


# Fiche n° 5

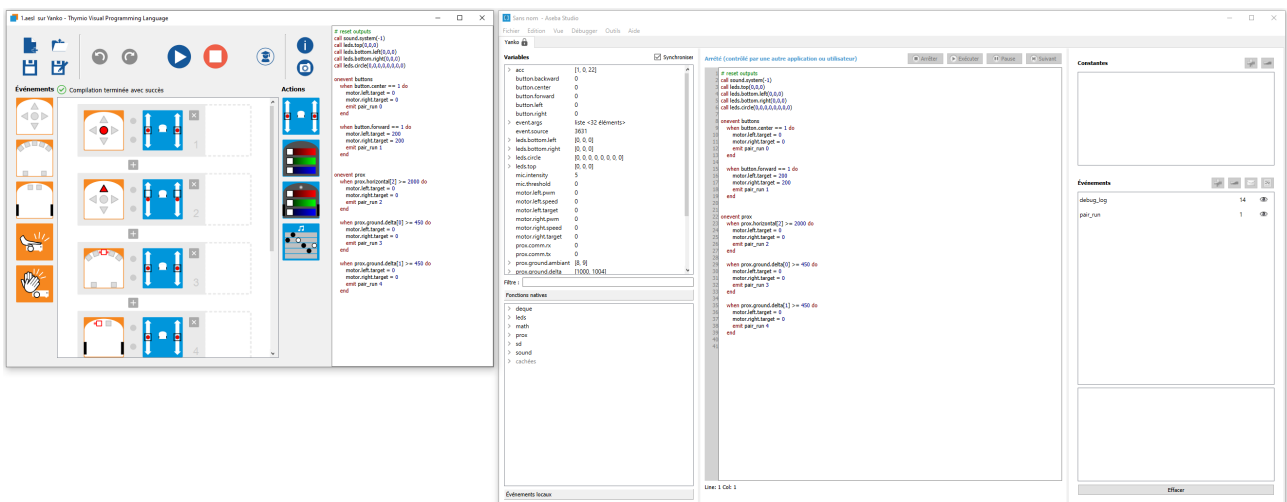


## Thymio II :

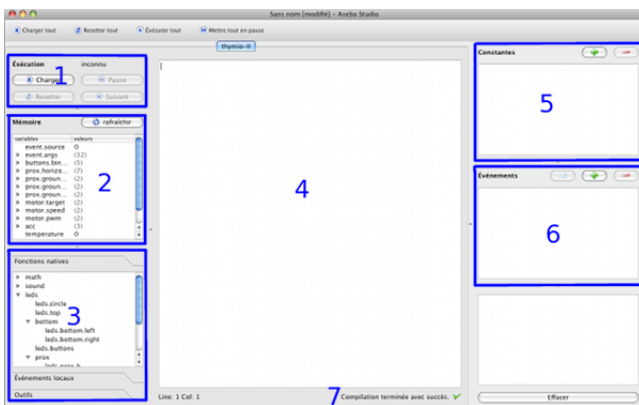
### La programmation en ligne de commande – partie 1 – Le VPL & Aseba Studio :

Il est possible de démarrer en même temps le VPL et Aseba Studio. Ceci permet de commencer à programmer en VPL, de voir les lignes de code se créer dans Aseba Studio, puis de modifier le code pour plus de précision dans Aseba Studio.

- 1) branche le robot à l'ordinateur à l'aide du câble USB.
- 2) Démarrer Thymio Suite
- 3) Lancer le VPL et en parallèle, lancer Aseba Studio 
- 4) Organise ton bureau de la manière à voir les 2 programmes en même temps comme ci-dessous :



### Le logiciel Aseba Studio se divise en 7 zones :



1. Ces boutons servent à envoyer au robot le code qu'on a écrit.
2. Cette fenêtre donne une vision d'ensemble de l'état des différents éléments du robot : capteurs, variables, ...
3. Ici on trouve les fonctions et les événements disponibles qu'on peut utiliser dans le code.
4. Au milieu on va écrire notre code. C'est ça qui détermine le comportement du robot.
5. Ici on peut déterminer des constantes.
6. Ici on peut ajouter nos événements.
7. Cette ligne nous indique si nos instructions sont complètes et correctement écrites.

Les parties du code sont regroupés par chapitres.

Par exemple, les parties du code concernant les boutons sont regroupées sous *onevent buttons*,

les parties du code concernant les capteurs de proximités se trouveront sous *onevent prox, ...*

### 1) onevent buttons :

Cette partie du code définit le rôle des boutons.

Les boutons sont définis ainsi :

<i>button.center</i>	→	<i>bouton central</i>	<i>button.backward</i>	→	<i>bouton arrière</i>
<i>button.forward</i>	→	<i>bouton avant</i>	<i>button.right</i>	→	<i>bouton droit</i>
			<i>button.left</i>	→	<i>bouton gauche</i>

Ces boutons peuvent avoir comme valeur soit 1 lorsque on appuie dessus, soit 0 si on n'appuie pas.

Ex : *when button.center == 1 do ...*

Ceci signifie que lorsque le bouton central est enfoncé, le robot doit effectuer les ordres qui suivent le *do ...*

Le *==* sert à vérifier si la valeur de la variable est bien 1.

Le *=* sert à fixer la valeur de la variable. (Ceci sera développé dans la Fiche n° XXXXXX)

### 1) onevent prox:


Cette partie du code correspond aux capteurs (de proximité et de sol)

Le *prox.horizontal[2]* correspond au capteur avant central,

Le *prox.horizontal[0]* correspond au capteur avant gauche,


Le *prox.horizontal[4]* correspond au capteur avant droit,

Pour savoir à quel numéro est associé un capteur, passe ton doigt devant un capteur et observe dans la fenêtre n°2 les valeurs changer.

 Plus la valeur associée est grande, plus le robot est proche de l'objet.

Le *prox.ground.delta[0]* correspond au capteur de sol gauche

Le *prox.ground.delta[1]* correspond au capteur de sol droit

 A l'inverse de *prox.horizontal[ ]*, plus la valeur de *prox.ground.delta[ ]* est grande, plus le sol est éloigné.

## Activité n° 10 - En Avant – en Arrière – à droite – à gauche :

Commence par programmer en VPL les comportements suivants :

- Lorsque l'on appuie sur le bouton CENTRAL, le robot stoppe
- Lorsque l'on appuie sur le bouton AVANT, le robot avance

Ensuite ferme le VPL et observe ton code :

La ligne 1 commence par un #, ce qui signifie que ce qui suit est un commentaire et ne sera pas exécuté.

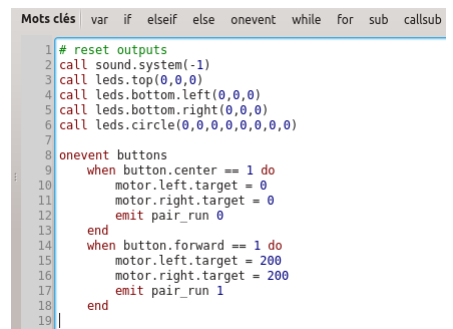
Les lignes 2 → 6 demandent au robot d'éteindre les LED et le son.

La ligne 8 ouvre le chapitre de programmation des boutons.

Les lignes 9 → 18 correspondent à la programmation des boutons central et avant.

**motor.left.target = 0**, signifie que le moteur gauche est arrêté. La valeur doit être comprise entre -500 et 500

On peut supprimer les lignes 12 et 17 et si l'on veut simplifier le code, aussi les lignes 1 → 7



```
Mots clés  var  if  elseif  else  onevent  while  for  sub  callsub
1  # reset outputs
2  call sound.system(-1)
3  call leds.top(0,0,0)
4  call leds.bottom.left(0,0,0)
5  call leds.bottom.right(0,0,0)
6  call leds.circle(0,0,0,0,0,0,0)
7
8  onevent buttons
9      when button.center == 1 do
10         motor.left.target = 0
11         motor.right.target = 0
12         emit pair_run 0
13     end
14     when button.forward == 1 do
15         motor.left.target = 200
16         motor.right.target = 200
17         emit pair_run 1
18     end
19
```

- Complète ensuite ton code pour que le bouton ARRIÈRE fasse reculer le robot.

Tu vas voir que lorsque tu commences à écrire **when**, il va automatiquement écrire **do** et **end**. Lorsque tu vas écrire **motor**, il va te proposer les différentes possibilités, ce qui aide énormément à écrire les codes.

- Complète ensuite ton code pour programmer les boutons DROIT et GAUCHE.

Une fois terminé, sauvegarde ton fichier en *Activité\_10\_tonprénom* et effectue une capture d'écran et enregistre-la en *Activité\_10\_tonprénom*

## Activité n° 11 - Éviter les obstacles :

Commence par programmer en VPL les comportements suivants :

- Lorsque l'on appuie sur le bouton CENTRAL, le robot stoppe
- Lorsqu'il y a un obstacle devant, le robot stoppe

Ensuite ferme le VPL et observe ton code :

La ligne 8 marque le début du chapitre *onevent buttons* qui définit le rôle des boutons.

La ligne 15 marque le début du chapitre *onevent prox* qui définit le comportement du robot en fonction des capteurs.

```
Mots clés  var  if  elseif  else  onevent  while  for  sub  callsub
1  # reset outputs
2  call sound.system(-1)
3  call leds.top(0,0,0)
4  call leds.bottom.left(0,0,0)
5  call leds.bottom.right(0,0,0)
6  call leds.circle(0,0,0,0,0,0,0)
7
8  onevent buttons
9    when button.center == 1 do
10     motor.left.target = 0
11     motor.right.target = 0
12     emit pair_run 0
13   end
14
15  onevent prox
16    when prox.horizontal[2] >= 2000 do
17     motor.left.target = 0
18     motor.right.target = 0
19     emit pair_run 1
20   end
21
```

a) (*facile*) Complète ensuite le chapitre *onevent buttons* pour que le bouton AVANT fasse avancer le robot.

Fixe la puissance des moteurs gauche et droit à 55.

b) (*moyen*) Complète ensuite le chapitre *onevent prox* pour que le robot tourne à droite s'il détecte un obstacle à gauche et qu'il tourne à gauche s'il détecte un obstacle à droite.

c) (*difficile*) Ensuite complète ton code pour que le robot ait le comportement suivant : Le robot se déplace en ligne droite. S'il y a un obstacle à gauche, il tourne à droite et dès qu'il n'y a plus d'obstacle, il recommence à aller en ligne droite.

- Ajoute le code nécessaire pour que le robot évite aussi les obstacles à sa droite.

Une fois terminé, sauvegarde ton fichier en *Activité\_11\_tonprénom* et effectue une capture d'écran et enregistre-la en *Activité\_11\_tonprénom*



Ce document est publié par le DIP Genève sous licence Creative Commons  
Utilisation et adaptation autorisée sous conditions. Auteur: [yanko.michel@edu.ge.ch](mailto:yanko.michel@edu.ge.ch)