



Cet enseignement s'inscrit dans le prolongement de l'enseignement d'algorithmique, d'informatique et de programmation dispensé au collège en mathématiques et en technologie. On approfondit ces notions et cette pratique de la programmation à travers les activités liées aux thèmes du programme : internet ; le *Web* ; les réseaux sociaux ; les données structurées et leur traitement ; localisation, cartographie et mobilité ; informatique embarquée et objets connectés ; la photographie numérique.



# Thèmes du programme

## Thématiques du programme

- \* *Internet*
- \* *Le Web*
- \* *Les réseaux sociaux*
- \* *Les données structurées et leur traitement*
- \* *Localisation, cartographie et mobilité*
- \* *Informatique embarquée et objets connectés*
- \* *La photographie numérique*

L'enseignement de sciences numériques et technologie en classe de seconde a pour objet de permettre d'appréhender les principaux concepts des sciences numériques, mais également de permettre aux élèves, à partir d'un objet technologique, de comprendre le poids croissant du numérique et les enjeux qui en découlent. La numérisation généralisée des données, les nouvelles modalités de traitement ou de stockage et le développement récent d'algorithmes permettant de traiter de très grands volumes de données numériques constituent une réelle rupture dans la diffusion des technologies de l'information et de la communication. Cette révolution multiplie les impacts majeurs sur les pratiques humaines.



Par exemple, l'actuel mobile multifonction est un objet technologique qui permet, comme le téléphone du XXe siècle, de téléphoner, mais qui sert également à bien d'autres activités : envoyer des messages, photographier, filmer, enregistrer, chercher et partager une information, écouter de la musique, regarder des vidéos, repérer où l'on se trouve, réserver des billets de train, vérifier son rythme cardiaque, programmer le chauffage de son appartement, etc. Ainsi, il est devenu une interface universelle d'accès à l'information et de commande d'autres objets.



Malgré leur grande variété, ces avancées se fondent toutes sur l'universalité et la flexibilité d'un petit nombre de concepts en interaction :

- \* les **données**, qui représentent sous une forme numérique unifiée des **informations** très diverses : textes, images, sons, mesures physiques, sommes d'argent, etc. ;
- \* les **algorithmes**, qui spécifient de façon abstraite et précise des traitements à effectuer sur les données à partir d'opérations élémentaires ;
- \* les **langages**, qui permettent de traduire les algorithmes abstraits en **programmes** textuels ou graphiques de façon à ce qu'ils soient exécutables par les machines ;
- \* les machines, et leurs systèmes d'exploitation, qui permettent d'exécuter des programmes en enchaînant un grand nombre d'instructions simples, assurent la persistance des données par leur stockage et de gérer les communications. On y inclut les objets connectés et les réseaux.

À ces concepts s'ajoute un élément transversal : les interfaces qui permettent la communication avec les humains, la collecte des données et la commande des systèmes



Cet enseignement a vocation à multiplier les occasions de mise en activité des élèves, **sous des formes variées** (exposés, travaux en groupe, mini-projets, productions individuelles ou collectives, etc.) qui permettent de développer des compétences transversales :

- \* faire preuve d'autonomie, d'initiative et de créativité ;
- \* présenter un problème ou sa solution, développer une argumentation dans le cadre d'un débat ;
- \* coopérer au sein d'une équipe ;
- \* rechercher de l'information, apprendre à utiliser des sources de qualité, partager des ressources ;
- \* faire un usage responsable et critique des sciences et technologies numériques.

The screenshot displays the mBlock software interface, which is used for programming microcontrollers like the Arduino. The window title is "mBlock - 创客工场(v3.3.0) - Serial Port Connected - Not saved". The interface is divided into several sections:

- Scripts Panel:** Contains various block categories such as Motion, Looks, Sound, Pen, Data&Blocks, Events, Control, Sensing, Operators, and Robots. The "Robots" category is currently selected.
- Block Palette:** Lists the available blocks for the "Auriga" robot, including "Auriga Program bluetooth mode", "run forward at speed", "set motor", "set servo", "set stepper motor", "set encoder motor", "encoder motor on board", "set 7-segments display", "set led on board", "set led strip", "play tone on note", "stop tone", and "show face".
- Script Area:** The main workspace where blocks are assembled into a program. The visible script consists of:
  - "Auriga Program bluetooth mode" (dropdown)
  - "run forward at speed 100" (dropdown)
  - "wait 1 secs" (dropdown)
  - "run forward at speed 0" (dropdown)
- Code Editor:** Shows the C++ code generated from the blocks:

```
01 }
02
03 double angle_rad = PI/180.0;
04 double angle_deg = 180.0/PI;
05
06
07
08 void setup(){
09   attachInterrupt(Encoder_1.GetIntNum(), isr_process_encoder1,
10   RISING);
11   attachInterrupt(Encoder_2.GetIntNum(), isr_process_encoder2,
12   RISING);
13   move(1,100);
14   delay(1000*1);
15   move(1,0);
16
17 }
18
19 void loop(){
20
21   Encoder_1.Update_speed();
22   Encoder_2.Update_speed();
23
24
25 }
```
- Serial Monitor:** Displays the output of the program, showing binary data received from the robot:

```
4-27 17:47:17.352 < 56 65 72 73 69 6f 6e 3a 20 30 39 2e : binary mode
30 31 0d 0a
4-27 18:24:11.232 < 56 65 72 73 69 6f 6e 3a 20 30 39 2e 30 31 2e 30
30 31 0d 0a
```



# Notions transversales de programmation

Un langage de programmation est nécessaire pour l'écriture des programmes : un langage simple d'usage, interprété, concis, libre et gratuit, multiplateforme, largement répandu, riche de bibliothèques adaptées aux thématiques étudiées et bénéficiant d'une vaste communauté d'auteurs dans le monde éducatif est nécessaire. Au moment de la conception de ce programme, le langage choisi est Python version 3 (ou supérieure).

Contenus	Capacités attendues
Affectations, variables Séquences Instructions conditionnelles Boucles bornées et non bornées Définitions et appels de fonctions	Écrire, exécuter et mettre au point un programme.
<b>Exemples d'activités</b>	
— Illustrer ces notions par des activités liées aux différents thèmes du programme.	

Au collège (cycle 4), les élèves ont découvert et pratiqué les éléments fondamentaux d'algorithmique et de programmation. Le programme de seconde de mathématiques approfondit l'apprentissage de la programmation. Une coordination avec le cours de mathématiques est donc nécessaire pour déterminer à quel moment des éléments de programmation peuvent être utilisés en sciences numériques et technologie.

The image displays two software windows side-by-side. The left window is the Arduino IDE, titled 'Arduino\_Py | Arduino 1.8.3', showing a C++ sketch for controlling an LED. The code includes comments in French and a loop that reads serial data from a Python program and toggles an LED. The right window is a Python IDE titled 'python pgm.py - D:\Work Table\Arduino with Python\Simple LED communicatio...', showing the corresponding Python code that establishes a serial connection, reads data from the Arduino, and controls the LED. A 'Python 2.7.9 Shell' window is overlaid on the Python code, showing the execution output: 'Hi!, I am Arduino' followed by a prompt to enter 1 to turn the LED on or 0 to turn it off. The shell shows successful execution for both inputs.

```
Arduino_Py | Arduino 1.8.3
File Edit Sketch Tools Help
Arduino_Py
/*
 * Code to blink an LED using Python
 * Code by: Aswint Raj, Dated: 8-9-2017
 * Website: www.circuitdigest.com
 */

int data;

void setup() {
  Serial.begin(9600); //initialize serial COM at 9600 baudrate
  pinMode(LED_BUILTIN, OUTPUT); //make the LED as output of the Arduino
  digitalWrite (LED_BUILTIN, LOW);

  Serial.println("Hi!, I am Arduino");
}

void loop() {
  while (Serial.available()){
    data = Serial.read();
  }

  if (data == '1')
    digitalWrite (LED_BUILTIN, HIGH);

  else if (data == '0')
    digitalWrite (LED_BUILTIN, LOW);
}

Ln: 23 Col: 0
```

```
python pgm.py - D:\Work Table\Arduino with Python\Simple LED communicatio...
File Edit Format Run Options Windows Help
#Program to Control LED of Arduino from Python
#Code by: Aswint Raj, Dated: 8-9-2017
#Website: www.circuitdigest.com

import serial #Serial module used for Serial communication
import time #Required module for delay functions

ser = serial.Serial('COM18',9600) #Create Serial port object called as ser
ser.write('Hi!') #Send the message for the communication to get established

while True:
    print(ArduinoSerial.read()) #read the serial data and print it as line
    print(ArduinoSerial.read() to turn ON LED and 0 to turn OFF LED")

    if (var == '1')
        ArduinoSerial.write('1')
        time.sleep(1)

    if (var == '0')
        ArduinoSerial.write('0')
        time.sleep(1)

var = raw_input() #get input from user
print "you entered", var #print the input for confirmation

if (var == '1')
    ArduinoSerial.write('1')
    print ("LED turned ON")
    time.sleep(1)

if (var == '0')
    ArduinoSerial.write('0')
    print ("LED turned OFF")
    time.sleep(1)

Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Hi!, I am Arduino

Enter 1 to turn ON LED and 0 to turn OFF LED
1
you entered 1
LED turned ON
0
you entered 0
LED turned OFF
1
you entered 1
LED turned ON
Ln: 23 Col: 0
```

L'enseignement de sciences numériques et technologie aide à mieux comprendre les enjeux scientifiques et sociétaux de la science informatique et de ses applications, à adopter un usage réfléchi et raisonné des technologies numériques dans la vie quotidienne et à se préparer aux mutations présentes et à venir de tous les métiers.