

CYCLE 4
Séance S1

Comment suivre un animal à distance ?

NIVEAU
TROISIEME

➤ **Mise en situation :**

Nous avons vu qu'un piège photographique utilisait une caméra (élément n°7).



➤ **Description de la situation :**

Le piège photographique capture une image (ou une vidéo) qui est ensuite traitée par une intelligence artificielle pour identifier automatiquement l'animal. Grâce à sa connectivité 4G, l'appareil transmet instantanément les fichiers et les alertes sur le smartphone de l'utilisateur.

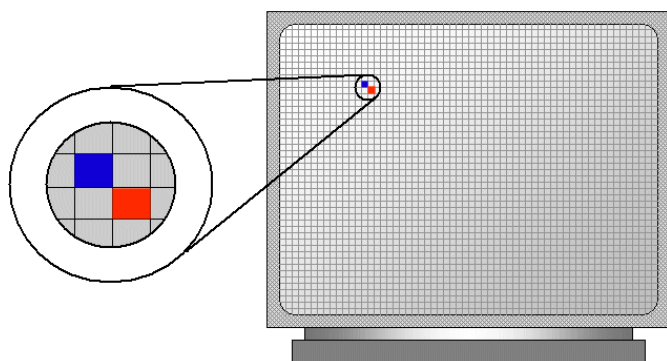
➤ **Problématique :**

Comment programmer la carte microcontrôleur pour reconnaître automatiquement un ours ?

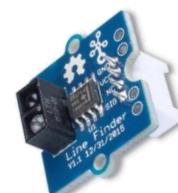
ACTIVITE 1

➤ **Résolution de problème :**

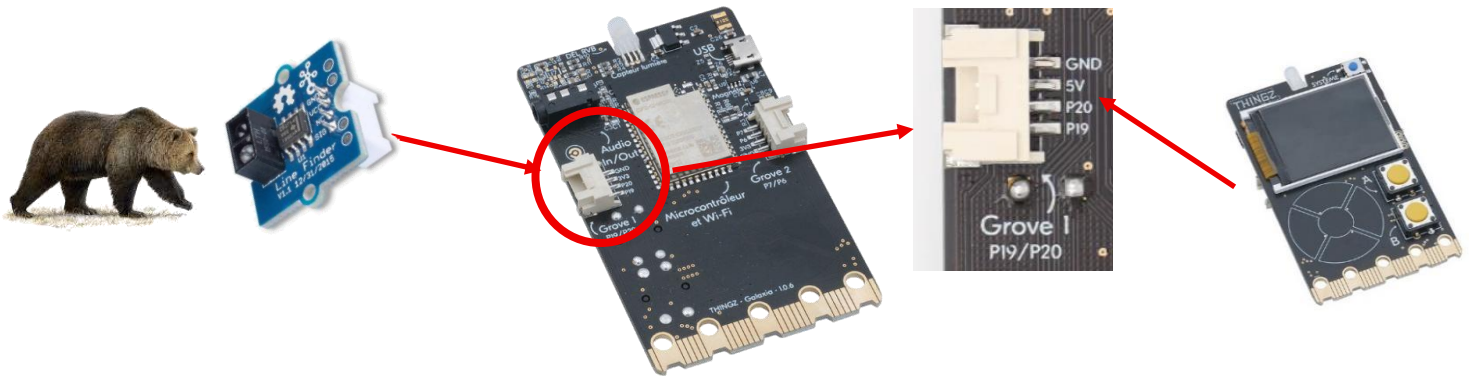
Une image classique se compose de plusieurs pixels (petits points lumineux qui composent l'image).



Pour simplifier l'étude, le système de prise de vue sera réduit à un capteur optique d'un seul pixel, qui jouera le rôle de la caméra IA.



Ce capteur optique sera connecté sur une carte microcontrôleur sur le connecteur P19.

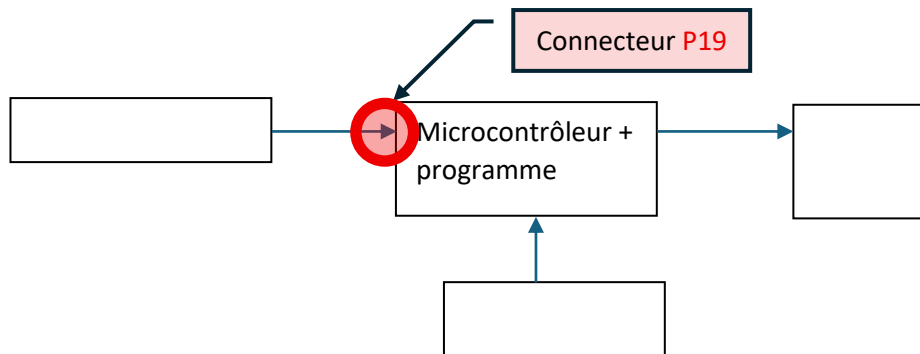


L'objectif est d'afficher sur l'écran de la carte Galaxia le texte « ours » toutes les secondes, dès qu'un ours est détecté et reconnu par le capteur optique. On affichera aussi le temps (en secondes) écoulé au moment où la photo a été prise.

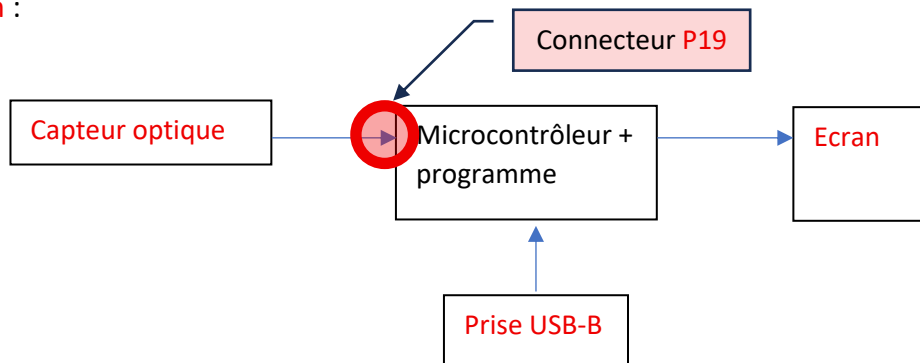
Comment fonctionne un capteur optique ? Quelles informations renvoi-t-il ?

Connecter le capteur optique sur le port Grove 1 (gauche). Le numéro de connecteur associé sera le P19.

- **Q1. Compléter** le diagramme de bloc interne avec les éléments suivants (*Ecran/ prise USB / capteur optique / P19*) :

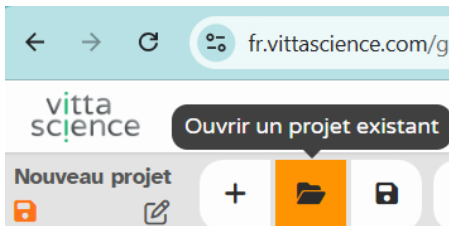


Correction :



- **Q2. Tester** le programme fourni « *PROG_forme_signal_capt_optiqueP19_S1* » dans la carte Galaxia :

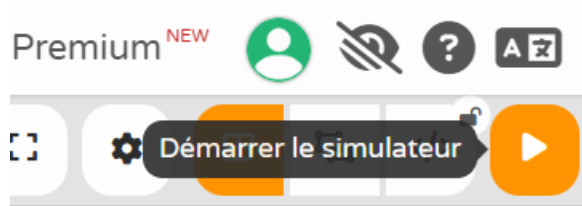
- Se connecter à l'adresse : <https://fr.vittascience.com/galaxia/>



- Ouvrir le programme ressource

PROG_forme_signal_capt_optiqueP19_S1

- Démarrer le simulateur



- Faire varier la position du capteur sur la ligne noire virtuelle
- Et observer le résultat.



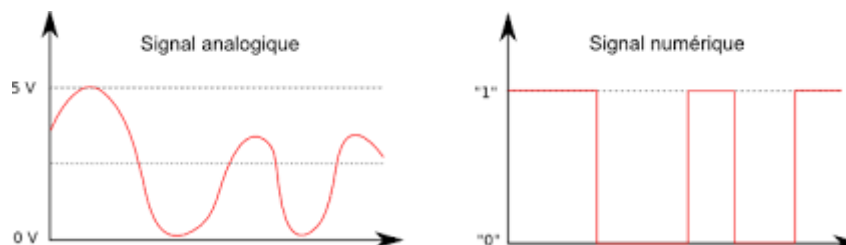
Combien de valeurs différentes issues du capteur optique s'affichent sur l'écran ?

Seules deux valeurs s'affichent sur l'écran. Soit 0 soit 1



Est-ce que le signal issu du capteur optique est un signal analogique ou numérique ?

C'est un signal numérique logique, car il n'y a que deux valeurs différentes : 0 et 1.



Quelle est la valeur du capteur optique qui permet de dire qu'un ours a été détecté quand il passe devant le capteur ?

Un ours est détecté si la valeur du capteur optique = 1.

➤ **Bilan de mes recherches :**

Ce capteur optique ne renvoie que deux valeurs : 0 ou 1. Si le capteur, qui simule le fonctionnement de la caméra IA, renvoie la valeur 1, cela signifie qu'un ours a été détecté.

ACTIVITE 2

- **Proposer** un algorithme simple du fonctionnement en langage naturel.

On lit la donnée du capteur optique connecté sur P19.

Si cette donnée est égale à 1 alors on affiche « ours » et le moment de la prise de vue pendant 1s.

Sinon on recommence.

- Compléter l'algorithme suivant avec les éléments ci-dessous :

Attendre 1s

Afficher « ours détecté »

capt_optique_IA ← état du capteur optique IA lu

capt_optique_IA = 1

DEBUT

TANT QUE Vrai (= répéter toujours)

.....

SI ALORS

.....

Afficher le temps (moment où a été prise la photo)

.....

FIN SI

FIN TANT QUE

FIN

Correction

DEBUT

TANT QUE Vrai (= répéter toujours)

capt_optique_IA ← état du capteur optique IA lu

SI capt_optique_IA = 1 ALORS

Afficher « ours détecté »

Afficher le temps (moment où a été prise la photo)

Attendre 1s

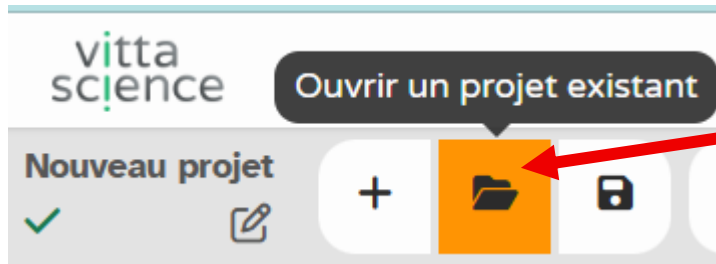
FIN SI

FIN TANT QUE

FIN

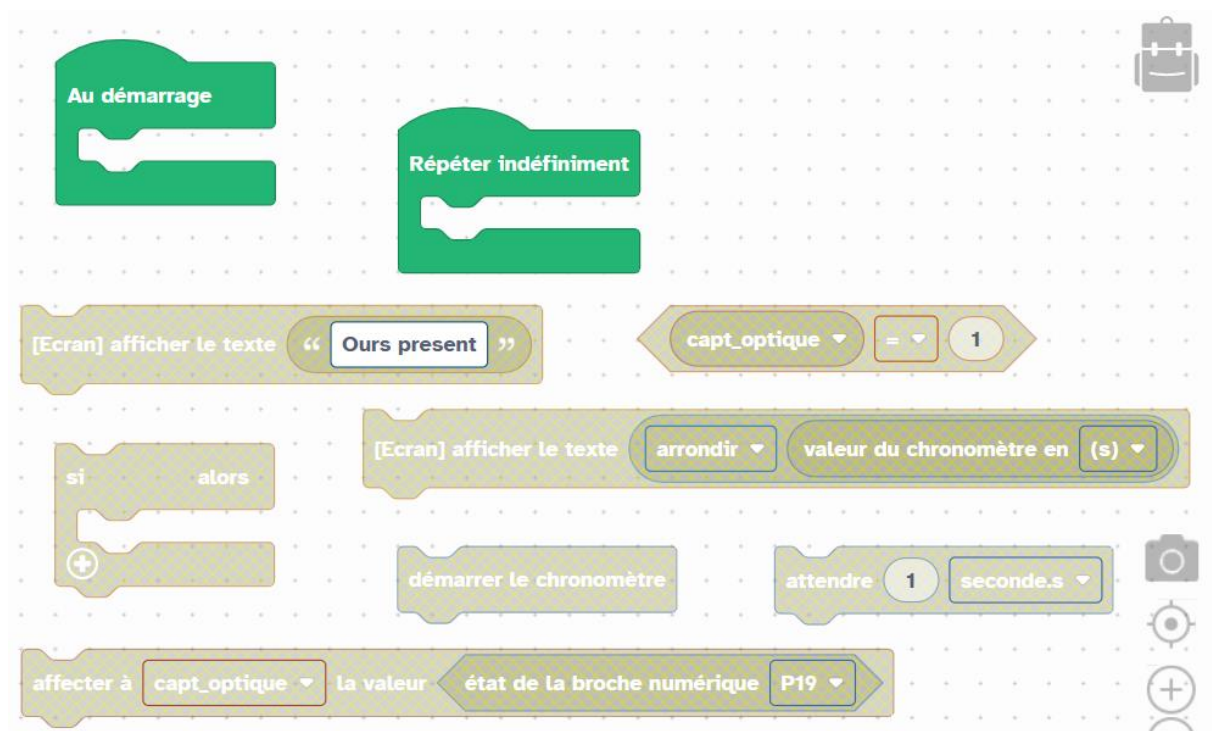
- **Modifier** le programme fourni « PROG_eleve_S1.py » :

Se connecter à l'adresse : <https://fr.vittascience.com/galaxia/>

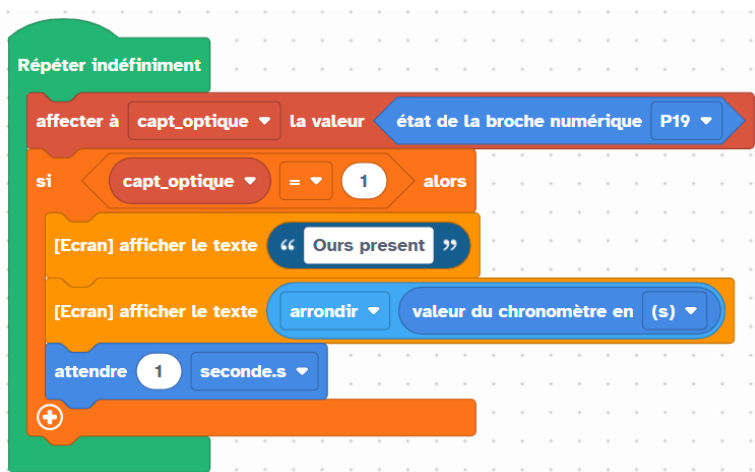


Ouvrir le fichier :
PROG_eleve_S1.py

Organiser les blocs pour respecter l'algorithme proposé.



Correction du programme :



➤ **Bilan de mes recherches :**

On peut créer un programme utilisant un capteur afin de réaliser une détection simplifiée d'un animal à partir d'un algorithme.

➤ **Bilan commun :**

Avant de programmer, il est nécessaire de connaître les connexions et les signaux échangés. Pour cela, on peut utiliser un **diagramme de bloc interne**

Pour concevoir le programme, on rédige dans un premier temps un **algorithme en langage naturel**. On utilise ensuite une **variable** afin de mémoriser l'état du capteur. Enfin, une **instruction conditionnelle** (un test) permet d'effectuer un choix en fonction de cette valeur binaire, ici 0 ou 1.