



# CENTRE SCOLAIRE SAINTE-JULIENNE

## TA 13 – Utilisation des Leds

### Exercices Thymio – Série 1 – Enoncés

#### **Mise en situation**

Une institutrice primaire souhaite faire découvrir la programmation robotique à ses élèves, elle t'a demandé de lui fournir les solutions qui répondent à ses besoins.

#### **Objets d'apprentissage**

<b>Appliquer</b>	<b>Transférer</b>
<ul style="list-style-type: none"><li>• Appliquer les règles de syntaxe et les conventions spécifiques à un langage de programmation</li><li>• Déclarer une variable en appliquant les règles et les conventions</li><li>• Utiliser des fonctions prédéfinies (bibliothèque) en vue d'animer un objet réel ou virtuel</li><li>• Tester la séquence d'instructions conçue</li><li>• Commenter des lignes de codes</li></ul>	<ul style="list-style-type: none"><li>• Écrire un logigramme d'actions d'un objet réel ou virtuel intégrant structure répétitive et opérateurs logiques</li><li>• Améliorer une séquence pour répondre à un besoin défini</li><li>• Corriger une séquence défaillante proposée pour atteindre un but défini</li></ul>
<b>Connaître</b>	
<ul style="list-style-type: none"><li>• Expliquer la notion d'expression</li><li>• Expliquer la notion d'instruction</li><li>• Expliquer la notion de séquence</li><li>• Caractériser les types de données</li><li>• Expliquer la notion de variable</li><li>• Expliquer la notion d'affectation</li><li>• Différencier les opérateurs logiques dont "et", "ou", "non"</li><li>• Expliquer la notion de répétition</li></ul>	

#### **Tâches à accomplir:**

1. Dessiner sur papier l'enchaînement des traitements;
2. Tester et corriger avec Aseba votre solution;
3. Enregistrer la solution finale.

## **Actuateurs et capteurs à utiliser:**

Thymio possède de **nombreuses LEDs** réparties sur tout son corps. La plupart sont associées à des capteurs et permettent de mettre en évidence leur activation: l'allumage des LEDs est lié par défaut aux valeurs des capteurs. Cependant, si elles sont utilisées dans le code, cette utilisation supplante celle par défaut.

Des fonctions natives permettent de contrôler les différentes LEDs. Pour toutes les LEDs, la plage de valeurs d'intensité s'étend de 0 (éteint) à 32 (complètement allumé).

### **Le cercle de LEDs du dessus**

8 LEDs jaunes forment un cercle sur le robot, autour des boutons.

*Activation par défaut* : reflète les valeurs de l'accéléromètre. Eteintes à l'horizontale, leur intensité augmente en penchant le robot. La LED allumée est toujours celle au point le plus bas.

***leds.circle(led 0, led 1, led 2, led 3, led 4, led 5, led 6, led 7)*** où led 1 commande l'intensité de la LED avant du robot (numérotation dans le sens des aiguilles d'une montre).

### **Les LEDs RGB**

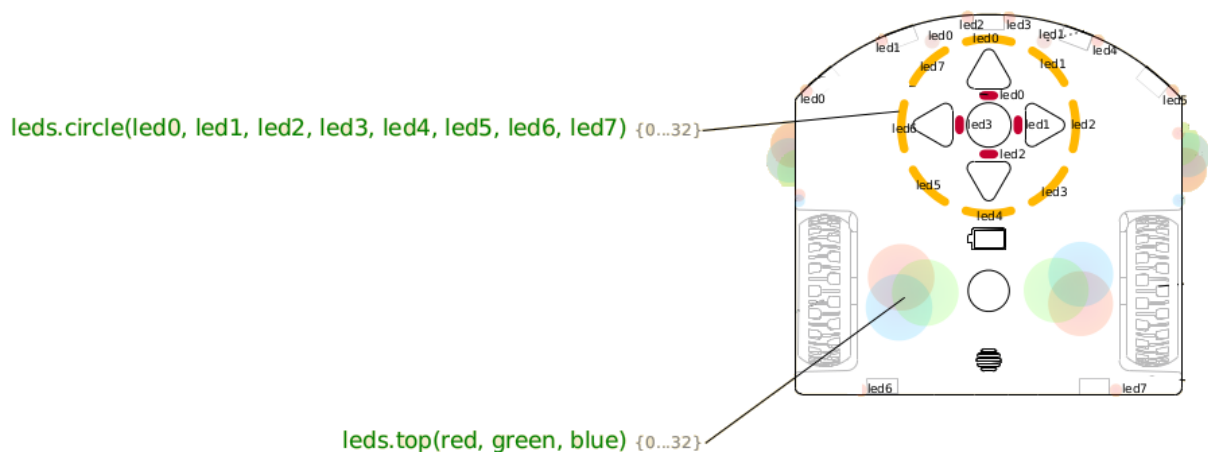
Il y a deux LEDs RGB sur le robot pilotées ensemble, ce sont celles qui indiquent le comportement du robot. Deux autres LEDs RGB dessous sont pilotables séparément.

*Activation par défaut*: éteintes dans le mode Aseba.

***leds.top(red, green, blue)*** commande les valeurs de rouge, vert et bleu respectivement, pour les LEDs du dessus.

***leds.bottom.left(red, green, blue)*** commande les valeurs de rouge, vert et bleu respectivement, pour la LEDs du dessous à gauche.

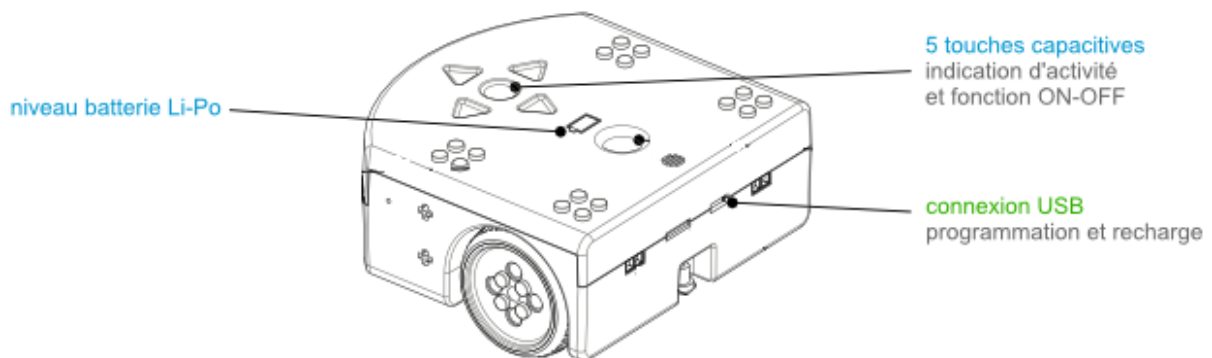
***leds.bottom.right(red, green, blue)*** commande les valeurs de rouge, vert et bleu respectivement, pour la LEDs du dessous à droite.



Thymio possède **5 boutons capacitifs** correspondant aux flèches et au bouton central. Un tableau de 5 variables, `buttons.binary`, contient l'état de ces boutons (1 = appuyé, 0 = relâché):

- **`button.backward`**: flèche arrière
- **`button.left`**: flèche gauche
- **`button.center`**: bouton central
- **`button.forward`**: flèche avant
- **`button.right`**: flèche droite

Thymio met à jour ce tableau à une fréquence de 20 Hz, et génère l'événement `button` après chaque mise à jour. En outre, pour chacune de ces touches, quand elle est appuyée ou relâchée, un événement correspondant avec le même nom est émis.



### **Ex 1**

Afficher la couleur « Vert » de la led du dessus de Thymio quand les boutons gauche et droit sont enfoncés, la couleur rouge dans le cas contraire.

Afficher la couleur « Bleu » quand le bouton du centre est appuyé.

Enregistrer la solution dans un fichier nommé EX01.aesl qui se retrouve dans un dossier nommé Série1.

#### Hypothèse:

Nous considérons que la couleur « Bleu » est affichée par défaut.

### **Ex 2**

Reprendre l'EX01.

Déclarer une variable  $i$  (valeur par défaut  $\rightarrow 0$ ) qui représentera le niveau d'intensité à afficher pour les 8 leds en cercle.

Augmenter la valeur de  $i$  de 1 quand on appuie sur le bouton droit et diminuer de 1 quand on appuie sur le bouton de gauche.

Vérifier que  $i$  est compris entre 0 et 32 ; si oui afficher la led en vert sinon en rouge.

Enregistrer la solution dans un fichier nommé EX02.aesl qui se retrouve dans un dossier nommé Série1.

### **Ex 3**

Reprendre l'EX02.

Créer la procédure « afficher » qui allume la led du dessus en vert quand  $0 < i < 32$  et en rouge dans le cas contraire. De plus, elle affiche l'intensité des 8 leds en cercle grâce à la valeur de  $i$ .

Créer la procédure « augmenter » qui incrémente  $i$  de 1 si  $i < 32$  et appelle la procédure « afficher ».

Créer la procédure « diminuer » qui décrémente  $i$  de 1 si  $i > 0$  et appelle la procédure « afficher ».

Appeler la procédure « augmenter » quand le bouton droit est enfoncé.

Appeler la procédure « diminuer » quand le bouton gauche est enfoncé.

La valeur de  $i$  est réinitialisée à 0 et la procédure « afficher » est appelée quand le bouton du centre est appuyé.

Enregistrer la solution dans un fichier nommé EX03.aesl qui se retrouve dans un dossier nommé Série1.

### **Ex 4**

Reprendre l'EX03.

Définir une constante MIN dont la valeur est 0.

Définir une constante MAX dont la valeur est 32.

Utiliser les deux constantes dans tout test de limite de la valeur de  $i$ .

Afficher la led correspondant à la valeur de  $i$ , et non plus son le degré d'intensité de toutes les leds, dans la procédure « Afficher ».

Enregistrer la solution dans un fichier nommé EX04.aesl qui se retrouve dans un dossier nommé Série1.