



Unité 3 : Luminosité, IF, et WHILE

Compétence1 : Mesures de luminosité

Dans cette leçon, nous étudions le capteur de lumière intégré, BRIGHTNESS, et utiliserons l'instruction **DispAt** pour afficher les lectures du capteur.

Objectifs :

- Lire les mesures de Luminosité
- Introduire la boucle **While**

Dans les leçons précédentes, nous n'avons envoyé que des instructions au Hub TI-Innovator™ afin d'avoir un impact sur ses périphériques intégrés (LUMIÈRE, COULEUR et SON).

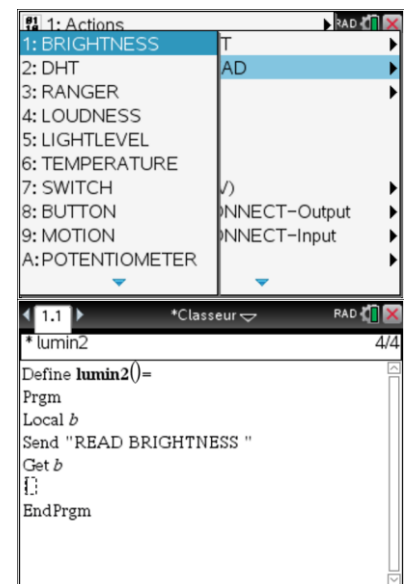
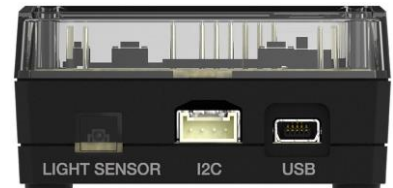
Dans cette unité, nous allons travailler avec le capteur de lumière embarqué, et utiliser la valeur lue dans notre programme pour créer un « compteur de lumière ». Le capteur de lumière produit des valeurs comprises entre 0 et 100 sous forme décimale.

L'obtention de la valeur du niveau de lumière à partir du Hub TI-Innovator nécessite deux instructions :

- **Send "READ BRIGHTNESS"**
- **Get <var>**

Mise au point du programme :

1. Commencer un nouveau programme et le nommer LUMIN1.
2. Nous utiliserons la variable **b** to pour stocker la valeur de la luminosité (BRIGHTNESS), ainsi nous déclarons cette variable comme **Local** dans le programme. Ceci est optionnel.
3. Sélectionner **menu > Hub > Send "READ... > BRIGHTNESS**. Appuyer sur la touche **enter**.
4. Sélectionner **menu > Hub > Get**.
5. Taper le nom de la variable **b** (aucune parenthèse n'est utilisée ici).



Fonctionnement du programme :

- **READ BRIGHTNESS** Demande au Hub TI-Innovator de lire le niveau de luminosité et stocke cette valeur dans une mémoire tampon « buffer » intégrée.
- **Get b** est une commande pour obtenir la valeur du tampon du Hub TI-Innovator. Cette instruction transfère la valeur de la mémoire tampon sur le Hub TI-Innovator vers la variable **b** dans la TI-Nspire. Vous pouvez utiliser n'importe quel nom autorisé de variable.

Ce document est mis à disposition sous licence Creative Commons

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>





Conseil à l'enseignant : Une mémoire tampon « buffer » est une zone de mémoire sur le Hub TI-Innovator qui conserve temporairement une valeur. Celle-ci est mise à jour lorsqu'une nouvelle demande de lecture est envoyée, ainsi il est fortement recommandé de faire succéder la commande **READ** par une instruction **Get** afin de conserver la valeur lue sur le Hub TI-Innovator dans une variable de la calculatrice. Il est aussi possible de conserver les valeurs lues depuis le Hub TI-Innovator et de les stocker dans une liste pour une future analyse, mais cela dépasse le cadre de cette introduction.

La boucle While :

La boucle **While...EndWhile** (menu > **Control** >) Est utilisé pour traiter un bloc de code alors qu'une condition est vraie. Une condition est une instruction logique qui peut être évaluée comme vraie ou fausse. Les opérateurs relationnels et les opérateurs logiques se trouvent sur la touche **ctrl + = key**.

Les opérateurs relationnels sont =, ≠, <, >, ≤, et ≥.

Les opérateurs logiques sont **and**, **or**, **not**, et **xor**.

Ces opérateurs peuvent être utilisés ensemble afin de composer des instructions composées **x>0 and y>0**.

Nous allons utiliser une simple boucle **While** qui s'arrête lorsque la luminosité atteint une valeur inférieure à 1. Pour terminer le programme, couvrir simplement avec votre main le capteur de lumière sur la face latérale du Hub TI-Innovator.

Ajouter une boucle While

1. Avant l'instruction **Send** de votre programme, ajouter la commande :
b:=2
 - Cette instruction est utilisée pour initialiser la boucle. Tant que la condition **b>1** est vraie, la boucle continue la lecture du capteur de lumière. Dès qu'elle devient fausse, c'est à dire lorsque la quantité de lumière reçue par le capteur est insuffisante ou que celui-ci est recouvert avec la main, la boucle et le programme se terminent.

```

1.1 *Classeur RAD 2/5
* lumin2
Define lumin2()=
Prgm
Local b
b:=2
Send "READ BRIGHTNESS "
Get b
EndPrgm

```

Nous voulons inclure les instructions **Send** et **Get** à l'intérieur d'une boucle **While**. Il existe une méthode pratique pour réaliser cela.

2. Sélectionner d'abord les instructions **Send** et **Get** en maintenant enfoncée la touche **shift** et en utilisant les touches de directions
3. Maintenant avec les instructions sélectionnées appuyer sur **menu > Control > While**
4. Cette action place **While** au début des instructions et **EndWhile** juste après elles.

```

1.1 *Classeur RAD 6/6
* lumin2
Define lumin2()=
Prgm
Local b
b:=2
While
Send "READ BRIGHTNESS "
Get b
EndWhile
EndPrgm

```

Ce document est mis à disposition sous licence Creative Commons



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>



10 Minutes de Code

TI-NSPIRE™ CX AVEC LE TI-INNOVATOR™ HUB

5. Ajouter **b>1** après le **While** ainsi votre boucle doit ressembler à celle-ci :

```
While b>1
  Send "READ BRIGHTNESS"
  Get b
EndWhile
```

UNITE 3 : COMPETENCE 1

NOTES DU PROFESSEUR

```
1.1 *Classeur RAD 3/6
* lumin2
Define lumin2()=
Prgm
Local b
b:=2
While b>1|
Send "READ BRIGHTNESS "
Get b
EndWhile
EndPrgm
```

Conseil à l'enseignant : C'est toujours une bonne idée d'écrire complètement une boucle avant le corps de celle-ci afin d'éviter des erreurs de structure.

Souvenez-vous que **End** n'est pas la fin du programme mais plutôt la fin de la structure de contrôle. (**If...Then...End**, **For...End**, **While...End**, et **Repeat...End**). Dans des programmes plus importants, lorsqu'il y a de nombreuses instructions **End**. Le processeur identifie parfaitement à quelle structure de contrôle, la commande **End** fait référence. Mais c'est au programmeur de concevoir un code construit avec rigueur.

Ajouter l'instruction **DispAt** après l'instruction **Get** et avant le **EndWhile** de la boucle comme montré ci-contre en sélectionnant **menu > I/O > DispAt**.

Afficher un message approprié ainsi que le contenu de la variable **b** :

DispAt 1, "Luminosité = ", b

```
1.1 *Classeur RAD 6/7
* lumin2
Define lumin2()=
Prgm
Local b
b:=2
While b>1
Send "READ BRIGHTNESS "
Get b
DispAt 1,"Luminosité = " ,b|
EndWhile
EndPrgm
```

- Faire fonctionner le programme (**ctrl+R**) avec TI-Innovator Hub connecté.
- Vous devriez voir une séquence de valeurs défilé vers le bas de l'application Calculs. Ces valeurs se modifient en fonction de l'intensité lumineuse lue par le capteur.
- Pour terminer la boucle (et le programme), couvrez le capteur de lumière situé à l'extrémité du Hub TI-Innovator afin que la valeur de luminosité affichée soit inférieure à 1

```
1.1 1.2 *Classeur RAD
lumin2()
Luminosité = 0.98883
Terminé
```

Conseil à l'enseignant : Les variables locales (**Local**) n'existent que pour l'exécution du programme. Elles ne sont pas créées dans le problème actuel. Si une variable utilisée dans le programme n'est pas déclarée comme étant locale lorsque le programme s'exécute, alors elle le sera dans le problème courant. C'est parfois un avantage, mais cela peut aussi être problématique dans d'autres situations.

Pour fixer les sauts d'affichage du mot "Luminosité," afficher le mot et sa valeur sur des lignes séparées.

```
DispAt 1, "Luminosité = " (ajouter des espaces pour tout placer à gauche)
DispAt 2, b
```

Ce document est mis à disposition sous licence Creative Commons



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>