

## Lampe frontale

F. GIROD

### Compétences visées

Un des objectifs de l'enseignement de SNT est d'écrire et développer des programmes pour répondre à des problèmes concrets. A travers le thème "**informatique embarquée et objets connectés**", nous pouvons notamment travailler les compétences suivantes dans l'activité proposée :

- Écrire des programmes simples d'acquisition de données.
- Gérer des entrées/sorties à travers les ports utilisés par le système.
- Écrire et développer des programmes pour répondre à des problèmes.
- Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs.

### Situation déclenchante

Une lampe frontale propose en général plusieurs modes :

- éteinte ;
- lumière faible ;
- lumière plus soutenue ;
- lumière rouge ;
- lumière rouge clignotante.

Comment reproduire un fonctionnement de ce type ?

Ici, une [vidéo](#) illustrant le projet.



### Problématique

Comment reproduire le fonctionnement d'une lampe frontale ?

1. Identifier un capteur possible ;
2. Identifier un actionneur possible ;
3. Mettre en place un algorithme modélisant le fonctionnement d'une lampe frontale ;
4. Le mettre en application.

### Matériel disponible

Les élèves disposent, en plus de leur TI-83 Premium CE :

- d'un TI-Innovator HUB ;
- d'un capteur de distance à ultrasons.



## Fiche méthode

*F. GIROD*

### Déroulement possible du projet

#### En amont du projet :

Les propositions suivantes permettent de préparer le projet et de le rendre possible sur 2 séances d'1h30. Elles peuvent aussi être préparées par l'enseignant sous forme de fiche et/ou d'un exposé au tableau.

- Un groupe d'élèves peut préparer une fiche ou présenter un rapide exposé illustrant les principaux points pour maîtriser la programmation en TI Basic en s'appuyant sur les ['10 mn de code'](#).
- Un groupe d'élèves peut expliquer le **mode de codage RGB** pour les couleurs.
- Un groupe d'élèves peut décrire le **TI-Innovator HUB**.
- Un groupe d'élèves peut présenter quelques spécificités du capteur à ultrasons à l'aide d'une fiche associée à la vidéo du composant [RANGER](#).

#### Différentes évolutions possibles pour le projet :

Les propositions suivantes donnent des pistes pour gérer les différences de vitesse d'exécution des élèves.

On peut aussi considérer ces différentes étapes comme une manière de séquencer le projet pour ceux qui en auraient besoin.

- 1<sup>ère</sup> étape : on allume une lumière en passant sa main devant le capteur à ultrasons.
- 2<sup>ème</sup> étape : on allume et on éteint une lumière en passant sa main devant le capteur à ultrasons : création d'un compteur et discrimination selon sa parité.
- 3<sup>ème</sup> étape : on actionne plusieurs modes (au choix des élèves):
  - On passe sa main devant le capteur : une lumière peu intense s'allume ;
  - On passe à nouveau sa main : une lumière plus intense s'allume ;
  - On passe à nouveau sa main : une lumière rouge s'allume ;
  - On passe à nouveau sa main : la lumière s'éteint ;
  - On passe à nouveau sa main : le cycle recommence.
- 4<sup>ème</sup> étape : on met en place un mode "clignotement".

#### A la suite du projet :

Synthétiser ce travail par rapport au cours de SNT :

- le/les capteurs et le/les actionneurs mis en jeu ;
- l'algorithme qui gère les données d'entrée ;
- le langage de programmation utilisé.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



## Fiche méthode

F. GIROD

### Proposition de résolution

#### Choix du capteur :

- Capteur de distance à ultrasons : en passant sa main à proximité, la distance mesurée passe en dessous d'un seuil qui entraîne le passage d'un mode à l'autre. C'est ce capteur qui a été choisi ici.
- Capteur de luminosité (brightness) : en cachant ce capteur, la luminosité passe en dessous d'un seuil qui entraîne le passage d'un mode à l'autre. On peut l'utiliser si on ne dispose pas de capteur à ultrasons ou si on veut proposer une alternative.

#### Choix de l'actionneur :

- LED RGB du HUB : elle pourra délivrer une lumière blanche plus ou moins intense ou une lumière colorée.

### Etapas de résolution

#### Un algorithme possible pour ce projet et le programme correspondant :

Connecter le capteur de distance (RANGER) au HUB

Initialiser les variables C et N à 0

"répéter indéfiniment"

lire la valeur donnée par le capteur à ultrasons  
stocker cette valeur dans la variable D  
afficher la distance D

Si  $D < 0,1$ , alors :

$C = C + 1$

Fin si

$N = \text{reste de la division euclidienne de } C \text{ par } 5$   
Afficher N

Si  $N = 0$ , alors :

tout éteindre

Fin si

Si  $N = 1$ , alors :

émettre une lumière blanche peu intense

Fin si

```
NORMAL FLOTT AUTO REEL RAD MP
EDIT MENU: [a.1pha] [f5]

PROGRAM: LAMPE
:Send("CONNECT RANGER 1 TO
IN 1")
:0→C
:0→N
:While 1
:Send("READ RANGER 1")
:Get(D)
:Disp D
:If D<0.1
```

```
NORMAL FLOTT AUTO REEL RAD MP
EDIT MENU: [a.1pha] [f5]

PROGRAM: LAMPE
:If D<0.1
:Then
:C+1→C
:End
:reste(C,5)→N
:Disp N
:If N=0
:Then
:Send("SET COLOR 0 0 0")
```

```
NORMAL FLOTT AUTO REEL RAD MP
EDIT MENU: [a.1pha] [f5]

PROGRAM: LAMPE
:If N=0
:Then
:Send("SET COLOR 0 0 0")
:End
:If N=1
:Then
:Send("SET COLOR 100 100 1
00")
:End
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus !



Ce document est mis à disposition sous licence Creative Commons

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

© Texas Instruments 2017 / Photocopie autorisée

## Fiche méthode

F. GIROD

```

Si N=2, alors :
  émettre une lumière blanche intense
Fin si

Si N=3, alors :
  émettre une lumière rouge intense
Fin si

Si N=4, alors :
  émettre une lumière rouge clignotante
Fin si

Fin Tant que
    
```

```

NORMAL FLOTT AUTO RÉEL RAD MP
EDIT MENU: [α][Pha.] [f5]

PROGRAM:LAMPE
:If N=2
:Then
:Send("SET COLOR 255 255 2
55")
:End
:If N=3
:Then
:Send("SET COLOR 255 0 0")
:End
    
```

```

NORMAL FLOTT AUTO RÉEL RAD MP
EDIT MENU: [α][Pha.] [f5]

PROGRAM:LAMPE
:If N=4
:Then
:Send("SET COLOR 255 0 0")

:Wait 0.1
:Send("SET COLOR 0 0 0")
:Wait 0.1
:End
:End
    
```

## Remarques

- **While 1** est une condition toujours vérifiée ; ainsi, cela revient à "répéter indéfiniment".

On peut sortir de ce programme en appuyant sur la touche "on" de la calculatrice.

- Les affichages effectués dans le programme (distance D, valeur de N) peuvent être supprimés (ou **commentés** par la présence d'un guillemet en début de ligne, comme sur cette copie d'écran) une fois le programme vérifié.

Leur présence peut être utile lors de la création du programme pour contrôler le bon fonctionnement de chaque étape.

- Les élèves pourront présenter leur projet lors d'un exposé oral, ou par le biais d'une vidéo explicative.

- La fonction **reste()** utilisée dans ce programme s'obtient par la séquence suivante : touche **math** / menu **NBRE** / choix **0**

```

NORMAL FLOTT AUTO RÉEL RAD MP
EDIT MENU: [α][Pha.] [f5]

PROGRAM:LAMPE
:Send("CONNECT RANGER 1 TO
IN1")
:0→C
:0→N
:While 1
:Send("READ RANGER 1")
:Get(D)
:"Disp D
:If D<0.1
    
```

```

NORMAL FLOTT AUTO RÉEL RAD MP
EDIT MENU: [α][Pha.] [f5]

MATH NBRE CMPLX PROB FRAC
2↑arrondir(
3:ent(
4:partDéc(
5:partEnt(
6:min(
7:max(
8:ppcm(
9:pgcd(
0:reste(
    
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus !



Ce document est mis à disposition sous licence Creative Commons

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

© Texas Instruments 2017 / Photocopie autorisée