

Voiture autonome

Compétences visées

Un des objectifs de l'enseignement de SNT est de développer et de coder des scripts PYTHON afin d'apporter une réponse à une problématique précise. A travers le thème " **informatique embarquée et objets connectés**", nous pouvons notamment travailler les compétences suivantes dans l'activité proposée :

- Coder des scripts simples d'acquisition de données.
- Gérer des entrées/sorties à travers les ports utilisés par le système.
- Écrire et développer des algorithmes pour résoudre une problématique.
- Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs.

Situation déclenchante



Le développement des logiciels embarqués est délicat, car il pose souvent des questions de temps-réel, c'est-à-dire de respect de temps de réponse imposé. Ceci conduit à des méthodes de programmation spécifiques.

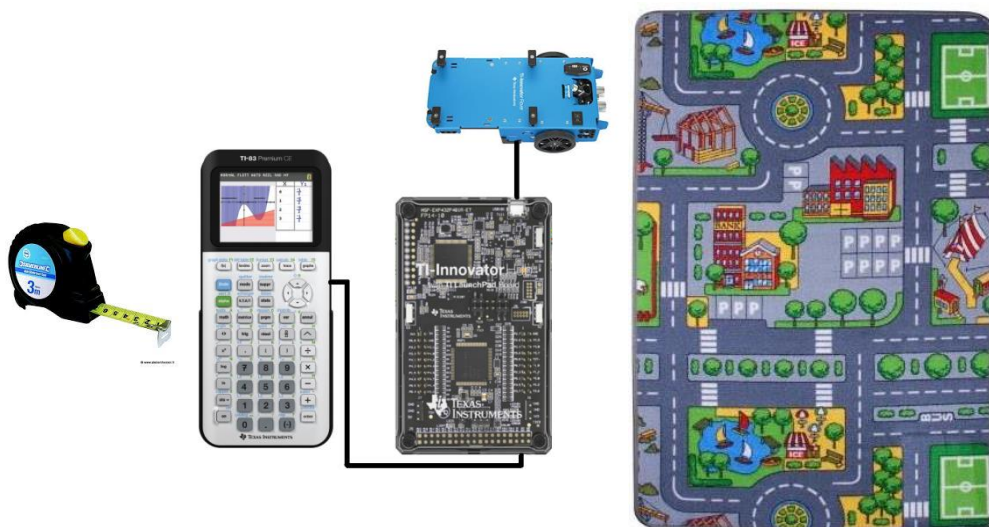
L'impact de l'informatisation des objets devient considérable, surtout depuis que leurs interfaces s'unifient. Le but est de fabriquer des machines d'utilisation facile permettant des fonctionnalités améliorées, voire complètement nouvelles comme la voiture autonome. Celle-ci utilise à la fois des techniques de systèmes embarqués pour son fonctionnement et sa navigation et de l'intelligence artificielle pour l'analyse en temps-réel de l'environnement à l'aide de capteurs variés (caméras, radars, ...etc.).

Problématique

Comment concevoir une solution technologique pour programmer le déplacement prédéfini d'un véhicule avec arrêt temporaire si un obstacle du type piéton est rencontré ?

Fiche méthode

Matériel nécessaire



- Calculatrice TI-83 Premium CE
- Câble (calculatrice/Hub)
- TI-Innovator Hub
- Rover
- Tapis de sol
- Mètre

Déroulement possible du projet

Travail de groupe possible. Chaque groupe disposera du matériel ci-dessus et devra concevoir une réponse à la problématique sous forme d'une démonstration munie d'une documentation qui explique les scripts pilotant la rover. A la fin du projet, les groupes pourront voter pour élire la production qui répond le mieux à la problématique.

La production devra répondre aux critères suivants :

Critère 1 : Déplacer le rover en suivant les routes au sol pour aller du point de départ au point d'arrivée (exemple ci-contre).

Critère 2 : S'arrêter si un obstacle du type piéton est rencontré sur la route.

Critère 3 : Allumer la diode RGB en rouge lors du freinage si un obstacle est rencontré.

Critère 4 : Après le passage des piétons, le rover doit finir le trajet jusqu'au point d'arrivée.

Point de départ



Point d'arrivée

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Fiche méthode

Proposition de résolution

Les élèves devront dans un premier temps mesurer les déplacements à effectuer à l'aide du mètre puis faire les conversions nécessaires. L'unité de référence pour les déplacements du rover est le décimètre.

Bibliothèques nécessaires (voir remarques ci-après).

Il faut saisir l'ensemble des déplacements du rover. L'unité d'angle est le degré et l'unité de mesure est le décimètre. L'ensemble des déplacements est envoyé dans une liste. Les déplacements sont ensuite traités dans l'ordre par le rover.

L'instruction `a=rv.ranger_measurement()` permet de lire la distance devant le rover et de la stocker dans `a`. Cette distance est en mètres.

L'instruction `rv.color_rgb(255,0,0)` permet d'allumer la diode RGB du rover avec la couleur rouge. L'intensité est ici de 255 (et peut varier entre 0 et 255).

La commande `rv.stop()` est exécutée immédiatement et n'est pas intégrée à la liste ordonnée des déplacements.

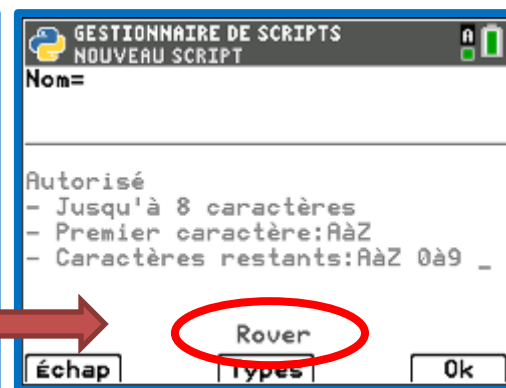
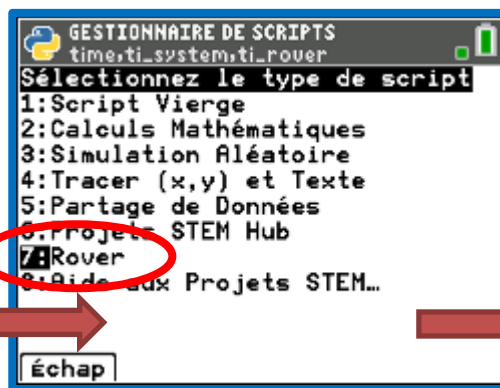
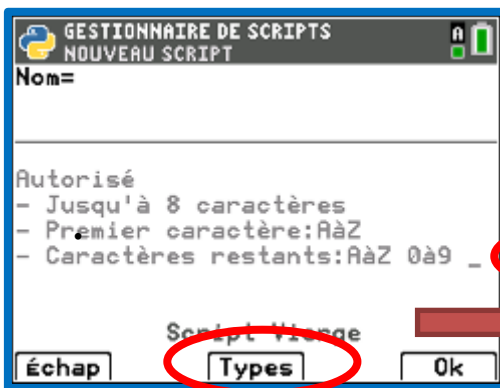
L'instruction `rv.resume()` permet de reprendre la liste des déplacements qui avait été interrompue par la commande STOP.

```
ÉDITEUR : VOITUR
LIGNE DU SCRIPT 0001
Rover
from time import *
from ti_system import *
import ti_rover as rv

def voiture():
    ..disp_clr()
    ..disp_at(8,"appuyer sur annul p
our arreter","left")
    ..rv.forward(6)
    ..rv.left(90)
    ..
    ..rv.forward(1)
    ..while not escape():
    ....a=rv.ranger_measurement()
    ....if a<0.2:
    .....rv.color_rgb(255,0,0)
    .....rv.stop()
    ....else:
    .....rv.color_off()
    .....rv.resume()
```

Utilisation des bibliothèques

- Lors de la création du script, sélectionner la rubrique **Types**, puis **Rover**. Cela permettra de charger automatiquement les trois bibliothèques suivantes : `time`, `ti_system` et `ti_rover`.

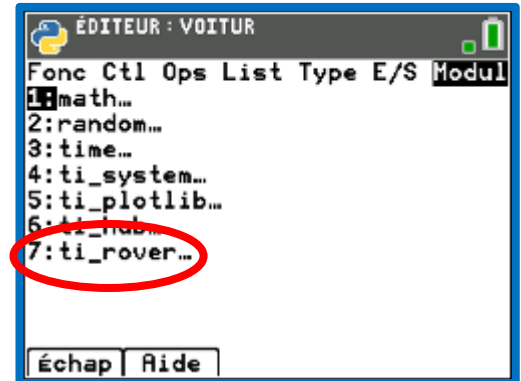


Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Fiche méthode

- Pour utiliser la bibliothèque `ti_rover` importée, appuyer sur la touche `f(x)`, sélectionner la rubrique **Modul** puis sélectionner **7 : ti_rover...**
- Au fur et à mesure que les bibliothèques sont importées, elles s'ajoutent aux bibliothèques déjà présentes (`math`, `random`, ...etc.) et permettent d'accéder aux instructions qu'elles contiennent.



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

