

Groupe 4

ISN 2013-2014

THYMIO II :



Suit une ligne

HOARAU Freddy
SEVERIN Coralie

Lycée Antoine Roussin
TS2

Sommaire :

Introduction

I- Nos recherches

- Notre première démarche

- * Comprendre le fonctionnement du Thymio au démarrage

- * La programmation

- Problèmes rencontrés

II- Travail de groupe

III- Résultats obtenus

- * Algorithme de suivi de ligne

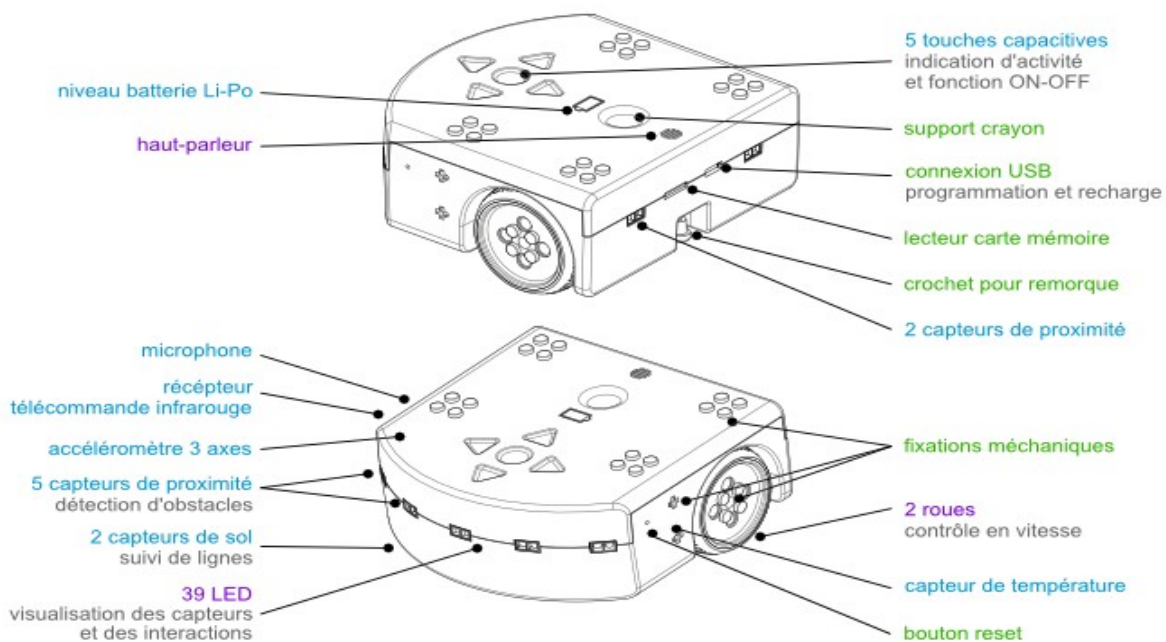
Introduction :

Le Thymio II est un robot éducatif. Il se base sur trois piliers :

- Une grande quantité de capteurs et d'actuateurs
- Une interactivité éducative basée sur la lumière et le touché
- Une programmation facile graphique et textuelle

Les capteurs principaux sont :

- Les capteurs de distance. Ils permettent de mesurer s'il y a un objet à proximité. Ils fonctionnent en émettant de la lumière infrarouge et en mesurant combien de lumière retourne en arrière.
- Les accéléromètres.
- Un microphone, qui permet de percevoir des sons.
- Un capteur de température (un thermomètre électronique)



actuateurs

capteurs

support

I – Nos recherches

Notre première démarche :

* Comprendre le fonctionnement du Thymio au démarrage, lorsqu'il s'allume et qu'il s'éclaire :

- Vert :

Bouton avant permet d'augmenter la vitesse

Bouton arrière permet de diminuer la vitesse

Bouton du milieu fait que le Thymio s'arrête

Lorsque l'un des 5 capteurs de proximité capte une présence le robot suit cette présence

- Magenta

Thymio II suit les ordres donnés par les boutons ou une télécommande

- Cyan :

Boutons avant et arrière + capteurs au dessus de la piste permettent de calibrer le niveau de noir.

Boutons gauche et droite + capteurs au dessus du blanc permettent de calibrer le niveau de blanc. La piste doit être au minimum de 4cm de large et avoir un contraste élevé, le mieux c'est noir sur blanc.

- Bleue

Il réagit au son. On peut commander le robot avec des clappements de main.

- 1 claps= tourne ou avance tout droit
- 2 claps= marche / arrêt
- 3 claps= fait un cercle

- Rouge :

Thymio II détecte les chocs, la chute libre et montre la direction de la gravité

- Jaune

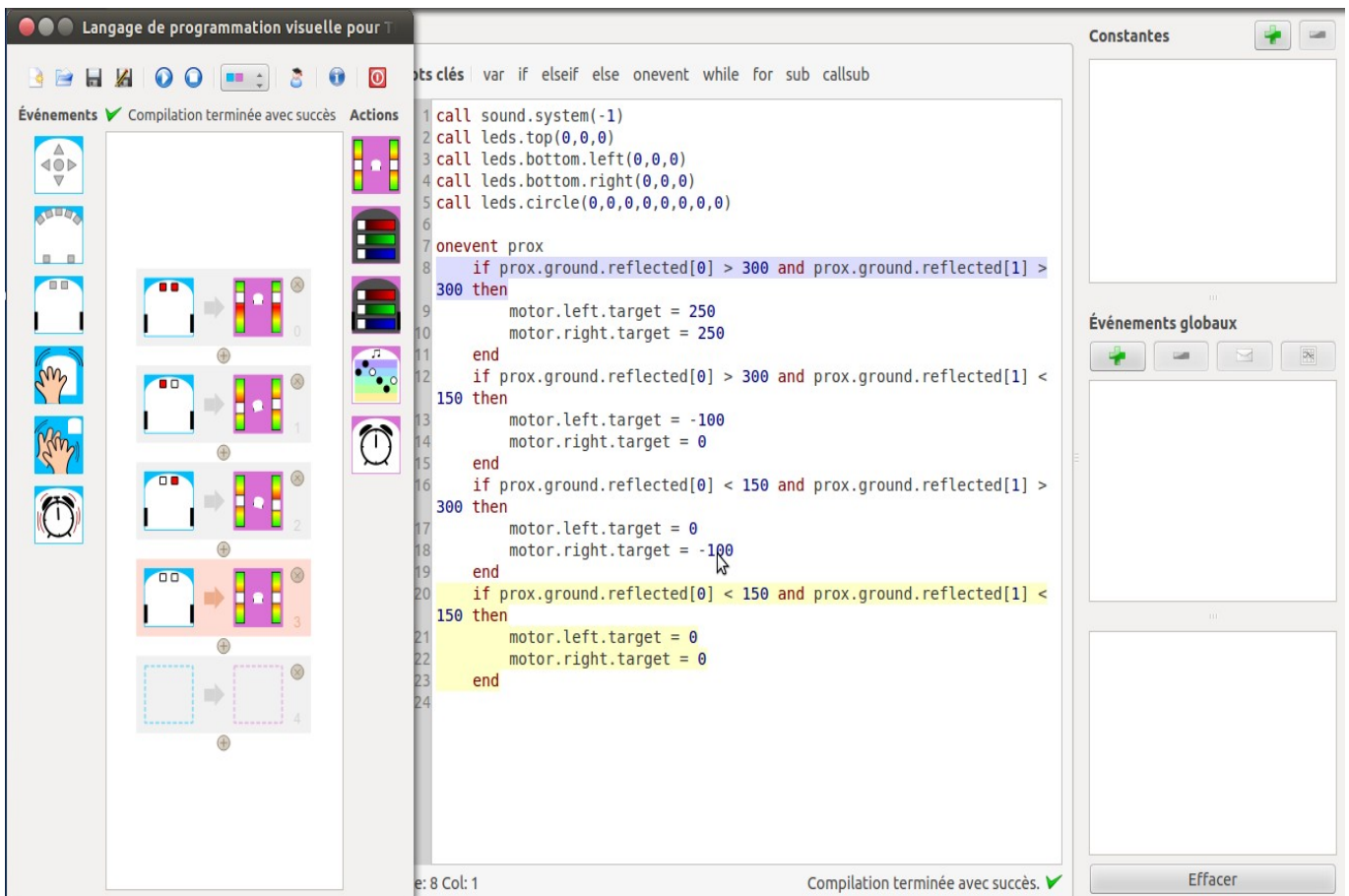
Plus il se rapproche de l'obstacle, plus ses capteurs de proximité renvoient de grande valeur

Un obstacle à gauche --> ralentit la roue droite et fait tourner le Thymio à droite

- 2 coefficients sont pris en comptes:

- Le premier ralentit si l'obstacle est frontal
- Le second fait tourner le Thymio si l'obstacle est sur le côté

Tout d'abord, pour comprendre le fonctionnement du Thymio, nous avons essayé de créer un programme visuellement avec vpl (Visual Programming Language), ce qui nous a permis de mieux comprendre le fonctionnement des capteurs.



```
1 call sound.system(-1)
2 call leds.top(0,0,0)
3 call leds.bottom.left(0,0,0)
4 call leds.bottom.right(0,0,0)
5 call leds.circle(0,0,0,0,0,0,0)
6
7 onevent prox
8   if prox.ground.reflected[0] > 300 and prox.ground.reflected[1] >
9     300 then
10      motor.left.target = 250
11      motor.right.target = 250
12   end
13   if prox.ground.reflected[0] > 300 and prox.ground.reflected[1] <
14     150 then
15      motor.left.target = -100
16      motor.right.target = 0
17   end
18   if prox.ground.reflected[0] < 150 and prox.ground.reflected[1] >
19     300 then
20      motor.left.target = 0
21      motor.right.target = -100
22   end
23   if prox.ground.reflected[0] < 150 and prox.ground.reflected[1] <
24     150 then
25      motor.left.target = 0
26      motor.right.target = 0
27   end
28 end
```

* Les problèmes rencontrés :

Par la suite, le problème a été de faire fonctionner le programme pour des bandes noires plus ou moins épaisses de tel sorte que pour toutes largeurs de bandes, le programme puisse fonctionner.

Le programme fonctionnant uniquement pour des bandes de 4cm de largeur minimum (distance entre les 2 capteurs), lors de notre premier essai sur la piste il n'a donc pas fonctionné. De ce fait il a fallu rechercher d'autre programme afin de modifié le notre.

II – Travail de groupe

Freddy s'est chargé d'analyser les différents programmes afin de comprendre là où notre programme était défectueux.

Et Coralie s'est occupée de modifier les différents programmes, de prendre ce qui était utile à la réalisation du programme finale , de commenter les programmes ainsi que de la réalisation des algorithmes.

III – Résultats

* Algorithme pour le suivi de ligne :

Variables: g0 (réflecteur du dessous gauche [0])
g1 (réflecteur du dessous droit [1])
ml (moteur gauche)
mr (moteur droit)

Début algorithme:

DébutSi:

Si $g0 < 150$ et $g1 < 150$

Si les deux capteurs du dessous détectent du noir,
alors les deux moteurs prennent la valeur 250

ml prend la valeur 250

mr prend la valeur 250

sinonsi $g0 > 300$ et $g1 > 300$

Sinon, si les deux capteurs détectent le blanc,
alors les deux moteurs s'arrêtent

ml prend la valeur 0

mr prend la valeur 0

sinonsi $g0 > 300$ et $g1 < 150$

Sinon, si le capteur du dessous gauche détecte du
blanc et le capteur du dessous droit du noir

ml prend la valeur 0


```
# Alors le moteur gauche prend la valeur 0,s'arrête.  
    mr prend la valeur -50  
  
# Et le moteur droit prend la valeur -50, le thymio  
se repositionne vers la droite  
    sinonsi  $g0 < 150$  et  $g1 > 300$   
# Sinon, si le capteur du dessous droit détecte du  
blanc et le capteur du dessous gauche du noir  
    ml prend la valeur -50  
# Et le moteur gauche prend la valeur -50, le  
thymio se repositionne vers la gauche  
    mr prend la valeur 0  
# Alors le moteur droit prend la valeur 0,s'arrête.  
    FinSi  
Fin algorithme
```

Conclusion :

Après plusieurs modifications apportées au programme, celui-ci fonctionne pour toutes les largeurs de bande au sol. Le Thymio est donc capable de suivre un trajet en se replaçant, suivant toujours la bande noire.

Même si le programme fonctionne pour toutes les largeurs de bande, il faut prendre en compte l'angle d'attaque du Thymio. Selon cet angle, le temps de réaction du Thymio sera (ou pas assez) rapide pour lui permettre de se replacer sur la bande. Plus la bande est fine, plus l'angle diminue

Bibliographie :

- <http://www.generationrobots.com/fr/401213-robot-mobile-thymio-2.html>
- <http://stefans-robots.net/pdfs/en/fuzzy-line-following-nxt-robot.pdf>
- <https://github.com/aseba-community/thymio-programming-exercises/blob/master/programs/line.aesl>
- <https://aseba.wikidot.com/fr>

Annexe

* Programme avec VPL :

```

onevent prox
  if prox.ground.reflected[0] < 150 and
prox.ground.reflected[1] < 150 then
! Si les capteurs du dessous détectent du noir alors thymio
s'avance
    motor.left.target = 250 ! Le moteur gauche avance a 250
    motor.right.target = 250 ! Le moteur droit avance a 250
  end
  if prox.ground.reflected[0] > 300 and
prox.ground.reflected[1] > 300 then
! Si les capteurs du dessous détectent du blanc alors thymio
s'arrete
    motor.left.target = 0 ! Le moteur gauche s'arrête
    motor.right.target = 0 ! Le moteur droit s'arrête
  end
  if prox.ground.reflected[0] > 300 and
prox.ground.reflected[1] < 150 then
! Si le capteur du dessous gauche (0) détecte du blanc, il ralentit
la vitesse du moteur droit
    motor.left.target = 0 ! Le moteur gauche s'arrête
    motor.right.target = -50 ! Le moteur droit recule et
ralenti, permet la replacement
  end
  if prox.ground.reflected[0] < 150 and
prox.ground.reflected[1] > 300 then
! Si le capteur du dessous droit (1) détecte du blanc, il ralentit la
vitesse du moteur gauche
    motor.left.target = -50 ! Le moteur gauche recule et
ralenti, permet le replacement
    motor.right.target = 0 ! Le moteur droit s'arrête
  end
end

```

* Programme de suivi de ligne 1 :

```

<!DOCTYPE aesl-source>
<network>

```

```

motor_change = (motor * CHANGE)/100
state = 0

```

```

<!--list of global events-->

<!--list of constants-->
<constant value="25"
name="CHANGE"/>
<constant value="150"
name="THRESHOLD"/>
<constant value="100" name="DIFF"/>
<constant value="20"
name="TIMER"/>
<constant value="100"
name="INCREMENT"/>

<!--show keywords state-->
<keywords flag="true"/>
<!--node thymio-II-->
<node name="thymio-II"># Thymio-II
program: follow a line
# Copyright 2013 by Moti Ben-Ari
# CreativeCommons BY-SA 3.0
# Constants
# THRESHOLD for sensing the tape
# DIFF difference to remember last
deviation
# CHANGE percentage change of
motors for steering
# TIMER timer for computing
direction

# Variables
var motor # The base value of the
motor power
var motor_change # The amount to
change when approaching and turning
var state # 0 = off, 1 = on
var diff # The difference between the
two ground sensors
var dir # The last direction: 1 = right,
-1 = left
# Initialization
motor = 0
if button.center == 0 then
  # If off, set state to 1 (on) and dirve
dir = 0
timer.period[0] = TIMER
call leds.circle(0,0,0,0,0,0,0,0)
call leds.buttons(0,0,0,0)
# Set the circle leds to indicate the motor
power
sub set_circle_leds
  if motor/100==0 then call
leds.circle(0,0,0,0,0,0,0,0) end
  if motor/100==1 then call
leds.circle(32,0,0,0,0,0,0,0) end
  if motor/100==2 then call
leds.circle(32,32,0,0,0,0,0,0) end
  if motor/100==3 then call
leds.circle(32,32,32,0,0,0,0,0) end
  if motor/100==4 then call
leds.circle(32,32,32,32,0,0,0,0) end
  if motor/100==5 then call
leds.circle(32,32,32,32,32,0,0,0) end

# Left and right button event handlers
# Increase or decrease motor power
within 0-500
onevent button.left
  if button.left == 0 then
    motor = motor - INCREMENT
    if motor < 0 then motor = 0 end
    motor_change = (motor *
CHANGE)/100
    callsub set_circle_leds
  end
onevent button.right
  if button.right == 0 then
    motor = motor + INCREMENT
    if motor > 500 then motor = 500 end
    motor_change = (motor *
CHANGE)/100
    callsub set_circle_leds
  end
# When center button released
onevent button.center
  # If both sensors no longer detect the
tape

```

```

forwards
  if state == 0 then
    state = 1
    motor.left.target = motor
    motor.right.target = motor
    # else if state is on, call stop the
motors
  else
    state = 0
    motor.left.target = 0
    motor.right.target = 0
  end
  motor_change = (motor *
CHANGE)/100
  callsub set_circle_leds
end
# Timer event occurs
onevent timer0
# Compute difference between the
values of the ground sensors
diff = prox.ground.delta[0] -
prox.ground.delta[1]
# If absolute value of the difference is
above the threshold
if abs diff >= DIFF then
  # Compute direction according to
the sign of the difference
  if diff > 0 then
    dir = 1
    call leds.top(0, 32, 0)
  else
    dir = -1
    call leds.top(0, 0, 32)
  end
else
  call leds.top(0, 0, 0)
end
# Drive straight
sub drive
  motor.left.target = motor
  motor.right.target = motor
  call leds.buttons(32,0,0,0)
end

```

```

if prox.ground.delta[0] > THRESHOLD
and
  prox.ground.delta[1] > THRESHOLD
then
  # Turn left or right depending on the
value in dir
  if dir == 1 then
    motor.left.target = motor +
motor_change
    motor.right.target = motor -
motor_change
    call leds.buttons(0,32,0,0)
  else
    motor.left.target = motor -
motor_change
    motor.right.target = motor +
motor_change
    call leds.buttons(0,0,0,32)
  end
  # If one of the ground sensors finds the
tape
  # turn the robot in the appropriate
direction
  elseif prox.ground.delta[0] >
THRESHOLD then
    motor.left.target = motor +
motor_change
    motor.right.target = motor -
motor_change
    call leds.buttons(0,32,0,0)
  elseif prox.ground.delta[1] >
THRESHOLD then
    motor.left.target = motor -
motor_change
    motor.right.target = motor +
motor_change
    call leds.buttons(0,0,0,32)
  # Otherwise, drive straight
else

```

```

# Proximity event occurs, call
subroutine drive
onevent prox
  if state == 1 then
callsub drive
  end
</node>
</network>

```

* Programme dans le Thymio :

Variables

```

var motor # Valeur de base pour la puissance des moteurs
var motor_change # valeur du changement lors d'une approche ou d'un virage
var state # 0 = off (Le Thymio est en arrêt), 1 = on (Le Thymio est en marche)
var diff # La difference entre les 2 grands capteurs au sol
var dir # La direction finale: 1 = droite, -1 = gauche

```

Initialisation

```

motor = 0 # La puissance des moteur est nulle
motor_change = (motor * 25)/100
state = 0 # Le thymio est en arrêt
dir = 0 # le thymio ne va dans aucune direction
timer.period[0] = 20 # Période pendant laquelle il y a le calcul de la direction
a prendre
call leds.circle(0,0,0,0,0,0,0,0) # Aucune led n'est allumée
call leds.buttons(0,0,0,0) # Aucune leds des boutons n'est activé

```

Désigner les leds en cercle pour indiquer la puissance du moteur

sub set_circle_leds

```

  if motor/100==0 then call leds.circle(0,0,0,0,0,0,0,0) end
# Si aucune led n'est allumée alors la vitesse des moteurs est nulle
  if motor/100==1 then call leds.circle(32,0,0,0,0,0,0,0) end
# Si la led frontale est allumée alors la vitesse des moteurs divisé par 100 vaut 1,
soit motor = 100
  if motor/100==2 then call leds.circle(32,32,0,0,0,0,0,0) end
# Si les led frontale et 2eme à droitesont allumées alors la vitesse des moteurs
divisé par 100 vaut 2, soit motor = 200
  if motor/100==3 then call leds.circle(32,32,32,0,0,0,0,0) end
# Si la led frontale,les 2eme et 3eme à droite sont allumées alors la vitesse des
moteurs divisé par 100 vaut 3, soit motor = 300
  if motor/100==4 then call leds.circle(32,32,32,32,0,0,0,0) end # Si la led
frontal, les 2eme, 3eme et 4eme à droite sont allumées alors la vitesse des

```

```
moteurs divisé par 100 vaut 4 soit motor = 400
  if motor/100==5 then call leds.circle(32,32,32,32,32,0,0,0) end # Si la led
frontal, les 2eme, 3eme,4eme et 5eme sont allumées alors la
                                vitesse des moteurs divisé par 100 vaut 5,
soit motor = 500
```

```
# boutons événements gauche et droit
# Augmente ou diminue la puissance du moteur entre 0 et 500
onevent button.left
  if button.left == 0 then
# Si on appuie sur le bouton gauche, on soustrait 100 à la puissance du moteur
    motor = motor - 100
    if motor < 0 then motor = 0 end
# Si la puissance est négative alors elle vaut 0
    motor_change = (motor * 25)/100
    callsub set_circle_leds
# Faire appel à la désignation des leds en cercle pour la puissance
  end
```

```
onevent button.right
  if button.right == 0 then
# Si on appuie sur le bouton droit, on ajoute 100 à la puissance du moteur
    motor = motor + 100
    if motor > 500 then motor = 500 end
# Si la puissance est supérieure à 500 alors elle vaut 500
    motor_change = (motor * 25)/100
    callsub set_circle_leds
# Faire appel à la désignation des leds en cercle pour la puissance
  end
```

```
# lorsque le bouton central est relâché
onevent button.center
  if button.center == 0 then                                # Si on appuie sur le bouton au centre
    if state == 0 then
# Si le thymio est en arrêt (off) , state prend 1 (on) et le thymio avance
      state = 1
      motor.left.target = motor
      motor.right.target = motor
    else
# Sinon si state vaut 1 ;le thymio est en marche (on); alors les moteurs
s'arretent
      state = 0
      motor.left.target = 0
```



```

    motor.right.target = 0
end
motor_change = (motor * 25)/100
# Calcul de la valeur de la puissance du moteur
    callsub set_circle_leds
# Faire appel à la désignation des leds en cercle pour la puissance
end

# L'évenement time apparait
onevent timer0
    diff = prox.ground.delta[0] - prox.ground.delta[1]
# calcul de la difference entre les valeurs des capteurs au sol
    if abs diff >= 100 then
# Si la valeur absolue de la difference est plus élevée que 150
# Calcule de la direction selon le signe de la différence
        if diff > 0 then      # Si la différence est négative, le thymio tourne à droite
            dir = 1
                call leds.top(0, 32, 0)          # le thymio s'éclaire vert
            else              # Sinon, le thymio tourne à gauche
                dir = -1
                call leds.top(0, 0, 32)          #le thymio s'éclaire bleu
            end
        else                  # Sinon le thymio avance tout droit
            call leds.top(0, 0, 0)              # le thymio ne s'éclaire d'aucune couleur
        end

# Avancer droit
sub drive
    if prox.ground.delta[0] > 150 and
# Si les deux capteurs ne detectent plus la bande
        prox.ground.delta[1] > 150 then
# Le thymio tourne a gauche ou a droite en fonction de la valeur de dir
        if dir == 1 then      # Si dir = 1 alors la vitesse du moteur gauche augmente
et celle du moteur droit diminue
            motor.left.target = motor + motor_change
            motor.right.target = motor - motor_change
            call leds.buttons(0,32,0,0)          # la led sous la flèche droite s'allume
        else                  # Si dir = 0 alors la vitesse du moteur droit
augmente et celle du moteur gauche diminue
            motor.left.target = motor - motor_change
            motor.right.target = motor + motor_change
            call leds.buttons(0,0,0,32)          # la led sous la flèche de gauche s'allume
        end                    # Si l'un des deux capteurs trouve la

```

bande,tourner le robot dans la direction appropriée

```
elseif prox.ground.delta[0] > 150 then # Si le capteur au sol gauche ne
detecte plus de bande, le moteur gauche augmente sa puissance tandis que le
moteur droit la diminue, le thymio tourne alors vers la gauche
```

```
motor.left.target = motor + motor_change
```

```
motor.right.target = motor - motor_change
```

```
call leds.buttons(0,32,0,0) # la led sous la flèche droite s'allume
```

```
elseif prox.ground.delta[1] > 150 then # Sinon, si le capteur droit ne détecte
plus la bande, La puissance du moteur gauche diminue et celle du moteur droit
augmente, le Thymio tourne vers la droite
```

```
motor.left.target = motor - motor_change
```

```
motor.right.target = motor + motor_change
```

```
call leds.buttons(0,0,0,32) # La led sous la flèche de gauche s'allume
```

```
else
```

```
# Sinon le Thymio avance tout droit, les deux moteurs on la même puissance
```

```
motor.left.target = motor
```

```
motor.right.target = motor
```

```
call leds.buttons(32,0,0,0) # La led sous la flèche frontale s'allume
```

```
end
```

```
# événement de proximité apparait,appel à 'subroutine drive'
```

```
onevent prox
```

```
if state == 1 then
```

```
# Si le thymio est en marche, alors faire appel a sub drive
```

```
callsub drive
```

```
end
```