

Suivie de Lignes

En utilisant le capteur prox.ground.delta[0] (qu'on renommera lecture_de_la_ligne) qui nous affiche un nombre qui varie selon la couleur que le capteur détecte entre le noir ou blanc. Ensuite on doit s'assurer que le robot parcourt la ligne en restant sur un niveau du gris donné.



Programmation

On choisit une valeur de gris (entre 200 et 700 dans cet exemple), on programme avec le principe de boucle fermée on fait recours à la programmation événementielle

```
lecture_de_la_ligne = prox.ground.delta[0]
lecture_du_code_barre = prox.ground.delta[1]

if switch==1 then
  if lecture_de_la_ligne<200 then #si gauche voit du noir, alors il vire à droite
    motor.lefttarget=50
    motor.righttarget=0
  elseif lecture_de_la_ligne>700 then #si gauche voit du blanc, alors il vire à gauche
    motor.lefttarget=0
    motor.righttarget=50
  else #sinon il va tout droit
    motor.lefttarget=100
    motor.righttarget=100
  end
end
```

Lecture du code-barre

Dans cette étape, on utilise le capteur prox.ground.delta[1] qui va lire deux codes-barres le premier qui va nous donner la distance et une couleur entre la fin de la piste et le trou puis le second qui contient la direction du green.



Programmation

Avec une variable cpt (compteur) et la fonction Timer period qu'on met à 375, si thymio a détecté l'amorce on compte deux fois le compteur pour chaque case de code-barre, et chaque fois on vérifie si elle est noire ou blanche après on stocke les valeurs dans un tableau. En fonction, des valeurs du tableau on aura des couleurs et des distances décodées pour le premier code-barre et un angle pour le deuxième.

Code barre	Valeur entière avec codage défini	angle par rapport à la ligne de parcours en degré, sens direct
1000	0	0 deg
1100	1	+45 deg
1010	2	-45 deg
1110	3	+90 deg
1001	4	-90 deg
1101	5	+120 deg
1011	6	-120 deg
1111	7	0 deg

Conclusion

Le module robot mobile nous a permis d'apprendre le langage robotique avec de la programmation événementielle. Ce projet nous donne un avant-goût sur le monde réel.

```
#Lecture du premier code barre
if amorce==1 and nombre_du_code_lu <2 then
  if cpt==2 then
    if lecture_du_code_barre <400 then
      bit[1]=0
    else
      bit[1]=1
    end
    call leds.top(0,0,0)
  elseif cpt==4 then
    if lecture_du_code_barre <400 then
      bit[2]=0
    else
      bit[2]=1
    end
    call leds.top(0,0,0)
  elseif cpt==6 then
    if lecture_du_code_barre <400 then
      bit[3]=0
    else
      bit[3]=1
    end
  end
  #fin du code barre
  emit data[bit[1],bit[2],bit[3]]
end
```

La distance est calculée avec cette formule:

$$d = \sum_{i=0}^2 b_i \cdot 2^i$$

