

La Smart Camera de Makeblock

La Smart Camera de Makeblock est une caméra « intelligente ». Elle intègre un système d'analyse des images qui lui permet de reconnaître les couleurs, suivre une ligne ou même lire certains codes-barres.

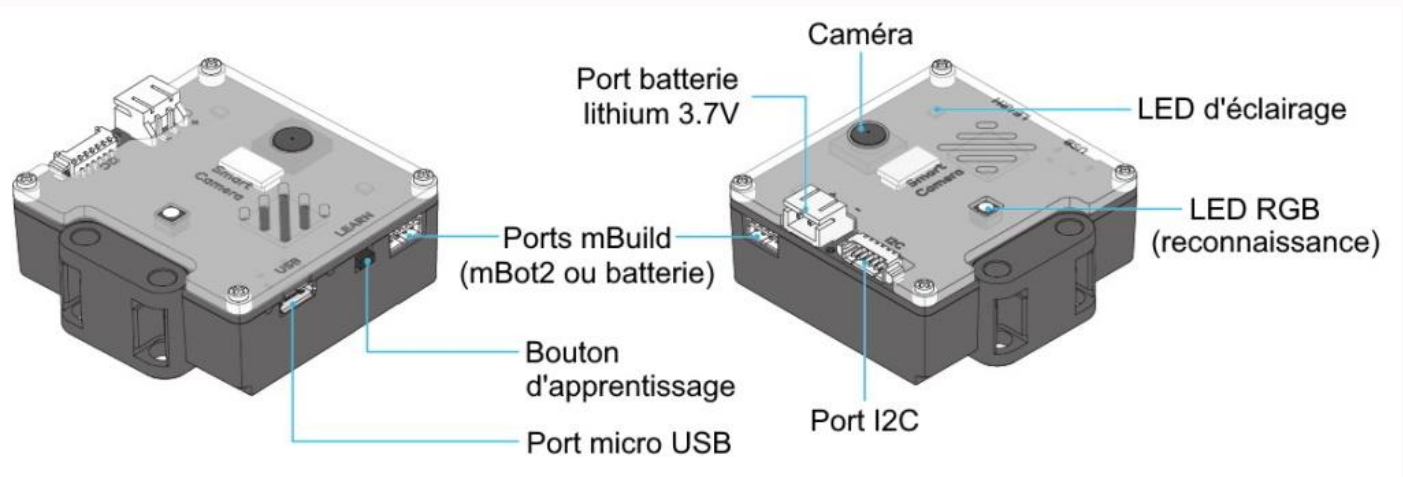
Contenu de la boîte

La boîte est de très bonne qualité, vous y trouverez le système de fixation (pièces métalliques, vis et écrous), les câbles pour relier entre eux les différents éléments, des balles vertes et jaunes/oranges, un assortiment codes barre (sous formes de cartes et d'autocollants) et bien sûr la caméra accompagnée de sa batterie rechargeable.



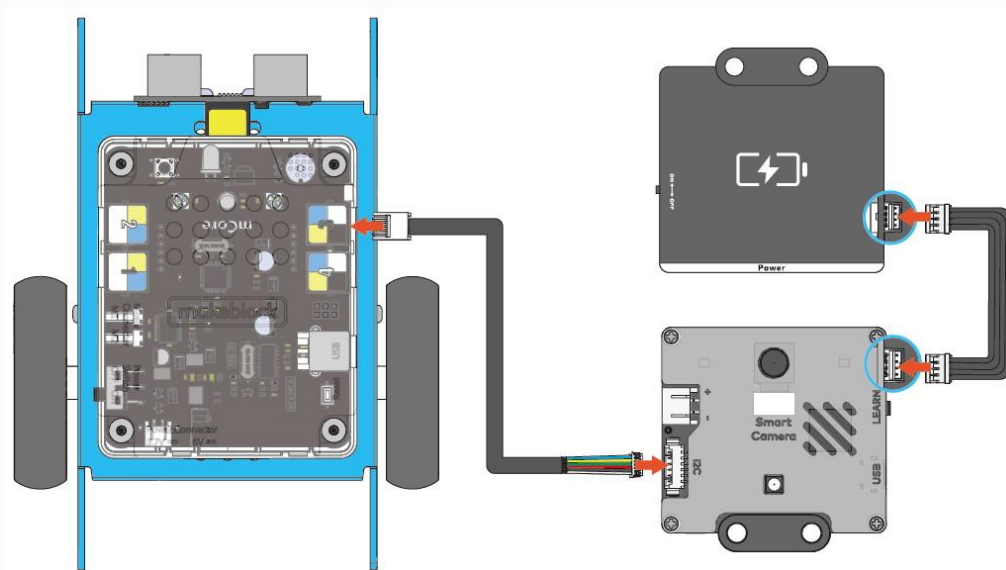
La caméra

La caméra possède un bouton d'apprentissage, une LED témoin (RGB) et une LED blanche très lumineuse. Le robot mBot sera branché sur le port I2C (avec un câble adaptateur vers le port RJ25 du robot). Comme la plupart des modules Makeblock récents, la caméra est équipée de 2 connecteurs mBuild. L'un d'eux permet de brancher la batterie rechargeable fournie dans le pack et l'autre pourrait être relié à une carte Halocode ou un robot mBot2 (mais je n'ai pas testé). Vous pouvez également alimenter la caméra avec une batterie lithium 3,7V (non fournie).



Branchement

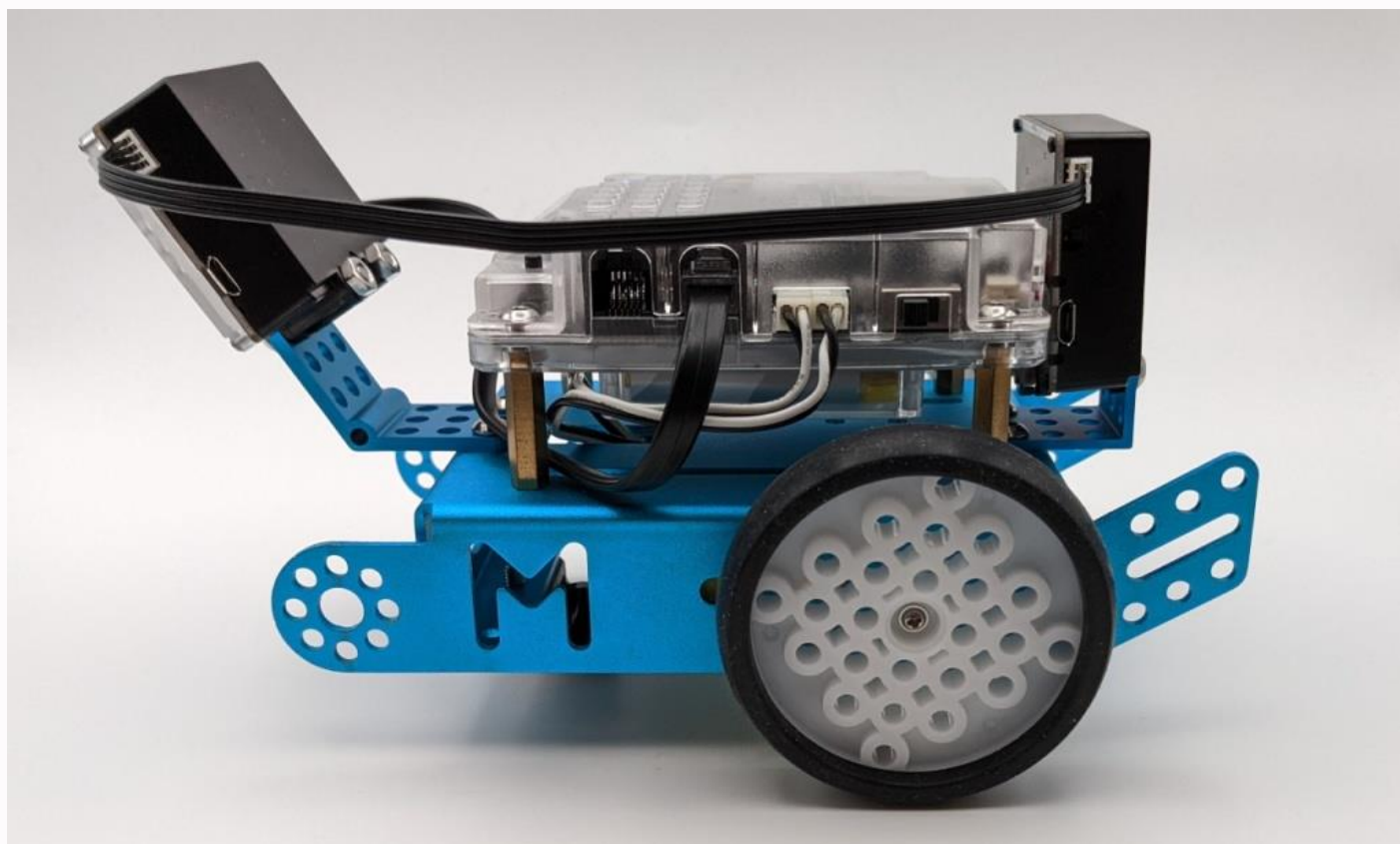
Voilà le schéma de branchement de la caméra au robot mBot. Vous n'êtes pas obligé d'utiliser le port n°3, il suffira simplement de l'indiquer dans votre programme.



Suivi de couleurs

La caméra est capable de mémoriser jusqu'à 7 couleurs différentes. Elle pourra ensuite les reconnaître et suivre un objet (de cette couleur).

Comme indiqué dans la documentation de Makeblock, j'ai fixé la Smart Camera à l'avant avec l'équerre inclinée à 45° et la batterie à l'arrière avec l'équerre à 90° (ça fonctionne très bien et c'est très facile puisque tous les éléments sont fournis dans la boîte).



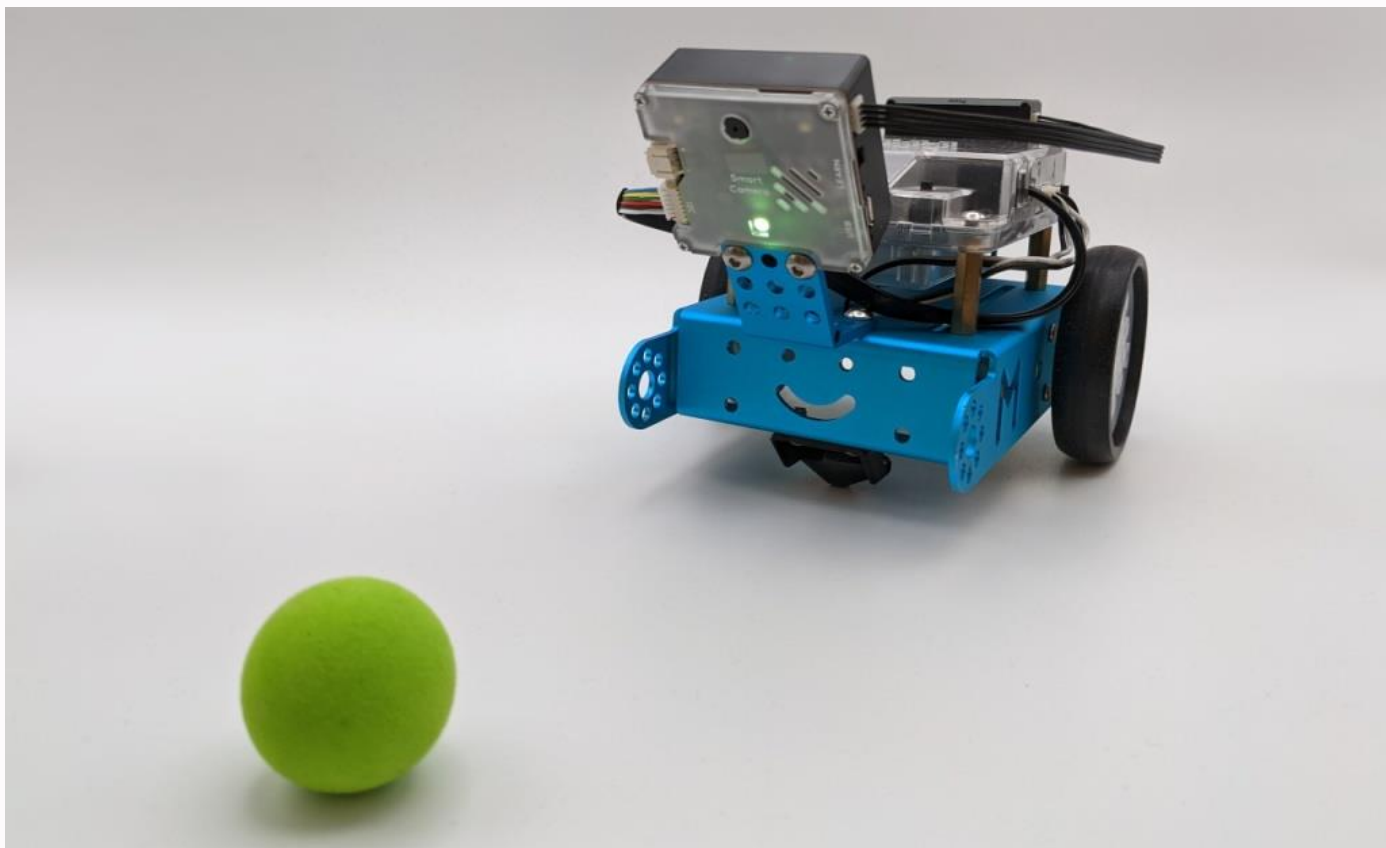
Pour faciliter la reconnaissance des couleurs, choisissez plutôt des objets monochromes, suffisamment grands et avec des couleurs bien marquées. Attention, la couleur change en fonction de la lumière. Pour éviter les problèmes, utilisez (si possible) la caméra dans le même environnement lumineux que pour l'apprentissage des couleurs. Faites aussi attention aux autres objets pouvant se trouver dans le champ de vision de la caméra. S'ils ont une couleur proche, il y a un risque de tromper la caméra.



Mémorisation des couleurs

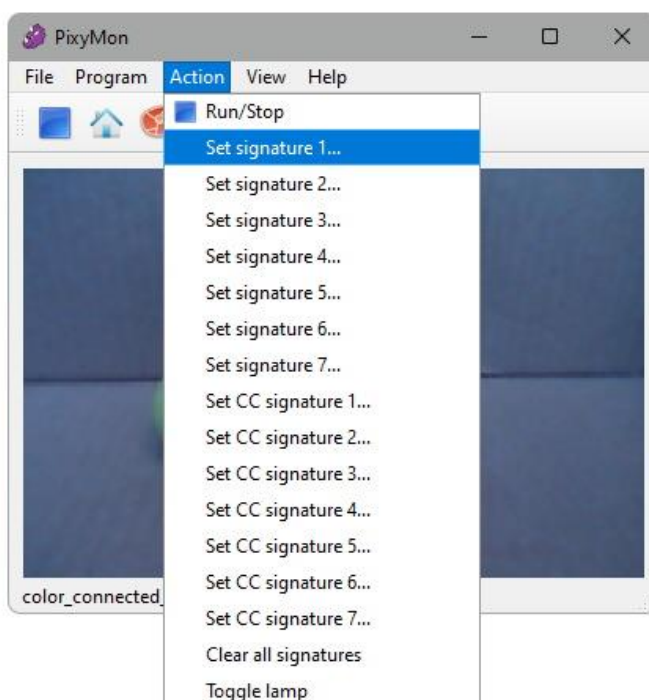
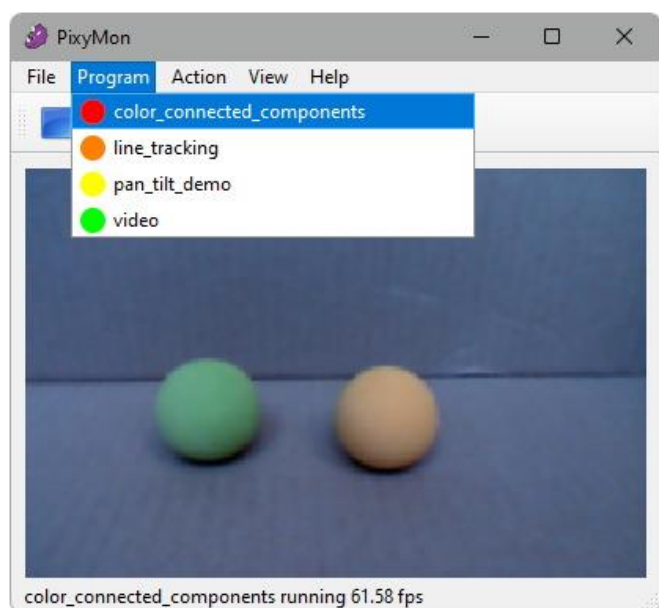
L'apprentissage des couleurs peut se faire directement depuis l'interface (le bouton poussoir) de la caméra :

- Allumez la caméra (avec l'interrupteur situé sur la batterie),
- Appuyez longuement sur le bouton d'apprentissage (la LED RGB s'allume successivement en rouge, orange, jaune, vert, cyan, bleu et violet).
- Relâchez le bouton
- Placez un objet de couleur devant la caméra, la LED RGB change de couleur (à peu près la même que l'objet)
- Appuyez brièvement sur le bouton pour apprendre la couleur
- Maintenant, en présence de l'objet, la LED RGB s'allume de la couleur correspondante (et si on éteint la caméra, les informations sont conservées).

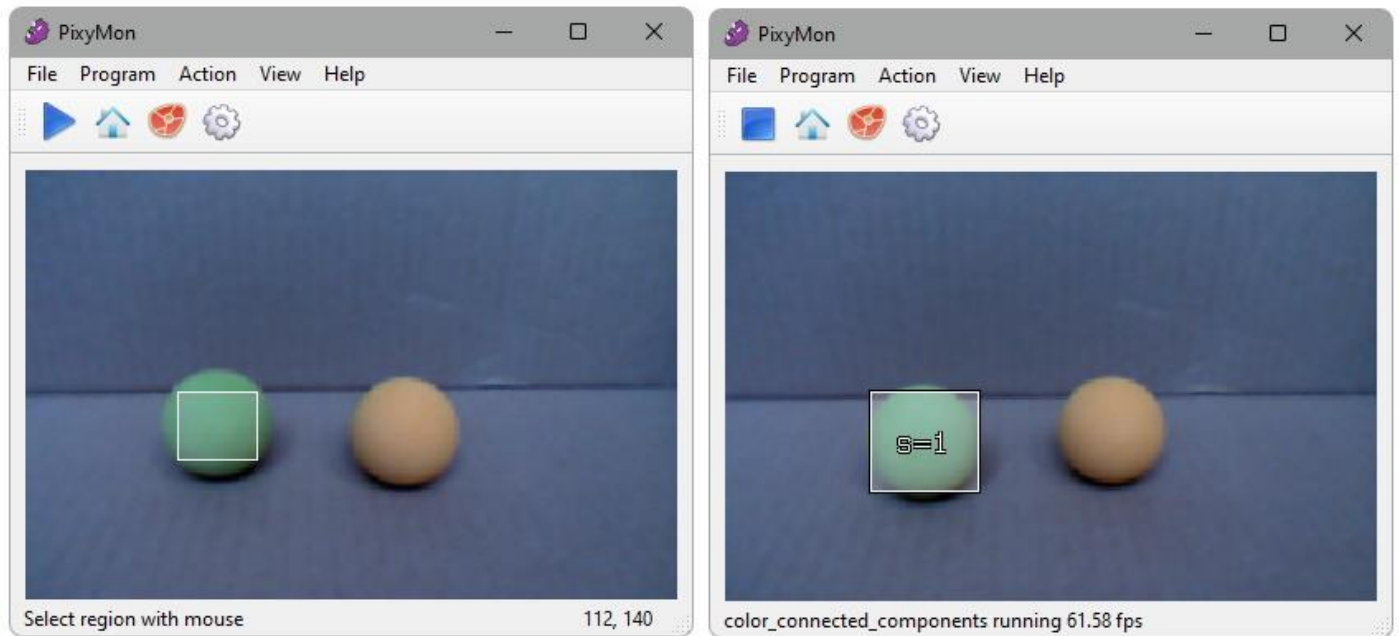


Cette méthode fonctionne, mais comme la Smart Camera ne possède pas d'écran, il est difficile de se rendre compte de ce qu'elle « voit » réellement. Après quelques essais on ne sait plus quel numéro correspond à chacune des couleurs et en cas d'erreur, il est impossible de tout effacer pour recommencer le processus d'apprentissage sur de bonnes bases.

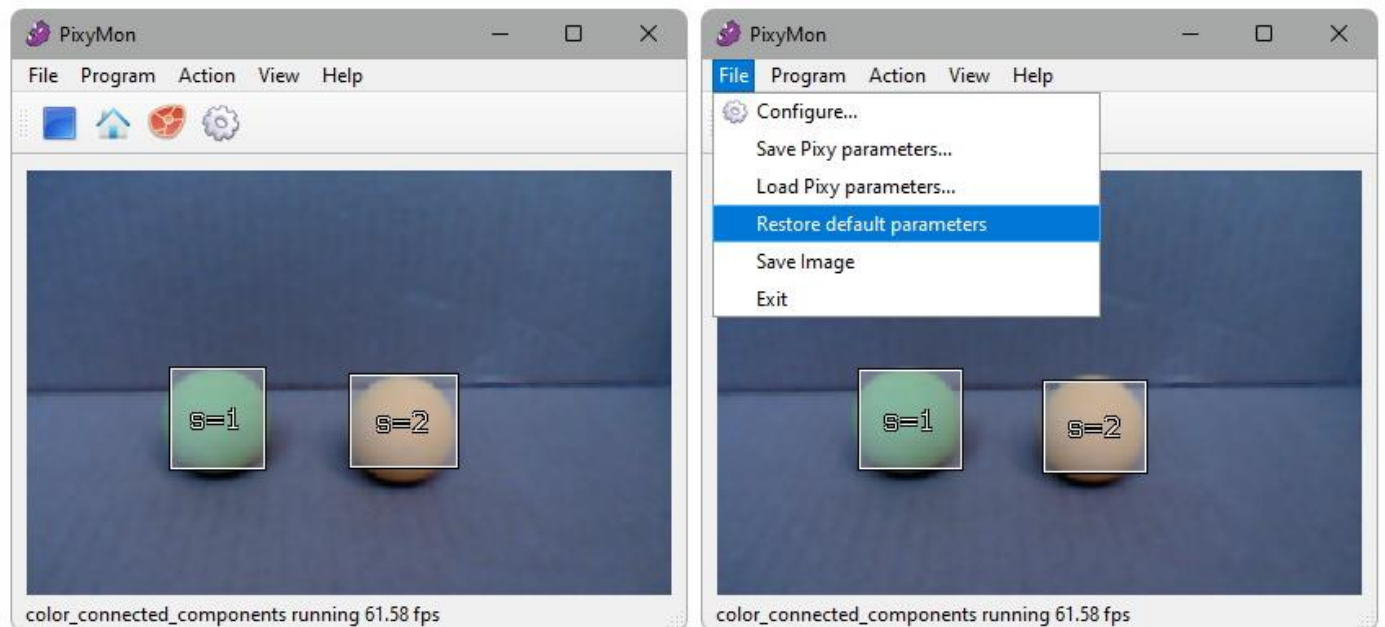
Heureusement, il existe une autre solution. **Téléchargez (et installez) PixyMon (v2)**, lancez le logiciel et reliez la caméra à l'ordinateur (en USB). L'image apparaît telle qu'elle est vue par la caméra, la résolution est très faible mais ça n'a pas vraiment d'importance (on ne va pas l'utiliser pour filmer le mariage de notre cousine). Allez dans **Program**, sélectionnez **color_connected_components** et ensuite allez dans **Action – Set signature 1...** pour mémoriser une couleur.



Encadrez la zone avec la souris (en cliquant avec le bouton gauche). Lorsque vous relâchez, la couleur est enregistrée (en cas d'erreur, il suffit de recommencer cette opération).

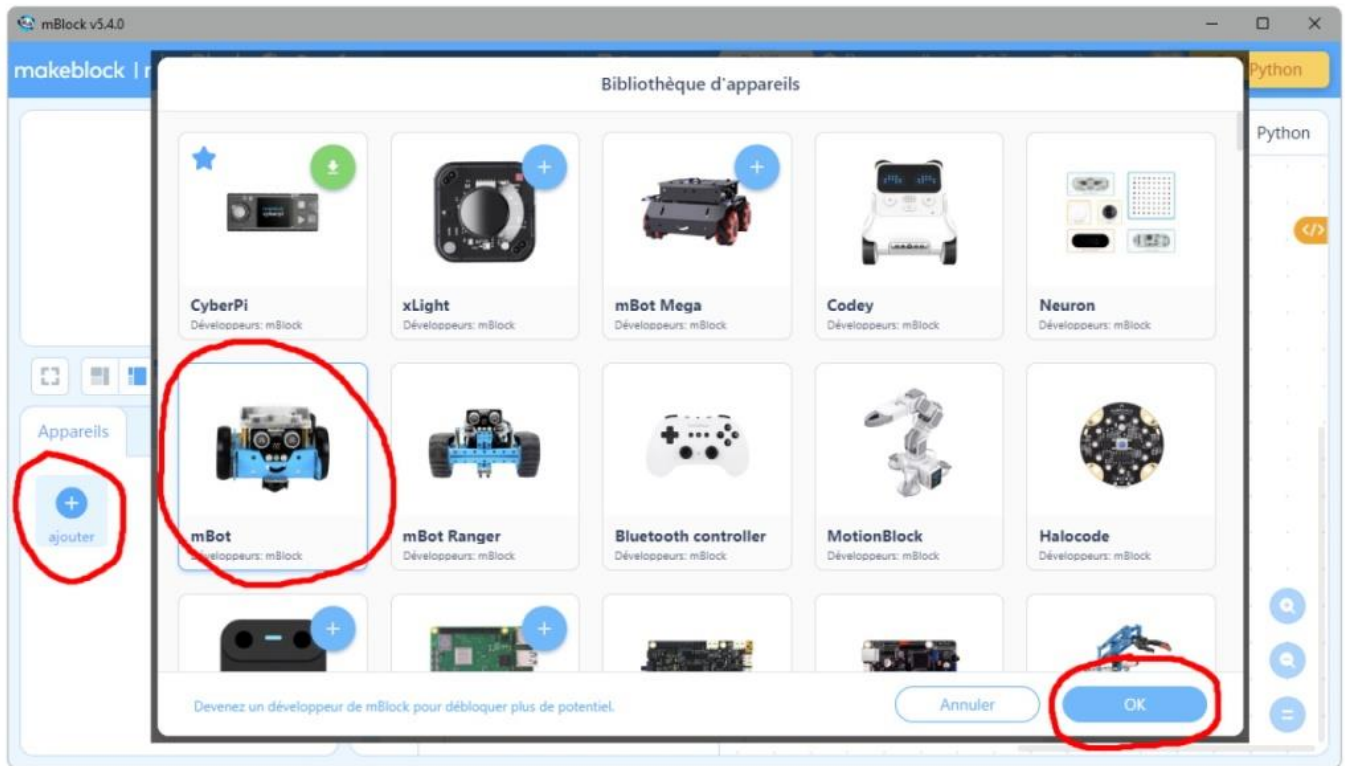


Vous pouvez enregistrer d'autres couleurs, effacer une seule couleur (**Action – Set CC signature x...**), toutes les couleurs mémorisées (**Action – Clear all signatures**) ou restaurer la configuration par défaut (**File – Restore default parameters**).

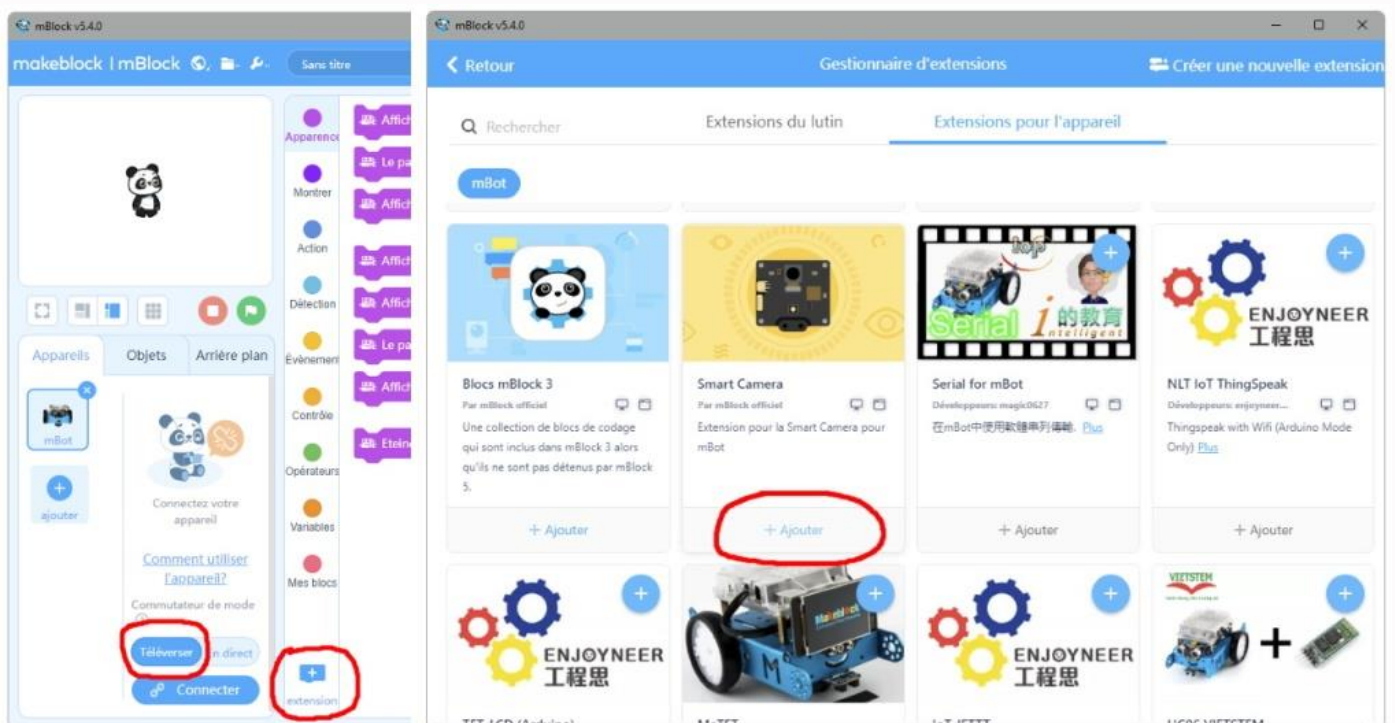


Programmation du robot

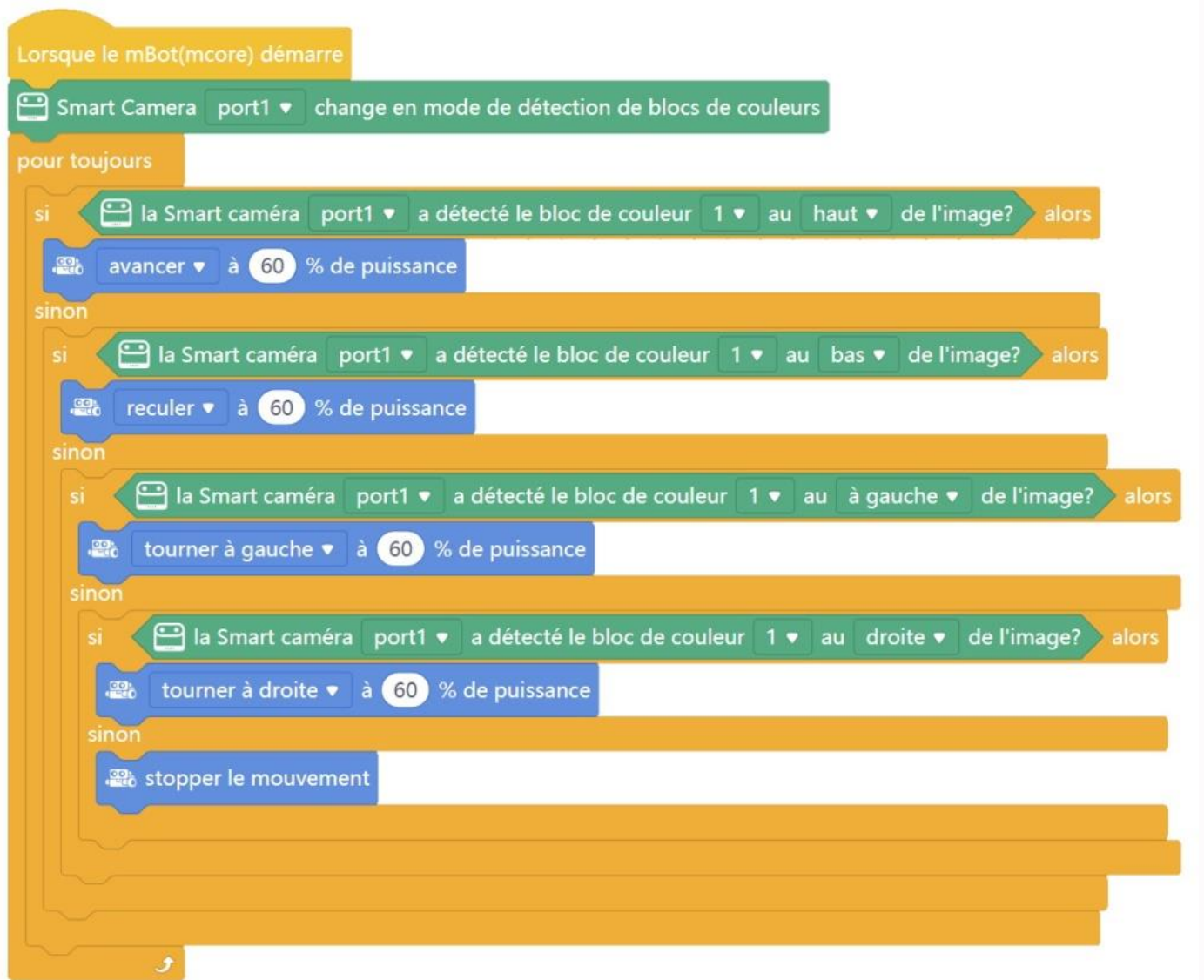
Téléchargez et installez la dernière version de mBlock. Lancez le programme, cliquez sur **ajouter** (à gauche), sélectionnez **mBot** et cliquez sur **OK**.



Pour obtenir les blocs nécessaires à la programmation de la Smart Camera, cliquez sur **extension**, puis sur **Ajouter** (en bas de **Smart Camera**). Sélectionnez ensuite le mode **Téléverser** (ça ne fonctionne pas en mode **En direct**).



Ces 2 exemples de code sont inspirés de la documentation Makeblock (le lien est en bas de la page). Ils permettent de suivre la couleur 1, mais vous pouvez facilement les modifier pour qu'ils en suivent une autre. Le premier programme est très facile à comprendre.



Mais le second est beaucoup plus performant.

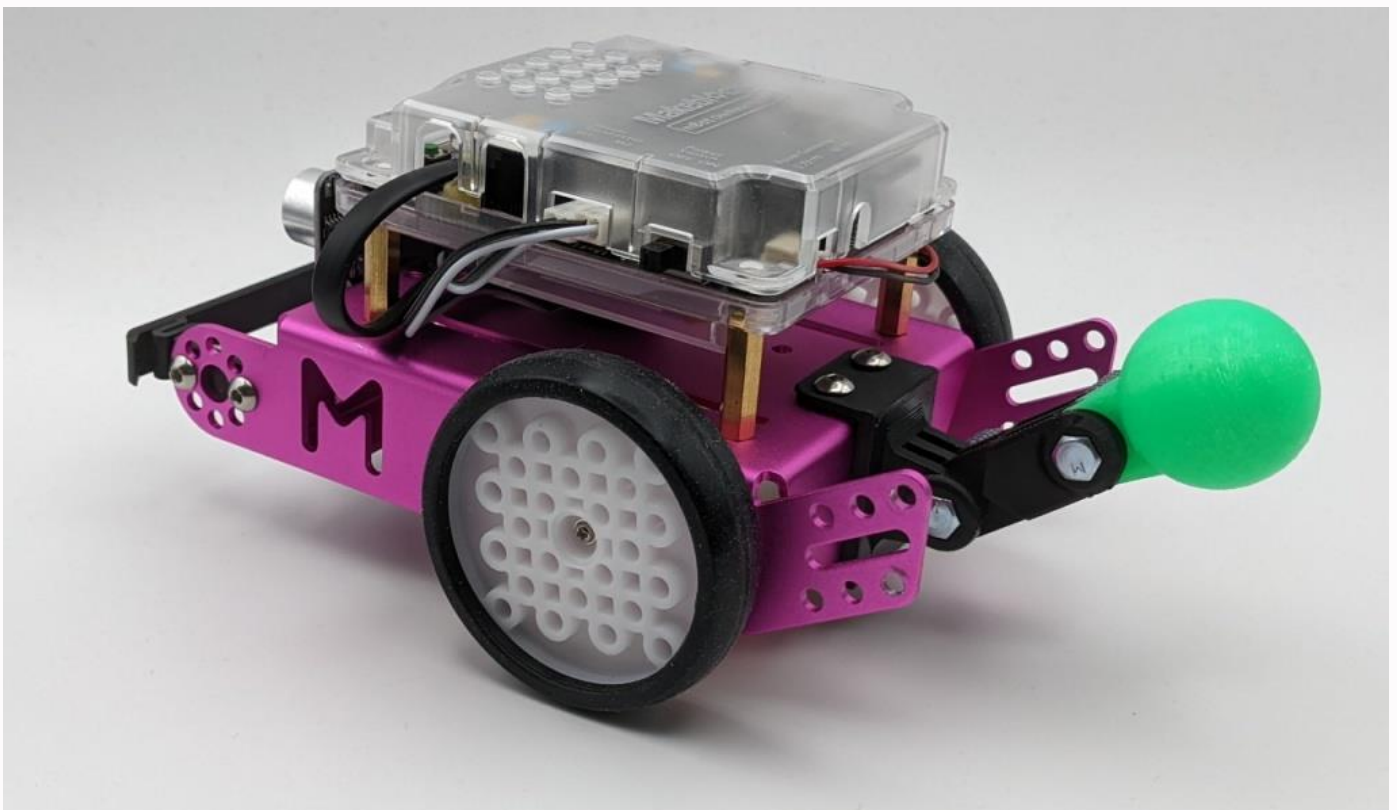

```

Lorsque le mBot(mcore) démarre
  Smart Camera port1 ▾ définit le coefficient Kp pour la vitesse différentielle du moteur à 0.3
  Smart Camera port1 ▾ change en mode de détection de blocs de couleurs
  pour toujours
    si Smart Camera port1 ▾ a détecté un bloc de couleur 1 ▾ ? alors
      allumer la lumière tout ▾ avec la couleur rouge 0 vert 100 bleu 0
    sinon
      allumer la lumière tout ▾ avec la couleur rouge 0 vert 0 bleu 0

    définir correction X ▾ à Smart Camera port1 ▾ calcule la vitesse différentielle du moteur (suivi automatique du bloc de couleur 1 ▾ à x ▾ de l'axe 160 )
    définir correction Y ▾ à Smart Camera port1 ▾ calcule la vitesse différentielle du moteur (suivi automatique du bloc de couleur 1 ▾ à y ▾ de l'axe 170 )
    définir roueGauche ▾ à 3 * correction Y + correction X
    définir roueDroite ▾ à 3 * correction Y - correction X
    si roueGauche > -25 et roueGauche < 25 alors
      définir roueGauche ▾ à 0
    si roueDroite > -25 et roueDroite < 25 alors
      définir roueDroite ▾ à 0
    roue gauche tourne à roueGauche % de puissance, roue droite à roueDroite % de puissance
  
```

Suivi d'un autre robot

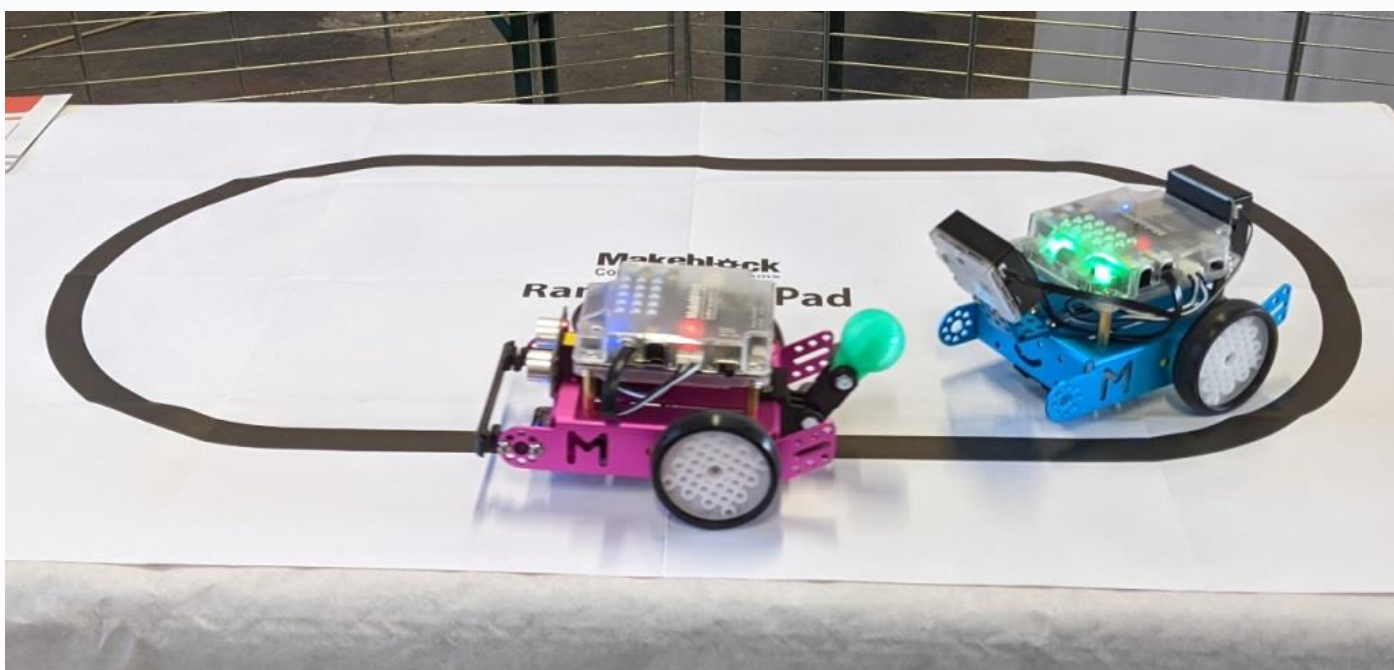
J'ai modélisé une boule et un support pour la fixer au robot (avec des vis et des écrous M4).



Si cela vous intéresse, [les fichiers 3D sont disponibles sur Printables.](#)

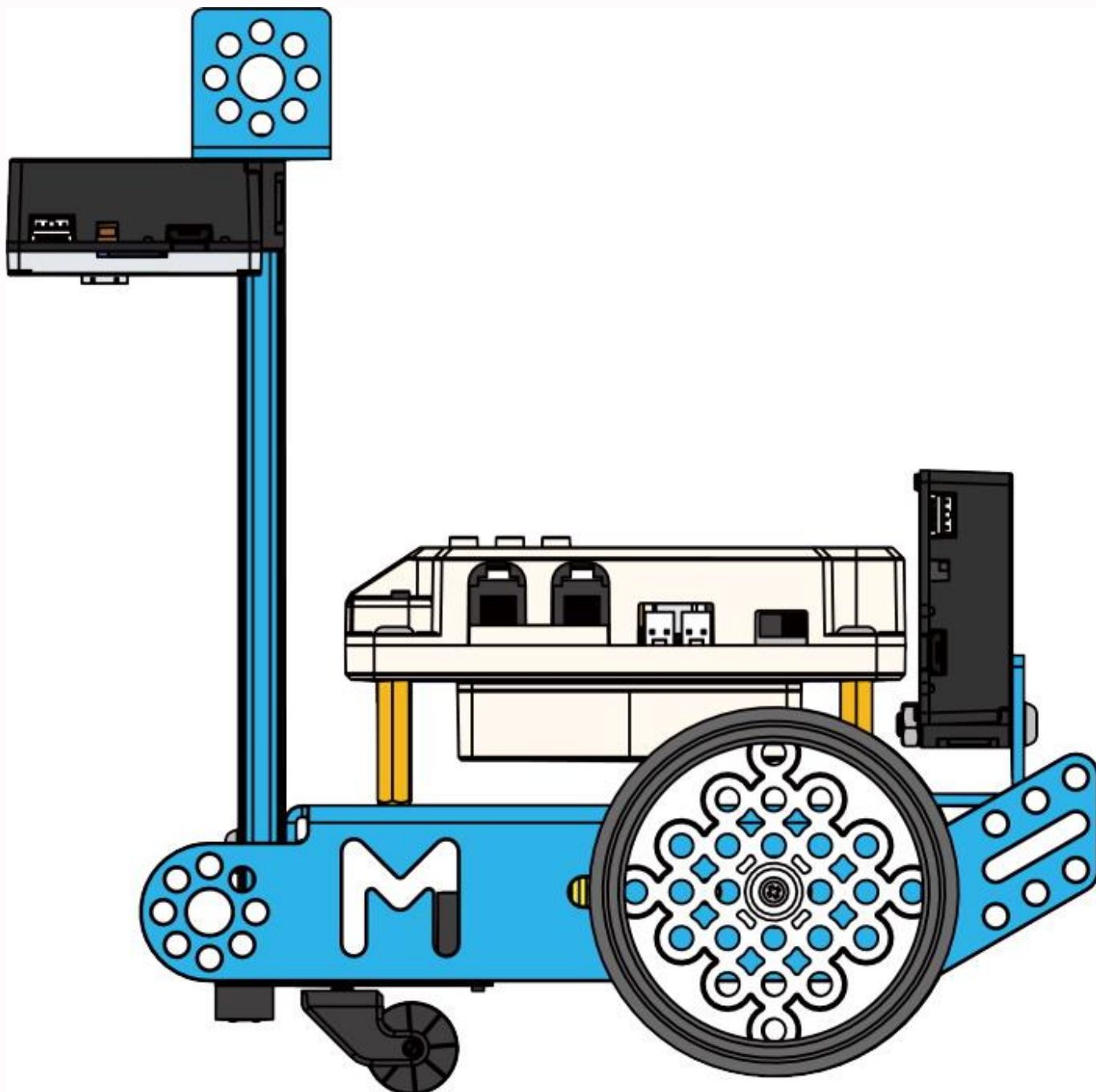


Le robot bleu suit la boule verte du robot rose (en mode suivi de ligne).



Suivi de ligne

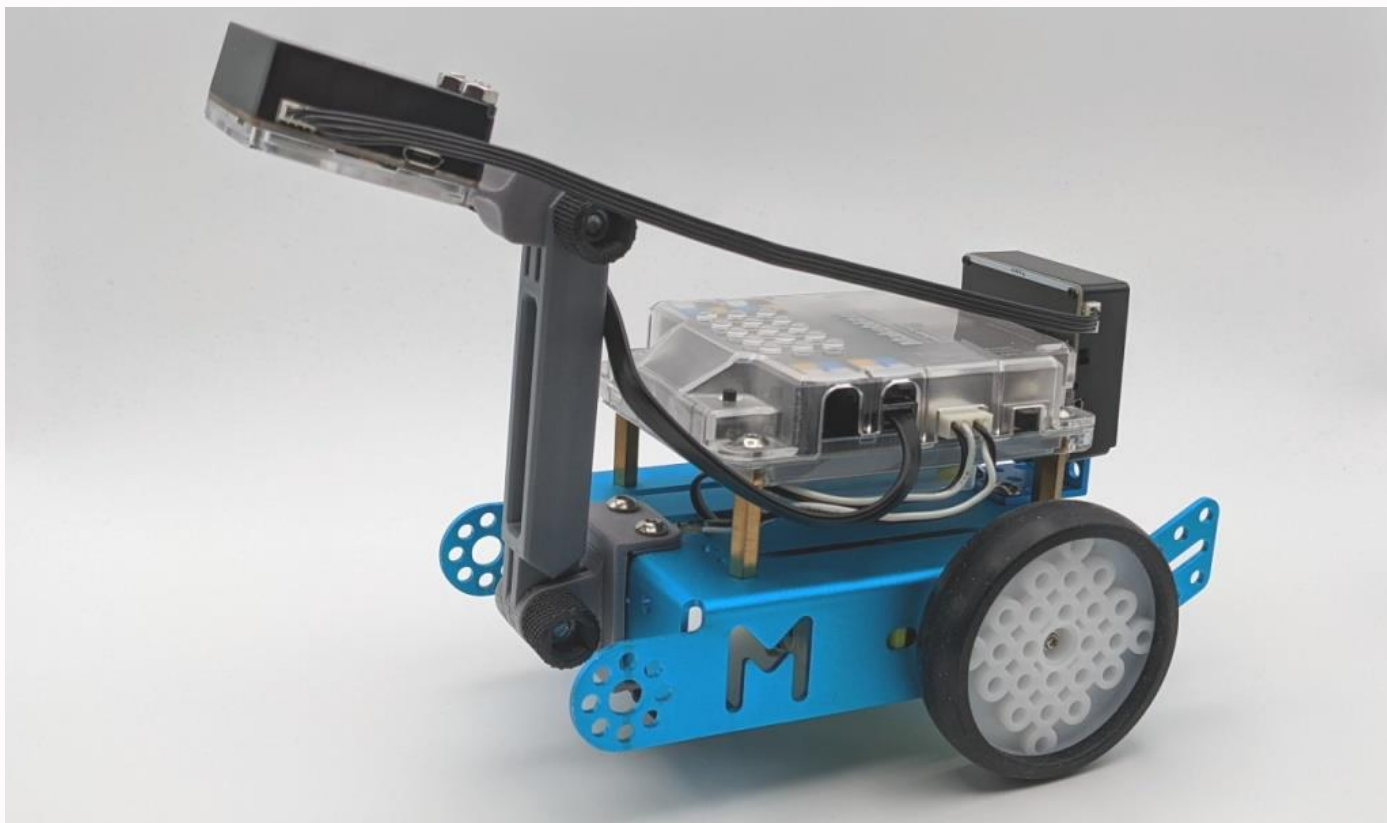
La caméra est également capable de détecter une ligne (pour que le robot puisse la suivre). D'après la documentation officielle, il faut que la caméra soit plus haute, mais également parallèle à la ligne.



Malheureusement, le grand mât qui apparaît sur le schéma n'est pas fourni dans la boîte et en plus la longueur du câble d'alimentation ne permettrait pas de placer aussi haut la caméra (ou alors, il faudrait déplacer la batterie).

Il faudra donc bricoler un support, mais si vous disposez d'une imprimante 3D, vous pouvez utiliser les mêmes fichiers que moi ([disponibles ici](#)).

Pour que cela fonctionne, il faut régler avec soin l'angle de la caméra, l'allumer avant le robot, attendre quelques instants pour que le robot détecte la ligne et appuyer ensuite sur le bouton (pour que le robot démarre).

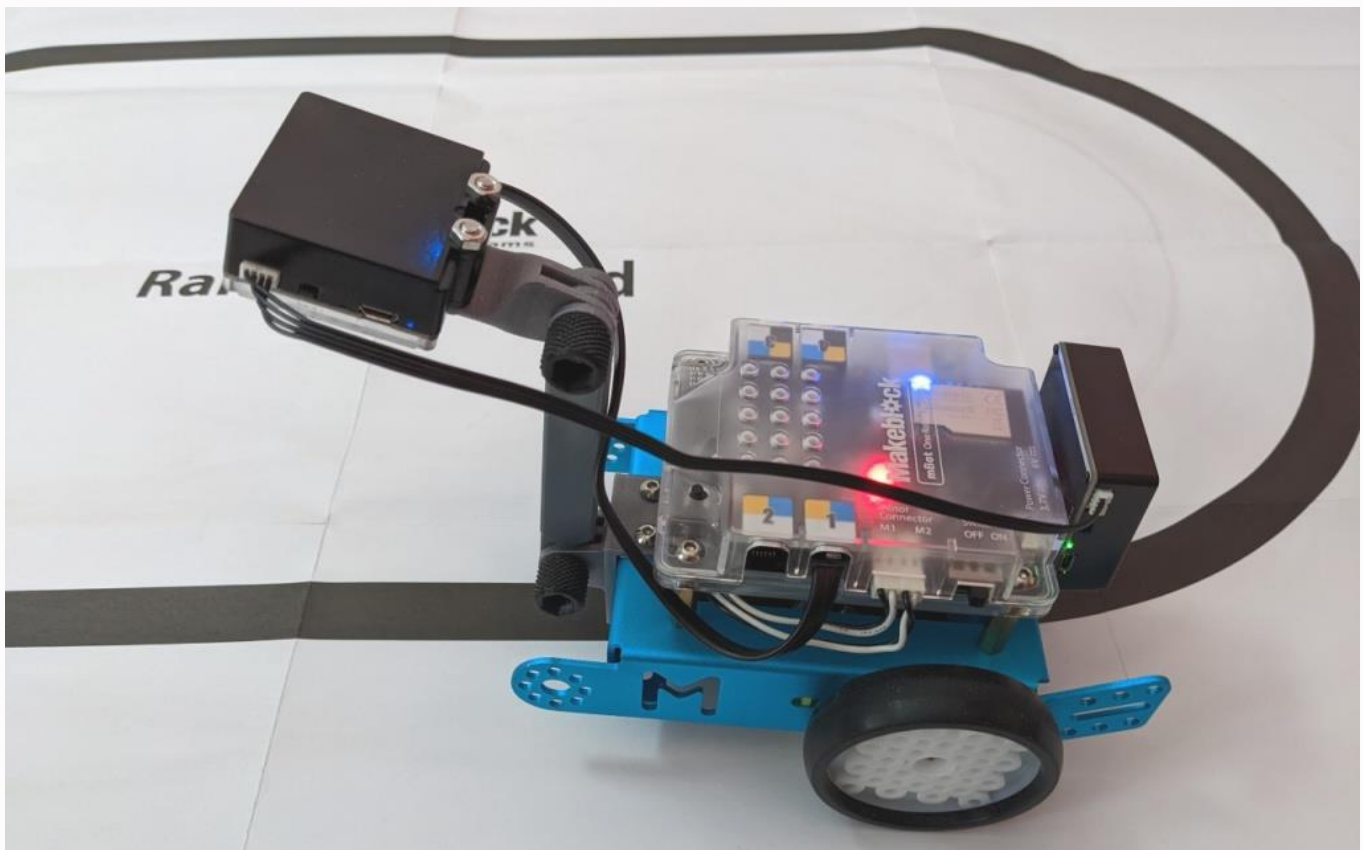


Vous pouvez utiliser cet exemple de code (inspiré par celui de MakeBlock) :

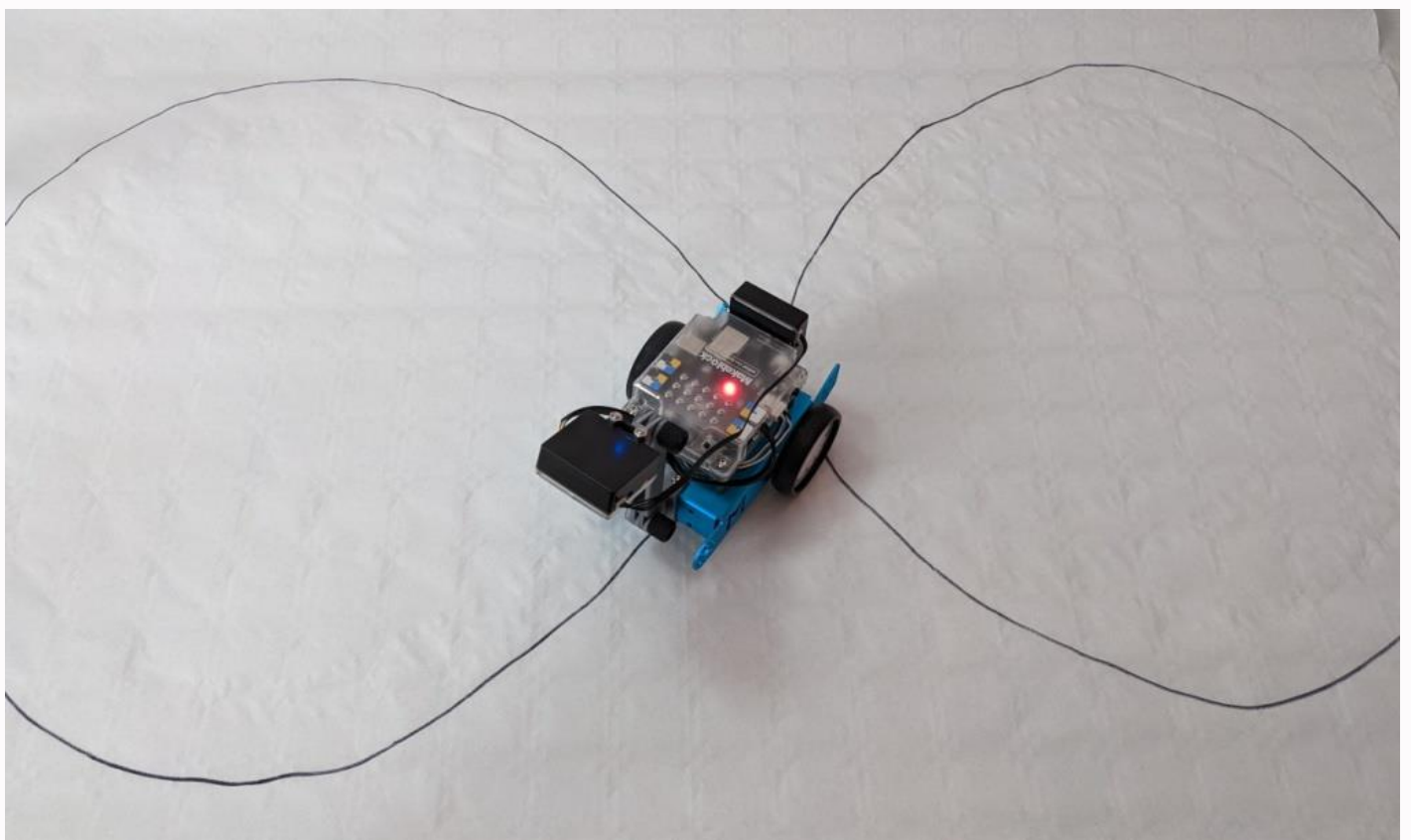
```

Lorsque le mBot(mcore) démarre
  Smart Camera port1 change en mode suivi de ligne/étiquette
  Smart Camera port1 définit le coefficient Kp pour la vitesse différentielle du moteur à 0.6
  Smart Camera port1 définit le mode de suivi de ligne à ligne foncée sur fond clair
  définir vitesse à 60
  attendre jusqu' à sur appui du bouton Carte pressé ?
  pour toujours
    définir correction à Smart Camera port1 calcule la vitesse différentielle du moteur (cible la section suivi de ligne)
    définir roueGauche à vitesse + correction
    définir roueDroite à vitesse - correction
    roue gauche tourne à roueGauche % de puissance, roue droite à roueDroite % de puissance
  
```

Sur une piste classique, la Smart Camera est un peu décevante puisqu'un simple capteur de suivi de ligne est bien plus performant (mais c'est normal, puisque les pistes sont destinées à ce type de capteur).

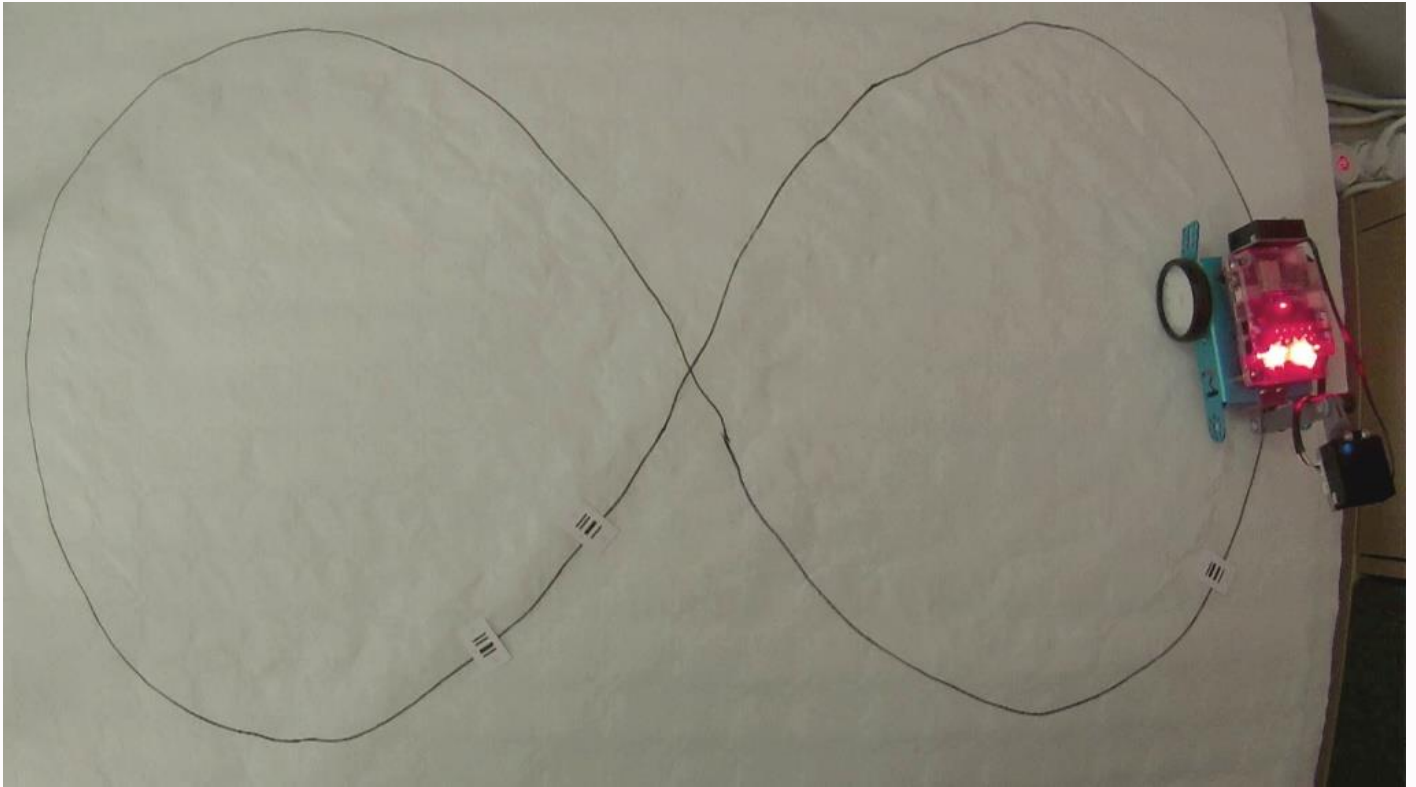


Cependant, la caméra est capable de détecter un simple trait sur une grande feuille. Cela permet de tester rapidement de nouveaux tracés et ouvre de nouvelles possibilités. Par exemple, vous pouvez fabriquer une piste géante en quelques minutes en déployant une nappe en papier sur le sol et en traçant rapidement une ligne au feutre.



Code barres

La caméra peut également reconnaître les étiquettes (codes barres). Elles sont fournies avec la Smart Camera, mais pouvez aussi les imprimer vous même (le lien est à la fin de l'article). Ensuite, il suffit simplement de les disposer sur la piste pour que la caméra puisse les lire. Attention, si l'étiquette est à l'envers, elle ne sera pas prise en compte.



Voilà par exemple comment changer la couleur des LED ou stopper le robot (avec les étiquettes) :

Lorsque le mBot(mcore) démarre

Smart Camera port1 change en mode suivi de ligne/étiquette

Smart Camera port1 définit le coefficient Kp pour la vitesse différentielle du moteur à 0.6

Smart Camera port1 définit le mode de suivi de ligne à ligne foncée sur fond clair

définir vitesse à 60

attendre jusqu'à sur appui du bouton Carte pressé ?

LED tout affiche la couleur

pour toujours

si Smart Camera port1 a détecté l'étiquette (5) arrêt alors

LED tout affiche la couleur

stopper le mouvement

sinon

définir correction à Smart Camera port1 calcule la vitesse différentielle du moteur (cible la section suivi de ligne)

définir roueGauche à vitesse + correction

définir roueDroite à vitesse - correction

roue gauche tourne à roueGauche % de puissance, roue droite à roueDroite % de puissance

si Smart Camera port1 a détecté l'étiquette (10) A alors

LED tout affiche la couleur

si Smart Camera port1 a détecté l'étiquette (11) B alors

LED tout affiche la couleur