

Intégration d'Arduino à Unity3D

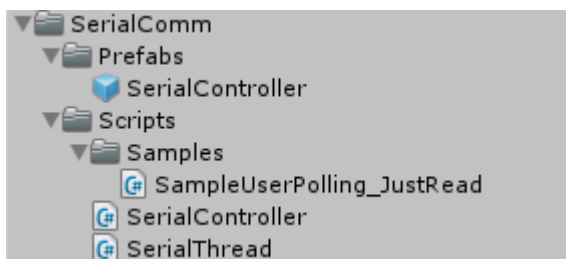
I. Configurations préliminaires

Partie Unity3D

Pour des raisons de compatibilité, on doit passer l'API du projet en .NET 2.0 :

Edit >> Project settings >> Player >> Other Settings >> API compatibility level
Passer en .NET 2.0

On insère les assets nécessaires à la communication entre Unity et Arduino comme suit :



(SerialComm est à la racine du projet)

On introduit un prefab *SerialController* auquel on associe le script *SerialController*. Dans l'inspecteur du script, il faut vérifier que le *Port Name* correspond à celui sur lequel la carte Arduino est branchée.

Ajouter un GameObject (empty), qu'on peut renommer *SampleUserPolling* et lui associer le script *SampleUserPolling_JustRead*. Dans l'inspecteur, associer le script *SerialController* au paramètre *SerialController* du script précédent.

Partie Arduino

Pour la partie Arduino, il suffit juste de téléverser le code fourni sur une carte Arduino. Une fois téléversé, il suffit de relier la carte Arduino via le port USB passé en paramètre public du script Unity *SerialController*. Une fois build en exécutable, il semble que le port USB sur lequel la carte est branchée importe peu mais il n'y a aucune garantie à l'exception du port configuré.

Le code a été écrit pour une Arduino UNO et ce modèle est donc recommandé.

II. Paramètres configurables

Nos scripts permettent avant tout la configuration de deux paramètres : la vitesse et l'intervalle de temps avant accélération.

Pour ce qui est de la vitesse, l'inspecteur du GameObject *SampleUserPolling* permet un changement de celle-ci.

Pour l'intervalle de temps avant accélération, il faut éditer le script *SampleUserPolling_JustRead* aux lignes 64 et 67. Les intervalles correspondent aux conditions des boucles *if*, réglés de base sur 3 et 6 secondes. Ces mêmes boucles *if* permettent de configurer l'augmentation de la vitesse dans le temps, initialement réglée sur 2 et 5 fois la vitesse de base.

Il est également possible d'éditer le script Arduino si vous disposez de l'IDE associée.

III. Fonctionnement en mode Run

L'application est écrite pour gérer un seul mouvement à la fois, il est donc recommandé d'éviter d'appuyer sur deux boutons à la fois.

Les mouvements gauche, droite, avant, arrière se font selon la direction du regard mais dans un plan horizontal tandis que les mouvements haut et bas se font toujours selon la direction verticale absolue de la scène.

Le script Arduino actuel associe ses entrées aux sorties suivantes :

Bouton poussoir connecté à l'entrée 8 : « bas »

Bouton poussoir connecté à l'entrée 9 : « haut »

Bouton poussoir connecté à l'entrée 10 : « gauche »

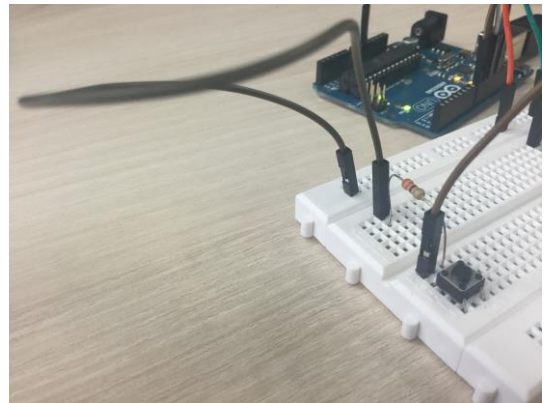
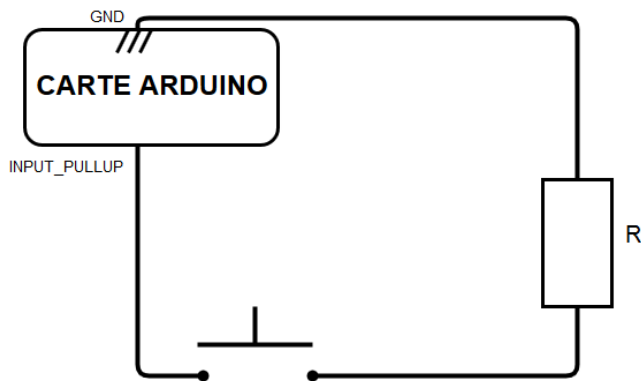
Bouton poussoir connecté à l'entrée 11 : « droite »

Bouton poussoir connecté à l'entrée 12 : « arrière »

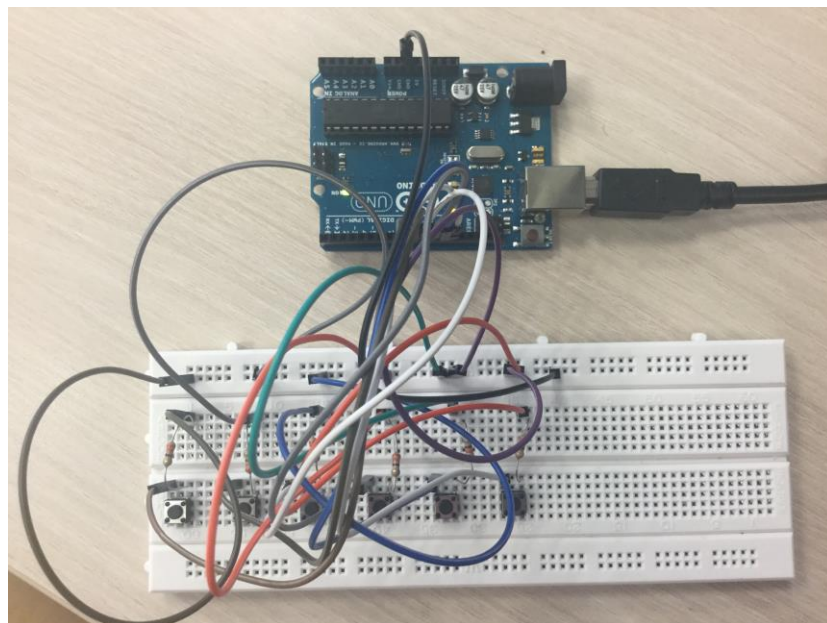
Bouton poussoir connecté à l'entrée 13 : « devant »

iv. Hardware

Pour donner une idée de ce à quoi ressemble la partie électronique voici quelques images :



Montage pour un bouton



ensemble du montage