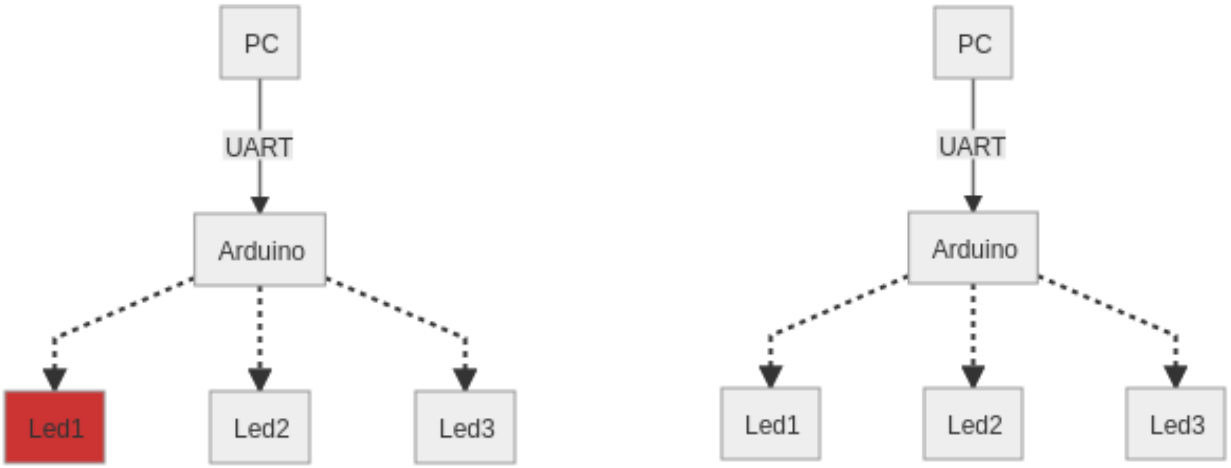
## Game 4

On the serial terminal of the Arduino IDE, display a string indicating which LED is lighting up.




## Game 5

Write a string of the type: `LED1 ON` OR `LED1 OFF` in the serial terminal to turn on|off the led.

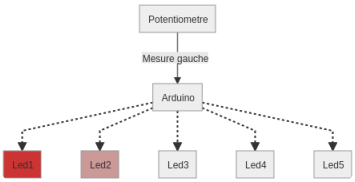


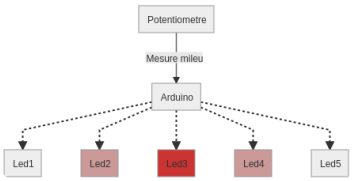
## Game 6

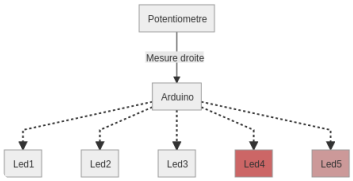

Display the distance measurement made by the ultrasonic sensor on the serial terminal.

## Game 7

Discover the potentiometer: on a light indicator composed of several LEDs, “move” the light signal along the chaser according to the movements of the potentiometer.



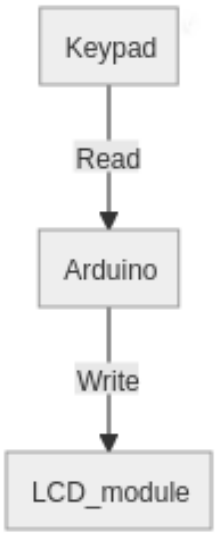




## Game 8


Same as before: a motor is positioned according to the movements of the *potard*.

Find out about the different types of motors *servo, stepper, DC, Brushless*, …

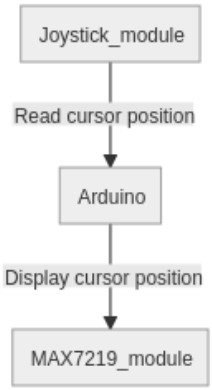
## Game 9




Read the digits pressed on a keypad and display them on the LCD module.

Assign operations to buttons: `* <-> delete; # <-> add; ...`

## Game 10




Read the position of the joystick on 2 axes and display the closest corresponding LED on the LED matrix.

The main objectives are:

- ✓ discover *Unity* ;
- ✓ become familiar with its interface ;
- ✓ compile a first application.

Here is an algorithm to help you with this bootstrap:

1. Online research should guide you.
2. If you cannot find an answer to your problem, ask your neighbor.
3. If they are not available or do not have the answer, check out the Forum .
4. If none of these solutions work, repeat step 1 more willingly.

## Games 1 & 2

Install the latest version of Unity and then start a 3D project. In the latter, two objects make a revolution (speed *v1*) around a common center of rotation.

Then, have each object perform its own rotation (speeds *v2* and *v3*), around their center of gravity (or not !?). Then control the animation with keyboard keys :

- ✓ stop/restart the revolution of an object ;
- ✓ stop/restart the proper rotation of an object ;
- ✓ speed up/slow down ;
- ✓ modify the position of the centers of revolution/rotation ;
- ✓ do lots of other amazing things…

A single script is allowed for the 3 *gameobjects*, so you must skillfully *expose* their speeds in the *inspector*.

## Game 3

Now build your mobile application, *Android* or *iOS* (requires Xcode). Create a *button* to stop/restart the revolution of objects.

Then control the revolution of objects with the *touch* on the phone.

## Game 4

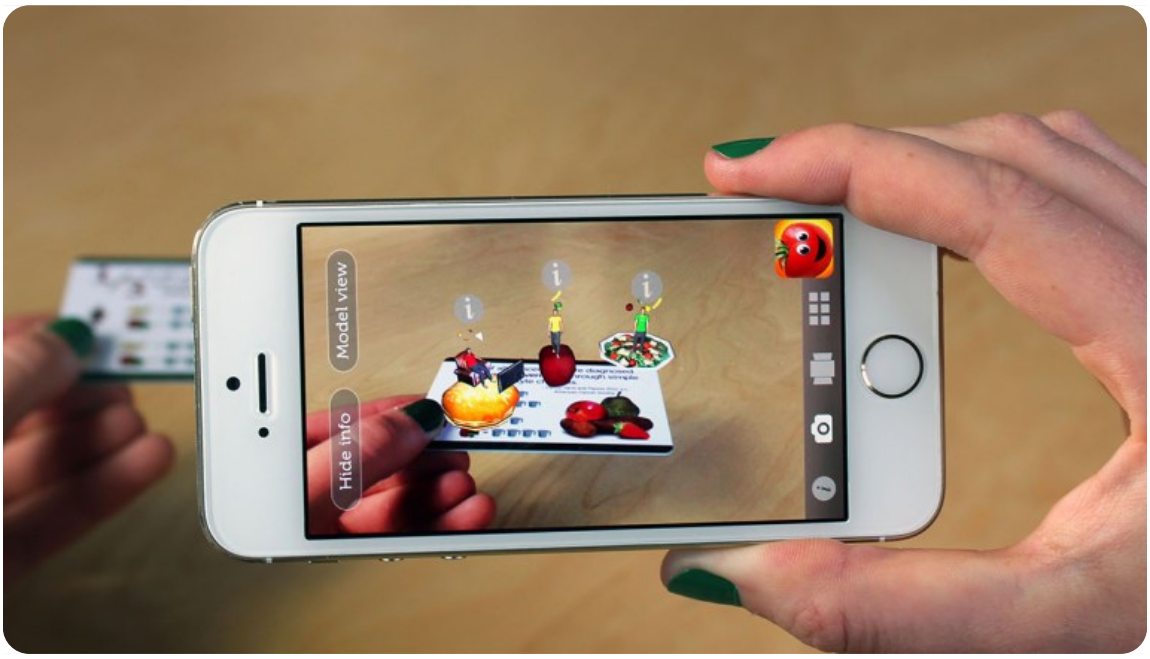
Let's get virtual! *Unity* integrates *AR Foundation*, a framework for augmented reality. Its advantage is to serve as an interface between the developer and the SDKs of the different manufacturers (ARCore, ARKit, Magic Leap XR and Windows).

Take control of this framework: put a cube on the ground in *AR* and create 3 sliders that control its color.

*samples* and information are available on the Unity Github account or on Learn, unity's learning platform.

## Game 5

Eventually, put one (or more) 3D model(s) in *AR* on one of your cards (student, identity, …).




Then do the same on **at least** 4 different flat objects (images, QRcode, map, …).

You will have to find a way to link objects with their image conterpart. If only there was some king of *dictionary*.

v 1.4