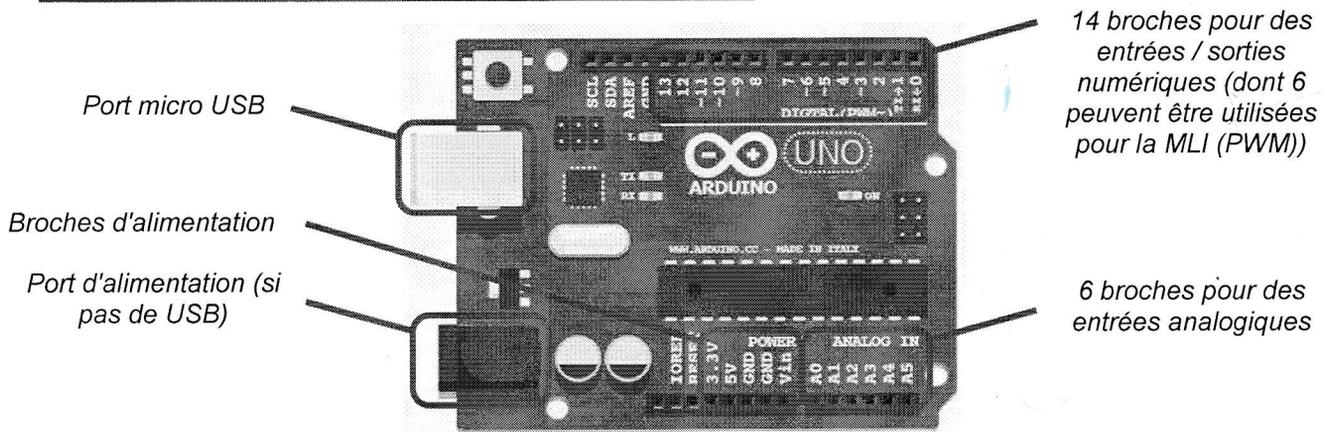


Objectif(s) :

- ✓ Décrire et analyser le comportement d'un système.
- ✓ Traduire le comportement d'un système.

# LANGAGE DE PROGRAMMATION ARDUINO

## I. BROCHAGE DE LA CARTE ARDUINO UNO



## II. PRINCIPALES INSTRUCTIONS

Structures	Variables	Fonctions
void setup() void loop()  <b>Structures de contrôle :</b> if if ... else for while do ... while break continue return goto  <b>Opérateurs arithmétiques :</b> = (assignement d'une valeur) + (addition) - (soustraction) * (multiplication) / (division)  <b>Opérateurs de comparaison :</b> == (égal à) != (différent de) < (inférieur) <= (inférieur ou égal à) > (supérieur à) >= (supérieur ou égal à)  <b>Opérateurs logiques :</b> && (et)    (ou) ! (non)	<b>Constantes :</b> HIGH (état haut) LOW (état bas) INPUT (entrée) OUTPUT (sortie)  <b>Type de données :</b> void (vide) boolean (booléen) char (caractère) byte (octet) int (entier) float (nombre décimal)	<b>Logiques :</b> pinMode() digitalRead() digitalWrite()  <b>Analogiques :</b> analogRead() analogWrite()  <b>Gestion du temps :</b> millis() micros() delay() delaymicroseconds()  <b>Mathématiques :</b> min() max() abs() constrain() map() pow() sqrt()  <b>Communication série :</b> Serial.begin() Serial.print() Serial.println()

D'autres instructions sont disponibles, ainsi que de nombreuses bibliothèques. Le lien internet <https://www.arduino.cc/en/Reference/HomePage> permet d'obtenir de l'aide sur l'ensemble des instructions.

Le lien internet <https://www.arduino.cc/> propose également de nombreux tutoriels.

### III. PRINCIPALES STRUCTURES

Structure	Algorithme	SYNTAXE ARDUINO
SI ... ALORS ... SINON ...		<pre>... if (Condition 1) { Action A } else { Action B } ...</pre>
SI ... ALORS ... SINON SI ... ALORS ... SINON ...		<pre>... if (Condition 1) { Action A } else if (Condition 2) { Action B } else { Action C } ...</pre>
SI ... ALORS ...		<pre>... if (Condition 1) { Action A } ...</pre>
FAIRE ... JUSQU'A ...		<pre>... do { Action A } while (Condition 1);</pre> <p><i>(Remarque : le langage Arduino ne permet pas la structure « Faire ... jusqu'à », mais il permet la structure « Faire ... tant que », du coup il faut utiliser le complément de la condition 1.)</i></p>
TANT QUE ... FAIRE ...		<pre>... while (Condition 1) { Action A } ...</pre>
POUR ... FAIRE ...		<pre>... for (variable = valeur initiale; <sup>variable</sup> * = valeur finale; variable = variable + pas) { Action A } ...</pre>