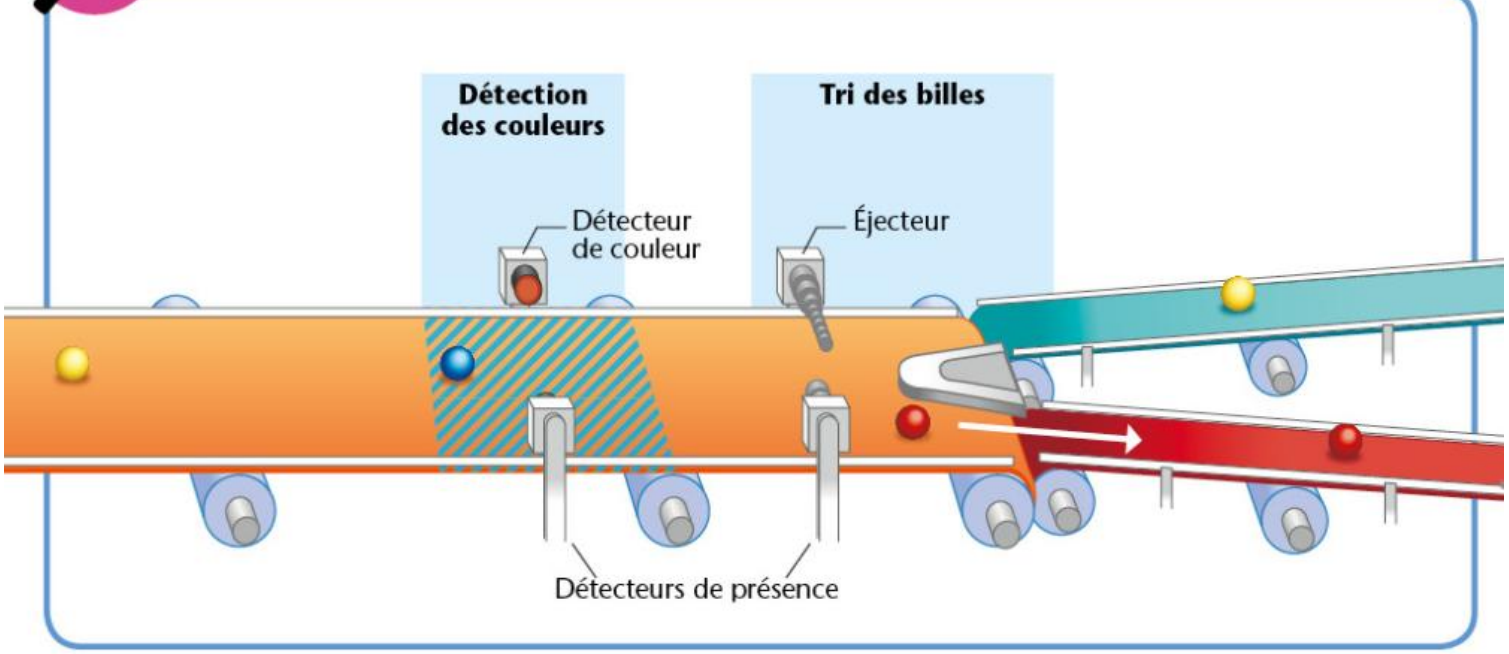
7

Programmation

- **CT 4.2 -** Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème simple.
- **CS 5.7 -** Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous problèmes afin de structurer un programme de commande.

COLLÈGE JULES FERRY

J'analyse des situations



1 Comment le microprocesseur qui gère le système peut-il déterminer la couleur de chaque bille ?

.....

.....

2 Comment le système peut-il être informé du passage d'une bille ?

.....

.....

3 Comment le système peut-il savoir comment éjecter chaque bille dans la bonne zone ?

.....

.....

1 Répéter plusieurs fois la même séquence d'instruction

❑ CT 4.2

Définitions

Pour réaliser plusieurs fois la même séquence d'instruction, on utilise les **boucles**. Il existe 3 catégories de boucles, ci-dessous illustrées avec le logiciel de programmation Blockly :
- celle qui répète des instructions tant qu'un critère n'est pas vérifié,
- celle qui répète des instructions jusqu'à ce qu'un critère soit vérifié,
- celle qui exécute des instructions en faisant varier une variable d'une valeur à une autre.

Les boucles infinies sont très souvent utilisées pour programmer les systèmes embarqués dont le rôle est d'effectuer la même tâche tant qu'ils sont allumés.

1 Reliez chaque proposition au bloc de programme qui lui correspond.

La boucle s'arrête quand une entrée du microcontrôleur change d'état.	
La boucle s'arrête quand la valeur est atteinte.	
La boucle ne s'arrête jamais.	
La boucle s'arrête après 5 passages.	

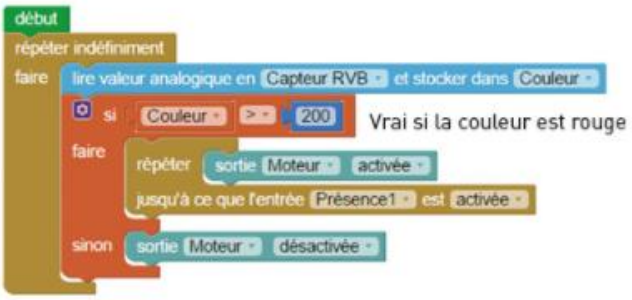
2 Que fait le programme ci-contre ?

.....

.....

.....

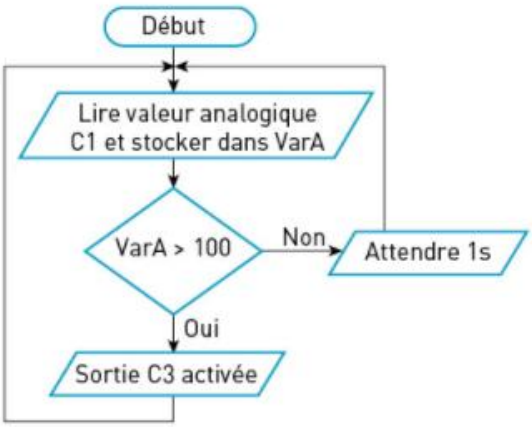
.....



Si un programme en Blockly permet de mettre en œuvre un algorithme, la mise au point ou la phase de réflexion peut se faire sur papier à l'aide d'un algorigramme.

3 Réalisez, sous Blockly, le programme correspondant à l'algorigramme ci-contre, prenez une carte PICAXE de type 18M2.

4 Modifiez ce programme pour que les instructions réalisées quand l'entrée « VarA » est plus petite que 100, soit :
– répéter 3 fois : sortie C2 activée, attendre 1 s, sortie C2 désactivée, attendre 1 s.



2 Réaliser une instruction sous condition et utiliser une variable

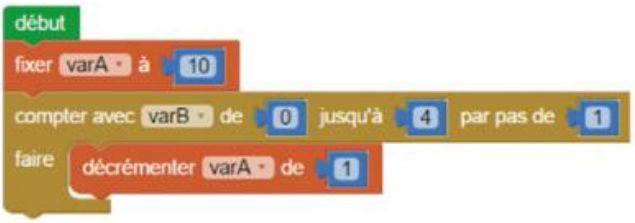
❑ CT 4.2

Définitions

Une autre structure de contrôle élémentaire est la **condition**. Si la condition est vraie alors les instructions sont réalisées, sinon on passe.
Dans un programme, des **variables** peuvent être utilisées pour stocker des informations comme l'état d'une porte (ouverte ou fermée).
Il est important d'initialiser les variables au début du programme pour imposer la valeur initiale.

1 Que vaut « varA » à la fin du programme ci-contre ?

- ☐ 4
- ☐ 5
- ☐ 6



2 Que vaut « varA » à la fin du programme ci-contre?

- ☐ 4
- ☐ 20
- ☐ 16

Le programme ci-contre est l'ébauche du programme permettant de gérer l'ouverture et la fermeture automatique d'un poulailler selon qu'il fait jour ou nuit.

3 Expliquez à quelle action correspond ce début de programme.

.....

.....

.....

4 Saisissez ce programme sur Blockly puis modifiez-le pour qu'il ouvre la porte si la luminosité est supérieure à 245, en créant la variable « Moteur sens2 » qui permet d'ouvrir la porte.

5 Que se passe-t-il lorsque la luminosité est inférieure à 10 et que la porte est déjà fermée ?

.....

.....

6 Pour éviter ce problème, ajoutez une variable « porte_ouverte » qui sera égale à 0 quand la porte sera fermée et à 1 quand la porte sera ouverte.
Modifiez le programme pour que le moteur ne force pas.

3 Décomposer un problème

❑ CT 5.7

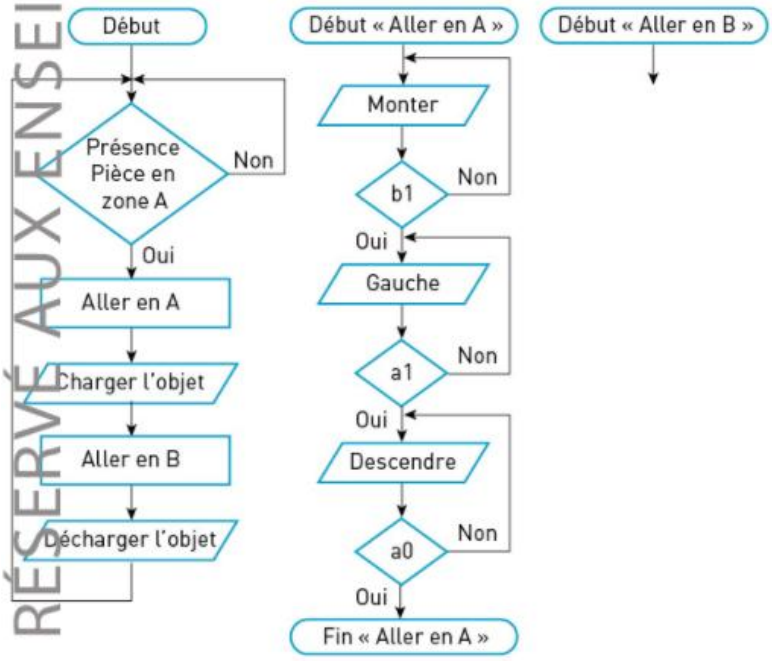
Définitions

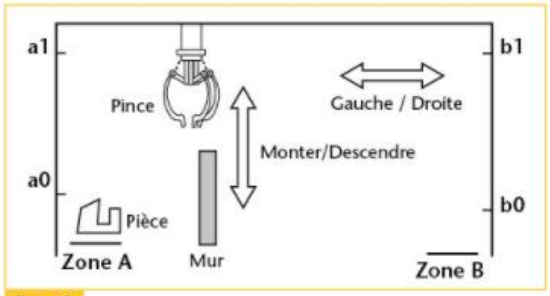
En informatique, il faut essayer de décomposer le programme à réaliser en **sous-fonctions** ou tâches élémentaires. En séparant les difficultés, la résolution du problème sera plus aisée, la lecture du programme principal sera plus simple, chaque tâche pourra être réutilisée plusieurs fois dans le programme.

Le programme principal ci-contre correspond à la gestion du système de la porte du poulailler de l'exercice précédent.

1 Proposez un nom à chacune de ses sous-fonctions.

On considère un système automatisé de déplacement de pièces (Doc. 2) d'une zone A vers une zone B à l'aide d'une pince. Des capteurs permettent de savoir si la pince est en zone A en bas (a0) ou en haut (a1), si la pince est en zone B en bas (b0) ou en haut (b1). Les commandes permettant de déplacer la pince sont « Monter », « Descendre », « Gauche » et « Droite ».





Doc. 2

L'algorigramme ci-dessus décrit le comportement attendu. Il est composé d'un algorigramme principal et de sous-fonctions.

2 En observant l'algorigramme, décrivez le comportement du système.
On supposera que la pince se situe en zone B au début du programme et qu'une pièce est présente en zone A.

3 Complétez l'algorigramme ci-dessus correspondant à la sous-fonction « Aller en B ».

La situation

Comment programmer une machine qui trie des objets par couleur ?

On souhaite réaliser le programme gérant le fonctionnement du trieur de bille décrit en première page. On suppose que vous avez à disposition :
- un capteur RVB (= capteur de couleur)
- deux détecteurs de présence
- deux éjecteurs

Doc. 3 Éjecteur

Comme sur le schéma p. 41, le système est composé d'un tapis roulant qui amène des billes de couleurs régulièrement espacées. Les billes passent devant un capteur RVB qui permet de déterminer si la bille est rouge ou non, puis elles avancent devant les éjecteurs associés chacun à un capteur de présence. Le système doit trier les billes rouges en les éjectant dans la bonne zone.

L'espace entre deux billes est suffisamment grand pour permettre l'éjection de la première avant que la seconde ne passe devant le capteur RVB.

Pour réaliser le programme, il sera nécessaire de le décomposer en plusieurs sous-fonctions. Il pourrait être utile de créer des algorigrammes pour bien décrire le comportement voulu. Chaque sous-fonction sera réalisée séparément puis l'ensemble du travail sera mis en commun afin de simuler le comportement global du système.