



# Capteurs de luminosité et Conditionnelle

Programmation par bloc - 1 période

Programmation - Conditionnelle - Capteurs - Lumière - Météo

## Compétences

### Mathématique

- Interpréter un tableau de nombres, un graphique, un diagramme
- Vérifier le résultat d'une opération

### Éducation par la technologie

- Imaginer des dispositifs expérimentaux simples et prendre des initiatives
- Dans le cadre d'une énigme, agencer les indices en vue de formuler au moins une question, une supposition ou une hypothèse
- Élaborer un concept, un principe, une loi
- Formuler des hypothèses à partir de l'observation d'un phénomène, d'une information médiatisée, d'un événement fortuit... pour préciser une énigme à résoudre
- Imaginer des dispositifs expérimentaux simples et prendre des initiatives
- Proposer une solution à l'énigme et la confronter avec la situation de départ
- Proposer au moins une piste de résolution possible
- Réfléchir aux pratiques mises en œuvre, évaluer la démarche suivie
- Confirmer ou infirmer un raisonnement par des arguments vérifiés
- Réaliser une brève communication orale, un petit exposé scientifique sur les résultats d'une recherche, en utilisant un média

---

## Objectifs

L'élève sera capable de :

- Manipuler un système tangible (micro:bit)
- Utiliser une interface de programmation par blocs
- Comprendre le concept de structure conditionnelle
- Trouver et reconnaître, dans l'interface de programmation, les blocs liés aux structure conditionnelle et capteur de luminosité
- Manipuler des structures conditionnelles dans un exercice donné





- Comprendre ce qu'est un capteur de lumière (présent dans de nombreux objets numériques) et ses potentialités
- Exploiter les potentialités d'un capteur de luminosité dans un exercice donné
- Résoudre un problème donné et proposer une solution
- Tester une solution
- Comprendre pourquoi une solution ne fonctionne pas et la corriger
- Reconnaître, à travers les solutions des autres, les variantes de solution possibles
- Savoir expliquer sa solution (ses choix)

## Description de l'activité

Une importante partie de la programmation est de pouvoir dire "Quand" un programme doit effectuer une certaine action. Les instructions conditionnelles permettent de le faire en indiquant une condition dans laquelle une action sera faite et éventuellement ce que le programme doit faire si la condition est fausse. En programmation, les structures conditionnelles font, avec les variables, partie des briques de bases sur lesquelles reposent tous les langages de programmation.

Lors de cette activité, les élèves devront créer un affichage météo en utilisant le capteur de lumière.


## Matériel et ressources

Matériel nécessaire :	Ressources nécessaires :
<ul style="list-style-type: none"> <li>• Par élève (ou groupe de 2)                             <ul style="list-style-type: none"> <li>– 1 micro:bit </li> <li>– 1 câble </li> <li>– 1 ordinateur</li> </ul> </li> <li>• 1 ordinateur et un projecteur pour le professeur (ou tableau interactif)</li> </ul>	<ul style="list-style-type: none"> <li>• Fiche explicative - La boussole, le thermomètre et les capteurs de luminosité du micro:bit</li> <li>• Fiche explicative - Les conditionnelles</li> <li>• feuille d'aide "L'affichage météo"</li> </ul>



## Déroulement

### Étape 1 : Théorie (10 min)

 *Si plus de temps est disponible, il peut être très intéressant de prévoir des activités en "débranché" pour découvrir les conditionnelles.*

Qui	Fait quoi	Matériel
Le professeur	Explique le concept de conditionnelle	
Le professeur	Explique ce qu'est un capteur de lumière et ses potentialités	
Le professeur	Présente, sur l'application web, les différents blocs correspondant aux conditionnelles et au capteur de lumière	ordinateur et projecteur
Le professeur	Montre un exemple de condition (par exemple : si la lumière est $\geq 40$ , il sourit, sinon, il boude)	ordinateur et projecteur

### Étape 2 : Programmation de l'affichage météo (20-30 min)

Qui	Fait quoi	Matériel
Le professeur	Distribue la feuille d'aide "L'affichage météo" aux élèves	feuille d'aide "L'affichage météo"
Le professeur	Distribue un micro:bit (avec le code pré-installé) et un câble micro USB par élève	micro:bit et câbles USB
Les élèves	Programment l'affichage météo en suivant les instructions données sur la feuille d'aide	ordinateur et feuille d'aide
Les élèves	Testent leur solution et la corrigent si besoin	micro:bit, câble USB et ordinateur

### Étape 3 : Conclusion (5-10 min)

Qui	Fait quoi	Matériel
Les élèves	Échangent de place avec d'autres élèves/groupes et testent leurs solutions	micro:bit, câble USB et ordinateur
Les élèves	Regardent le code qui est devant eux et essayent de repérer les différences avec le leur	micro:bit, câble USB et ordinateur
Tous	Discutent des différentes solutions possibles	

### Pour aller plus loin

Si les élèves ont bien compris comment exploiter les possibilités offertes par les conditionnelles et les capteurs, ils peuvent rajouter de nouvelles fonctionnalités (voir "pour aller plus loin" sur la feuille d'aide).



## Solutions & Problèmes

L'élève est libre dans la résolution de l'activité mais voici un exemple de solution :

Si la luminosité est au-dessus de 40,  
ALORS affiche un Soleil.

SINON SI la luminosité est au-dessus de 15,  
ALORS affiche un nuage

SINON affiche une lune.

Les capteurs de luminosité renvoient une valeur entre 0 et 255. Une valeur supérieure à 40 correspond à une luminosité normale en intérieur.

En cliquant sur l'engrenage du bloc 'SI', il est possible de créer des blocs 'SI...ALORS ... SINON SI...' ('IF...THEN ... ELSE IF...' en anglais) permettant de créer un code plus clair et plus simple comme ceci :

Les élèves peuvent ensuite rajouter des fonctionnalités liées au thermomètre comme l'affichage de la valeur par ex. :

Si la luminosité est au dessus de 40  
ET la température au dessus de 15,  
ALORS affiche un parapluie



## Évaluation

### Manipuler un système tangible (micro:bit) [./1]

	non	oui
Appuyer sur les boutons	0	1

### Utiliser une interface de programmation par blocs [./0.5]

	non	oui
Ajouter et emboîter des blocs	0	0.5

### Comprendre le concept de structure conditionnelle [./4]

	non	oui
Prendre conscience qu'une structure conditionnelle sert à faire une action si une condition est remplie (si... alors... (sinon...))	0	2
Prendre conscience que les conditions sont soit vraies soit fausses (aucune autre possibilité)	0	1
Prendre conscience que plusieurs structures conditionnelles peuvent se suivre (et être emboîtée)	0	1

### Trouver et reconnaître, dans l'interface de programmation, les blocs liés aux structures conditionnelles et au capteur de lumière [./2.5]

	non	oui
Trouver le menu contenant les blocs liés aux structures conditionnelles	0	0.5
Reconnaître les blocs de condition et leur différence (si-alors et si-alors-sinon)	0	0.5
Savoir que les blocs '<' et '=' servent à comparer des éléments	0	0.5
Savoir que les blocs 'et' et 'ou' servent à tester plusieurs conditions en même temps	0	0.5
Trouver le bloc du "niveau d'intensité lumineuse"	0	0.5

### Manipuler des structures conditionnelles dans un exercice donné [./3]

	non	oui
Utiliser un si-alors-sinon	0	1
Utiliser une comparaison comme condition	0	1
Emboîter un 'si' dans un 'sinon'	0	1

### Comprendre ce qu'est un capteur de lumière (présent dans de nombreux objets numériques) et ses potentialités [./1]

	non	oui
Prendre conscience que ça capte la lumière	0	0.5
Prendre conscience que la mesure est entre 0 et 255	0	0.5



**Exploiter les potentialités du capteur de lumière dans un exercice donné [./1]**

	non	oui
Trouver le bloc "niveau d'intensité lumineuse"	0	0.5
Afficher la luminosité	0	0.5

**Résoudre un problème donné et proposer une solution [./3]**

	non	oui
Afficher un soleil si la luminosité est supérieure à 40	0	1
Afficher un nuage si la luminosité est entre 15 et 40	0	1
Afficher une lune si la luminosité est inférieure à 15	0	1

**Tester une solution [./1]**

	non	oui
Tester la solution (sur le micro:bit ou l'émulateur)	0	1

**Comprendre pourquoi une solution ne fonctionne pas et la corriger [./1]**

	non	oui
Ne pas avoir d'erreur ou les corriger s'il y en a	0	1

**Reconnaître, à travers les solutions des autres, les variantes de solution possibles [./1]**

	non	oui
Voir les différences avec d'autres codes qui fonctionnent	0	1

**Savoir expliquer sa solution (ses choix) [./1]**

	non	oui
Expliquer sa solution aux autres	0	1