



Dans cette première leçon de l'unité 1, Vous allez utiliser l'éditeur de programme pour écrire un ensemble de commandes qui contrôle une diode sur le Hub TI-Innovator™.

Objectifs :

- Utiliser l'éditeur de programme de le TI-83 Premium CE
- Utiliser la commande **Send(** pour contrôler une diode lumineuse sur le Hub TI-Innovator
- Introduire la commande Wait
- Temporiser le Hub TI-Innovator et la calculatrice

Connecter le Hub TI-Innovator à la TI-83 Premium CE. L'écran de la calculatrice s'allume. Sur le Hub TI-Innovator, une lumière verte s'allume, indiquant que le Hub TI-Innovator est sous tension et prêt à recevoir des commandes.

Afin d'apprendre à programmer une calculatrice avec le Hub TI-Innovator, vous serez conduits à apprendre la programmation de deux mondes connectés: la calculatrice et le Hub TI-Innovator.



Pour écrire un programme dans la calculatrice, vous utiliserez la touche `[prgm]` pour coller des instructions dans le programme.

- La commande **Send(** est utilisée pour envoyer des commandes au Hub TI-Innovator afin de produire une action physique (allumer une lampe, produire un son, faire tourner un moteur, etc.).
- Les commandes du Hub TI-Innovator se trouvent également en appuyant sur la touche `[prgm]` à partir du menu **HUB**.

Conseil à l'enseignant : Programmer la calculatrice pour contrôler le Hub TI-Innovator requiert l'OS v5.2 ou supérieur sur TI-83 Premium CE et l'application HUB. Il y a un menu nouveau menu **HUB** menu dans l'éditeur de programme qui vous aide à construire les commandes à donner au Hub TI-Innovator

Notre premier programme demandera au Hub TI-Innovator Hub d'allumer une diode rouge (DEL) pendant 5 secondes.

Pour commencer l'écriture d'un nouveau programme appuyer sur la touche `[prgm]` > **Nouveau**, entrer un nom pour le nouveau programme (nous choisissons LUMIERE1), et appuyer sur **entrer**.



Conseil à l'enseignant: N'appuyez pas sur les touches alpha pour entrer le nom du programme. La calculatrice est déjà en mode alpha ce qui est indiqué par le curseur. Pour taper le nombre 1 à la fin du nom du programme, appuyer sur la touche alpha afin d'éteindre le mode alpha. Si vous essayez de réutiliser un nom de programme qui existe déjà, cela éditera le programme au lieu d'en créer un nouveau.





10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

Votre programme est constitué d'une seule ligne de code.

Send("SET LIGHT ON TIME 5")

LIGHT est le nom de la DEL rouge.

Pour créer la commande, vous devez :

1. Presser la touche **[prgm]** et sélectionner le menu **HUB**.
2. Sélectionner l'item **Send("SET...** dans le menu.
3. Sélectionner l'item **LIGHT** dans le menu.

UNITE 1: COMPETENCE 1

NOTES DU PROFESSEUR

```
NORMAL FIXE7 AUTO REEL RAD MP
CTL E/S COULEUR EXEC HUB
1:Send("SET...
2:Send("READ...
3:Settings...
4:Wait
5:Get(
6:eval(
7:Send("CONNECT-Output...
8:Send("CONNECT-Input...
9↓Ports...
```

Conseil à l'enseignant: Les programmes sont sauvegardés dès qu'ils sont entrés. Pour effacer des lignes, utiliser la touche ANNUL. Pour effacer des caractères ou des commandes, utiliser la touche SUPPR. Pour insérer des lignes vides, mettre le curseur en fin de ligne, et appuyer sur [insérer] puis entrer.

Vous ne pouvez pas taper les commandes de programmation lettre par lettre. Les commandes ne sont pas éditables. Tous les mots clés sont sélectionnés à partir de menus. Le texte affiché est en fait juste un symbole utile à la programmation.

Votre programme devrait ressembler à l'écran de droite. Compléter l'instruction avec le reste de la commande.

Send("SET LIGHT ON TIME 5")

```
NORMAL FIXE7 AUTO REEL RAD MP
PROGRAM:LUMIERE1
:Send("SET LIGHT
```

Conseil à l'enseignant: Vous pouvez écrire à l'intérieur des instructions du Hub TI-Innovator qui sont entre guillemets de la commande **Send(** . Ces instructions sont des chaînes de caractères qui sont envoyées au Hub afin d'être exécutées.

Vous pouvez obtenir les mots **ON** et **TIME**:

1. Appuyer sur la touche **[prgm]**, et sélectionner le menu **HUB**.
2. Sélectionner Settings.

Ne pas oublier les guillemets (**[alpha]** **[+]**) et les parenthèses à la fin de l'instruction. Le caractère espace est sur la touche 0 (zéro).

```
NORMAL FLOAT AUTO REAL RADIAN MP
Settings
1:ON
2:OFF
3:TO
4:TIME
5:BLINK
6:TEMPERATURE
7:HUMIDITY
8:CW
9↓CCW
```

Conseil à l'enseignant: Quelques instructions de la calculatrice peuvent être différentes lorsqu'on utilise l'éditeur de programmes: La touche **MATH** fait référence aux fonctions mathématiques à utiliser dans un programme. Le **CATALOG** contient la liste de toutes les instructions classées par ordre alphabétique. La touche **MODE** permet d'accéder à l'ensemble des réglages de la calculatrice. D'autres sont accessibles par d'autres procédés.





10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

Le programme complet est affiché sur l'écran de droite.

La commande **Send()** enverra la chaîne de caractères située entre guillemets eu Hub TI-Innovator Hub.

UNITE 1: COMPETENCE 1

NOTES DU PROFESSEUR

```

NORMAL FIXE7 AUTO REEL RAD MP
PROGRAM:LUMIERE1
:Send("SET LIGHT ON TIME
5")

```

Conseil à l'enseignant : Attention à l'usage de la touche ANNUL. Si elle est utilisée dans l'éditeur, Elle effacera la ligne de code en cours — et on ne peut pas revenir en arrière. Mais si vous avez un menu à l'affichage de l'écran, ANNUL agit comme la touche d'un ordinateur : Cela fera disparaître le menu de l'écran et affichera de nouveau le code du programme.

Pour faire fonctionner le programme :

1. Quitter l'éditeur de programme en tapant [quitter] (2nd **MODE**).
2. Appuyer sur la touche **prgm** et sélectionner le programme à partir du menu **EXÉC**. Le nom du programme est collé dans l'écran courant.

Assurez vous de bien avoir connecté le Hub TI-Innovator à la calculatrice. Appuyer sur **entrer** pour faire fonctionner le programme.

```

NORMAL FIXE7 AUTO REEL RAD MP
PRgmLUMIERE1
..... Fait

```

Si l'instruction est écrite correctement, alors la diode rouge clignotera pendant 5 secondes. S'il y a une erreur, la diode clignotera une fois, puis un signal sonore se fera entendre.

La calculatrice affiche « Fait » lorsque le programme se termine. Notez la fin de l'exécution du programme avant que la lampe ne s'éteigne. Si vous souhaitez synchroniser le temps de fonctionnement du programme avec celui de la mise sous tension de la diode, vous devez demander à la calculatrice d'attendre (Wait) pendant la même durée où la diode est allumée.

Conseil à l'enseignant: Dans les commandes Send de ce programme, il y a en fait deux instructions : Send(et SET. Send est une instruction donnant un état logique au port USB. La commande SET envoyé au Hub TI-Innovator Hub place la diode en état d'extinction ou de mise sous tension.

Pour ajouter une commande à un programme, vous devez l'éditer.

1. Appuyer sur **prgm** et utiliser les flèches de direction pour accéder au menu **ÉDIT**
2. Sélectionner le nom du programme dans la liste.
3. Utiliser les touches de direction pour placer le curseur à la fin du code.
4. Appuyer sur **entrer** pour ajouter une nouvelle ligne.
5. Ajouter la commande **Wait 5** en appuyant sur la touche **prgm**, puis flèche vers le bas afin d'atteindre **Wait** sur le menu HUB et en tapant 5 puis **entrer**.

```

NORMAL FIXE7 AUTO REEL RAD MP
PROGRAM:LUMIERE1
:Send("SET LIGHT ON TIME
5")
:Wait 5

```

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-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

UNITE 1: COMPETENCE 1

NOTES DU PROFESSEUR

Quitter l'éditeur de programme, et faire fonctionner de nouveau le programme. Si le programme correspond à la dernière commande de l'écran courant, appuyer simplement sur `⏏` pour la rappeler. Le programme effectuera la même temporisation avant d'éteindre la diode.

Nous pouvons effacer l'élément **TIME** de l'instruction **Send** et contrôler le temps d'exécution du programme en utilisant la commande **Wait**.

Conseil à l'enseignant : Si un programme génère un message d'erreur, alors il y a un dysfonctionnement dans celui-ci. Face à un message d'erreur, deux options sont possible erreur: **1:Quitter** et **2:Voir**. **Quitter** renvoie la calculatrice à l'écran courant, et **Voir** place le curseur là où l'erreur s'est produite. Cela peut correspondre à la place exacte de l'erreur, mais pas toujours.

Pour éteindre la diode, utiliser la commande

Send("SET LIGHT OFF")

Ajouter les instructions pour permettre à la diode de clignoter plusieurs fois

```
NORMAL FIXE? AUTO REEL RAD MP
PROGRAM:LUMIERE1
:Send("SET LIGHT ON")
:Wait 1
:Send("SET LIGHT OFF")
:Wait 1■
```

Ce document est mis à disposition sous licence Creative Commons

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





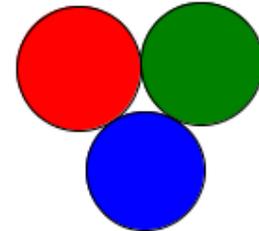
Dans cette seconde leçon de l'unité 1, vous apprendrez à utiliser les instructions pour entrer données dans un programme afin de contrôler la couleur d'une diode (DEL) sur le Hub TI-Innovator™.

Objectifs:

- Utiliser les commandes Prompt et Input
- Contrôler la couleur d'une diode DEL

La couleur d'une Diode Electroluminescente DEL est construite selon trois « canaux » : Rouge Vert et Bleu. D'où le nom de diode RVB "RVB DEL".

Pour obtenir une couleur particulière, nous devons ajuster le mélange de chacune d'entre elles.



Dans ce programme, vous expérimenterez le fonctionnement de la diode RVB. Vous entrerez une valeur pour chaque couleur Rouge, Vert et Bleu afin de faire clignoter la diode.

En plus de l'étude des couleurs de la diode RVB, Nous présentons le premier programme qui accepte une entrée de l'utilisateur, alors que celui-ci est en fonctionnement **Prompt**.

```
NORMAL FLOTT AUTO REEL RAD MP
CTL E/S COULEUR EXEC HUB
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
8:Effécran
9↓EffTable
```

Conseil à l'enseignant : Il existe deux commandes pour entrer les données: **Input** et **Prompt**. Prompt est plus simple est affiche sous forme de question le nom de la variable à entrer. L'instruction Input est plus souple, et autorise le programmeur à fournir davantage d'information. Nous verrons cela dans une autre leçon.

La commande **Prompt** autorise l'utilisateur à entrer des variables pendant l'exécution d'un programme. Elle est appelée **Prompt** car elle fournit un message visuel (le nom de la variable suivi d'un point d'interrogation) lorsque cette commande est exécutée, l'utilisateur entre la valeur demandée en réponse.

Les syntaxes possibles de l'instruction sont :

- **Prompt** <variable>
- **Prompt** <variable1>,<variable2>,<variable3>,...

Les variables numériques de la TI-83 Premium CE sont les lettres A..Z et Θ (theta). On peut aussi leur affecter des nombres décimaux et des nombres complexes

Dans ce programme, nous utiliserons la commande Prompt pour les couleurs rouge, vert et bleu une valeur comprise dans la gamme (0 – 255) et ensuite la transmettre à la diode RVB. Afin d'être efficace, les trois valeurs seront incluses dans une seule commande.

Conseil à l'enseignant : Les couleurs de la diode peuvent être réglées selon deux façons. Vous pouvez soit envoyer les trois commandes de la couleur regroupées en une seule ("SET COLOR # # #") ou bien utiliser séparément la commande **Send**(pour contrôler chaque canal de couleur, COULEUR.ROUGE, COULEUR.VERTE, et COULEUR.BLEU. Dans tous les cas, les valeurs permises sont dans l'intervalle 0..255. Le nombre total de couleur possible est donc $256^3=16777216$.

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-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

Mise au point du programme.

1. Appuyer sur la touche `[prgm]` > **Nouveau**, entrer un nom pour le programme (nous utilisons le nom COULEUR1 ici), et appuyer sur **entrée**.
2. Ajouter une commande **Effécran** en appuyant sur la touche `[prgm]` puis flèche de direction pour atteindre le menu **E/S**, et sélectionner **8: Effécran**.
3. Ajouter une commande **Prompt** en appuyant sur `[prgm]`, puis flèche de direction jusqu'au menu **E/S**, et sélectionner **2: Prompt**.
4. Ajouter la variable R en sélectionnant `[ALPHA]` `[X]`.
5. Continuer en ajoutant les variables pour le Vert et le Bleu. Ne pas oublier les guillemets entre les noms des variables.

UNITE 1: COMPETENCE 2

NOTES DU PROFESSEUR

```
NORMAL FLOTT AUTO REEL RAD MP
PROGRAM: COULEUR1
: Effécran
: Prompt R,V,B
: █
```

Maintenant nous allons utiliser la commande **Send(** pour envoyer une instruction au Hub TI-Innovator.

1. Appuyer sur la touche `[prgm]` puis utiliser les touches de direction pour atteindre le menu **HUB**.
2. Sélectionner **Send("SET...)** puis choisir **COLOR**.

Note: Vous ne pouvez pas envoyer les variables *R V B* comme les valeurs de la couleur avec la commande **Send(** car les lettres *R, V* et *B* seraient envoyées au Hub TI-Innovator Hub à la place des valeurs de ces variables.

```
NORMAL FLOTT AUTO REEL RAD MP
PROGRAM: COULEUR1
: Effécran
: Prompt R,V,B
: Send("SET COLOR
```

Nous avons besoin d'utiliser une fonction spéciale, **eval()**, conçue pour convertir la valeur d'une expression de la calculatrice en une chaîne de caractères que le Hub TI-Innovator peut traiter.

Compléter le programme :

1. Ajouter la fonction **eval(** en appuyant sur la touche `[prgm]`, puis flèche de direction jusqu'au menu **HUB**, et sélectionner **eval(**.
2. Taper la lettre R et la parenthèse de droite.
3. Ajouter un espace (`[ALPHA]``[0]`).
4. Refaire la même opération deux fois pour le V et B. Ne pas oublier d'ajouter un espace entre !

```
NORMAL FLOTT AUTO REEL RAD MP
CTL E/S COULEUR EXEC HUB
1: Send("SET...
2: Send("READ...
3: Settings...
4: Wait
5: Get(
6: eval(
7: Send("CONNECT-Output...
8: Send("CONNECT-Input...
9↓ Ports...
```

5. Finalement après la troisième instruction **eval(B**, ajouter les guillemets et la parenthèse droite afin de fermer la commande **Send(**.

La commande complète sera :

Send("SET COLOR eval(R) eval(V) eval(B)")

```
NORMAL FLOTT AUTO REEL RAD MP
PROGRAM: COULEUR1
: Effécran
: Prompt R,V,B
: Send("SET COLOR eval(R) e
val(V) eval(B)")
```





10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

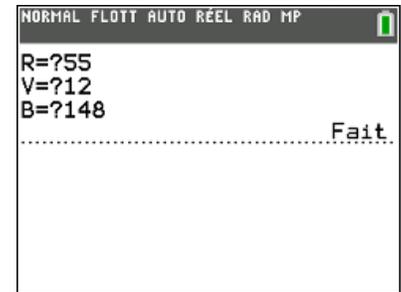
UNITE 1: COMPETENCE 2

NOTES DU PROFESSEUR

Faire fonctionner le programme :

1. [2nde] [mode] pour retrouver l'écran d'accueil en sélectionnant.
2. Presser [PRGM] et sous le menu **EXÉC**, sélectionner le nom du programme.
3. Presser [entrer] pour exécuter le programme.
4. Vous obtiendrez trois invitations après le nettoyage de l'écran, une pour R, une pour V, et une pour B.
5. Entrer trois nombres entre 0 et 255 pour chaque variable.
6. Après avoir entré la dernière valeur, observer la couleur de la diode RVB sur le Hub TI-Innovator.

Pour essayer différentes valeurs, appuyer sur [entrer] afin de relancer le programme, et entrer de nouvelles valeurs.



Conseil à l'enseignant : l'envoi des valeurs 0 0 0 à la diode RVB devrait l'éteindre. Mais la DEL est aussi utilisée pour signaler que le Hub est prêt à être employé, la diode s'illumine alors en vert.

Essayez d'envoyer 1 1 1 et observez de près celle-ci afin de constater que la diode est constituée de trois luminophores. La diffusion de la DEL permet de mélanger efficacement les couleurs : couvrir la DEL à l'aide d'un morceau de papier blanc afin de mieux observer le mélange des trois couleurs. Intéressons-nous en particulier aux valeurs 255 255 0 (rouge et vert) afin d'obtenir du JAUNE. Votre défi est de fabriquer du ORANGE
Noter que la couleur de la DEL subsiste après la fin de l'exécution du programme. Une fois la fin du programme est réalisée, la DEL reste allumée de la même couleur jusqu'à ce que le HUB soit débranché ou bien qu'une autre commande SET COLOR soit envoyée.

Ce document est mis à disposition sous licence Creative Commons

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





Dans cette troisième leçon de l'unité 1, vous allez apprendre une autre méthode pour entrer une donnée dans un programme afin de contrôler un SON sur le Hub TI-Innovator™.

Objectifs :

- Utiliser l'instruction Input
- Contrôler la fréquence et la temporisation d'un (SON)

Le Hub TI-Innovator comporte un haut-parleur embarqué appelé 'SOUND'.

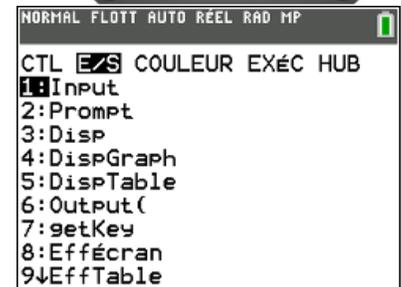
Vous contrôlez le son émis par ce haut-parleur en envoyant une valeur de fréquence. La fréquence d'un son se mesure en Hertz (Hz), ou 'cycles par seconde'.



La commande **Input**, comme **Prompt**, se trouve en appuyant sur la touche `PRGM` menu **E/S**. Elle est aussi employée afin que l'utilisateur fournisse une entrée. Mais elle contient une fonctionnalité qui permet au programmeur de créer un message plus explicite que la simple commande **Prompt** étudiée précédemment.

Syntaxe de la commande: **Input** <Chaîne> , <Variable>

Dans ce programme sur le SON, nous utilisons la commande **Input**.



Conseil à l'enseignant : Le haut-parleur ne dispose pas d'amplificateur. Le son émis est donc de faible intensité. Cela peut permettre de contrôler le bruit dans une classe remplie de Hub TI-Innovator

La syntaxe de la commande SET SOUND est :

SET SOUND fréquence durée

La fréquence peut prendre ses valeurs de 1 à ???

La durée s'exprime en secondes.

Mais l'intervalle de fréquences audible est plus limité. Une fréquence intéressante est 5 Hz. Vous entendrez le haut-parleur cliqueter 5 fois en une seconde. Cela vous aidera à comprendre la notion de fréquence en Hz (cycles par seconde). Aux fréquences plus hautes, vous entendrez des notes de musique. C'est ainsi que fonctionne votre audition. Les ondes compriment les molécules d'air dans l'environnement proche du haut-parleur et notre cerveau interprète ces vibrations comme des sons.





10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

Mise au point du programme SON

1. Commencer un nouveau programme et le nommer SON1.
2. Ajouter la commande **Effécran** et **Input** à partir du menu E/S.
3. Après la commande **Input** utiliser [A-LOCK] (2nd[ALPHA]) pour taper la chaîne de caractères "Frequence? ".
4. Sortir du mode Verr A pour taper la virgule.
5. Puis ajouter la variable qui représentera la fréquence, F
6. Ajouter une autre commande **Input** afin de proposer à l'utilisateur d'entrer une valeur fixant la durée d'émission du son.

Comme avec le programme COULEUR de la compétence précédente, nous devons utiliser la fonction **eval()** afin d'évaluer les variables F et T.

Finishing up the SOUND program

1. Sélectionner **Send("SET...** et **SOUND** en appuyant sur la touche [PRGM]
2. Utiliser les touches de direction pour atteindre le menu **HUB**, et sélectionner **1: Send("SET....**
3. Puis choisir **6:SOUND**
4. Sélectionner **eval**(en appuyant sur [PRGM][.]
5. Utiliser les touches de direction pour atteindre le menu **HUB**, et sélectionner **6:eval(.**
6. Ajouter la variable F, et fermer en utilisant les parenthèses.
7. Ajouter un espace ([ALPHA][0]) puis une autre fonction **eval**(pour la variable T.
8. ajouter la variable T, et fermer en utilisant les parenthèses.
9. Fermer la commande **Send**(à l'aide de guillemets et de parenthèses.

Faire fonctionner le programme

Entrer la fréquence 440 et une durée de 5. Cela jouera une note de 440 Hz pendant 5 secondes. Cela signifie que le haut-parleur vibrera 440 fois par seconde pendant 5 secondes.

Dans un environnement bruyant, vous pourriez avoir à tenir le Hub TI-Innovator près de votre oreille pour entendre le son.

Presser [ENTER] afin de relancer le programme avec une autre valeur de fréquence et de temps. Expérimentez avec d'autres fréquences.

Conseil à l'enseignant : Cette activité est une bonne opportunité afin d'expérimenter la notion de spectre audible. Les hautes et basses fréquences sont en dehors du spectre audible, mais le haut-parleur peut tout de même fournir une réponse qui permette d'interpréter la notion de fréquence. Si vous entrez 5 pour la fréquence et 2 pour le temps vous entendrez 10 clics en 2 secondes. Les hautes fréquences ne sont pas favorables aux oreilles humaines. Nous vous laissons le soin de découvrir la gamme audible des fréquences. Dans une leçon ultérieure, nous nous intéresserons à la progression géométrique formée par les notes de musique. Pour les musiciens 440 Hz est la

Ce document est mis à disposition sous licence Creative Commons



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

UNITE 1: COMPETENCE 3

NOTES DU PROFESSEUR

```
NORMAL FLOTT AUTO REEL RAD MP
PROGRAM:SON1
:Effécran
:Input "FREQUENCE ?",F
:Input "TEMPS ?",T
```

```
NORMAL FLOTT AUTO REEL RAD MP
PROGRAM:SON1
:Effécran
:Input "FREQUENCE ?",F
:Input "TEMPS ?",T
:Send("SET SOUND eval(F) e
val(T)")
```

```
NORMAL FLOTT AUTO REEL RAD MP
FREQUENCE ?440
TEMPS ?5
```



10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

UNITE 1: COMPETENCE 3

NOTES DU PROFESSEUR

fréquence du LA de la quatrième octave.

Ce document est mis à disposition sous licence Creative Commons

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





Écrire un programme qui contrôle un feu de circulation.

Objectifs :

- Contrôler la diode RVB pour simuler les feux de circulation en utilisant une seule lampe
- Créer une séquence d'instructions avec des contrôles de synchronisation appropriés

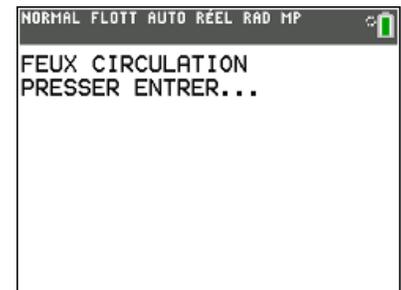
Votre travail consiste à écrire un programme qui contrôle un feu de circulation. La lumière sera simulée en utilisant la couleur de DEL sur le Hub TI-Innovator™.

La DEL de couleur doit passer du vert au jaune au rouge et du rouge au jaune au vert. Vous fixerez la temporisation.

Votre programme aura une séquence d'instructions qui simulent le changement du rouge au vert au jaune au rouge. Une structure de contrôle de séquence dans la programmation est un ensemble d'états qui sont traités l'un après l'autre, de haut en bas, sans interruption.

Utiliser **Disp** et **Pause**

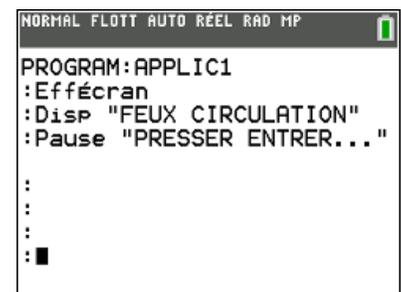
La commande **Disp** affiche un message sur l'écran d'accueil de la calculatrice. On peut s'en servir pour afficher la valeur d'une variable **Disp X**, ou on peut afficher une chaîne comme dans l'exemple de l'écran de droite. vous trouverez **Disp** en appuyant sur la touche **PRGM** puis utiliser les touches de direction pour atteindre le menu **E/S** et sélectionner **3:Disp**.



La commande **Pause** affiche aussi un message mais arrête la calculatrice en cessant de traiter toutes les autres instructions jusqu'à ce que l'utilisateur appuie sur **entree**. Vous pouvez trouver **Pause** en appuyant sur **PRGM** et en sélectionnant **8:Pause**.

Créer un titre à l'écran

1. Commencer un nouveau programme et le nommer APPLIC1.
2. Ajouter une instruction **Effécran** à partir du menu **PRGM** **E/S**.
3. Ajouter **Disp** en appuyant sur **PRGM** puis utiliser les touches de direction pour atteindre le menu **E/S** et sélectionner **3:Disp**.
4. Entre guillemets ajouter un titre pour l'écran du programme.
5. Ajouter la commande **Pause** en appuyant sur **PRGM** et en choisissant **8:Pause**
6. Entre guillemets, ajouter le texte *PRESSER ENTRER* comme montré sur l'écran de droite.



Conseil à l'enseignant : Pour quelles raisons trois positions sont-elles utilisées sur un feu de circulation plutôt qu'une seule dont le changement de couleur signifierait la conduite à adopter ? Tout simplement afin que les personnes Daltoniennes puissent identifier l'état de la circulation en fonction de la position de la couleur allumée.





10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

Mise au point des couleurs

Tout d'abord nous avons mis la couleur au rouge en définissant les valeurs RVB à 255, 0, 0.

Dans cet exemple, sur l'écran de droite, nous utilisons la commande **Wait** afin de demander à la calculatrice d'attendre 5 secondes avant d'envoyer une nouvelle instruction au Hub TI-Innovator. La couleur rouge devrait persister Durant ce temps. Votre travail consiste à ajouter les instructions nécessaires afin de réaliser le vert, puis le jaune, puis le rouge à nouveau.

UNITE 1: APPLICATION

NOTES DU PROFESSEUR

```
NORMAL FLOTT AUTO REEL RAD MP
PROGRAM:APPLIC1
:Effécran
:Disp "FEUX CIRCULATION"
:Pause "PRESSER ENTRER..."
:
:Send("SET COLOR 255 0 0")
:Wait 5
:
```

Conseil à l'enseignant : Le code RVB pour le jaune est 255, 255, 0.

Ce document est mis à disposition sous licence Creative Commons

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





Dans cette première leçon de l'unité 2, vous allez apprendre à utiliser la boucle For à travers un programme permettant de faire clignoter une diode tout en affichant des informations sur l'écran de la calculatrice.

Objectifs :

- Apprendre à utiliser la boucle **For**
- Utiliser la commande Disp pour afficher du texte et des variables

Conseil à l'enseignant : Il y a trois sortes de boucles dans la TI-83 Premium CE en langage TI-Basic : **For**, **While** et **Repeat**. Toutes les trois exigent l'instruction **End** à la fin du corps de la boucle. **End** fait référence à la fin d'une boucle, mais pas à la fin d'un programme. Le mot clé **Stop** fait quant à lui référence à la fin d'un programme.

Écrivons un programme pour faire en sorte que la LUMIERE clignote un certain nombre de fois. Avec d'autres instructions d'entrée, vous pouvez également contrôler la temporisation de l'état de la lampe (allumé ou éteint).

Ce programme introduit le concept de boucle **For...End**.

```
NORMAL FLOTT AUTO RÉEL RAD MP
CTL E/S COULEUR EXÉC HUB
1:If
2:Then
3:Else
4:For(
5:While
6:Repeat
7:End
8:Pause
9↓Lb1
```

Mise au point du programme :

1. Commencer un nouveau programme et le nommer LUM2.
2. Ajouter la commande **Effécran** en appuyant sur [PRGM], puis en utilisant les touches de direction pour atteindre le menu **E/S**, et en sélectionnant **8:ClrHome**.
3. Ajouter l'instruction **Disp** en appuyant sur [PRGM], puis en utilisant les touches de direction pour atteindre le menu **E/S**, et en sélectionnant **3:Disp**.
4. Entre guillemets ajouter le mot "CLIGNOTEMENT"
5. Ajouter l'instruction **Input** en appuyant sur [PRGM], puis en utilisant les touches de direction pour atteindre le menu **E/S**, et en sélectionnant **1:Input**.
6. Entre guillemets ajouter « NOMBRE DE FOIS ? » comme montre ci-contre.
7. Puis ajouter une virgule et la variable N.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:LUM2
:Effécran
:Disp "CLIGNOTEMENT"
:Input "NOMBRE DE FOIS ?",
N
:
```

Conseil à l'enseignant : La boucle **For** requiert **For(var, début, fin, <pas>)** et **End** pour terminer. Le corps de la boucle est automatiquement itéré en fonction des valeurs que prend la variable de la boucle depuis la valeur initiale jusqu'à celle de fin. Le pas prend par défaut la valeur 1 si rien n'est précisé.





10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

Ajouter la boucle For :

1. Ajouter la commande **For(** en appuyant sur la touche **[PRGM]** et en sélectionnant **4:For(**.
2. Compléter l'instruction avec les autres arguments, **I,1,N**). Cette commande signifie "Pour **I** variant de **1** à **N** par pas de **1**".
3. Presser **[entrer]** quelques instants pour insérer des lignes vides nous les compléterons plus tard, et ajouter la commande **End** au programme.

Ne vous préoccupez pas du nombre de lignes ajoutées. Vous pouvez toujours ajouter des lignes si vous en avez besoin, et des lignes vides n'ont pas d'effet lorsque vous exécutez le programme.

Le bloc de commande entre **For** et **End** est appelé le "corps de la boucle". C'est cette portion de code qui sera exécutée **N** fois grâce au travail de la boucle **For**.

Pour insérer une ligne vierge dans un programme, placer le curseur à la fin de la ligne dessous l'endroit où vous souhaitez insérer une ligne, puis appuyer sur **[insérer]** (**[2nd]** **[suppr]**) et enfin **[entrer]**.

Nous voulons que la lumière clignote **ON** et **OFF N** fois. Nous voulons aussi que le programme affiche le nombre de clignotements.

Nous commencerons le corps de la boucle avec **Disp I**, variable qui contrôle la boucle.

Nous ajoutons maintenant les instructions pour allumer et éteindre la lampe (ON et OFF) comme montré sur l'écran de droite.

1. Ajouter la commande **Send(SET LIGHT** depuis le menu **[prgm] HUB**.
2. Ajouter le mot **ON** depuis le menu **[prgm] HUB SETTINGS**.
3. Ne pas oublier les guillemets et parenthèses pour terminer l'instruction.
4. Ajouter **Wait** (en secondes) depuis le menu **[prgm] HUB** menu afin que la calculatrice attende avant l'envoi de la commande suivante.

Dans notre programme nous fixons un délai de 1 seconde mais vous pouvez utiliser toutes les valeurs que vous souhaitez y compris des décimales.

5. Quitter l'éditeur de programme pour le faire fonctionner.

Voyez-vous la diode clignoter ? Si la temporisation est trop courte, vous ne percevrez pas le clignotement.

Vous ajouterez ensuite des commandes pour contrôler la vitesse du clignotement..

UNITE 2: COMPETENCE 1

NOTES DU PROFESSEUR

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:LUM2
:Effécran
:Disp "CLIGNOTEMENT"
:Input "NOMBRE DE FOIS ?",
N
:
:For(I,1,N)■
```

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:LUM2
:Effécran
:Disp "CLIGNOTEMENT"
:Input "NOMBRE DE FOIS ?",
N
:
:For(I,1,N)
:
:
:End■
```

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:LUM2
:Effécran
:Disp "CLIGNOTEMENT"
:Input "NOMBRE DE FOIS ?",
N
:
:For(I,1,N)
:Disp I
:
:End
```

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:LUM2
N
:
:For(I,1,N)
:Disp I
:Send("SET LIGHT ON ")
:Wait 1
:Send("SET LIGHT OFF ")
:Wait 1
:End
```

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-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

UNITE 2: COMPETENCE 1

NOTES DU PROFESSEUR

Voici un défi: ajouter des instructions **Input** en haut du programme (avant la boucle **For**) pour fixer les deux valeurs d'attente **Wait** à utiliser.

Quittez l'éditeur de programmes et exécuter à nouveau le programme. Observer le clignotement et les valeurs affichées sur l'écran de la calculatrice.

Ce document est mis à disposition sous licence Creative Commons



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



Dans cette seconde leçon de l'unité 2, vous allez apprendre à contrôler l'intensité des trois couleurs de la diode RVB qui équipe le Hub TI-Innovator™.

Objectifs :

- Utiliser la boucle For pour contrôler chaque canal RVB de la diode

Le Rouge, Vert et Bleu prennent leurs valeurs dans l'intervalle (0 à 255) envoyées à la diode RVB, ces trois valeurs déterminent la luminosité de chaque canal. Ce programme montre comment faire varier progressivement la quantité de chaque couleur par transitions successives à travers les 16 million (256^3) de couleurs possibles. Vous utiliserez de nouveau la boucle **For**(dans votre programme.

Créer un programme de changement de couleurs

1. Commencer un nouveau programme et le nommer COULEUR2.
2. Ajouter les instructions **Effécran** et **Disp** ainsi que le titre, comme montré sur l'écran de droite.
3. Ajouter l'instruction **Input** entre guillemets ainsi que le mot *WAIT ?* Puis ajouter une virgule et la variable *W*. Cette variable sera utiliser pour temporiser la commande **Wait** . Ainsi plus le nombre est élevé, plus le temps d'attente et plus vite sera exécutée la commande suivante.
4. Comme montré sur l'écran de droite, ajouter une entrée **Input** afin de fixer le pas (**STEP**(de la boucle **For** ceci afin d'accélérer un peu les choses.

Notre programme fera progressivement (en fonction des valeurs d'attente et du pas) augmenter l'intensité ROUGE, puis ajoutera du VERT, puis reprendra graduellement la couleur ROUGE, puis ajoutera du BLEU, enlèvera du VERT, puis ajoutera du ROUGE au BLEU, puis enlèvera le BLEU, pour finalement enlever la composante ROUGE. Ce programme est assez long, et vous pouvez l'exécuter après avoir terminé chaque boucle.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:COULEUR2
:Effécran
:Disp "CHANGER LES COULEUR
S"
:Input "WAIT ?",W
:Input "PAS ?",S
```

Une bonne habitude à prendre lorsqu'on on travaille sur des programmes plus compliqués est d'insérer chaque boucle complète de sorte que vous n'oublierez aucune instruction de fin de boucle plus tard. Il suffit de laisser quelques lignes vierges dans le corps de la boucle en appuyant sur la touche « entrer » quelques fois.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:COULEUR2
:Effécran
:Disp "CHANGER LES COULEUR
S"
:Input "WAIT ?",W
:Input "PAS ?",S
:For(
:
:
:End
```

Compléter la première boucle :

1. Ajouter le reste des commandes de la boucle **For**(pour aller de 0 à 255. Nous utilisons les variables de la boucle **I** et ajoutons **S** pour le pas.
2. Ajouter la commande **Send**("SET COLOR à partir du menu **[prgm] HUB**.
3. Utiliser la fonction **eval**(à partir du menu **[prgm] HUB** pour la variable **I** et fixer les canaux VERT et BLEU à 0.
4. Utiliser les guillemets et parenthèses pour fermer les instructions.
5. Continuer avec la commande **Send**(et **Wait** en utilisant la variable **W** déjà appelée plus tôt par l'instruction **Input**.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:COULEUR2
:Disp "CHANGER LES COULEUR
S"
:Input "WAIT ?",W
:Input "PAS ?",S
:For(I,0,255,S)
:Send("SET COLOR eval(I) 0
0")
:Wait W
:End
```

Conseil à l'enseignant : Après avoir complété la première boucle, vous pouvez exécuter le programme pour voir la diode briller progressivement vers le ROUGE. Utiliser une petite valeur comme 0.1 et un pas assez grand comme 10.





10 Minutes de Code

TI-83 PREMIUM CE AVEC HUB TI-INNOVATOR™

Après la fin de votre première boucle **For** vous pouvez utiliser la commande **Pause** avec un message pour que vous puissiez admirer le rouge vif de la DEL.

Maintenant, nous allons construire une autre boucle **For** pour ajouter du VERT à la DEL. Mais cette fois, nous voulons contrôler uniquement le canal VERT et ne pas toucher le canal ROUGE. Nous pouvons le faire de deux façons:

Send(SET COLOR 255 eval(I) 0")

(puisque nous savons que le ROUGE est allumé est que le BLEU est éteint)

ou

Send("SET COLOR.GREEN eval(I)")

Cette seconde instruction contrôle uniquement le canal VERT et n'affecte pas les canaux ROUGE et BLEU. Dans les deux cas, notez que nous pouvons réutiliser la variable **I** de la première boucle **For**.

Sur l'image de droite, notez que nous faisons le choix d'utiliser **Send("SET COLOR.GREEN eval(I)")**.

Ajouter la commande **Wait** dans le corps de la boucle en utilisant la variable **W**.

Ajouter la commande **Pause** après le **End** de la boucle pour admirer la nouvelle couleur. Quelle est cette couleur?

Maintenant, nous voulons diminuer progressivement la quantité de rouge, afin de retrouver seulement le VERT.

Pour diminuer dans une boucle **For**(nous commençons par le plus grand nombre, pour aller vers le plus petit nombre, et utilisons un pas négatif.

For(I, 255, 0, -S)

Commence à 255 et soustrait **S** à chaque pas de la variable **I** jusqu'à ce que celle-ci soit inférieure à 0 lorsque la boucle se termine. Assurez-vous d'utiliser le signe  'négatif' des nombres et non pas l'opérateur de soustraction. Cela renverrait un message d'erreur.

Nous voulons seulement changer le canal ROUGE, aussi nous utiliserons l'instruction COLOR.RED dans la commande **Send**. Le reste de la boucle est similaire à la première déjà construite. L'image de droite montre seulement la suite des commandes entrées.

Pouvez-vous compléter chacune de ces instructions ? Sinon, reportez-vous à l'étape suivante.

UNITE 2: COMPETENCE 2

NOTES DU PROFESSEUR

```

NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM: COULEUR2
: Input "WAIT ?", W
: Input "PAS ?", S
: For(I, 0, 255, S)
: Send("SET COLOR eval(I) 0")
: Wait W
: End
: Pause "PRESSER ENTRER"
:

```

```

NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM: COULEUR2
: End
: Pause "PRESSER ENTRER"
:
: For(I, 0, 255, S)
: Send("SET COLOR.GREEN eval(I)")
: Wait W
: End
: Pause "PRESSER ENTRER"

```

```

NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM: COULEUR2
:
: For(I, 0, 255, S)
: Send("SET COLOR.RED")
: Wait
: End
: Pause
:
:
:

```





10 Minutes de Code

TI-83 PREMIUM CE AVEC HUB TI-INNOVATOR™

Voici la section terminée qui supprime le progressivement ROUGE. A la fin de cette boucle, vous devriez voir une couleur VERT vif.

1. Maintenant faire une boucle pour ajouter du BLEU.
2. Puis ajouter une boucle pour enlever le VERT.
3. Puis ajouter une boucle pour ajouter de nouveau du ROUGE.
4. Quelle couleur voyez-vous à la fin de ces boucles ?
5. Ajouter ensuite une boucle pour enlever le BLEU.
6. Finalement faire une dernière boucle pour enlever le ROUGE.
 - Quelle est la couleur de la DEL à la fin du programme ?
 - Que se passe-t-il si les trois canaux sont à 0 0?

Conseil à l'enseignant : Lorsque le ROUGE et le BLEU sont allumés, nous avons du MAGENTA. A la fin du programme, la DEL pourrait être éteinte. En effet, les dernières valeurs envoyées à la DEL pourraient ne pas être 0. Afin d'être certain que la diode est éteinte à la fin du programme, **Send("SET COLOR 0 0 0")**.

UNITE 2: COMPETENCE 2

NOTES DU PROFESSEUR

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:COULEUR2
:For(I,255,0,-S)
:Send("SET COLOR.RED eval(
I)")
:Wait W
:End
:Pause "PRESSER ENTRER"
:
:
:
```





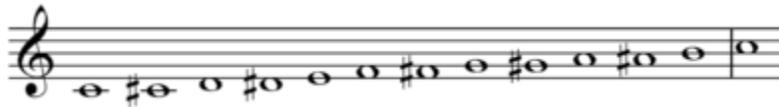
Unité 2: Boucle For

Compétence 3: Boucle à travers des notes de musique

Dans cette troisième leçon pour l'unité 2, vous apprendrez quelle relation existe entre les fréquences et l'échelle musicale. Vous écrirez un programme afin de jouer des notes que les musiciens utilisent depuis quelques siècles.

Objectifs:

- Expliquer l'origine de la racine douzième de 2 dans la gamme Pythagoricienne.
- Ecrire un programme qui joue les notes successives de la gamme.



Un peu de théorie musicale

Les notes de musique sont déterminées par la fréquence d'un objet vibrant, comme un haut-parleur, membrane de tambour, ou une corde de guitare ou piano. Les notes de l'échelle musicale suivent une relation mathématique particulière. Il y a 12 notes ou « pas » dans une octave. Si la fréquence d'une note est F , alors la fréquence de la note suivante est $F \times \sqrt[12]{2}$.

On multiplie donc la fréquence d'une note par $\sqrt[12]{2}$ ou $2^{1/12}$ (la racine douzième de 2) effectuée 12 fois, cette opération revient à doubler la fréquence d'origine, de sorte que la dernière note dans l'octave ait une fréquence de $F \times (2^{1/12})^{12} = 2 \times F$. Par exemple si la fréquence d'une note est 440 Hz, la fréquence de la même note à l'octave supérieure est 880 Hz, et celle de l'octave inférieure est 220 Hz.

L'oreille humaine a tendance à entendre les deux notes d'une octave à part comme étant essentiellement «les mêmes», en raison des harmoniques étroitement liées.

Pour cette raison, les notes d'une octave sont données par le même nom dans le système occidental de la musique. Ainsi la note DO (C) est également DO (C) à l'octave supérieure. Les intervalles entre ces notes sont appelés des «demi-tons».

Dans ce projet, vous allez utiliser le rapport $2^{1/12}$ pour générer les douze notes d'une octave.

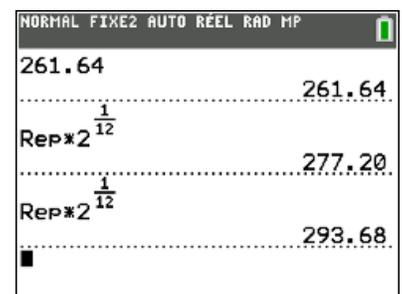
Le DO médium (C) a une fréquence de 261.64Hz. Une octave au-dessus le DO médium (C), aura une fréquence de $2 \times 261.64\text{Hz}$ or 523.28Hz. Il y a 12 « pas » (demi tons) entre ces notes, et chaque demi ton est le $2^{1/12}$ du son suivant ou précédent quant à sa fréquence.

Sur l'écran de droite, nous avons entré 261.64. Ainsi l'instruction suivante consiste juste à effectuer une multiplication par $2^{1/12}$.

La calculatrice fournit **Rép** au début, car le symbole de multiplication nécessite un argument en face de celui-ci. Appuyer simplement sur la touche `[entree]` afin de créer les séquences de calcul comme sur l'écran de droite.

Nous prendrons en compte ce principe répétitif dans notre programme. Si vous continuez la progression, la douzième réponse sera 523,28, deux fois la valeur de départ, car :

$$F \times (2^{1/12})^{12} = 2 \times F.$$





Conseil à l'enseignant : Le calcul peut devenir rapidement complexe si vos étudiants ne sont pas familiarisés avec les calculs sur les puissances et racines. Avoir un clavier à portée de main et montrer l'instrument vous permettra d'illustrer le principe de construction d'un demi ton en incluant les touches noires du clavier, jusqu'à la construction des douze notes de l'octave.

Mise au point du programme:

1. Commencer un nouveau programme et le nommer SON2.
2. Ajouter les instructions **Effécran** et **Disp** et entrer le message entre guillemets.
3. Commencer l'échelle des fréquences à, 261.64 en stockant cette valeur dans la variable F en utilisant la touche **[STO]**.
4. Cette variable représentera chacune des notes de la gamme.

```
NORMAL FIXE2 AUTO REEL RAD MP
PROGRAM:SON2
:Effécran
:Disp "JOUONS LA GAMME"
:261.64→F
```

Mise au point de la boucle For :

1. Ajouter une boucle **For**(allant de 1 à 12 (pour les 12 notes).
2. Ajouter la commande **Send**("**SET SOUND** à partir du menu **HUB**.
3. Ajouter **eval**(pour la variable F comme montré sur l'écran de droite. Attention à ne pas oublier les guillemets et les parenthèses

```
NORMAL FIXE2 AUTO REEL RAD MP
PROGRAM:SON2
:Effécran
:Disp "JOUONS LA GAMME"
:261.64→F
:
:For(I,1,12)
:Send("SET SOUND eval(I)")
:
:End
```

Evaluer la fréquence:

1. Multiplier F par $2^{1/12}$, et stocker de nouveau le résultat dans F
F*2^(1/12)→F
- Cette instruction prend la valeur actuelle de F et change la fréquence de la note suivante sur l'échelle.
2. Quitter l'éditeur de programme pour le faire fonctionner.

```
NORMAL FIXE2 AUTO REEL RAD MP
PROGRAM:SON2
:Disp "JOUONS LA GAMME"
:261.64→F
:
:For(I,1,12)
:Send("SET SOUND eval(I)")
:
:F*2^(1/12)→F
:
:End
```

Modifier le programme :

Essayer d'ajouter l'instruction **TIME** à la commande **SEND**("**SET SOUND** et assurez-vous d'incorporer une commande **Wait** de même valeur afin que chaque note s'achève correctement avant de passer à la suivante.

Si une nouvelle commande est reçue par le Hub TI-Innovator avant qu'il ne termine sa dernière tâche, le Hub TI-Innovator traitera la nouvelle commande au lieu de terminer l'actuelle.

```
NORMAL FIXE2 AUTO REEL RAD MP
PROGRAM:SON2
:Disp "JOUONS LA GAMME"
:261.64→F
:
:For(I,1,12)
:Send("SET SOUND eval(I)")
:
:F*2^(1/12)→F
:Wait 0.5
:End
```





Nous allons utiliser le générateur de nombres pseudo aléatoires TI-83 Premium pour créer de la musique

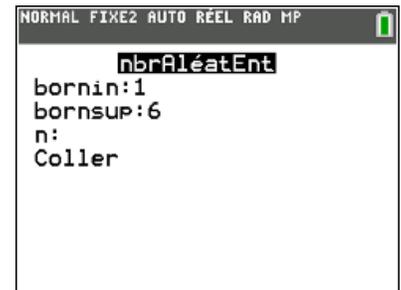
Objectifs :

- Utiliser la boucle **For**(pour contrôler le nombre de notes
- Utiliser le générateur de nombres aléatoires pour créer des notes de musique

Votre travail consiste à compléter un programme qui demande le nombre de notes à jouer, et utilise ensuite une boucle **For** (afin de jouer le nombre donné de notes aléatoires. Comme la note est en cours de lecture, la fréquence doit être affichée sur l'écran de la calculatrice en utilisant l'instruction **Disp**.

Dans cette application nous utiliserons la fonction **nbrAléaEnt()** de la TI-83 Premium CE.

1. A partir de l'écran d'accueil localiser **nbrAléaEnt** dans le menu **math>PROB** .
 - Cette instruction prend deux (ou trois) arguments. Dans un premier temps, vous obtiendrez un «assistant».
2. Entrer les valeurs inférieures et supérieures et laisser le champ de **n**: vide.
 - Il est utilisé pour créer une liste de **n** nombres.
3. Choisir **Coller** pour mettre l'instruction dans l'écran d'accueil et appuyer sur la touche « **entrer** ».
4. A partir d'ici, utiliser la touche de déplacement vers le haut pour rappeler la commande précédente, appuyer sur la touche « **entrer** » fin de la réutiliser et modifier les deux arguments.



Nous combinerons la fonction **nbrAléaEnt()** avec nos notes de musique pour créer un programme qui génère aléatoirement des notes fondées sur la progression géométrique de raison $2^{1/12}$ la partie importante du code est :

```

:For(I,1,10)
:  nbrAléatEnt(0,59)→N
:  55*2^(N/12)→F
:  Send("SET SOUND eval(F) T
IME 0.5")
:  Wait 0.5
:End

```

Comme vous pouvez le voir dans le tableau, la fréquence de la note LA (A) de la première octave est de 55Hz. L'intervalle 0 à 59 est utilisé pour les 60 notes du tableau. Noter l'usage de **N** en $2^{(N/12)}$ pour générer la **N**ième note depuis le LA de la première octave (A1). Lorsque **N** est à zéro, alors la fréquence est 55Hz, puisque 2^0 is 1.

Notes	Frequency (octaves)				
A	55.00	110.00	220.00	440.00	880.00
A#	58.27	116.54	233.08	466.16	932.32
B	61.74	123.48	246.96	493.92	987.84
C	65.41	130.82	261.64	523.28	1046.56
C#	69.30	138.60	277.20	554.40	1108.80
D	73.42	146.84	293.68	587.36	1174.72
D#	77.78	155.56	311.12	622.24	1244.48
E	82.41	164.82	329.64	659.28	1318.56
F	87.31	174.62	349.24	698.48	1396.96
F#	92.50	185.00	370.00	740.00	1480.00
G	98.00	196.00	392.00	784.00	1568.00
G#	103.83	207.66	415.32	830.64	1661.28





10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

Conseil à l'enseignant : Un programme minimal est montré sur l'écran de droite. Noter dans la boucle **For** la présence des paramètres TIME et la commande Wait.

UNITE 2: APPLICATION

NOTES DU PROFESSEUR

```
NORMAL FIXE2 AUTO REEL RAD MP
PROGRAM:APPLIC2
:
:For(I,1,10)
:nbrAléatEnt(0,59)→N
:55*2^(N/12)→F
:Send("SET SOUND eval(F) T
IME 0.5")
:Wait 0.5
:End
```

Ce document est mis à disposition sous licence Creative Commons

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





Dans cette première leçon de l'unité 3, nous étudions le capteur de luminosité intégré, et introduisons la commande **Output**(en TI-Basic afin d'illustrer une technique efficace permettant d'afficher des nombres de caractéristiques (nombre de décimales) différents **Output**(.

Objectifs:

- Lire la valeur de la luminosité recue par un capteur
- Introduire la boucle **While**
- Utiliser la commande **Output**(
- Introduire **versChaîne**(et la concaténation

Dans les leçons précédentes, nous avons seulement envoyé des instructions au Hub TI-Innovator™ afin d'obtenir une action sur les composants qui équipent ce système (LUMIERE, COULEUR, et SON).

Dans cette unité, nous allons travailler avec le capteur de lumière intégré afin de créer un appareil capable de mesurer la quantité de lumière reçue. Le capteur de lumière produit des valeurs dans l'intervalle (0 à 100) sous forme décimale.

Pour obtenir une mesure de la grandeur proportionnelle à la quantité de lumière reçue par le Hub TI-Innovator la commande exige deux arguments.

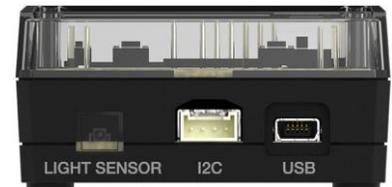
- **Send("READ BRIGHTNESS")**
- **Get(B)**

Mise au point du programme :

1. Commencer un nouveau programme et le nommer LUMIN1.
2. Ajouter la commande **Effécran** et **Disp** pour afficher le titre comme montré sur l'écran de droite.
3. Appuyer sur la touche `prgm` et utiliser les flèches de direction pour atteindre le menu **HUB**.
4. Sélectionner **2: Send("READ...)** puis choisir **1: BRIGHTNESS**.
5. Appuyer sur la touche `prgm` et utiliser les flèches de direction pour atteindre le menu **HUB**.
6. Sélectionner **5: Get(**

Comment cela fonctionne t-il ?

- **READ BRIGHTNESS** demande au Hub TI-Innovator de lire le niveau de luminosité et stocke la valeur dans sa mémoire tampon "buffer"
- **Get(B)** est une commande pour obtenir la valeur depuis le Hub TI-Innovator. Cette instruction transfère la mesure de la mémoire tampon vers une variable B dans la TI-83 Premium CE. B peut être remplacée par n'importe quelle variable numérique, A..Z and Θ (theta).



```
NORMAL FIXE2 AUTO RÉEL RAD MP
PROGRAM:LUMIN1
:Effécran
:Disp "CAPTEUR DE LUMIERE"
:
:Send("READ BRIGHTNESS ")
:Get(B)
:
:
:█
```

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...End** (menu `prgm` CTL) est utilisée pour réaliser une portion de code tandis qu'une *condition* est vraie.



10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

Une *condition* est une instruction logique pouvant être évaluée comme étant vraie ou fausse. Les opérateurs relationnels et logiques se trouvent dans le menu **test** de la calculatrice (**2nd math**).

- Les opérateurs de relation sont =, ≠, <, >, ≤, et ≥.
- Les opérateurs logiques sont **et**, **ou**, **non**(, 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.

Une autre méthode pour arrêter brutalement un programme consiste à appuyer sur la touche **ON**. Vous verrez un message d'erreur: **ERROR: ARRÊT** en haut de l'écran avec deux options **1:Quitter** pour retourner à l'écran d'accueil et **2:Voir** l'éditeur de programme s'ouvre alors en la ligne où a été appelée l'interruption **ON** stoppe le programme. C'est aussi une bonne méthode pour continuer l'édition d'un programme.

```
NORMAL FIXE2 AUTO RÉEL RAD MP
ERREUR: ARRÊT
1:Quitter
2:Voir
Arrêt de l'opération
demandé.
```

Ajouter une boucle While :

1. Avant la commande **Send**(de votre programme, ajouter l'instruction : **2→B**
While B>1 [utiliser le menu [test] pour le symbole >]
 - Cette commande initialise 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.
2. Le **End** de la boucle **While** doit aussi être entré. sous la commande **Get**(Ajouter enfin la commande **End** afin de marquer la fin de la boucle **While**.

```
NORMAL FIXE2 AUTO RÉEL RAD MP
PROGRAM:LUMIN1
:Disp "CAPTEUR DE LUMIERE"
:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)
:
:
:End
```

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.

3. Ajouter la commande **Output**(après l'instruction **Get**(et avant le **End** de la boucle comme montré sur l'écran de droite, en tapant **[prgm]**, puis en utilisant les touches de direction vers le menu **E/S** et en choisissant enfin **6: Output**(.
 - La commande **Output**(vous donne la possibilité de contrôler à l'écran l'emplacement où une information doit être affichée. La structure de la commande est : **Output**(<ligne#>, <colonne#>, <chaîne ou variable>)

```
NORMAL FIXE2 AUTO RÉEL RAD MP
PROGRAM:LUMIN1
:Disp "CAPTEUR DE LUMIERE"
:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)
:Output(5,10,B)
:
:End
```

Exemples :

- **Output(3,7,"HELLO")** placer la lettre "H" à la ligne 3, colonne 7 de l'écran d'accueil et le reste du mot suivra la lettre "H".



10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

- **Output(5,10,B)** placera la valeur de la variable B au début de la ligne 5, colonne 10 de l'écran d'accueil et les autres digits à la suite.
4. Quitter l'éditeur de programme et le faire fonctionner après avoir connecté le Hub TI-Innovator.
- Vous devriez voir le message de titre en haut de l'écran et une valeur dans le centre de l'écran qui change en fonction de l'intensité lumineuse lue par le capteur. Malheureusement, les chiffres que vous voyez peuvent ne pas être corrects !

La commande **Output**(n'efface pas les digits en excès si un nombre plus court est affiché après un précédent comportant plus de décimales. Par exemple, si une valeur affichée est 1.23456 et la valeur suivante à afficher est 55, alors vous verrez 5523456. La dernière modification propose une façon élégante afin de corriger ce problème.

Pour corriger le problème de «suivi des chiffres » nous convertissons la variable B dans une chaîne et ajoutons des espaces à la fin de celle-ci pour effacer complètement la valeur précédente affichée.

toString(se trouve dans le menu **prgm E/S**.

La commande correcte sera finalement : **Output(5,10,toString(B)+" ")**

Il y a 10 espaces entre les guillemets.

Maintenant, lorsque vous exécutez le programme, vous verrez que certaines valeurs sont plus courtes que les autres parce que les espaces que nous avons ajoutés à la représentation de la chaîne B effacent les chiffres précédents.

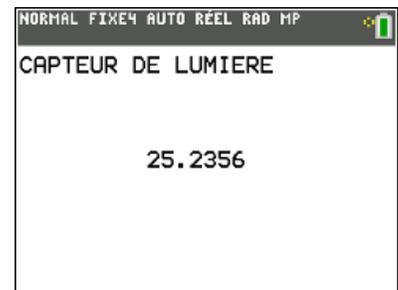
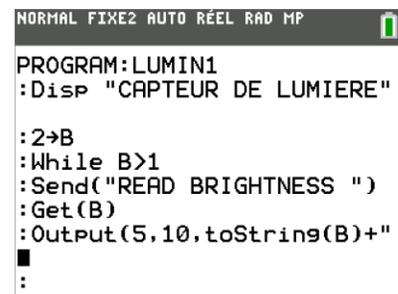
Concaténation:

Le signe "+" utilisé dans l'instruction **Output**(ne correspond pas à une addition : il est utilisé pour 'concaténer' (associer) deux chaînes de caractères. Les espaces (entre guillemets) sont ajoutées à la fin de la chaîne incluant la variable B.

Conseil à l'enseignant : La manipulation des chaînes de caractères n'est pas nécessaire pour ces leçons, mais il doit être clair que la connaissance des instructions permettant de travailler avec des chaînes de caractères constitue pour le programmeur une expérience indispensable.

UNITE 3: COMPETENCE 1

NOTES DU PROFESSEUR





Dans cette seconde leçon de l'unité 3, nous allons construire un interrupteur automatique de lumière dont l'état (ouvert ou fermé) correspond à la valeur de la lumière ambiante.

Objectifs :

- Lire la valeur de la luminosité
- Utiliser une boucle **While** infinie
- Utiliser **If...Then...Else...End** pour mettre sous (hors) tension une lampe en fonction de la luminosité

Ecrivons maintenant un programme qui consiste à réaliser un allumage automatique d'une lampe en fonction de la luminosité reçue

Notre programme va interroger le capteur de lumière du Hub TI-Innovator™ et allumer une diode ou une lampe lorsque la valeur lue correspond à une ambiance trop sombre, puis l'éteindra à partir d'un seuil de luminosité donné.

Mise au point du programme :

1. Commencer un nouveau programme et le nommer LUMIN2.
2. Ajouter les commandes **Effécran** et **Disp** ainsi que les informations du titre comme montré sur l'écran de droite.
3. Initialiser la variable **B** en ajoutant la commande **2→B**.
4. Ajouter une boucle **While** avec la condition **B>1**. (La valeur de la luminosité est vraiment très faible).
5. Ajouter **End** pour fermer la boucle **While**.

```
NORMAL FIXE4 AUTO REEL RAD MP
PROGRAM:LUMIN2
:Effécran
:Disp "INTERRUPTEUR AUTO"
:2→B
:While B>1
:
:
:
:End
```

Pour terminer le programme, couvrir le capteur.

6. Dans le corps de la boucle **While** ajouter **Send("READ BRIGHTNESS")** et **Get(B)** à partir du menu **prgm HUB** comme montré ci-contre.

```
NORMAL FIXE4 AUTO REEL RAD MP
PROGRAM:LUMIN2
:Effécran
:Disp "INTERRUPTEUR AUTO"
:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)
:
:End
```

```
NORMAL FIXE4 AUTO REEL RAD MP
PROGRAM:LUMIN2
:Send("READ BRIGHTNESS ")
:If (B)
:Then
:
:Else
:
:End
:End
```

Instruction If

Votre Instruction **If** contient deux blocs de code. Un lorsque la condition est vraie et un autre lorsqu'elle est fausse.

La structure de l'instruction sur plusieurs lignes est :

If <condition>

Then

<Faire Si condition vraie>

Else

<Faire si condition fausse>





End

Vous pouvez ajouter plusieurs lignes vierges [insérer] [entrer].

Conseil à l'enseignant : Noter que **Then**, **Else**, et **End** ont leurs propres lignes. Ainsi les deux commandes de fin (**End**), une pour la boucle **While** et une autre pour la boucle **For** sont parfaitement identifiées par le processeur lors de l'exécution du programme.

Ecrivons maintenant la condition...

La valeur de la luminosité est stockée dans la variable **B** et prend ses valeurs dans l'intervalle 0 à 100. Qu'est-ce qu'une bonne valeur pour une ambiance sombre ? Nous choisissons 25, mais nous pouvons la modifier par n'importe laquelle entre 0 et 100. Vous pourriez améliorer le programme en utilisant l'instruction **Input** pour la valeur de déclenchement « trigger ». Assurez-vous de placer la commande **Input** avant le début de la boucle **While**.

```
NORMAL FIXE4 AUTO REEL RAD MP
PROGRAM:LUMIN2
:Send("READ BRIGHTNESS ")
:Get(B)
:If B<25
:Then
:
:Else
:
:End
:End
```

Le symbole '<' (inférieur à) se trouve dans le menu **[tests]** (2nd math)

Finalement, allumer puis éteindre la lampe **LIGHT ON** ou **OFF** dans le bloc **Then** et **Else** comme montré sur l'écran de droite.

```
NORMAL FIXE4 AUTO REEL RAD MP
PROGRAM:LUMIN2
:Send("READ BRIGHTNESS ")
:Get(B)
:If B<25
:Then
:Send("SET LIGHT ON ")
:Else
:Send("SET LIGHT OFF ")
:End
:End
```

Quitter l'éditeur et faire fonctionner le programme avec le Hub TI-Innovator connecté. Contrôler la lumière qui arrive sur la cellule du capteur, ainsi que la diode rouge qui s'allume ou s'éteint.

Il peut être intéressant de rajouter une instruction **Output** vue dans la leçon précédente, pour afficher la valeur de **B** sur l'écran de la calculatrice et préciser par ailleurs l'état de la diode allumée ou éteinte.

Output(9,1,"ON ") and **Output(9,1,"OFF")**





Dans cette troisième leçon de l'unité 3, nous allons utiliser la luminosité reçue par un capteur afin de contrôler la couleur d'une DEL

Objectifs :

- Lire un capteur de lumière et contrôler soit la COULEUR DEL, soit la fréquence d'un son issue d'un haut-parleur en fonction de la luminosité reçue.
- Utiliser des formules de conversion pour passer de la valeur de la luminosité à la valeur chromatique de couleur.

Nous allons construire un produit qui réagit à la luminosité de la pièce. Plus une partie de la pièce est lumineuse, plus la diode DEL l'est aussi. La partie délicate ici est de convertir la valeur de la LUMINOSITE en une valeur de couleur appropriée.

La LUMINOSITE **B** prend ses valeurs de 0 à 100.

La COULEUR **C** (sur les trois canaux) peut varier de 0 à 255.

Comment allons-nous convertir B en C?

Conseil à l'enseignant : Réponse : $2.55*B \rightarrow C$ fonctionnera, mais en général, ce type de conversion est utilisé pour calculer la pente d'une droite passant par les points (0,0) et (100,255).

$$(255-0) / (100-0) \rightarrow M: M*B \rightarrow C.$$

Si nous utilisons le Son, alors les points pourraient être (0,100) et (55,880) (Intervalle raisonnable pour les fréquences audio). Mais si nous voulons utiliser des notes de musique, alors (15,75) représente grossièrement le milieu des 60 touches d'un piano. Nous utiliserons alors la relation de conversion $2^{(C/12)}$ pour obtenir la note correcte.

De nombreuses notions mathématiques sont nécessaires et c'est là qu'est la beauté du codage.

Mise au point du programme :

1. Commencer un nouveau programme et le nommer LUMIN3.
2. Ajouter les commandes **Effécran** et **Disp** pour afficher le titre comme montré sur l'écran de droite.
3. Initialiser la variable B comme montré.
4. Ajouter une boucle **While** afin de lire la valeur de la luminosité, puis d'obtenir sa valeur à l'aide de l'instruction **Get(B)**. Nous réaliserons la conversion un peu plus tard.
5. Utiliser la variable **C** pour représenter la valeur de la couleur que nous allons transmettre aux trois canaux de la diode RVB. Le facteur de conversion est $2.55*B \rightarrow C$. Considérer cette formule avec deux valeurs. Lorsque B=0 alors C=2.55*0=0, et lorsque B=100 alors C=2.55*100=255.
6. Ajouter la commande **Send("SET COLOR** avant la fin de la boucle **End**. Cette instruction contrôle la luminosité ou la brillance de la diode.

```
NORMAL FIXE2 AUTO RÉEL RAD MP
PROGRAM:LUMIN3
:Effécran
:Disp "LUMIERE ET COULEUR"

:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)■
```

```
NORMAL FIXE2 AUTO RÉEL RAD MP
PROGRAM:LUMIN3

:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)
:2.55*B→C
:Send("SET COLOR
:
:End■
```



10 Minutes de Code

TI-83 PREMIUM CE AVEC LE HUB TI-INNOVATOR™

7. Finalement compléter la commande SET COLOR en utilisant l'instruction **eval(C)** trois fois (une fois pour chaque canal de couleur). Lorsque les trois canaux ont la même valeur, alors la diode est de "couleur" blanche et sa luminosité dépend de cette valeur.
8. Connecter le Hub TI-Innovator™, et faire fonctionner le programme. Changer la luminosité en pointant le capteur vers divers objets. Observer l'intensité lumineuse de la diode DEL.

Vous pourriez ajouter une instruction **Output**(afin de pouvoir lire les valeurs de **B** et

C.

Conseil à l'enseignant : Lorsque la diode de couleur n'est plus active, celle-ci prend la couleur VERT, cela indique que le Hub est prêt à recevoir des commandes.

Mais attention ! L'effet produit n'est pas correct ! Quel devrait être la couleur ou la teinte la plus brillante prise par la diode au plus sombre de la pièce ? Comment inverser cet effet ?

Un autre défi : Que diriez-vous de changer le programme de sorte que les différentes valeurs de luminosité produisent des couleurs différentes?

Conseil à l'enseignant : Utiliser $2.55*(100-B) \rightarrow C$ pour inverser l'effet.

UNITE 3: COMPETENCE 3

NOTES DU PROFESSEUR

```
NORMAL FIXE2 AUTO REEL RAD MP
PROGRAM: LUMIN3
:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)
:2.55*B→C
:Send("SET COLOR eval(C) e
val(C) eval(C)")
:End■
```



Dans cette application, vous allez écrire un programme pour contrôler le son provenant du haut-parleur en fonction de l'évolution de la luminosité détectée par un capteur de lumière et de votre propre mouvement de main d'où, Musique à la main!

Objectifs :

- Ecrire un programme qui convertit la luminosité en son
- Revoir les notes de musique, leur fréquence ainsi que la progression de celles-ci selon la racine douzième de 2.

Ecrire un programme qui lit la LUMINOSITE reçue par un capteur de lumière et joue un son différent en fonction de la luminosité. Il y a deux options possibles pour le son:

- Jouer une fréquence contenue dans une portion du spectre audible (peut-être 100Hz – 1000Hz)
- Jouer une note de musique (un des sons harmonieux spécifiques trouvé sur un piano ou un autre instrument de musique)

La première option consisterait à jouer simplement un bruit et la seconde à réellement faire de la musique, mais cela fait appel à des théories mathématiques plus complexes

Ce programme rend la TI-Innovator™ Hub se comporte comme un « theremin ».

Commençons

1. Débuter un nouveau programme et le nommer APPLIC3.
2. Ajouter les commandes **Effécran** et **Disp** afin d'afficher le titre comme montré sur l'écran de droite.
3. Ajouter une boucle **While**.
 - Dans la boucle, nous **Lisons (Read)** la luminosité (**BRIGHNESS**) issue du capteur et obtenons sa valeur **Get** dans une variable.
4. Ajouter la commande pour jouer le son.
 - Noter que nous utiliserons la variable **B** pour lire la LUMINOSITE et la variable **F** pour jouer le SON.

Votre travail consiste à compléter le code manquant qui convertit la LUMINOSITE en un *son* audible ou une *note de musique*.

Pour le son, utiliser une fréquence comprise dans l'intervalle 100 à 1000 (ou deux fréquences au choix).

Pour les notes de musique, essayer avec le LA (A1) (55Hz) et en remontant la gamme de 50 notes. (Vous devriez vous référer à l'activité de l'Unité 2, Compétence 3 –prgmSON2—qui joue les 12 notes sur une octave.

Pour les notes de musique, vous devrez convertir votre valeur en un nombre entier de sorte qu'une note «nombre» soit correctement représentée.

Vous pouvez utiliser la fonction **ent()** ou la fonction **arrondir(,0)**.

ent(X)→X donne le plus grand nombre entier inférieur à **X**.

arrondir(X,0)→X arrondi **X** à l'entier le plus proche.

```
NORMAL FIXE4 AUTO REEL RAD MP
PROGRAM:APPLIC3
:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)
:
:Send("SET SOUND eval(F)")
:Wait 0.2
:End
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
MUSIQUE A LA MAIN
B=24.024904
N=11
F=103.8261744
```





Conseil à l'enseignant :

Pour jouer juste un son, les étudiants doivent convertir B depuis (0,100) en F dans (100,1000).

B	F
0	100
100	1000

Calculer la pente de la droite passant par ces deux points, puis écrire l'équation de F en fonction de B.

$$M = (1000-100)/(100-0) = 9$$

soit

$$F = 9*B+100$$

Pour jouer les notes de musique $F*2^{(1/12)}$ donne la propriété des intervalles de notes. Si on donne $A1=55\text{Hz}$ et que nous voulions 50 notes, la première d'entre elles est #0 et la dernière est #49 (les programmeurs commencent toujours à compter à partir de 0). Aussi souvenez-vous que le nombre de notes doit être entier, ainsi nous devons utiliser `ent()` ou `arrondir()`. Nous utilisons deux lignes de commandes afin de rendre cette partie compréhensible.

$$N=\text{ent}(49B/100) \quad \text{--nombre de notes}$$

$$F=55*2^{(N/12)} \quad \text{--fréquence de la note}$$

