

Découvrir Scratch



Ce manuel explique l'environnement de programmation Studuino et comment l'utiliser. L'environnement de programmation Studuino étant en développement, ce manuel peut être amené à être modifié ou révisé.

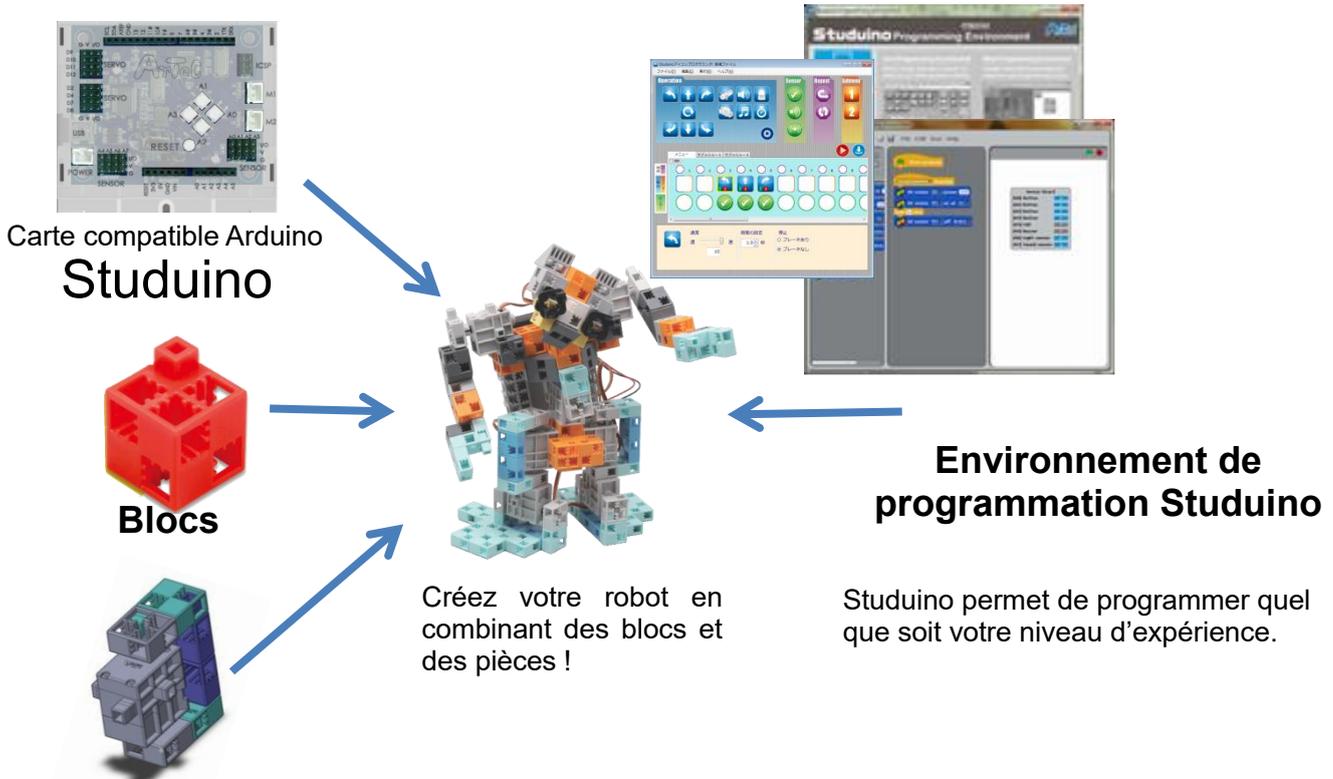
Sommaire

1. Environnements de programmation Studuino	4
1.1. Présentation et caractéristiques.....	4
1.2. Exigences de système	4
1.3. À propos de Studuino.....	5
2. Introduction	6
2.1. Installation des pilotes de périphérique USB	6
2.2. Installer et désinstaller l'environnement de programmation Studuino.....	6
3. Écran de démarrage	7
4. Scratch – Programmation par blocs	8
4.1. Présentation et caractéristiques.....	8
4.2. La palette du groupe de commandes et la palette de blocs	9
4.2.1. À propos de Scratch - la programmation par blocs.....	9
4.3. Champ de script	19
4.4. Le champ de condition	20
4.5. Le menu principal	20
4.6. Menus contextuels	30
5. Dépannage	31
5.1. Dépannage général.....	31
5.2. Scratch - Environnement de programmation par blocs	31
6. Animation	33
6.1. Présentation et caractéristiques.....	33
6.2. Programmer	34
6.2.1. Ajouter une scène.....	35
6.2.2. Ajouter des objets.....	37
6.2.3. Ajouter un costume.....	38
6.2.4. Programmer les objets	40
6.2.5. Utiliser les messages pour lier votre Studuino aux objets	43
6.2.6. Lier votre Studuino aux variables	46
6.2.7. Utiliser les meshes pour lier votre Studuino aux objets	50
6.2.8. Utiliser les meshes pour lier votre Studuino aux objets	53
6.3. Blocs de l'objet Studuino.....	56
6.4. En savoir plus sur les blocs	57
<input type="checkbox"/> Blocs de la palette Mouvement.....	57
<input type="checkbox"/> Blocs de la palette Contrôle	57
<input type="checkbox"/> Blocs de la palette Capteurs.....	58
<input type="checkbox"/> Blocs de la palette Opérateurs	58
<input type="checkbox"/> Blocs de la palette Variables	59
6.5. Le menu principal.....	59
<input type="checkbox"/> Le menu Fichier	59
● Nouveau.....	59
● Ouvrir... ..	59
● Enregistrer	60
● Enregistrer sous... ..	60
● Importer un projet.....	60
● Exporter un objet.....	61
● Options de paramétrage de la calibration.....	62
● Quitter.....	62
● Le menu Édition	63
● Restaurer.....	63
● Démarrer/arrêter le pas-à-pas	63
● Définir le mode pas-à-pas... ..	63
● Calibration du moteur... ..	63

● Paramètres des ports	63
● Héberger le mesh	63
● Rejoindre le mesh	63
● Montrer l'adresse IP	63
● Arrêter l'hébergement du mesh	64
● Montrer/Cacher les blocs Studuino	64
□ Le menu Exécuter	65
● Connecter/Déconnecter	65
□ Menu Aide	65
● FAQ	65
6.6. Menus contextuels	65
7. Expériences.....	66
7.1. Aperçu et fonctionnalités	66
7.2. Programmation	67
7.2.1. Paramétrage de Studuino.....	67
7.2.2. Programmation.....	68
7.2.3. Observer le programme en mode test.....	68
7.2.4. Transférer un programme	70
7.3. En savoir plus sur les blocs	72
7.4. Le menu principal.....	73
□ Menu Éditer.....	73
● Paramètres des ports... ..	73
7.5. Le menu contextuel	73
7.6. Pour les tablettes Windows	74
7.7. Studuino 2.0 (version complète).....	75
● Pièces optionnelles (version complète de Studuino, partie Environnement de programmation par icônes seulement).....	75

1. Environnements de programmation Studuino

1.1. Présentation et caractéristiques



Pièces

(moteurs, LED, avertisseurs sonores et capteurs)

Un kit robotique est composé d'une carte programmable Studuino, de blocs, de pièces et de l'environnement de programmation Studuino (Scratch et programmation simplifiée par icônes). Mélangez et combinez ces différents éléments pour créer vos propres robots ! Les programmes que vous créez dans l'environnement de programmation Studuino vous permettent de contrôler la façon dont votre robot se déplace et agit.

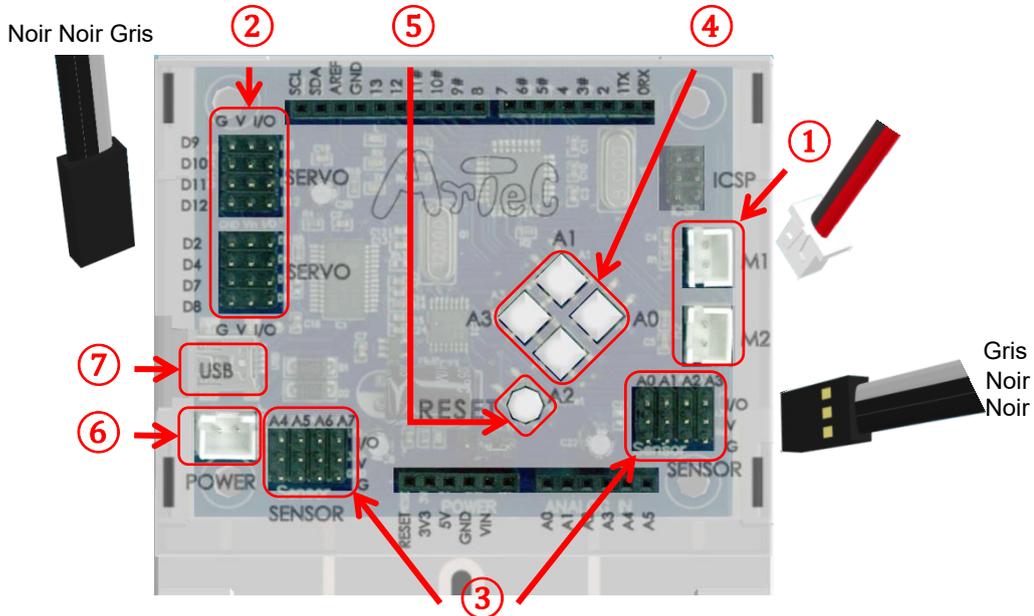
L'environnement de programmation Studuino est conçu de telle sorte que même les débutants puissent facilement créer des programmes sophistiqués. Il suffit de faire glisser et de relâcher les icônes ou les blocs qui contrôlent votre robot dans le champ réservé au programme. Ces programmes sont compatibles avec Arduino. Les programmes créés dans l'environnement Studuino peuvent, en effet, être convertis en langage Arduino que vous pouvez utiliser pour concevoir des programmes encore plus avancés.

1.2. Exigences de système

Élément	Spécificités
Système d'exploitation	Windows XP (SP2 ou version ultérieure) / Vista / 7 / 8 / 8.1, Mac OS X, Raspbian
Processeur (CPU)	Pentium 4, 2 GHz ou version ultérieure (ou équivalent) recommandé
Mémoire	256 MB ou plus
USB	USB 2.0 port
Logiciel	Microsoft .NET Framework 4.0 requis (installé automatiquement avec le logiciel Studuino)

1.3. À propos de Studuino

Cette partie traite des connecteurs Studuino utilisés lors de la construction d'un robot.



① **Connecteurs de moteur à courant continu**

Ces connecteurs sont destinés aux moteurs à courant continu. Jusqu'à deux moteurs à courant continu peuvent être connectés. Les noms de connecteurs sont M1 et M2.

② **Connecteurs des servomoteurs (connecteur numérique à plusieurs broches).**

Ces connecteurs sont destinés aux servomoteurs. Jusqu'à huit servomoteurs peuvent être connectés. Les noms de ces connecteurs sont D2, D4, D7, D8, D9, D10, D11 et D12. Lorsqu'un servomoteur est connecté, le fil de signal (fil gris) doit être du côté intérieur, comme indiqué ci-dessus.

③ **Capteurs/LED/avertisseurs sonores (connecteur analogique à plusieurs broches).**

Ces connecteurs à 12 broches sont destinés aux capteurs, LED ou avertisseurs sonores. Chaque connecteur a 3 broches. Les noms des connecteurs vont de A0 à A7. Tous les capteurs (hormis les capteurs tactiles ou les accéléromètres) peuvent être connectés aux connecteurs A0-A7. Les capteurs tactiles, LED et avertisseurs sonores peuvent être connectés à A0-A5. Les accéléromètres ne peuvent être connectés qu'aux connecteurs A4 et A5. Lorsqu'un(e) capteur/LED/avertisseur sonore est connecté, le fil de signal (fil gris) doit être du côté intérieur.

④ **Boutons-poussoirs**

Les boutons-poussoirs sont mappés sur A0-A3. Les connecteurs de capteur/LED/avertisseur sonore sur A0-A3 ne peuvent pas être utilisés avec un bouton-poussoir.

⑤ **Bouton de réinitialisation (Reset)**

Ce bouton réinitialise la carte. Appuyez sur ce bouton pour redémarrer Studuino lorsqu'il est initialisé mais qu'il ne fonctionne pas correctement ou lorsque le PC ne le reconnaît pas.

⑥ **Connecteur d'alimentation électrique**

Se connecte à une source d'alimentation. Les capteurs/LED/avertisseurs sonores/interrupteurs peuvent utiliser la tension fournie par l'USB, mais les moteurs à courant continu et les servomoteurs nécessitent l'alimentation de la prise d'alimentation.

Connectez un boîtier de piles à la prise d'alimentation lorsque l'USB est déconnectée.

⑦ **Prise de communication (mini-B USB).**

Se connecte à un câble pour communiquer avec le PC. Le câble mini-B USB inclus peut être remplacé par un câble disponible dans le commerce.

Les connecteurs ou interrupteurs suivants ne peuvent pas être utilisés en même temps :

- **Connecteur de moteur à courant continu M1 et connecteur de servomoteur D2, D4**
- **Connecteur de moteur à courant continu M2 et connecteur de servomoteur D7, D8**
- **Interrupteur à bouton-poussoir A0 et connecteur de capteur/LED/avertisseur sonore A0**
- **Interrupteur à bouton-poussoir A1 et connecteur de capteur/LED/avertisseur sonore A1**
- **Interrupteur à bouton-poussoir A2 et connecteur de capteur/LED/avertisseur sonore A2**
- **Interrupteur à bouton-poussoir A3 et connecteur de capteur/LED/avertisseur sonore A3**

Par exemple, vous ne pouvez pas utiliser un moteur à courant continu connecté sur M1 alors qu'un servomoteur est connecté sur D2 ou D4. Il se peut qu'ils ne fonctionnent pas correctement en raison d'interférences. De même, utiliser un servomoteur connecté sur D2 ou D4 alors qu'un moteur à courant continu est connecté sur M1 peut provoquer un dysfonctionnement du moteur à courant continu.

2. Introduction

Vous devrez installer les pilotes de périphérique USB et le logiciel de l'environnement de programmation pour utiliser l'environnement de programmation Studuino.

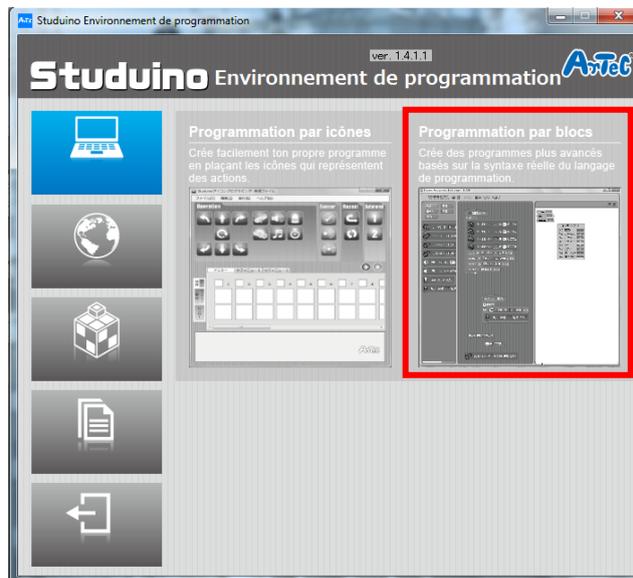
2.1. Installation des pilotes de périphérique USB

Vous devrez installer les pilotes de périphériques avant de programmer votre Studuino avec votre PC. Lisez le guide pour [installer le pilote de périphérique USB](#) pour obtenir des instructions sur la façon de procéder.

2.2. Installer et désinstaller l'environnement de programmation Studuino

Pour plus d'informations concernant l'installation ou la désinstallation de l'environnement de programmation Studuino pour Windows, Mac OS X ou Raspberry Pi, veuillez lire le guide pour [installer l'environnement de programmation Studuino](#).

3. Écran de démarrage



Seuls les utilisateurs de Windows peuvent ouvrir les environnements de programmation Studuino à l'aide de l'écran de démarrage. Vous verrez l'écran ci-dessus lors de l'ouverture de l'environnement de programmation Studuino sur Windows. Veuillez lire ce qui suit pour plus de détails.

- **Programmation par blocs (Scratch)**

Quand vous cliquez sur « Programmation par blocs », vous avez le choix entre 3 versions du logiciel (Robotique, Animation, Expériences).

- **Langue (Language)**

Choisissez cette icône  pour changer la langue du logiciel en français.

- **Exemples (Samples)**

Des extraits de vidéos, instructions et fichiers pour robots conçus avec des blocks, la carte et le logiciel Studuino et d'autres éléments.

- **Manuels (Manuals)**

Manuels pour l'environnement de programmation Studuino.

- **Sortie (Exit)**

Utilisez ce bouton pour fermer l'environnement de programmation Studuino.

4. Scratch – Programmation par blocs

4.1. Présentation et caractéristiques



L'environnement de programmation Studuino est basé sur l'environnement de programmation Scratch développé par le Massachusetts Institute of Technology. Les programmes de commandes du robot peuvent être créés en faisant glisser des blocs de la palette de blocs et en les attachant à d'autres blocs dans le champ de script.

Il est possible de créer des programmes avancés aussi sophistiqués que ceux écrits dans d'autres langages de programmation (tels que C) car les blocs sont compatibles avec des éléments de programmation standard tels que des branchements, des boucles et des variables. Vous pouvez également contrôler votre robot en temps réel en mode test. Utilisez le mode test pour créer un programme tout en vérifiant les valeurs des capteurs (à l'aide du tableau des capteurs), l'état du robot et plus encore (voir la section **5.5. du Menu principal**).

Après avoir créé votre programme, il peut être transféré à Studuino et exécuté en cliquant sur «Transférer».

Le programme de contrôle fini peut également être converti en langage Arduino pour une édition ultérieure (voir **Export en langage Arduino** dans la section **5.5. du Menu principal**).

L'environnement de programmation en blocs Studuino est décrit ci-dessous.

4.2. La palette du groupe de commandes et la palette de blocs

La palette du groupe de commandes et la palette de blocs contrôlent les blocs qui sont utilisés pour la programmation. Les palettes de blocs peuvent être visualisées en cliquant sur les différents boutons de la palette de commandes. Les boutons et les blocs de la palette du groupe de commandes sont décrits ci-dessous.

Bouton	Blocs
Mouvement	Blocs qui contrôlent les moteurs à courant continu, les servomoteurs, les avertisseurs sonores et les LED.
Contrôle	Éléments standard de programmation tels que les branchements, les boucles, les fonctions et les blocs de traitement
Capteurs	Blocs faisant appel aux valeurs des capteurs
Opérateurs	Blocs qui contrôlent les opérateurs arithmétiques et logiques basiques et avancés
Variables	Blocs qui contrôlent les variables, les listes et le traitement des variables et des listes

4.2.1. À propos de Scratch - la programmation par blocs

Dans l'environnement de programmation par blocs, il existe deux types de blocs, ceux avec des encoches en haut et en bas tels que  (blocs de traitement) et ceux sans encoches tels que  et  (blocs de paramètre). Les blocs de traitement, utilisés essentiellement pour le traitement des actions, sont joints pour créer les programmes contrôlant votre robot.



Les blocs s'attachent verticalement.



Le moteur à courant continu se déplace à la vitesse maximale pendant une seconde avant de s'arrêter.

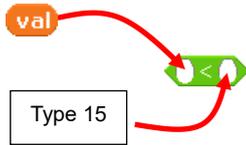
Les blocs avec des bords arrondis tels que celui-ci  déclarent des valeurs et sont principalement utilisés pour modifier les réglages d'autres blocs. Des blocs hexagonaux tels que  sont utilisés pour définir des conditions et modifier des blocs de traitement conditionnels tels que .



Associer un bloc de valeur d'un capteur de lumière à un bloc val dans un bloc de somme...



...crée un bloc utilisant la somme du capteur de lumière et de val



En définissant un opérateur inférieur à (<) en utilisant un bloc val et une valeur d'entrée de 15...



...cet opérateur vérifie si val est inférieur à 15.



En ajoutant un bloc qui vérifie si la valeur du capteur de lumière est inférieure à 50 et un bloc de moteur à courant continu à un bloc de branchement...



...ce branchement lancera le moteur à courant continu si la valeur du capteur de lumière est inférieure à 50.

La case d'entrée arrondie en ① et l'espace hexagonal en ② peuvent être utilisés pour modifier les paramètres du bloc.



La case d'entrée arrondie en ① peut accepter des blocs tels que 'Light Sensor A6 value' ou des entrées numériques. L'espace hexagonal en ② peut accepter un bloc tel que '<' pour changer ses paramètres. La section suivante explique les différents types de blocs de la palette de blocs.

■ Blocs de la palette Mouvement

Les blocs de la palette de mouvement sont utilisés pour contrôler les pièces de votre robot. Les blocs des pièces qui n'ont pas été configurés dans les paramètres des ports sont grisés et ne peuvent pas être placés dans le champ de script.



Connecté à Studuino



Pas connecté à Studuino

Consultez la description de chaque bloc ci-dessous.

- **Bloc de contrôle du servomoteur**

Contrôle un servomoteur connecté à votre Studuino.



Définit le servomoteur qui se connecte avec les connecteurs (D2-D12) spécifiés en ① avec un angle (0-180) spécifié en ②. Tout paramètre avec une valeur inférieure à 0 sera défini à 0 et toute valeur supérieure à 180 sera définie à 180.

- **Blocs de contrôle de moteur à courant continu**

Contrôle un moteur à courant continu connecté à votre Studuino.



Définit le moteur à courant continu connecté aux connecteurs (M1/M2) spécifiés en ① avec la vitesse (0-100) spécifiée en ②. Plus la vitesse est élevée, plus vite le moteur à courant continu fonctionne. Tout paramètre avec une valeur inférieure à 0 sera défini comme 0 et toute valeur supérieure à 100 sera définie comme 100.



Définit le moteur à courant continu connecté aux connecteurs (M1/M2) spécifiés en ① avec la vitesse (0-100) spécifiée en ②.

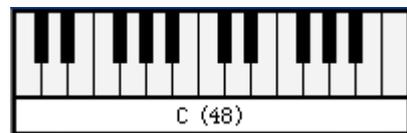


Définit le moteur à courant continu connecté aux connecteurs (M1/M2) spécifiés en ① avec la méthode d'arrêt (frein/roue libre) spécifiée en ②.

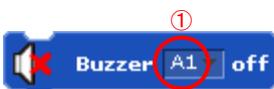
- **Blocs de contrôle de l'avertisseur sonore**



Définit l'avertisseur sonore connecté aux connecteurs (A0-A5) spécifié en ① avec un volume de sortie spécifié en ②. Cliquez sur ▼ en ② pour régler la note que vous souhaitez que l'avertisseur sonore joue. Vous pouvez sélectionner des gammes allant de C48 (130 Hz) à C108 (4186 Hz).



Vous pouvez également sélectionner des notes allant de C48 (130 Hz) à C72 (523 Hz). Si vous souhaitez utiliser des gammes supérieures à C72, saisissez-les directement à l'aide du clavier. La lecture des notes avec l'avertisseur sonore peut interférer avec le fonctionnement d'un moteur à courant continu connecté à M1. Si vous souhaitez utiliser un moteur à courant continu connecté à M1, utilisez un bloc pour arrêter l'avertisseur sonore avant de fixer un bloc de moteur à courant continu.



La liste déroulante en ① est utilisée pour arrêter tout avertisseur sonore connecté à (A0-A5).

- **Bloc de contrôle de LED**



La liste déroulante en ① est utilisée pour allumer ou éteindre toute LED connectée à (A0-A5).

- **Blocs de la palette « Contrôle »**

Les blocs de la palette « Contrôle » vous permettent de contrôler le flux de votre programme. Retrouvez la description de chaque bloc ci-dessous.



Ce bloc ouvre une fonction. Vous pouvez nommer la fonction en utilisant la liste déroulante en ①.



Utilisez la liste déroulante en ① pour faire appel à une fonction spécifique.



Attendez le nombre de secondes spécifié en ①.



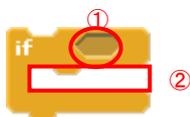
Le processus inséré en ① se produira en boucle indéfiniment.



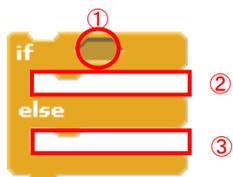
Le processus inséré en ① se répétera le nombre de fois spécifié en ②.



Vérifie en continu si la condition en ① a été respectée et exécute le processus en ② si c'est le cas. La condition en ① est spécifiée à l'aide d'un bloc de la palette des opérateurs.



Si la condition en ① a été respectée, le processus en ② est exécuté. La condition en ① est spécifiée à l'aide d'un bloc de la palette des opérateurs.



Si la condition en ① a été respectée, le processus en ② est exécuté. Si ce n'est pas le cas, le bloc exécute le processus en ③. La condition en ① est spécifiée à l'aide d'un bloc de la palette des opérateurs.



Attend jusqu'à ce que la condition en ① soit respectée. La condition en ① est spécifiée à l'aide d'un bloc de la palette des opérateurs.



Le processus en (2) se répète jusqu'à ce que les conditions pour le processus en (1) soient respectées. La condition en (1) est spécifiée à l'aide d'un bloc de la palette des opérateurs.



Ce bloc permet de faire fonctionner les servomoteurs des blocs insérés en (2) à la vitesse synchronisée insérée en (1). Ce bloc fonctionne uniquement avec des blocs de servomoteur et les autres blocs ne seront pas reconnus.

■ Blocs de la palette « Capteurs »

Les blocs de la palette « Capteurs » sont utilisés pour récupérer des valeurs de capteurs. Les blocs des pièces qui n'ont pas été configurés dans les paramètres des ports sont grisés et ne peuvent pas être placés dans le champ de script.

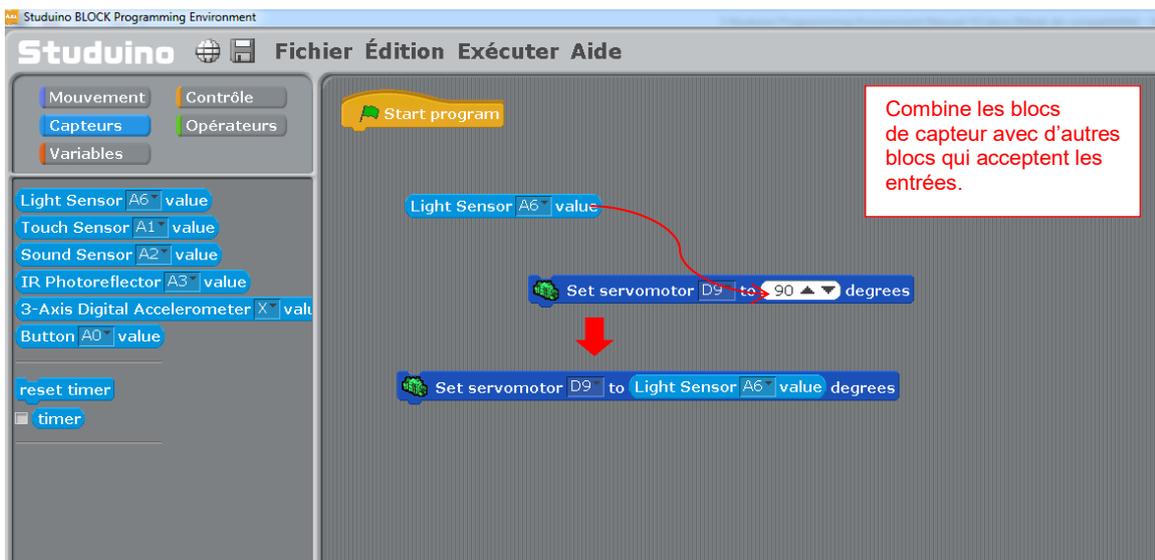


Connecté à Studuino



Pas connecté à Studuino

Les blocs des capteurs peuvent être combinés avec d'autres blocs et utiliser les valeurs d'autres blocs. L'image ci-dessous montre une combinaison entre un bloc de servomoteur et un bloc de capteur de lumière. Cette combinaison vous permet de modifier les angles du moteur en fonction de la quantité de lumière dans une pièce.



Consultez la description de chaque bloc ci-dessous.



(1)

Renvoie la valeur du capteur de lumière connecté à A0-A7 spécifié en (1). La valeur du capteur varie de 0 à 100.



(1)

Renvoie la valeur du capteur tactile connecté à A0-A5 spécifié en (1). La valeur du capteur est 0 lorsque le capteur est touché et 1 lorsqu'il est dans son état normal.



(1)

Renvoie la valeur du capteur de son connecté à A0-A7 spécifié en (1). La valeur du capteur varie



de 0 à 50.

Renvoie la valeur du capteur infrarouge connecté à A0-A7 spécifié en ①. La valeur du capteur varie de 0 à 100.



Renvoie la valeur d'une direction spécifiée (axe X/Y/Z) d'un accéléromètre connecté. La valeur du capteur varie de 0 à 100.



Renvoie la valeur d'un interrupteur à bouton-poussoir connecté à A0-A3 spécifié en ①. La valeur du capteur est 0 lorsque le bouton est poussé et 1 lorsqu'il est dans son état normal.



Réinitialise la minuterie à 0.



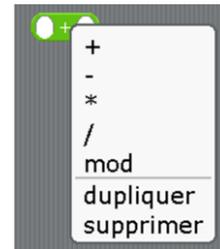
Renvoie la valeur de la minuterie en secondes.

■ Blocs de la palette « Opérateurs »

Les blocs de la palette « Opérateurs » réalisent des calculs sur la base des valeurs que vous insérez. Retrouvez la description de chaque bloc ci-dessous.



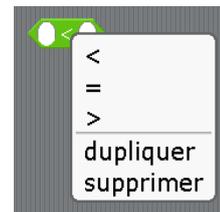
Additionne les valeurs insérées en ① et en ②. Les blocs d'opérations arithmétiques incluent des blocs de soustraction (-), de multiplication (*) et de division (/). Vous pouvez sélectionner les autres opérateurs arithmétiques dans le menu contextuel en faisant un clic droit sur le bloc.



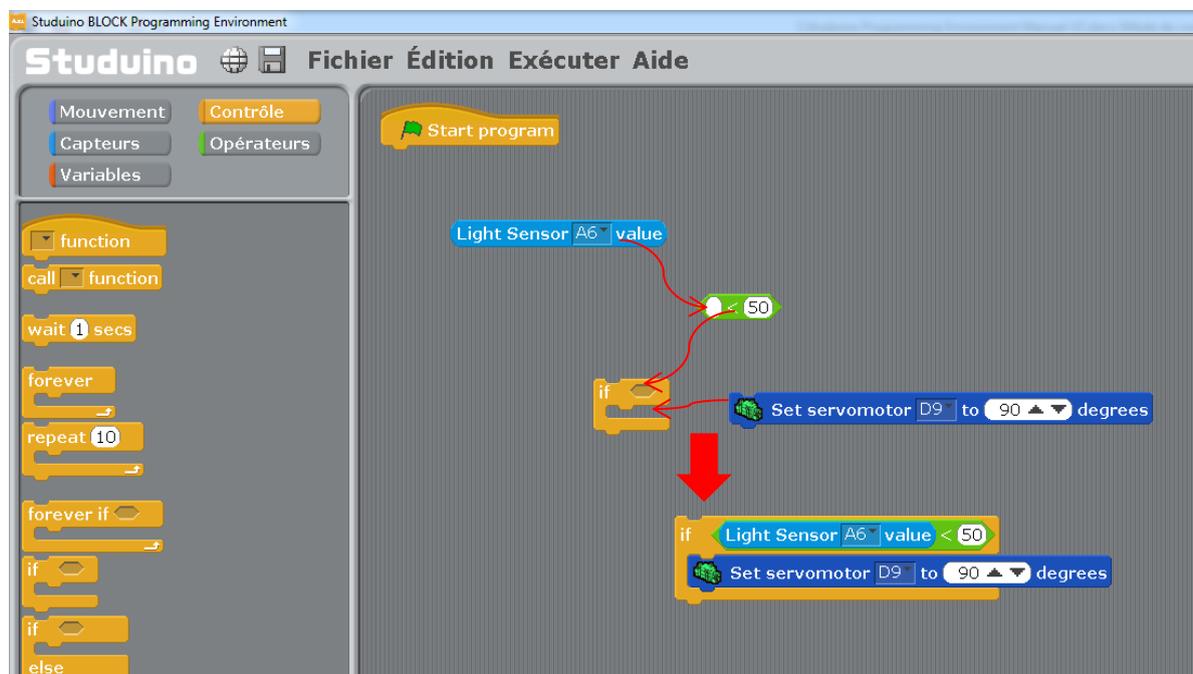
Renvoie un nombre aléatoire entre les valeurs définies en ① et ②.



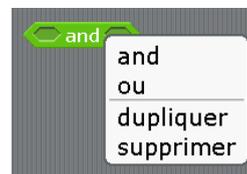
Détermine si la valeur définie en ① est inférieure à la valeur définie en ②. Les autres opérateurs comparatifs définissent si les valeurs sont égales (=) ou si une valeur est plus grande (>). Vous pouvez sélectionner les autres opérateurs comparatifs dans le menu contextuel en faisant un clic droit sur le bloc.



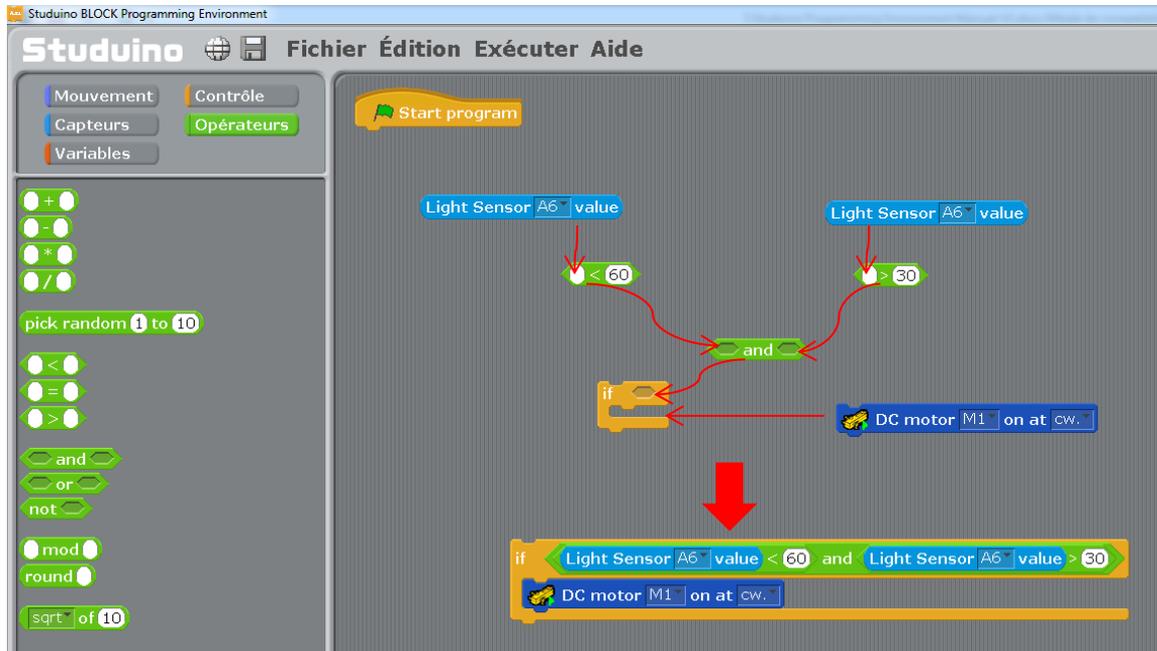
L'image ci-dessous montre une combinaison entre un capteur de lumière, une condition et un bloc de servomoteur. Vous pouvez utiliser cela pour régler l'angle du servomoteur à 90 degrés si la valeur du capteur de lumière est inférieure à 50.



Ce bloc est un opérateur «AND» et utilise les conditions établies en ① et ②. Les autres blocs d'opérateur logique sont «OR» et «NOT». Vous pouvez sélectionner les autres opérateurs logiques dans le menu contextuel en faisant un clic droit sur le bloc.



L'image ci-dessous montre une combinaison entre un opérateur logique, une condition et un bloc de moteur à courant continu. Vous pouvez utiliser cette combinaison pour faire avancer le moteur à courant continu lorsque la valeur sonore est comprise entre 30 et 60.



Ce bloc divise ① par ② et renvoie le reste. Comme pour le bloc , un clic droit sur ce bloc vous permettra de sélectionner d'autres opérateurs arithmétiques dans le menu contextuel.



Ce bloc renvoie l'entier le plus proche (nombre entier) pour la valeur définie en ①.



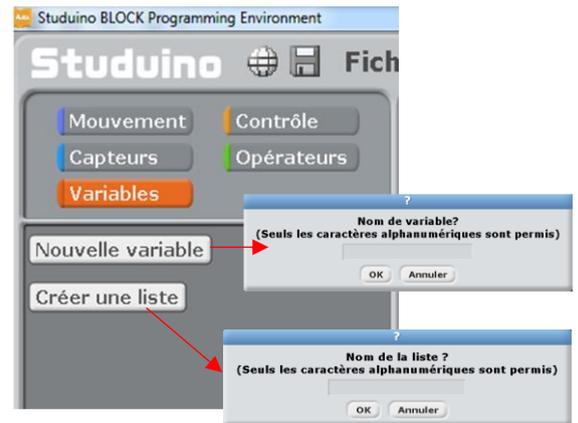
Ce bloc utilise la valeur en ① pour l'opération arithmétique spécifiée en ②. Vous pouvez choisir parmi des valeurs absolues, des racines carrées, des fonctions trigonométriques, des logarithmes et des exposants.



■ Blocs de la palette « Variables »

Les blocs de la palette « Variables » sont utilisés pour créer des variables et des listes. Cliquez sur le bouton «**Nouvelle variable**» et entrez un nom pour créer une variable. Cliquez sur le bouton «**Créer une liste**» et entrez un nom pour créer une liste. L'utilisation de caractères non alphanumériques (A-Z, a-z, 0-9) dans les noms de fonction, de variable et de liste entraînera une erreur de construction lors du transfert de votre programme (voir Transfert dans la section 5.5. du **Menu principal** pour plus de détails). Utilisez uniquement des caractères alphanumériques (A-Z, a-z, 0-9) pour les noms de fonction, variable et liste. Les valeurs des variables et des listes peuvent être de 3,4028235E + 38 à 3,4028235E + 38, ou d'un maximum de 32 bits (4 octets).

La section suivante explique les blocs de variables dans la palette de variables (pour une variable nommée "val").



val

Renvoie la valeur de la variable.

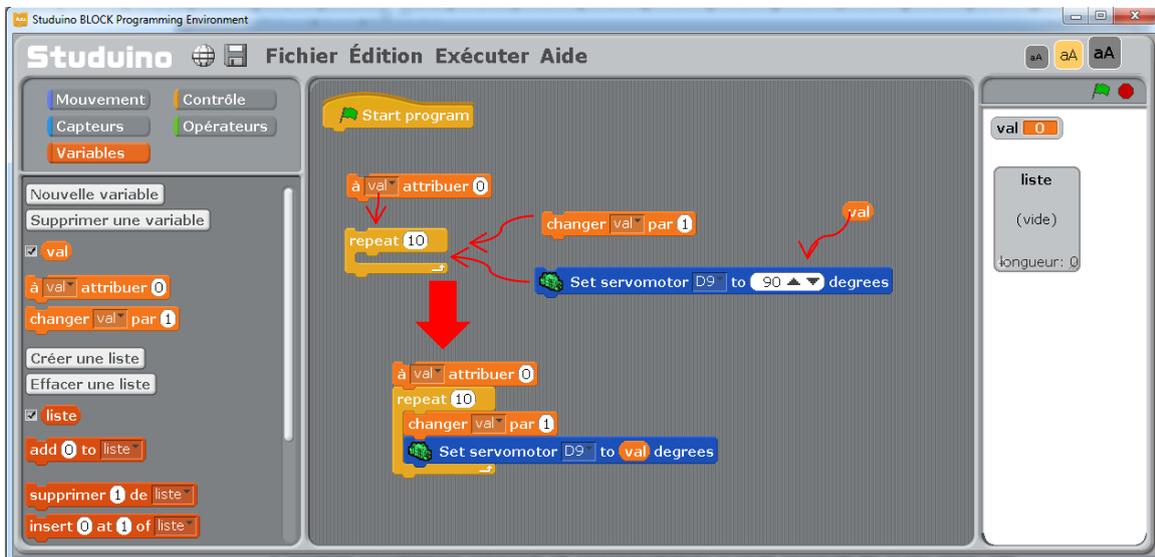
① à val attribuer ② 0

Définit la variable en ① en la valeur définie en ②.

② changer val par ① 1

Augmente la valeur définie en ① par la valeur définie en ②.

Vous pouvez créer jusqu'à 70 variables différentes.



L'image ci-dessus montre la combinaison d'une boucle et d'un bloc de servomoteur. Vous pouvez utiliser ceci pour créer un programme qui augmente de 10 la valeur dans la variable "val", ce qui va entraîner une répétition à 10 reprises du processus et augmenter l'angle du moteur de 10 à 100 par paliers de 10 degrés.

Les listes sont des blocs structurés qui vous permettent d'ajouter ou de retirer des valeurs comme bon vous semble. Les listes peuvent accueillir jusque 40 valeurs distinctes. La section suivante explique les blocs de listes dans la palette des variables (pour une liste nommée «liste»).

liste

Renvoie la première valeur dans la liste.

① ②
add 0 to liste

Ajoute la valeur (①) à la liste (②).

① ②
supprimer 1 de liste

Supprime la valeur qui se trouve en position (①) de la liste (②).

③ ② ①
insert 0 at 1 of liste

Insère la valeur 0 (③) à la liste (①) en position 1 (②).

② ① ③
replace item 1 of liste with 0

Remplace la valeur en position 1 (②) dans la liste (①) par la valeur définie en (③).

② ①
élément 1 de liste

Renvoie la valeur de l'élément se trouvant en position 1 (②) dans la liste (①).

①
longueur de liste

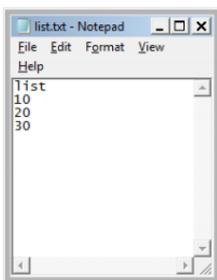
Renvoie la longueur de la liste (①).

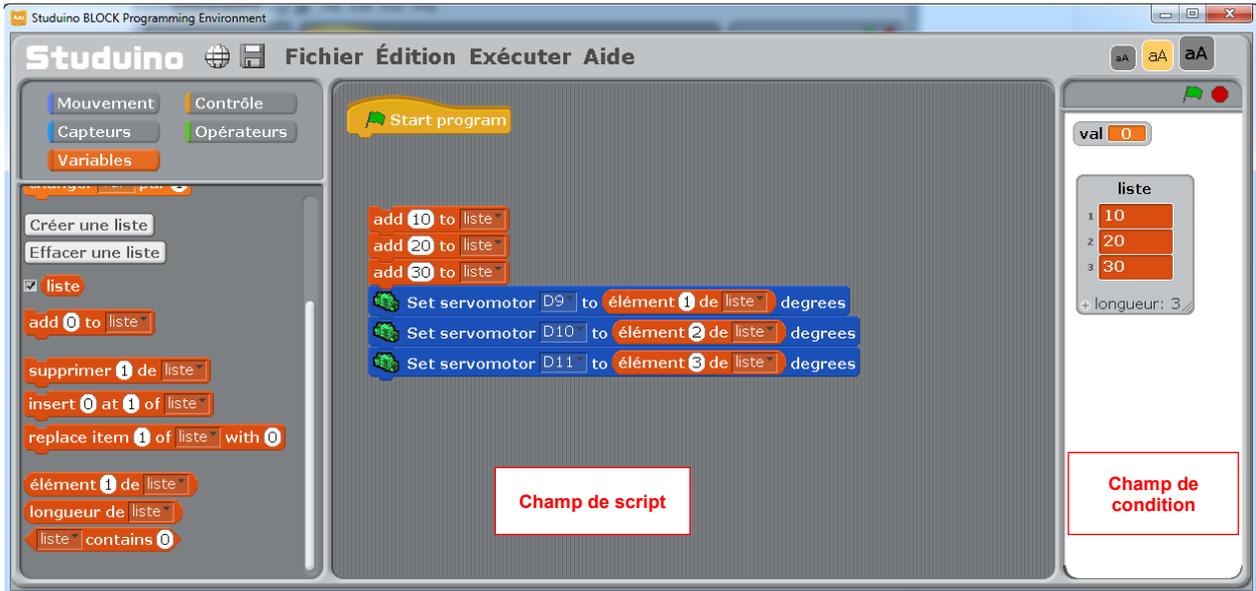
① ②
liste contains 0

Détermine si la liste (①) comprend la valeur 0 (②).

write liste to C:\Users\Chloe\Documents\liste.txt as text file

Enregistre la liste spécifiée en (①) en texte dans le nom de fichier et l'emplacement spécifiés en (②). Comme montré ci-dessous, le contenu de la liste sera présenté en-dessous du nom de la liste.





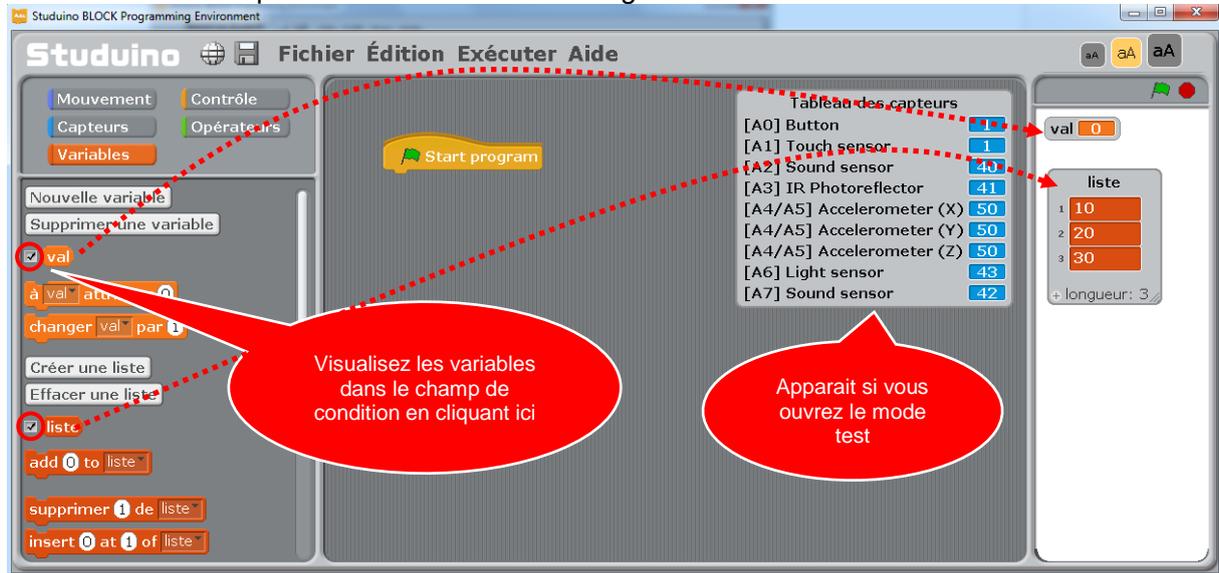
L'image ci-dessus ajoute les valeurs 10, 20 et 30 des 3 blocs précédents à la liste nommée «liste». Ce processus fait une liste qui comprend la valeur 10 en première position, 20 en deuxième position et 30 en troisième position. En utilisant cette liste, vous pouvez définir les angles du servomoteur dans les blocs suivants. Le servomoteur connecté à D9 aura ses angles réglés sur 10 degrés, le servomoteur en D10 sur 20 degrés et le servomoteur en D11 sur 30 degrés.

4.3. Champ de script

Le champ de script est l'endroit où vous faites glisser et relâchez des blocs pour créer votre programme. L'environnement de programmation par blocs débute avec un bloc  dans le champ de script. Ce bloc démarre votre programme. Tout programme que vous effectuez doit inclure ce bloc.

4.4. Le champ de condition

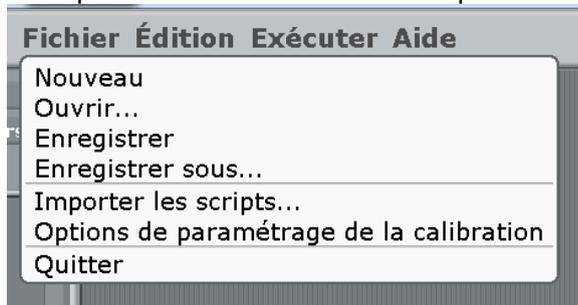
Le champ de condition affiche les valeurs des capteurs, les variables et les listes en mode test. Le lancement du test ouvrira le tableau des capteurs où vous pourrez visualiser les valeurs de n'importe quel capteur connecté à votre Studuino. Comme indiqué ci-dessous, vous pouvez également visualiser la modification des valeurs de toutes les variables ou listes dans le champ de conditions en cliquant sur la case se situant à gauche de celles-ci.



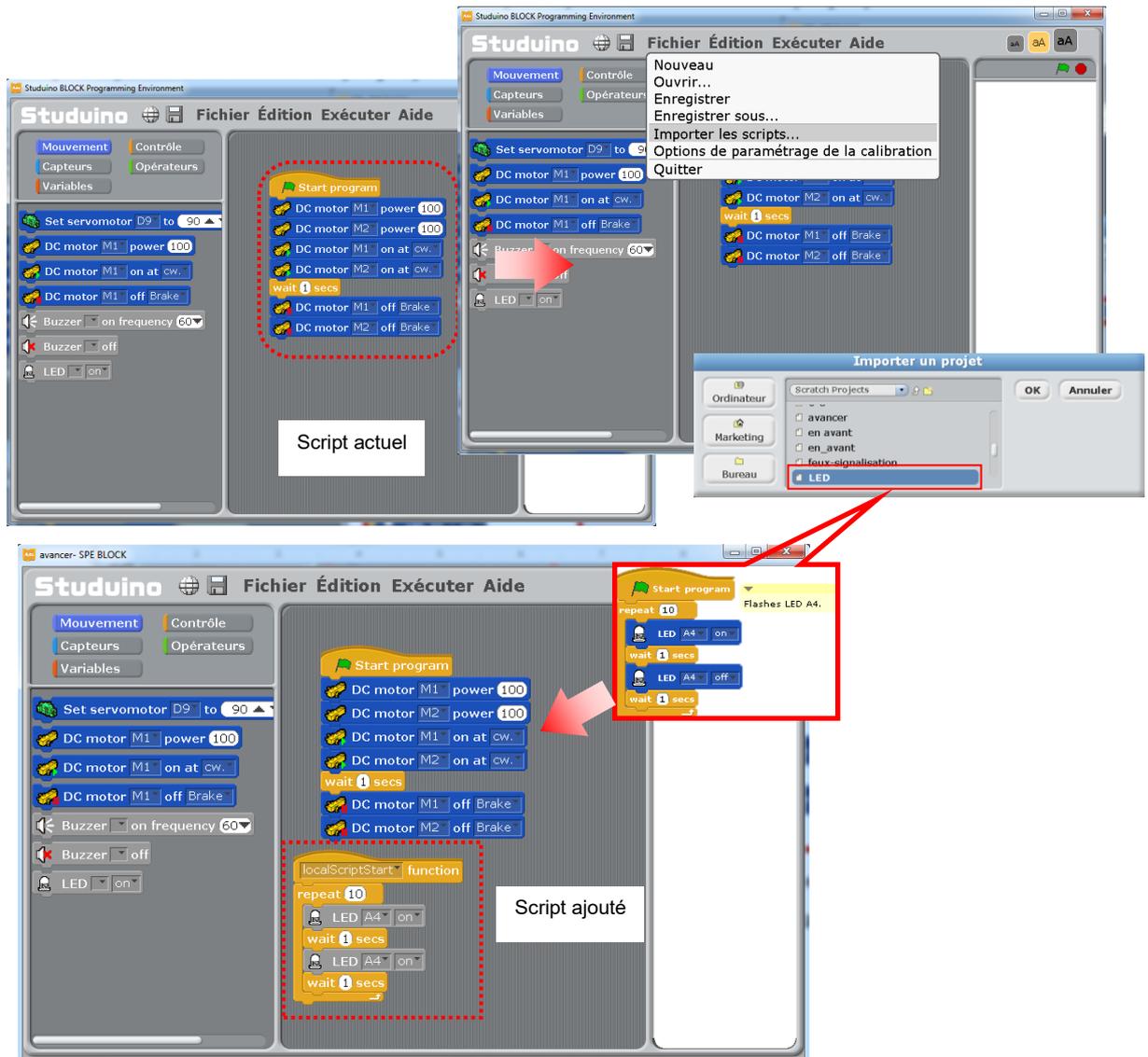
4.5. Le menu principal

■ «Menu Fichier»

Vous pouvez utiliser le menu Fichier pour sauvegarder, charger des projets et importer des scripts.

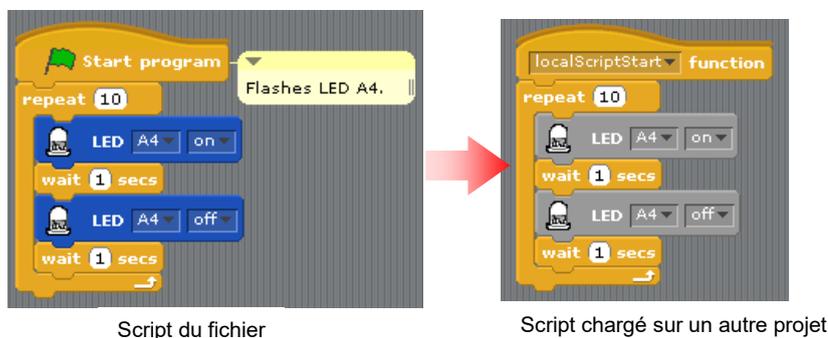


- «Nouveau»
Commencez un nouveau projet.
- «Ouvrir...»
Ouvrez un fichier préalablement sauvegardé.
- «Enregistrer»
Sauvegardez votre projet actuel.
- «Enregistrez sous...»
Sauvegardez votre programme sous un nom donné.
- «Importer les scripts...»
Chargez les scripts à partir d'un fichier de projet précédent.

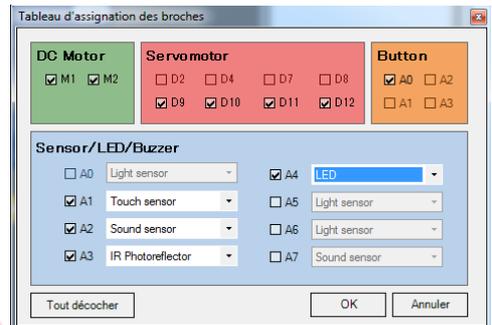
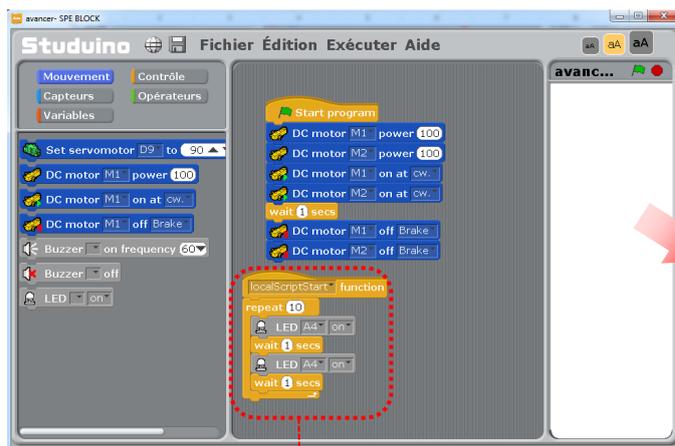


Prenez en compte ces trois éléments lors du chargement de scripts :

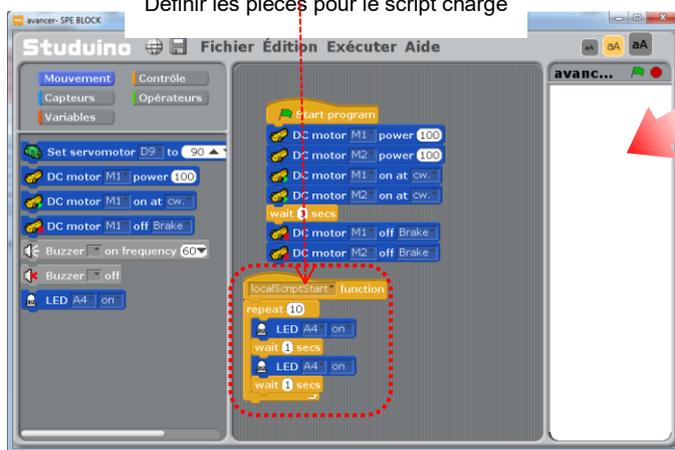
- 1 . Dans la mesure où l'environnement de programmation ne peut pas afficher plus de deux blocs de démarrage jaune, le programme sera chargé en tant que fonction «localScriptStart».
- 2 . Tous les commentaires pour le programme chargé seront supprimés.



- 3 . Les paramètres du port seront remplacés par ceux du script chargé et tous les blocs des pièces non configurées seront grisés. Ces blocs peuvent être activés en configurant les pièces correspondantes dans la boîte de dialogue des paramètres de port.

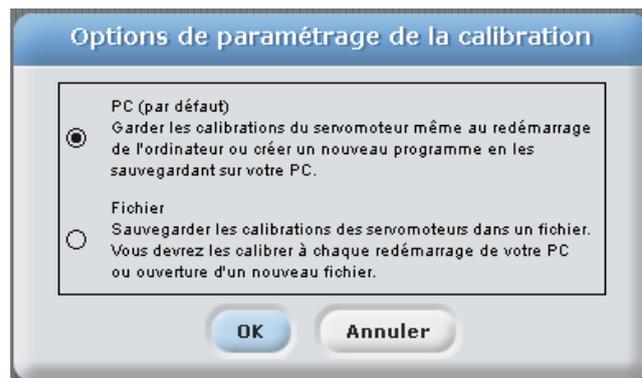


Définir les pièces pour le script chargé



- **Options de paramétrage de la calibration**

Cette option sert à spécifier si les réglages de calibration du moteur que vous avez faites doivent être enregistrés sur votre PC ou dans un fichier séparé.

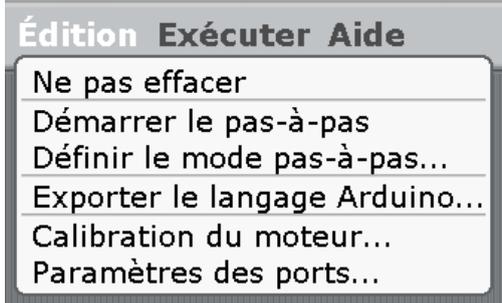


- **«Quitter»**

Clôture l'environnement de programmation par blocs.

■ «Menu Édition»

Ce menu d'édition peut être utilisé pour modifier votre programme.



- «Ne pas effacer»

Restaure un bloc supprimé.

- «Démarrer / Arrêter le pas-à-pas»

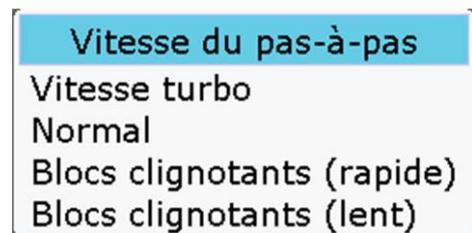
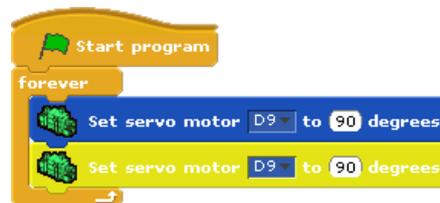
Commence à exécuter votre programme étape par étape. Lorsque vous utilisez le «pas-à-pas», le bloc actuellement en cours de traitement devient jaune. Vous pouvez modifier la vitesse d'exécution des étapes en utilisant le menu «Définir le mode pas-à-pas»

- «Définir le mode pas-à-pas...»

Utilisez ceci pour modifier la vitesse de traitement des étapes. La vitesse turbo est la plus grande vitesse et les blocs clignotants est la méthode la plus lente de traitement des étapes.

- «Exporter le langage Arduino...»

Convertit le programme dans votre champ de script en langage Arduino. Tous les blocs dans votre champ de script seront convertis en langage Arduino. Le code source que vous exportez en utilisant cette fonctionnalité peut être compilé et envoyé à votre Studuino en utilisant Arduino IDE.

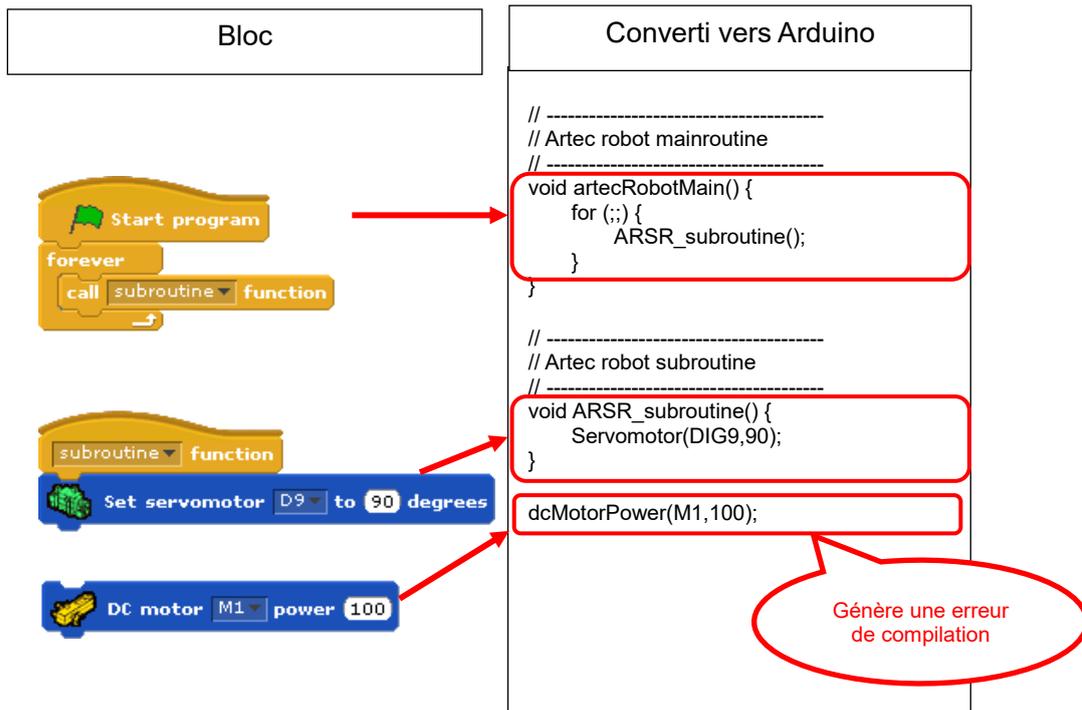


Tous les blocs qui ne sont pas attachés au bloc-maître



ou

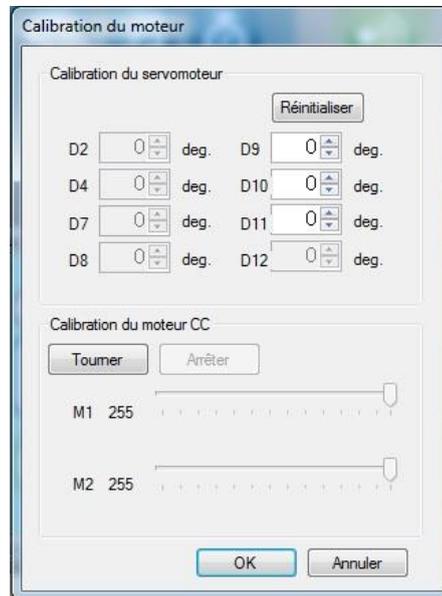
seront également convertis en langage Arduino. Utilisez Arduino IDE pour supprimer tout code de blocs non connectés avant de créer votre programme.



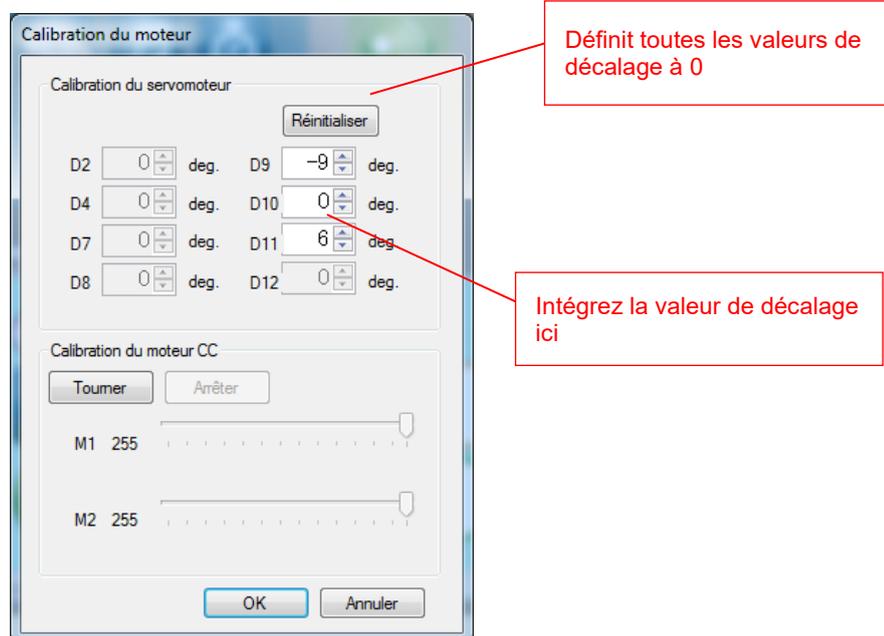
Les blocs non connectés aux fonctions (orphelins) seront tout de même traités par la fonction dans l'environnement de programmation par blocs, mais le code orphelin entraînera une erreur de compilation dans Arduino IDE. Les valeurs non connectées dans les blocs telles que  sont définies sur 0 lors de l'exportation vers le langage Arduino. Les conditions non connectées dans des blocs tels que  seront définies comme «fausses» lors de l'exportation vers le langage Arduino.

- «Calibration du moteur...»

Utilisez cette fonction pour ajuster les angles de votre servomoteur et les enregistrer dans un fichier. L'exécution de cette fonction démarrera le mode test et ouvrira la boîte de dialogue de **calibration du moteur**.



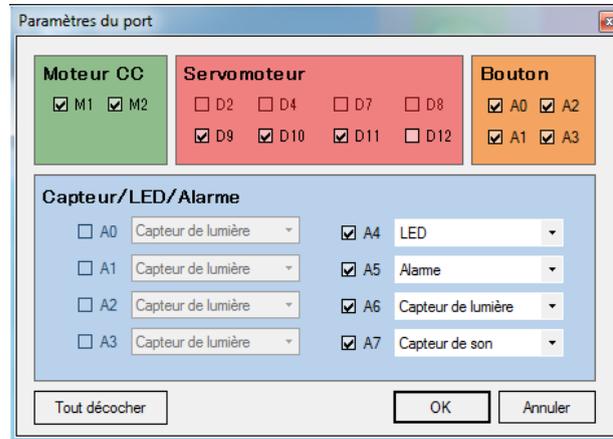
Lorsque la boîte de dialogue s'ouvre, tous les angles du servomoteur connecté sont réglés à 90 degrés par défaut.



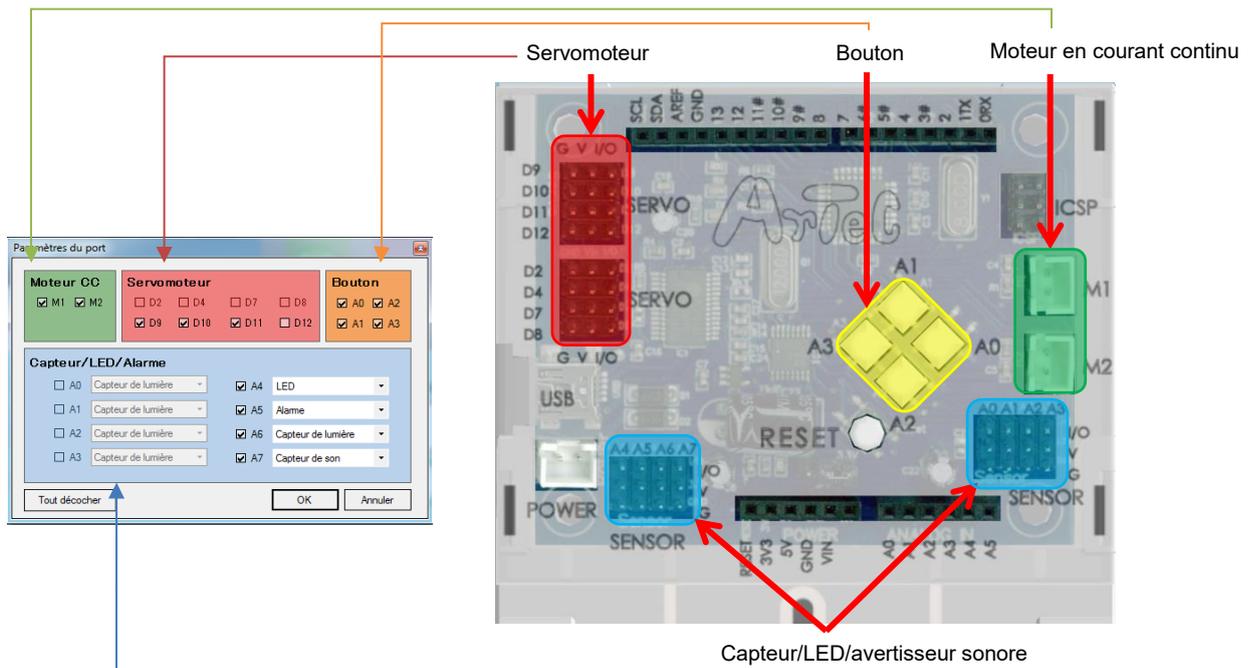
Observez votre servomoteur et réglez les valeurs jusqu'à ce que vous obteniez l'angle correct de 90 degrés. Cliquez sur OK lorsque vous avez terminé le calibrage de votre servomoteur. Les paramètres enregistrés seront utilisés pour les futurs programmes.

- «Paramètres des ports...»

Vous permet de voir les pièces connectées à votre Studuino dans l'environnement de programmation par blocs. En cliquant sur cette fonction, vous verrez la boîte de dialogue des paramètres des ports.



Comme le montre l'image ci-dessous, cette boîte de dialogue affiche tous les connecteurs et boutons du Studuino. Cochez tous ceux pour lesquels une pièce est connectée.



Tel qu'indiqué dans la section **1.3. À propos de Studuino**, les connecteurs ou boutons suivants ne peuvent pas être utilisés en même temps:

- Connecteur de moteur à courant continu M1 et connecteur de servomoteur D2, D4
- Connecteur de moteur à courant continu M2 et connecteur de servomoteur D7, D8
- Interrupteur à bouton-poussoir A0-A3 et connecteur de capteur/LED/avertisseur sonore A0-A3

Dans la mesure où ces combinaisons ne peuvent pas être utilisées simultanément dans la boîte

de dialogue des paramètres des ports, désélectionnez un côté pendant que l'autre est activé. Par exemple, lors de l'utilisation de D2 et D4 pour le servomoteur, les cases à cocher pour D2 et D4 ne seront disponibles qu'en décochant M1 pour le moteur à courant continu.

<Avertissement>

Pour l'accéléromètre, qui utilise deux connecteurs, tous les connecteurs correspondants (ou une combinaison compatible) doivent être cochés.

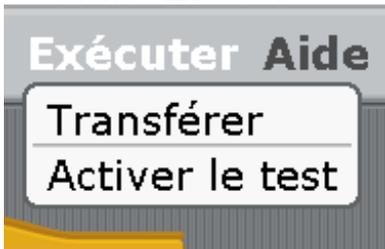
Partie	Combinaison de connecteur
Accéléromètre	A4, A5

Connecteurs utilisés lorsqu'un accéléromètre est branché sur le Studuino

Lorsque les cases des connecteurs A4 et A5 sont cochées, l'accéléromètre connecté au Studuino s'affichera dans la liste déroulante. Lorsque vous sélectionnez l'accéléromètre (ou une autre pièce qui utilise plusieurs connecteurs à la fois) pour l'un des connecteurs, les autres connecteurs qu'il utilise se sélectionneront automatiquement.



■ **Menu «Exécuter»**



Le menu «Exécuter» vous permet de lier votre Studuino pendant que vous créez un programme.

● **«Transférer»**



Le transfert compile tout programme que vous avez créé et l'envoie à votre Studuino. Vous verrez un message comme celui ci-contre si une erreur de construction se produit.

Une erreur de construction se produira si des blocs sont déconnectés des blocs maîtres

 ou de fonction  dans le champ de script. Voir ci-dessous pour plus de détails sur ce message.

Ce message apparaît s'il y a des blocs déconnectés des blocs maîtres ou de fonction dans le champ de script. Veuillez connecter ou supprimer des blocs orphelins dans le champ de script avant d'utiliser la fonction de transfert. Vous verrez également ce message lorsque les noms de fonctions, de variables ou de listes contiennent des caractères non alphanumériques. Renommez votre fonction, votre liste ou votre variable et réessayez.



Ce message apparaît si des blocs de fonction indéfinie se trouvent dans votre programme. Comme indiqué ci-dessous, tous les blocs appelant des fonctions doivent avoir une fonction définie. Lorsque vous êtes prêt à créer votre programme, placez et définissez un bloc de fonction  dans le champ de script ou supprimez les blocs de fonction orphelins avant d'utiliser un transfert.

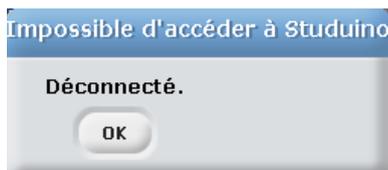


Ce message apparaît lorsque les deux messages d'erreur décrits ci-dessus se produisent en même temps. Lors de la préparation de la construction, veuillez connecter ou supprimer tout bloc orphelin et définissez des fonctions pour les blocs les utilisant avant d'utiliser la fonction de transfert.

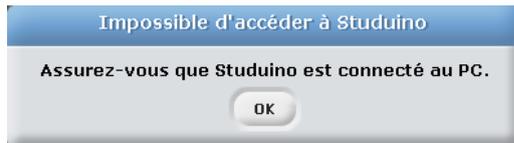
Une fois que vous avez créé votre programme, vous pouvez le transférer sur votre Studuino. Consultez les précisions ci-dessous concernant les messages que vous verrez lors du transfert de votre programme.



Ce message apparaît lors d'un transfert de programme. Ne déconnectez pas votre Studuino du PC lors du transfert d'un programme. Si la communication entre votre Studuino et votre PC est interrompue pendant un transfert, vous verrez les messages ci-dessous.



Ce message s'affiche si la communication entre le PC et Studuino est interrompue pendant un transfert. Reconnectez votre Studuino à l'ordinateur et essayez à nouveau.



Ce message s'affiche quand il n'y a pas de connexion entre votre ordinateur et Studuino. Si vous visualisez ce message, vérifiez la connexion entre votre PC et Studuino et réessayez.



Ce message apparaît lorsqu'une autre application interfère avec la communication entre votre ordinateur et Studuino. Fermez toutes les applications susceptibles de provoquer cette erreur et réessayez.

- **«Exécuter»**

Exécutez tout programme que vous avez transféré. Cela apparaît dans le menu après avoir transféré un programme avec succès.

- **«Activer/Désactiver le test»**

Vous pouvez utiliser le mode test pour communiquer avec votre Studuino et vos composants en temps réel. Assurez-vous que votre PC et Studuino sont connectés avant de démarrer le mode de test.



Cliquez sur «Activer le test» et vous verrez le message à gauche vous indiquant que le mode test est en cours d'initialisation. Ne débranchez pas votre Studuino du PC en mode test.

Si la communication entre votre Studuino et votre PC est interrompue lors d'un transfert, vous verrez les messages ci-dessous. Il se peut également qu'une erreur d'exécution provoque l'arrêt de l'environnement de programmation par blocs. Conservez votre Studuino connecté au PC en tout temps en mode test. Voir ci-dessous pour plus de détails sur ces messages d'erreur.



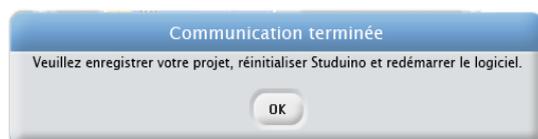
Ce message s'affiche si la communication entre l'ordinateur et le Studuino est interrompue pendant un transfert. Reconnectez votre Studuino à l'ordinateur et essayez à nouveau.



Ce message s'affiche quand il n'y a pas de connexion entre votre ordinateur et le Studuino. Si vous visualisez ce message, vérifiez la connexion entre votre PC et Studuino et réessayez.



Ce message apparaît lorsqu'une autre application interfère avec la communication entre votre ordinateur et Studuino. Fermez toutes les applications susceptibles de provoquer cette erreur et réessayez.



Ce message s'affiche si la communication entre l'ordinateur et le Studuino est interrompue pendant un transfert. Sauvegardez vos données et redémarrez le logiciel.

■ Menu «Aide»

Le menu «Aide» contient des informations concernant l'environnement de programmation par icônes.



- **Environnement de programmation par blocs**

Affiche la version du logiciel que vous utilisez actuellement.

4.6. Menus contextuels

Un clic droit sur les blocs dans le champ de script ou sur le champ de script lui-même affichera un menu contextuel.



- **« Nettoyer »**

Organise les blocs dans le champ de script.

- **« Enregistrer une image des scripts »**

Sauvegarde une image des blocs dans le champ de script comme dans le format .gif.

- **« Ajouter un commentaire »**

Ajoute un commentaire. Le fait de faire glisser le commentaire vers n'importe quel bloc ajoutera le commentaire à ce bloc.

Un clic droit sur n'importe quel bloc, à l'exception du bloc maître



un menu contextuel.

- **« Dupliquer »**

Duplique un bloc ou un groupe de blocs.



- **« Supprimer »**

Supprime un bloc ou un groupe de blocs.

5. Dépannage

La section suivante porte sur les méthodes de dépannage de ce logiciel. Dans le cas où les solutions fournies dans ce manuel ne résoudre pas votre problème ou si votre problème n'est pas abordé, contactez-nous directement à support@speechi.net.

5.1. Dépannage général

Cette section porte sur les méthodes de dépannage communes aux deux environnements de programmation Studuino (programmation par blocs et par icônes).

■ Mon moteur à courant continu ou servomoteur ne fonctionne pas en mode test

Les moteurs nécessitent l'alimentation du boîtier de batterie pour fonctionner correctement. Insérez les piles dans le boîtier de batterie et connectez-la à votre unité Studuino avant de mettre l'appareil sous tension.

■ Mon robot se réinitialise de manière inattendue

Des piles à plat peuvent provoquer la réinitialisation inopinée de votre robot. Remplacez les piles dans le boîtier de la batterie.

■ Mes valeurs de capteur se comportent étrangement

Chaque capteur est différent, ce qui peut provoquer des variations de valeurs de votre capteur.

5.2. Scratch - Environnement de programmation par blocs

■ J'obtiens une erreur en utilisant le mode test

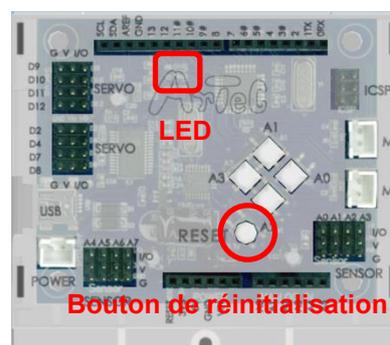
■ J'obtiens une erreur en utilisant le transfert

Il se peut que votre logiciel se clôture sans avertissement préalable si l'erreur ci-dessous se produit.

Titre	Message
Échec de l'accès au Studuino	Votre Studuino n'a pas pu être synchronisé avec votre ordinateur. Sauvegardez votre projet, quittez le logiciel, réinitialisez votre Studuino et redémarrez le logiciel.

Sauvegardez votre programme, quittez le logiciel et suivez les étapes ci-dessous pour réinitialiser votre Studuino.

1. Éteignez l'interrupteur d'alimentation sur le boîtier de la batterie et débranchez le câble USB de votre Studuino.
2. Attendez trois secondes.
3. Rebranchez votre câble USB et allumez l'interrupteur d'alimentation du boîtier de la batterie.
4. Appuyez sur le bouton de réinitialisation (Reset) dans votre Studuino. La réinitialisation provoquera l'allumage d'une LED verte sur votre carte Studuino. (voir image)



Essayez de commencer le mode test après avoir terminé les étapes ci-dessus.

Visualiser le message ci-dessous peut signifier que vos fichiers système de l'environnement de programmation Studuino ont été corrompus.

Titre	Message
Erreur du système	Le fichier système de votre environnement de programmation a été corrompu. Sauvegardez votre projet, quittez le logiciel, réinitialisez votre Studuino et redémarrez le logiciel.

L'erreur qui génère ce message peut provoquer l'arrêt de l'environnement de programmation par blocs sans avertissement préalable.

Sauvegardez votre programme, désinstallez le logiciel en utilisant les étapes de la section **2.3. Désinstaller le logiciel**, et réinstallez l'environnement de programmation.

Titre	Message
Échec de l'accès au Studuino	Erreur du port COM

Ce message s'affiche quand la connexion ne peut pas se faire entre votre ordinateur et votre Studuino. Vous devez redémarrer votre ordinateur. Consultez les parties dédiées aux Transfert et Mode test dans la section **5.5. Menu principal** pour obtenir plus d'informations sur la résolution du problème ci-dessus.

■ J'obtiens une erreur en utilisant le transfert (en compilant un programme)

Visualiser le message ci-dessous peut signifier que vos fichiers système de l'environnement de programmation Studuino ont été corrompus.

Titre	Message
Le programme ne peut être compilé	Une erreur s'est produite lors du processus d'archivage. Veuillez réinstaller.
	Une erreur s'est produite lors du processus de copie de l'objet 1. Veuillez réinstaller.
	Une erreur s'est produite lors du processus de copie de l'objet 2. Veuillez réinstaller.

L'erreur qui cause ce message peut provoquer l'arrêt de l'environnement de programmation par icônes sans avertissement préalable. Sauvegardez votre programme, désinstallez le logiciel en utilisant les étapes de la section **2.3. Désinstaller le logiciel** et réinstallez l'environnement de programmation.

■ J'obtiens un débordement de programme lors de la compilation de mon programme

La taille des programmes qui peuvent être réalisés dans l'environnement de programmation par blocs est basée sur les spécifications de votre Studuino. Ce message s'affichera lorsque la taille de votre programme dépassera les 15 kbytes (Koctets).

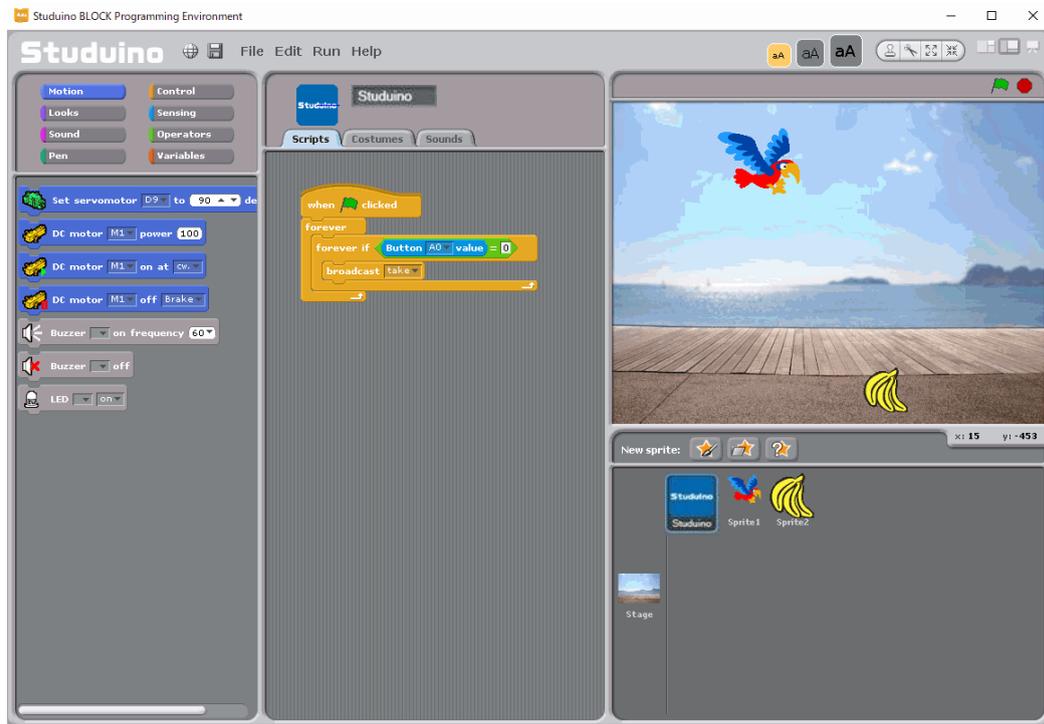
Titre	Message
Le programme ne peut être compilé	Votre script est trop important. Essayez des méthodes différentes pour diminuer la taille de votre programme.

Si ce message s'affiche, vous devrez rendre votre programme plus petit en utilisant d'autres blocs ou d'autres méthodes. Gardez à l'esprit que l'environnement de programmation par blocs n'est pas prévu pour le développement de programmes de grande échelle. En raison de la plus petite échelle de l'environnement de programmation Studuino, il se peut que vous deviez envisager d'utiliser d'autres logiciels tels que Arduino IDE ou Atmel Studio pour développer des programmes plus

importants.

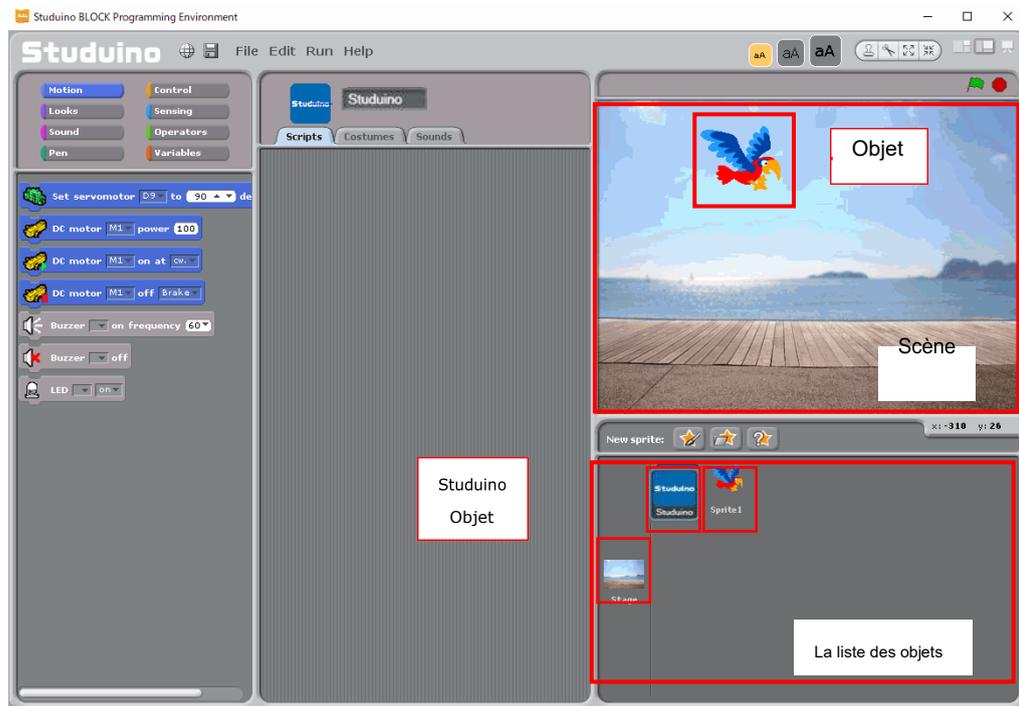
6. Animation

6.1. Présentation et caractéristiques



La partie Animation de l'environnement de programmation Studuino vous permet de créer des jeux et des animations qui s'associent avec votre Studuino. Les jeux que vous faites peuvent être utilisés pour contrôler votre robot ou peuvent être affectés par les valeurs de vos capteurs. Vous pouvez également interagir avec les réseaux de PC appelés meshes. Un mesh vous permet de vous connecter à une copie du logiciel s'exécutant sur un autre ordinateur et de créer des jeux avec la connectivité réseau. Vous pouvez aussi charger les programmes que vous avez faits dans la partie Robotique de l'environnement Studuino ou Scratch 1.4 (vous pouvez donc associer votre Studuino à des jeux que vous avez faits sur Scratch).

6.2. Programmer

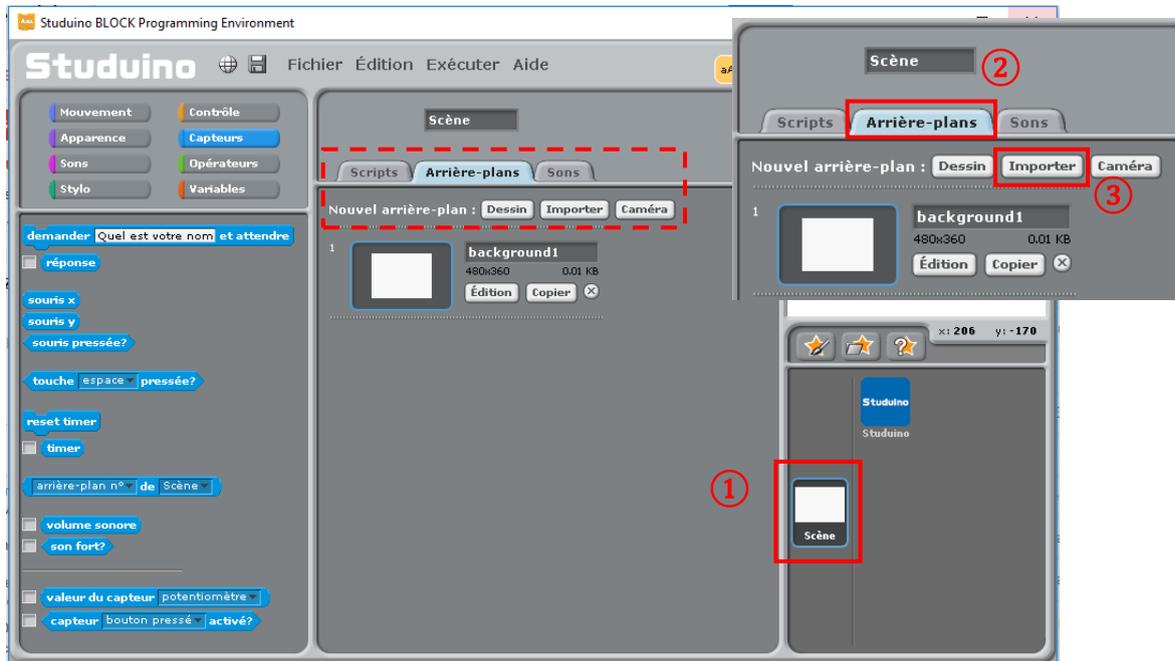


Les **objets** sont les personnages de votre jeu et la **scène** est l'espace dans lequel ils évoluent. Vous pouvez programmer les objets et la scène à faire différentes choses. Les icônes pour les objets et les scènes se trouvent dans la **liste des objets** en bas à droite de l'écran. Les objets Studuino dans la liste des objets sont uniquement utilisés pour contrôler votre Studuino et n'apparaîtront pas sur la scène.

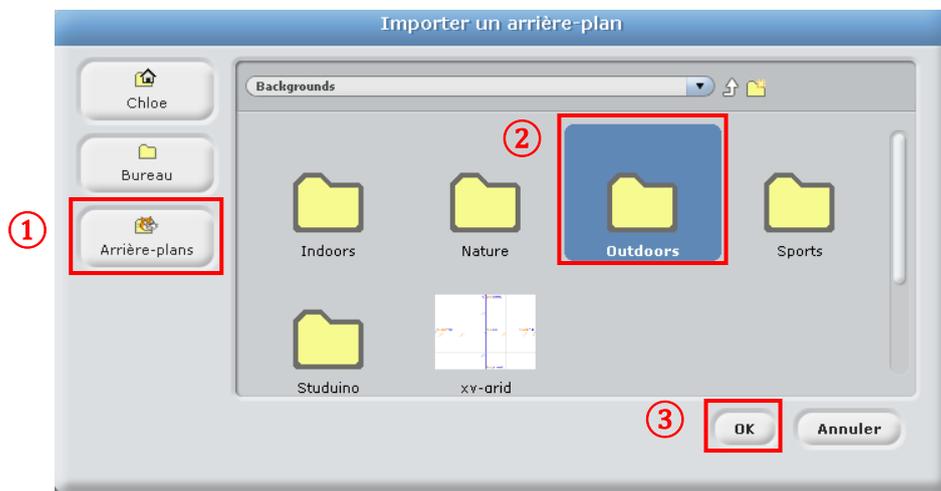
Suivez les étapes de la page suivante pour avoir un aperçu et apprendre à programmer avec ce logiciel.

6.2.1. Ajouter une scène

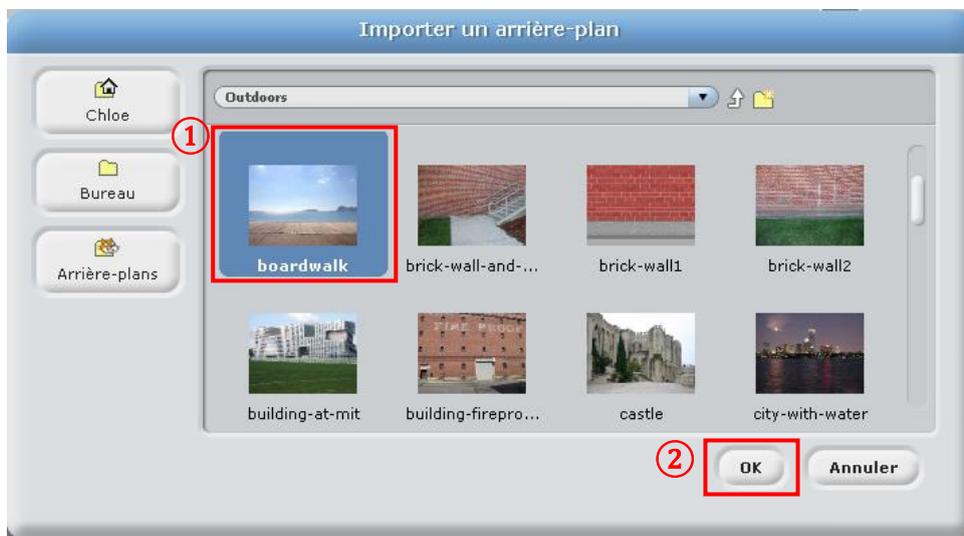
Cliquez sur l'icône **Scène** (①) dans la liste des objets. Puis, cliquez sur l'onglet **Arrière-plans** (②) et cliquez sur le bouton **Importer** (③).



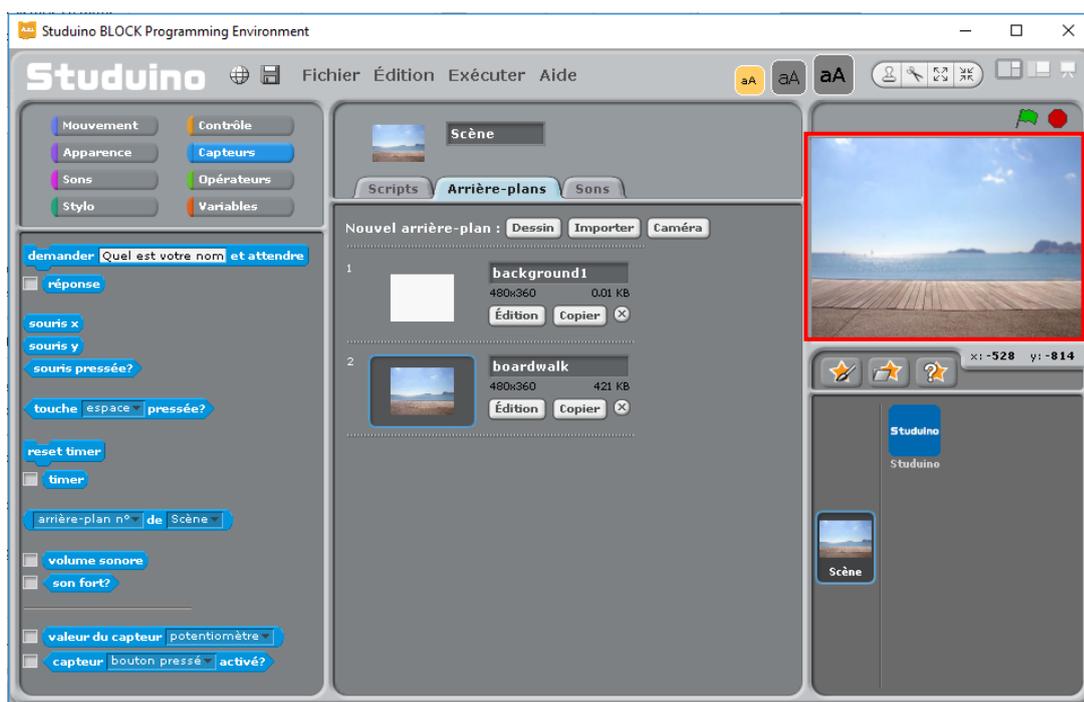
La fenêtre **Importer un arrière-plan** apparaît. Elle propose une liste de dossiers. Cliquez sur le bouton **Arrière-plans** (①), choisissez le dossier **Outdoors** (②) et cliquez sur le bouton **OK** (③).



Choisissez, dans la liste d'images, **boardtalk** (①) et cliquez sur **OK** (②).

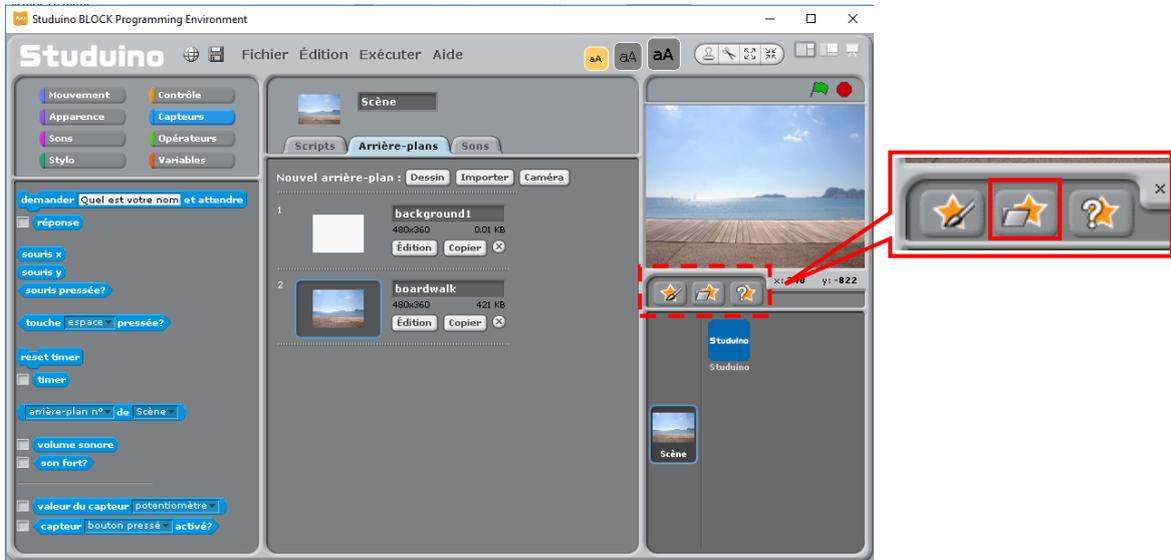


Un arrière-plan a été désormais ajouté à votre scène.

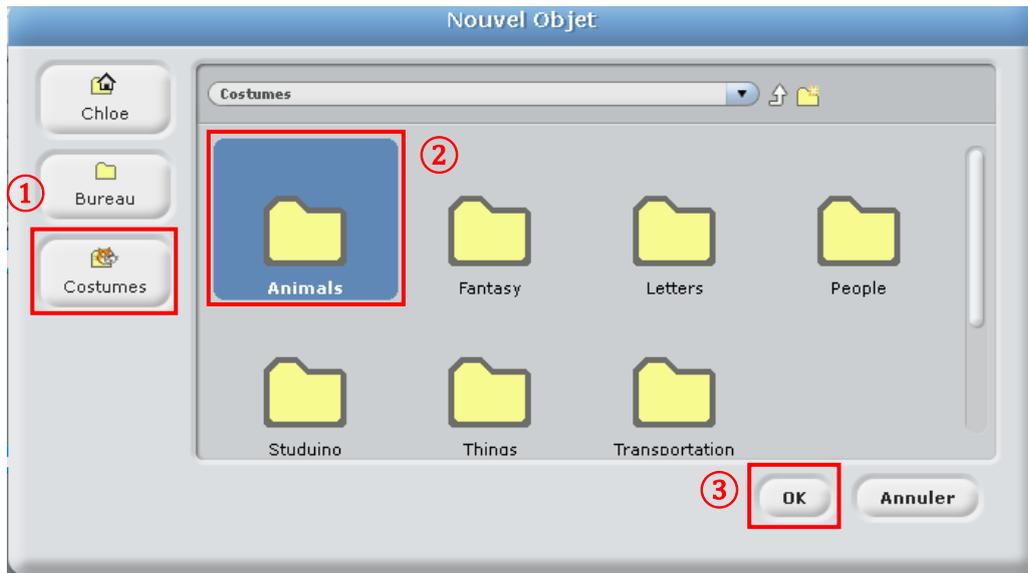


6.2.2. Ajouter des objets

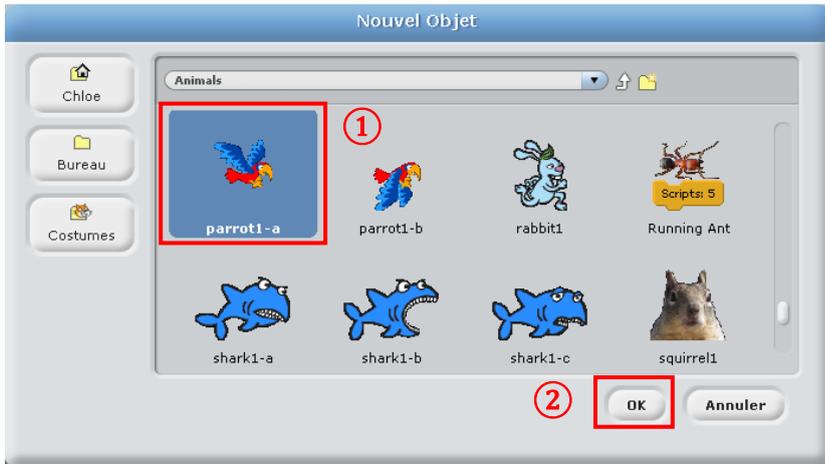
Sous la scène, cliquez sur le bouton du milieu (choisir un nouvel objet dans un dossier).



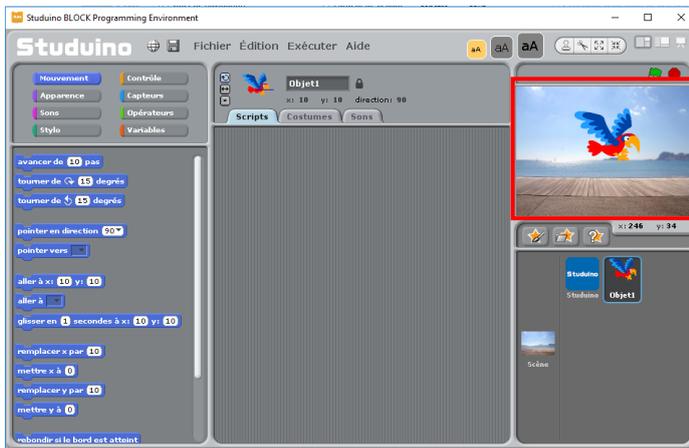
La fenêtre **Nouvel objet** apparaît. Elle propose une liste de dossiers. Cliquez sur le bouton **Costumes** (①), choisissez le dossier **Animals** (②) et cliquez sur le bouton **OK** (③).



Choisissez, dans la liste d'images, **parrot1-a** (①) et cliquez sur le bouton **OK** (②).

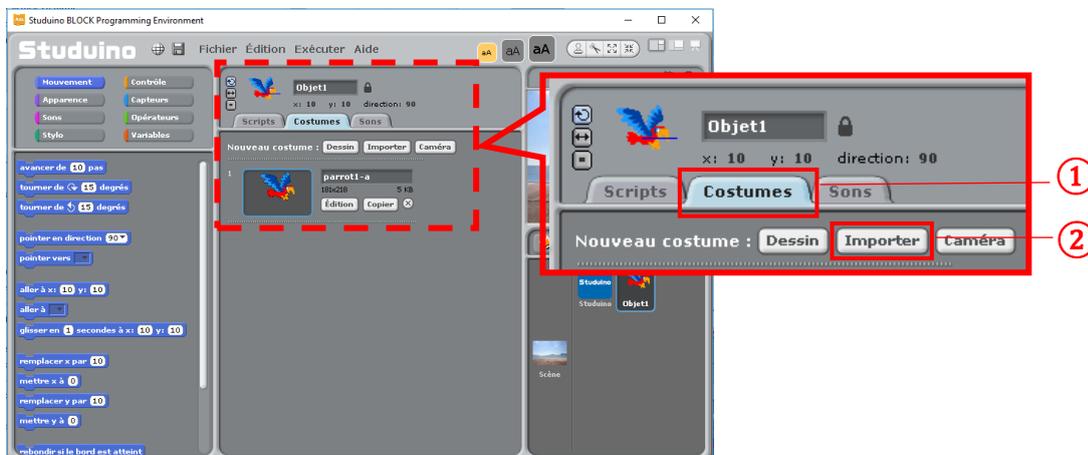


Un objet a été ajouté à votre scène.



6.2.3. Ajouter un costume

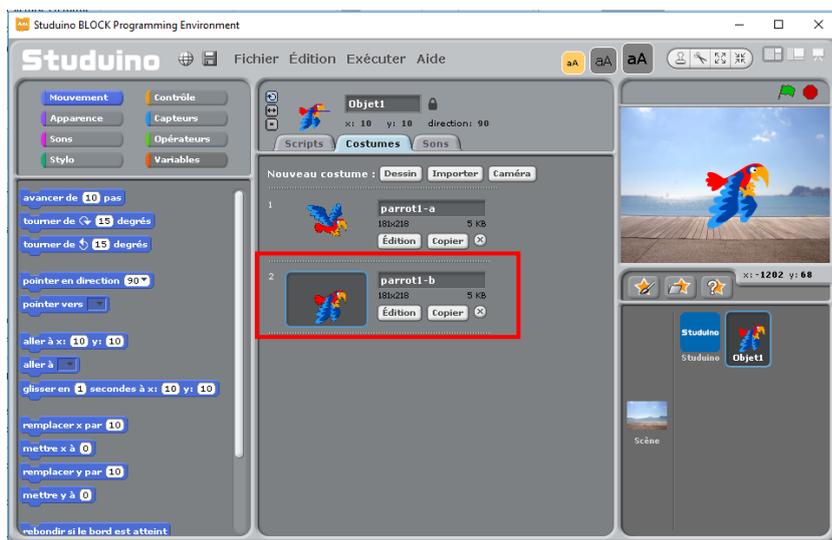
Cliquez sur l'onglet **Costumes** (①) dans le champ de script. Cliquez sur le bouton **Importer** (②).



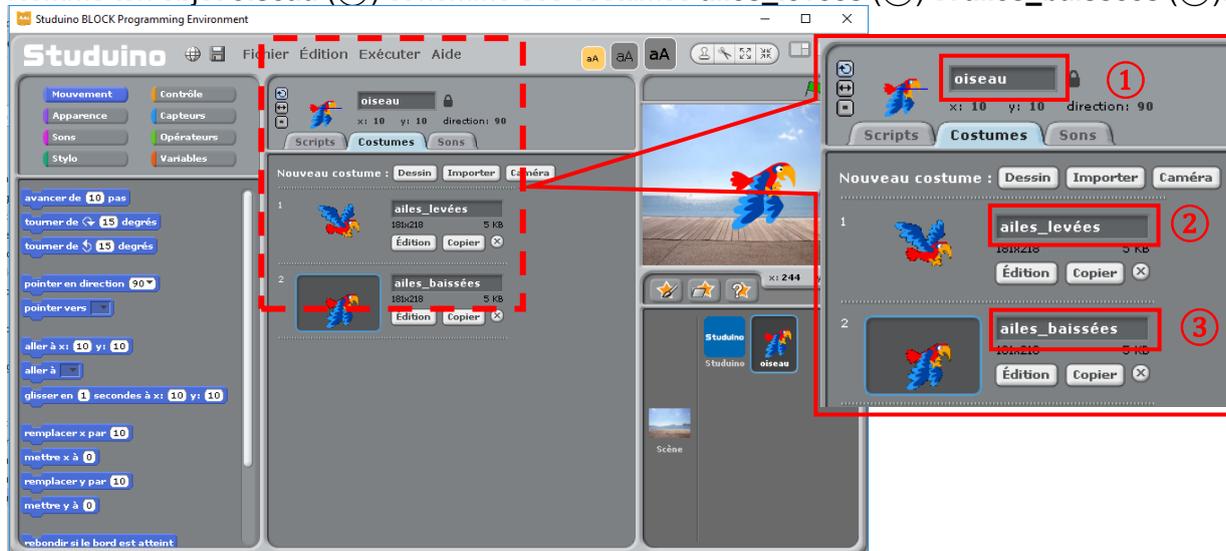
La fenêtre **Importer un costume** apparaît. Choisissez, dans la liste d'images, **parrot1-b** (①) et cliquez sur le bouton **OK** (②).



Un costume a été ajouté à l'objet.

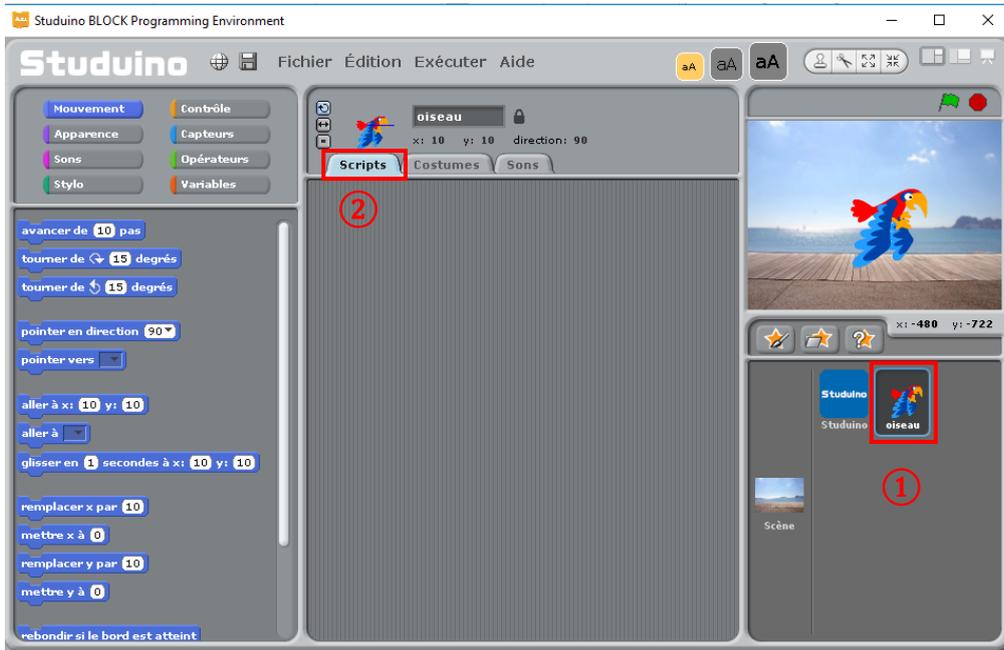


Nomme ton objet **oiseau** (①) et nomme ses costumes **ailles_levées** (②) et **ailles_baissées** (③).

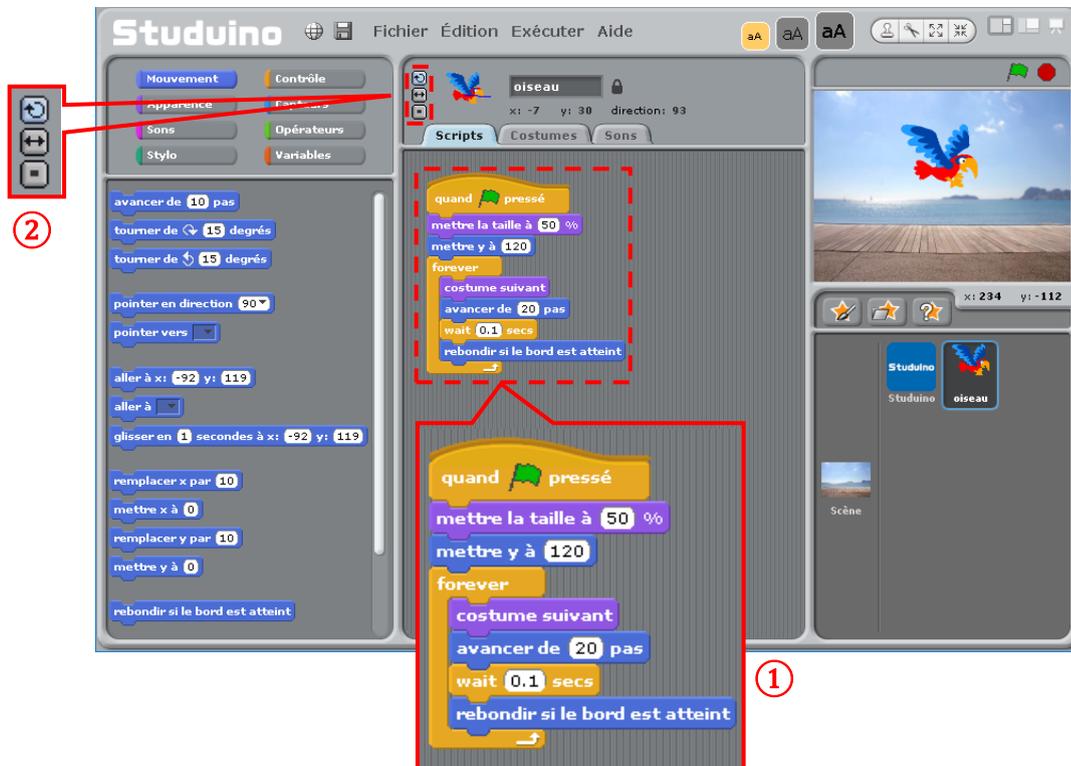


6.2.4. Programmer les objets

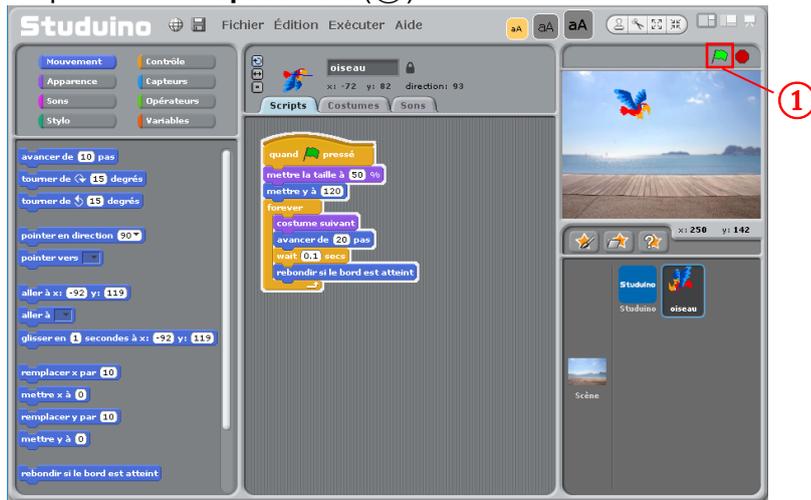
Cliquez sur l'icône oiseau (①). Puis cliquez sur l'onglet **Scripts** (②).



Reproduisez le programme montré ci-dessous (①) dans le champ de script. Cliquez sur "retournement gauche-droite uniquement" (②).

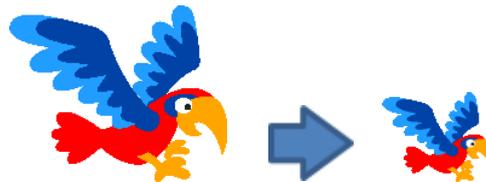
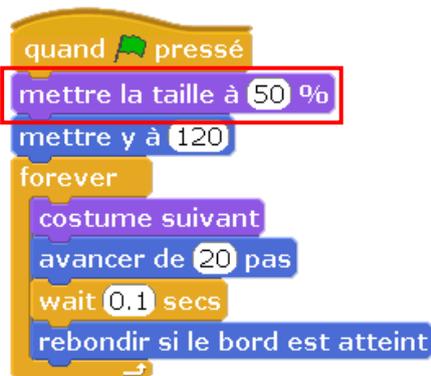


Cliquez sur le **drapeau vert** (1) : l'oiseau se mettra à voler sur la scène.

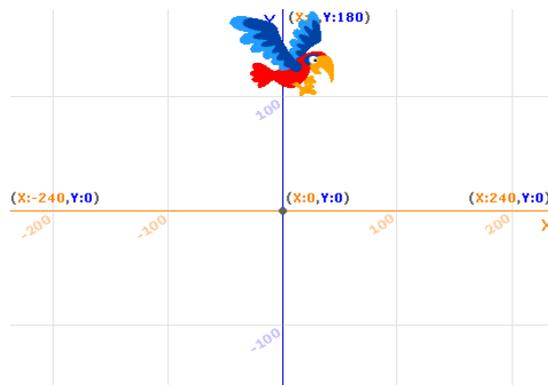


Suivez les étapes suivantes pour comprendre comment fonctionne le programme.

1. Cliquez sur le drapeau vert pour exécuter le bloc **mettre la taille à 50 %**. Ce bloc change la taille de l'objet.



2. Le bloc **mettre y à 120** s'exécutera ensuite. Ce bloc fait bouger l'objet de 120 degrés le long de l'axe Y. Toutes les scènes comportent les coordonnées montrées ci-dessous.



3. Le bloc `costume suivant` s'exécutera ensuite. Ce bloc règle le costume de l'objet sur **ails_baissées**.
 Le bloc `avancer de 20 pas` s'exécutera ensuite. Ce bloc fait avancer l'objet de 20 pas. Puis ce bloc `wait 0.1 secs`, qui met le programme en pause pendant 0.1 seconde, s'exécutera.

```

quand flag pressé
mettre la taille à 50 %
mettre y à 120
forever
  costume suivant
  avancer de 20 pas
  wait 0.1 secs
  rebondir si le bord est atteint
  
```



4. Le bloc `rebondir si le bord est atteint` fera faire demi-tour à l'objet quand il atteindra le bout.

```

quand flag pressé
mettre la taille à 50 %
mettre y à 120
forever
  costume suivant
  avancer de 20 pas
  wait 0.1 secs
  rebondir si le bord est atteint
  
```



5. Les étapes 3 et 4 se répéteront indéfiniment. Changer de façon continue le costume de l'oiseau pendant qu'il fait ses 20 pas donnera l'impression que l'oiseau vole.

```

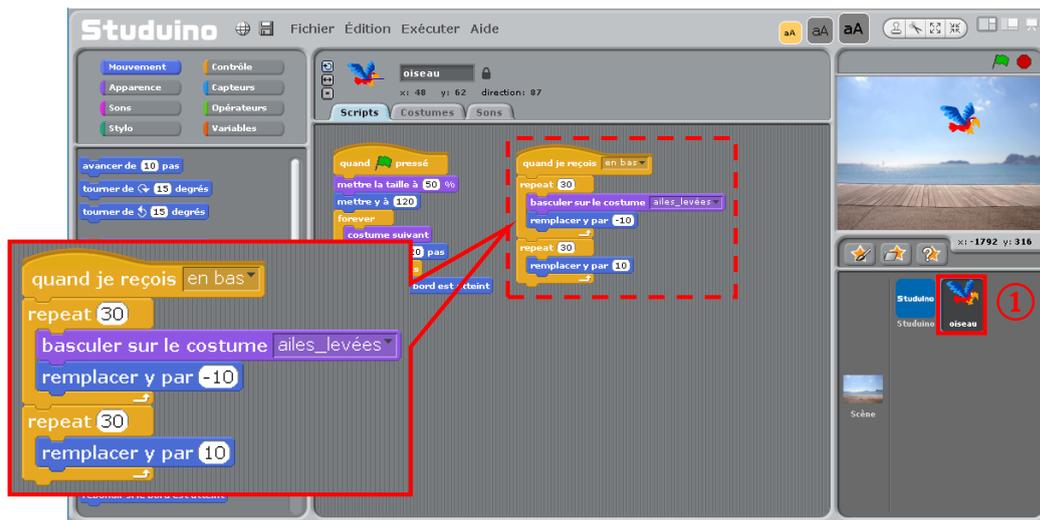
quand flag pressé
mettre la taille à 50 %
mettre y à 120
forever
  costume suivant
  avancer de 20 pas
  wait 0.1 secs
  rebondir si le bord est atteint
  
```



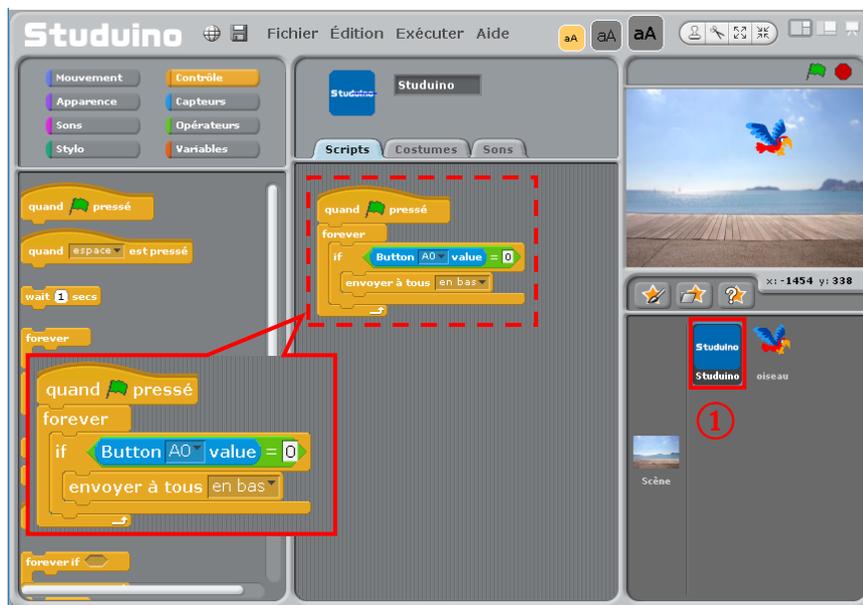
6.2.5. Utiliser les messages pour lier votre Studuino aux objets

Vous pouvez également créer des programmes qui se lient à votre Studuino en utilisant les blocs **quand je reçois** et **envoyer à tous** qui permettent à votre Studuino et aux objets oiseaux d'échanger des messages.

Cliquez sur l'icône de l'objet **oiseau** (①) dans la liste des objets, puis reproduisez le programme montré ci-dessous dans **champ de script** (②).



Cliquez sur l'icône **Studuino** (①) dans la liste des objets et reproduisez le programme, comme montré ci-dessous, dans le champ de script.



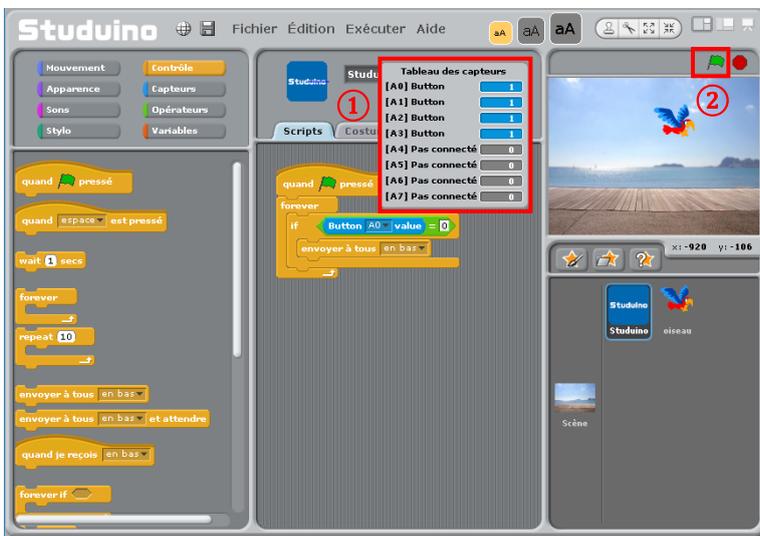
Choisissez **Connecter** dans le menu **Exécuter**.



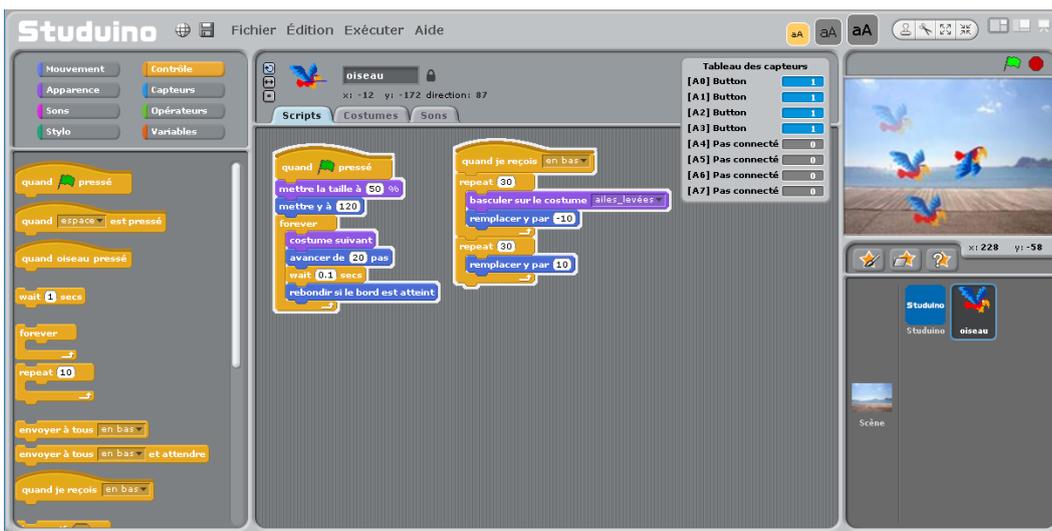
Le message ci-dessous apparaîtra à l'ouverture du mode test.



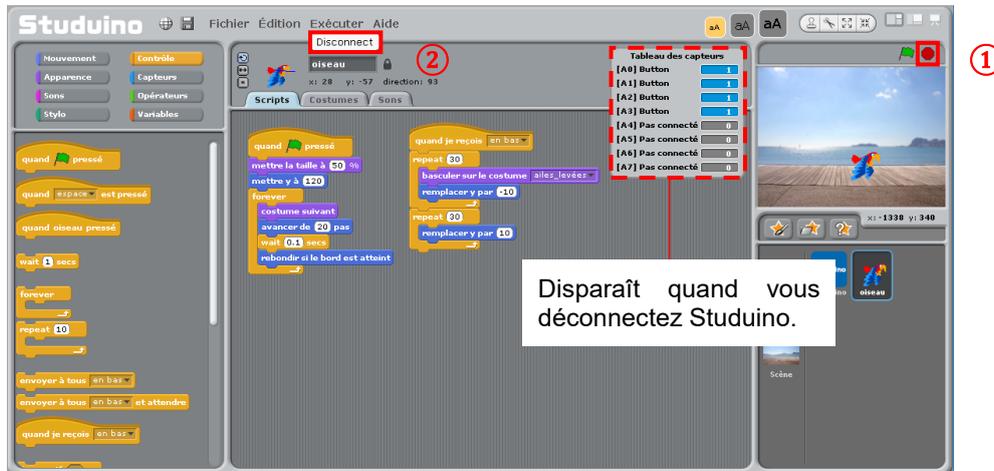
Quand le message disparaît, le **tableau des capteurs** (①) apparaît. Quand vous cliquez sur le **Drapeau vert** (②), l'oiseau volera à travers la scène.



Quand vous appuyez sur **A0** sur ton Studuino, l'oiseau se posera avant de reprendre son envol.



Appuyez sur le bouton **Stop** (①) pour arrêter l'oiseau. Choisissez **Déconnecter** (②) dans le menu **Exécuter** pour déconnecter votre Studuino de votre PC. Le **tableau des capteurs** disparaîtra.



Suivez les étapes suivantes pour voir comment fonctionne le programme.

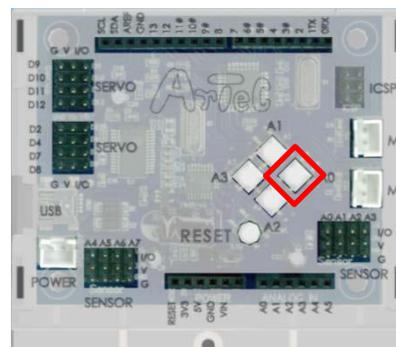
1. L'oiseau changera son costume en **ailles_baissées** quand il reçoit la commande **en bas**, en répétant 30 fois un pas de -10 sur l'axe Y.



2. Il répétera ensuite 30 fois un pas de 10 points sur l'axe Y pour retourner à sa hauteur initiale.



L'objet Studuino vérifiera **toujours** si la valeur de A0 est égale à 0 ou non (ce qui veut dire que vous avez appuyé sur le bouton). Il envoie la commande **en bas** quand vous appuyez sur A0.

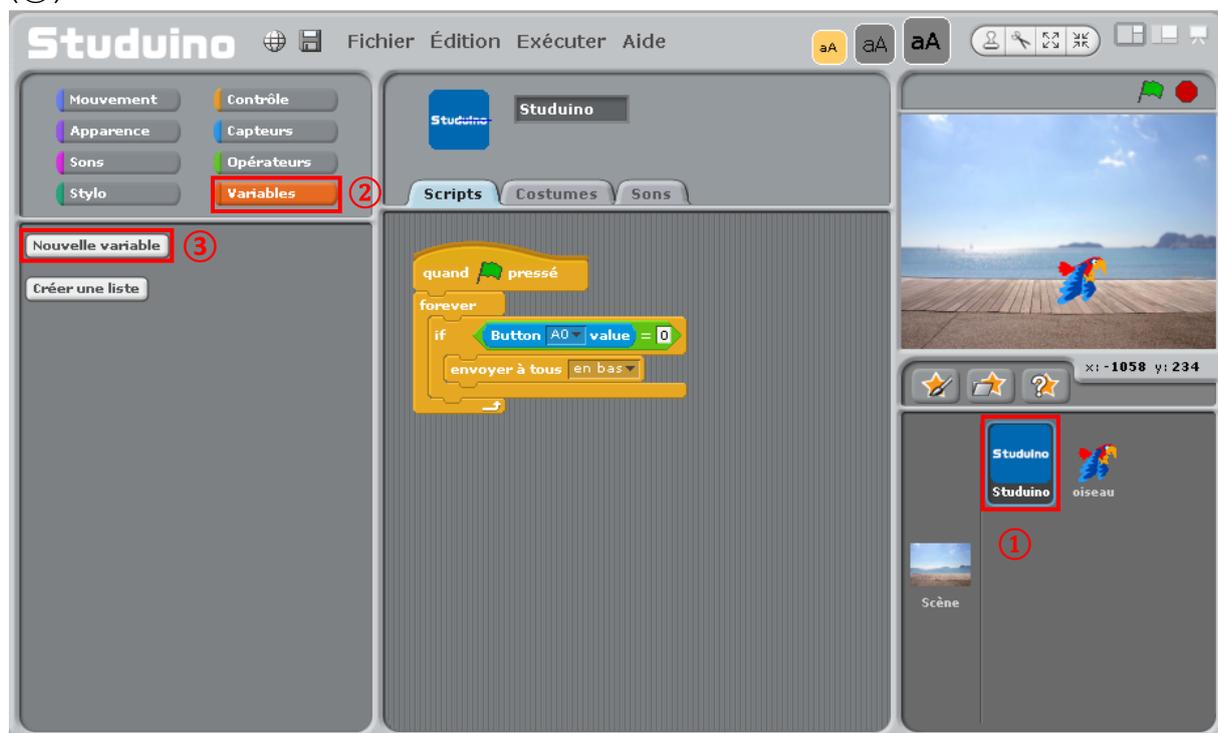


Studuino

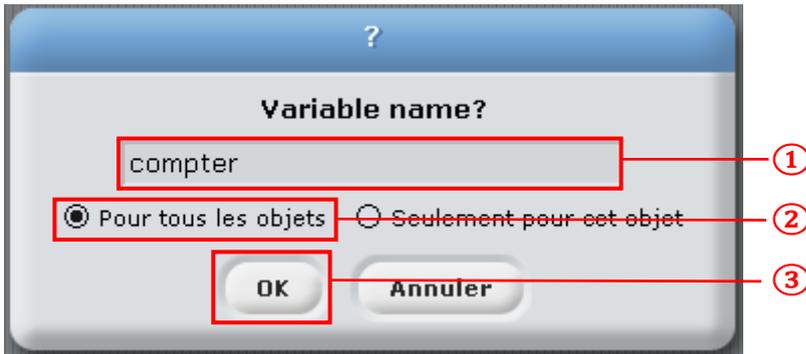
Dans cet exemple, l'objet **Studuino** envoie la commande **en bas** quand vous appuyez sur A0 et l'**oiseau** se pose et reprend son envol quand il reçoit le message. L'oiseau se posera et prendra donc son envol à chaque fois que vous appuyerez sur le bouton. Cet échange répété de messages permettra donc à votre Studuino de contrôler les objets

6.2.6. Lier votre Studuino aux variables

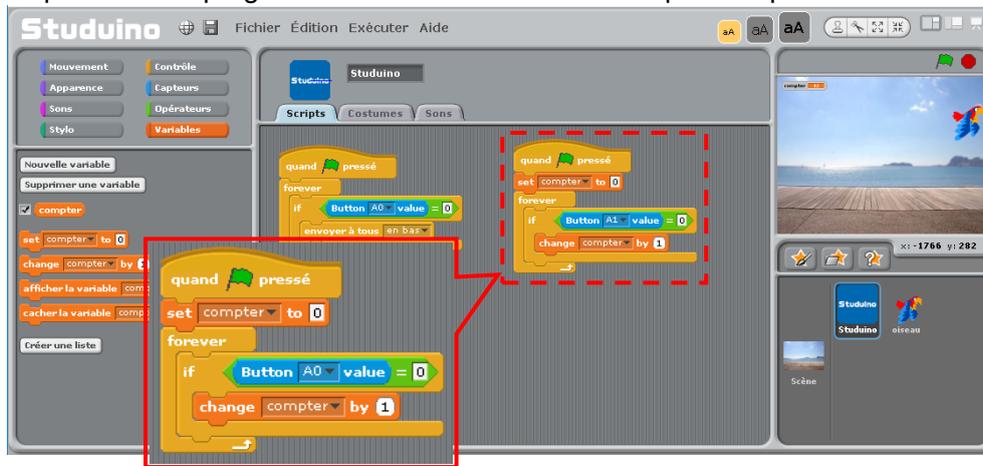
Vous pouvez aussi utiliser l'objet Studuino pour faire des variables. Cliquez sur l'icône de l'objet **Studuino** (①). Cliquez sur le bouton **Variables** (②) et cliquez sur le bouton **Nouvelle variable** (③).



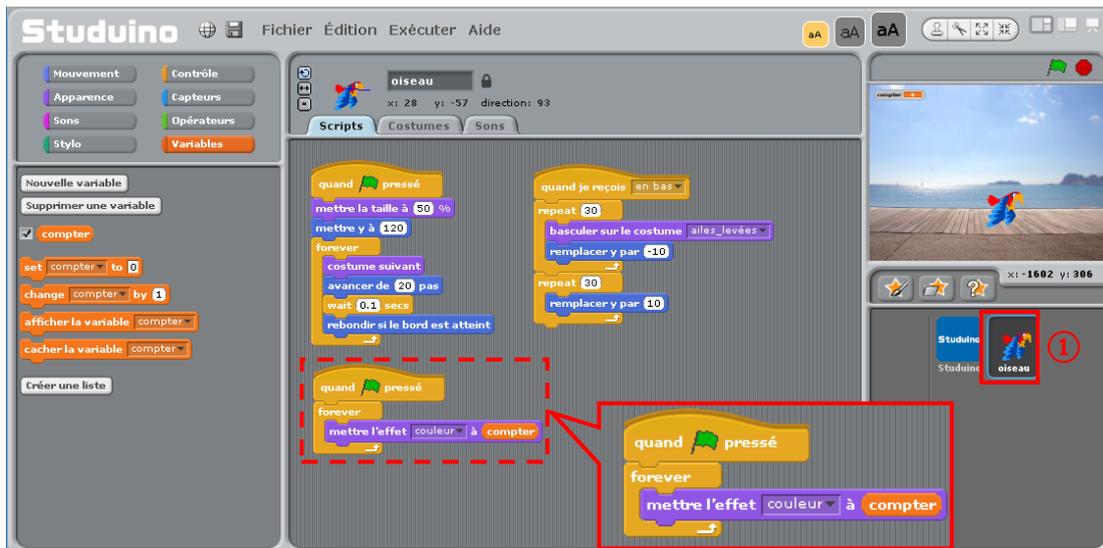
Nommez la variable **compter** (①) et choisissez **Pour tous les objets** (②). Puis cliquez sur **OK** (③).



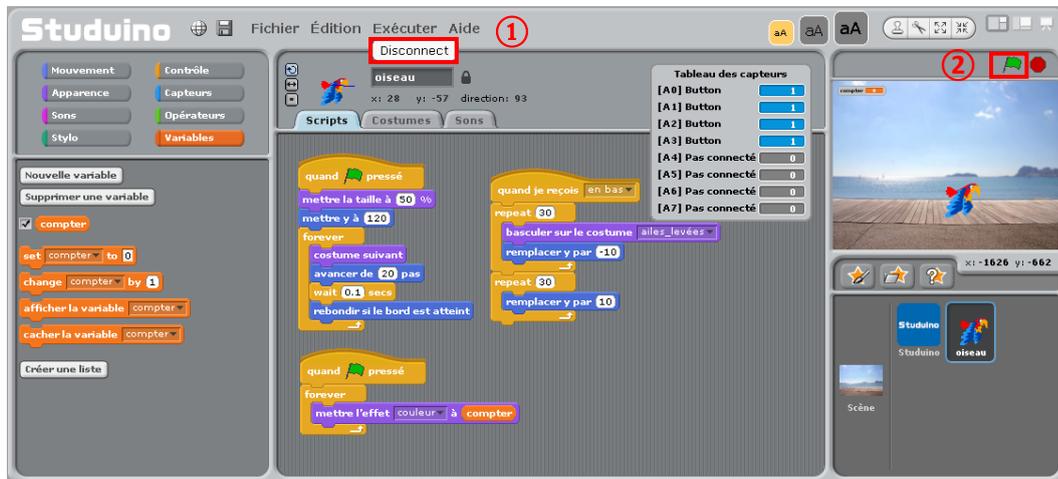
Reproduisez le programme ci-dessous dans le champ de script.



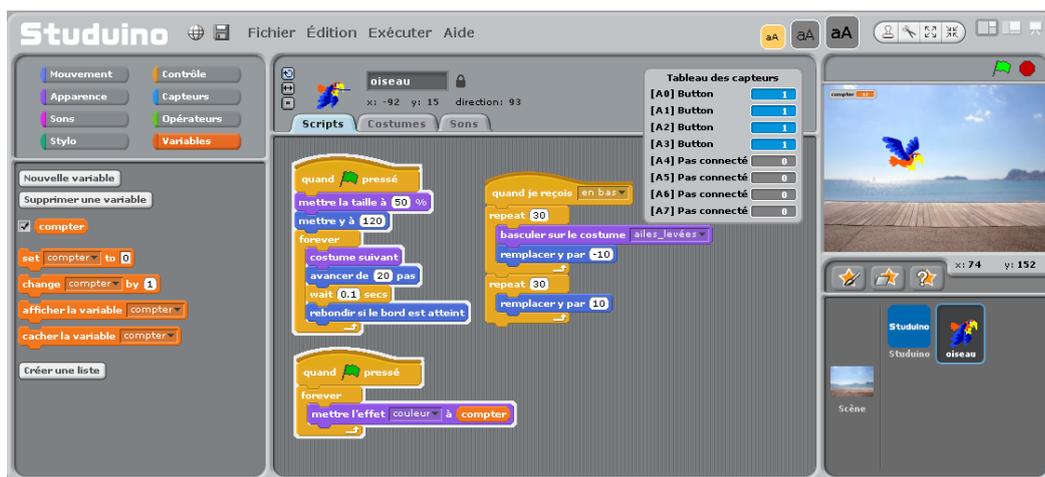
Cliquez sur l'icône de l'objet **oiseau** (①) dans la liste des objets pour ajouter le programme montré ci-dessous dans le champ de script.



Utilisez un câble USB pour connecter votre Studuino au PC et choisissez **Connecter** (①) dans le menu **Exécuter**. Dès qu'ils sont connectés, cliquez sur le **Drapeau vert** (②) pour faire voler l'oiseau dans la scène.



Quand vous appuyez sur le bouton **A1** de votre Studuino, l'oiseau change de couleur.



Arrêtez votre programme en cliquant sur le bouton **Stop** (①) sur l'image de droite). Puis déconnecter le Studuino du PC en cliquant sur **Déconnecter** dans le menu **Exécuter**. Le tableau des capteurs disparaîtra dès qu'ils seront déconnectés.

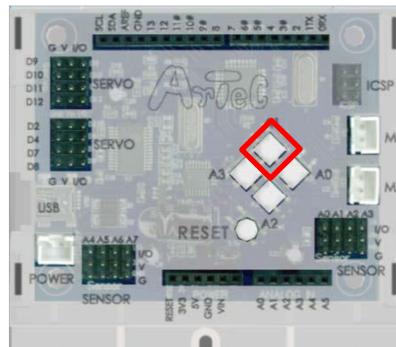
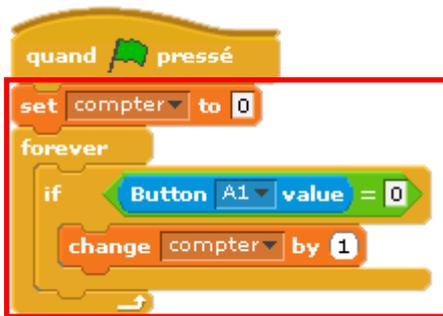


Comment le programme que vous avez ajouté fonctionne-t-il ?

La couleur de l'objet **oiseau** change selon la valeur qui se trouve dans la variable **compter**.



Dès que la variable **compter** est à **0**, l'objet **Stduino** vérifiera de façon continue (**forever**) si la valeur de A1 est de 0 ou non (elle est à 0, si vous avez appuyé sur le bouton). Un appui sur A1 ajoutera **1** à la valeur de la variable **compter**.



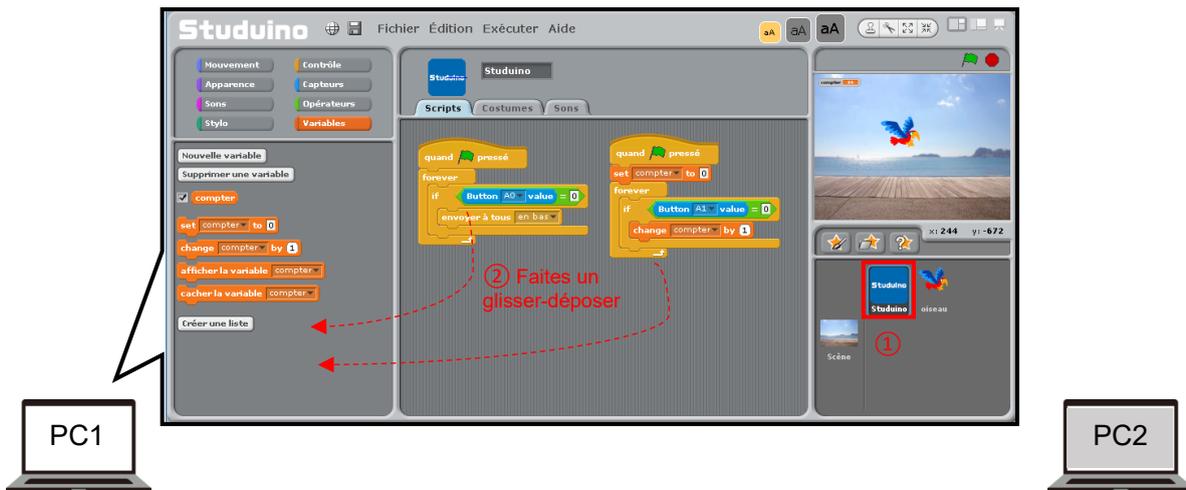
Stduino

Quand vous appuyez sur A1, l'objet **Stduino** mettra à jour la variable **compter** et l'objet **oiseau** utilisera cette variable pour changer ses couleurs à chaque fois que vous appuyez sur ce bouton. Cette utilisation des variables permet à votre Stduino de contrôler les objets.

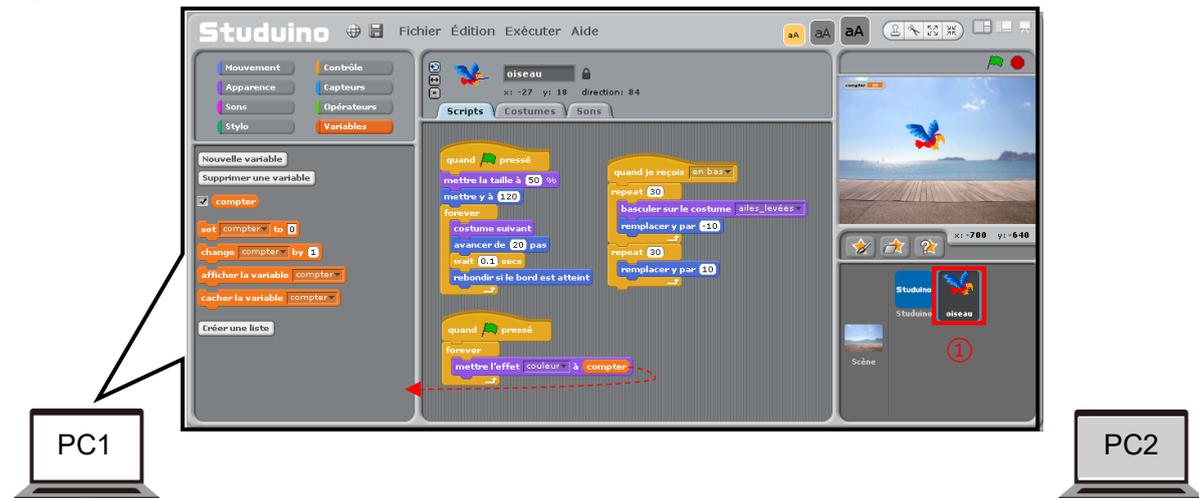
6.2.7. Utiliser les meshes pour lier votre Studuino aux objets

Ce logiciel permet d'utiliser les réseaux mesh pour lier plusieurs PC. Suivez les étapes suivantes pour utiliser ce logiciel sur deux PC. Les PC que vous utiliserez devront être connectés par un câble LAN ou un réseau Wifi. Nommons le PC que vous avez utilisé de 6.2.1 à 6.2.6 **PC1** et appelons le PC que vous allez configurer dans cette partie **PC2**.

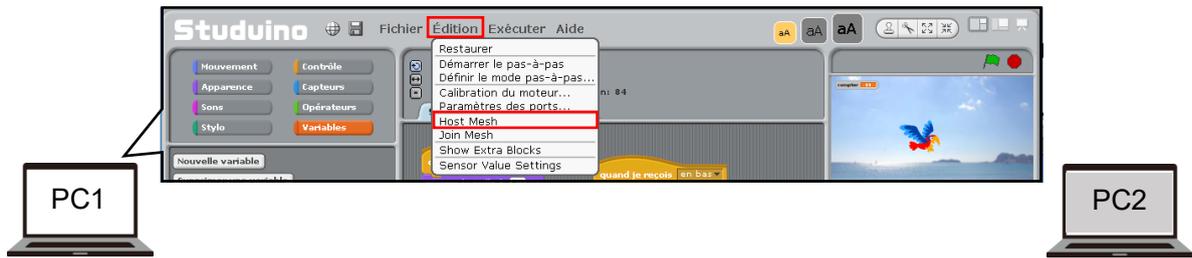
Sur le PC1, supprimez le programme de l'objet Studuino. Comme montré ci-dessous, cliquez sur l'icône de l'objet Studuino (①) et glissez les blocs **when clicked** (②) dans la palette de commandes pour les supprimer.



Vous devrez aussi cliquer sur l'icône de l'objet **oiseau** (①) et glisser le bloc de variable **compter** (②) dans la palette de commandes pour les supprimer.



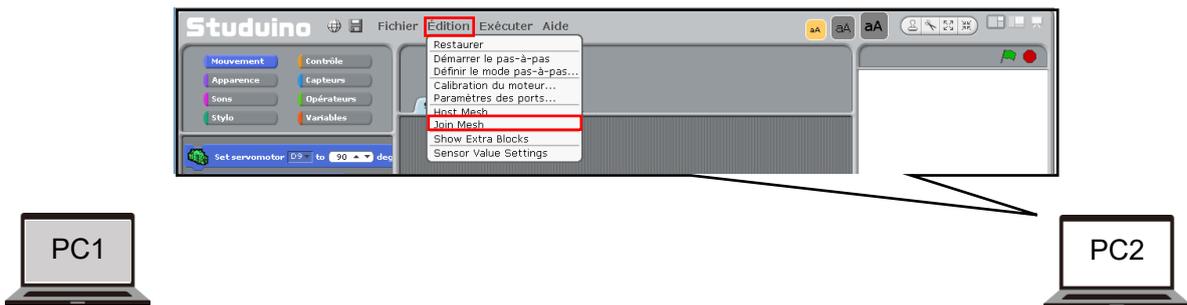
Il faudra ensuite configurer votre réseau. Dans le menu **Editer** du PC1 (①), cliquer sur **Héberger le mesh** (②).



Notez l'adresse IP (①), puis cliquez sur **OK** (②).



Dans le menu **Editer** (①) du PC2, cliquez sur **Rejoindre le mesh** (②).



Entrez l'adresse IP (①) et cliquez sur **OK** (②).

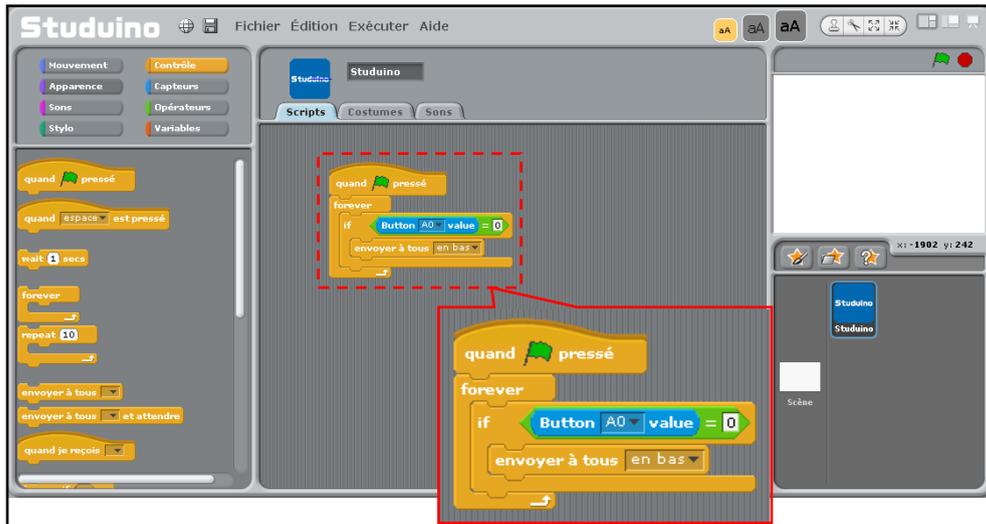


Si vous entrez une adresse IP incorrecte, vous verrez le message ci-dessous.



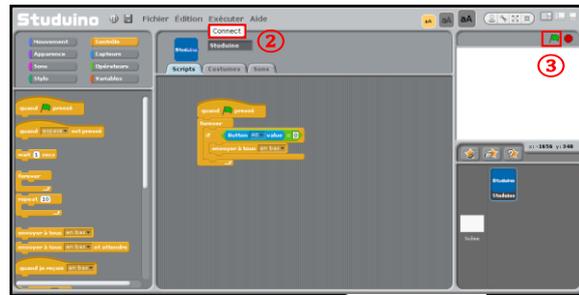
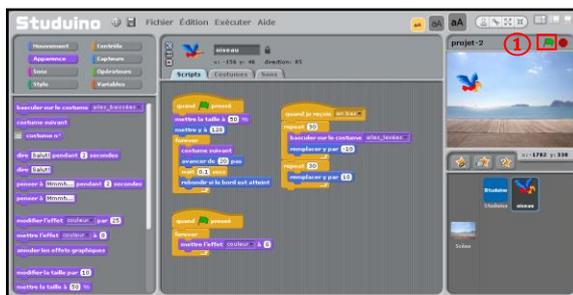
Si ce message n'apparaît pas, cela signifie que le PC2 a rejoint le mesh avec succès.

L'envoi de messages entre chaque PC permet à votre Studuino d'établir la liaison entre les objets à travers le mesh. Faisons, dans le champ de script du PC2, le programme ci-dessous.



Suivez les étapes ci-dessous pour exécuter votre programme :

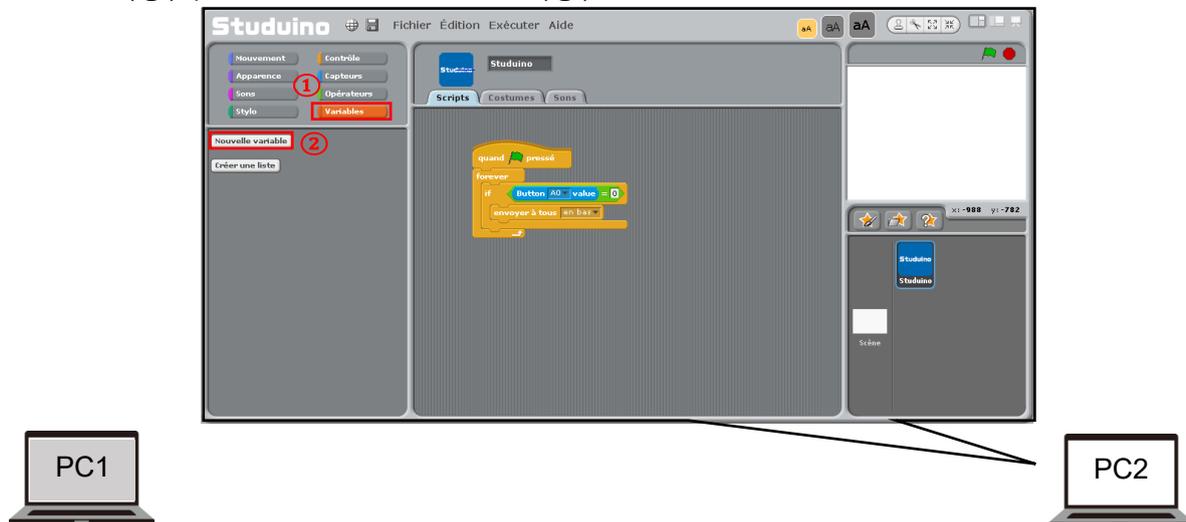
1. Cliquez sur le **Drapeau vert** (①) sur le **PC1** : votre oiseau commencera à voler sur la scène.
2. Utilisez un câble USB pour connecter votre Studuino au **PC2**.
3. Sélectionnez **Connecter** dans le menu Exécuter du **PC2**.
4. Cliquez sur le **Drapeau vert** (③) dès que le Studuino est connecté au **PC2**.



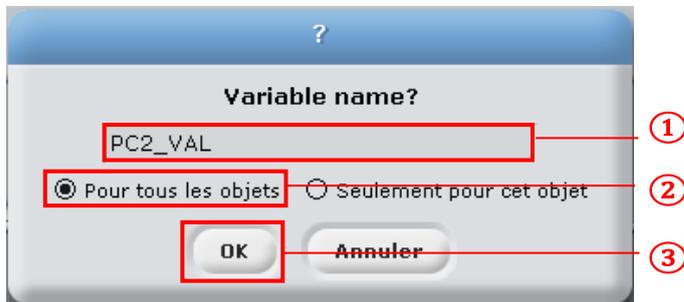
Appuyez sur le bouton **A0** de votre Studuino et voyez sur le **PC1** si l'oiseau se pose avant de reprendre son envol. Vous avez donc utilisé un Studuino connecté au PC2 pour contrôler l'oiseau sur le PC1. Cette façon d'utiliser les meshes et les messages vous permet d'utiliser un Studuino pour contrôler des objets sur un réseau.

6.2.8. Utiliser les meshes pour lier votre Studuino aux objets

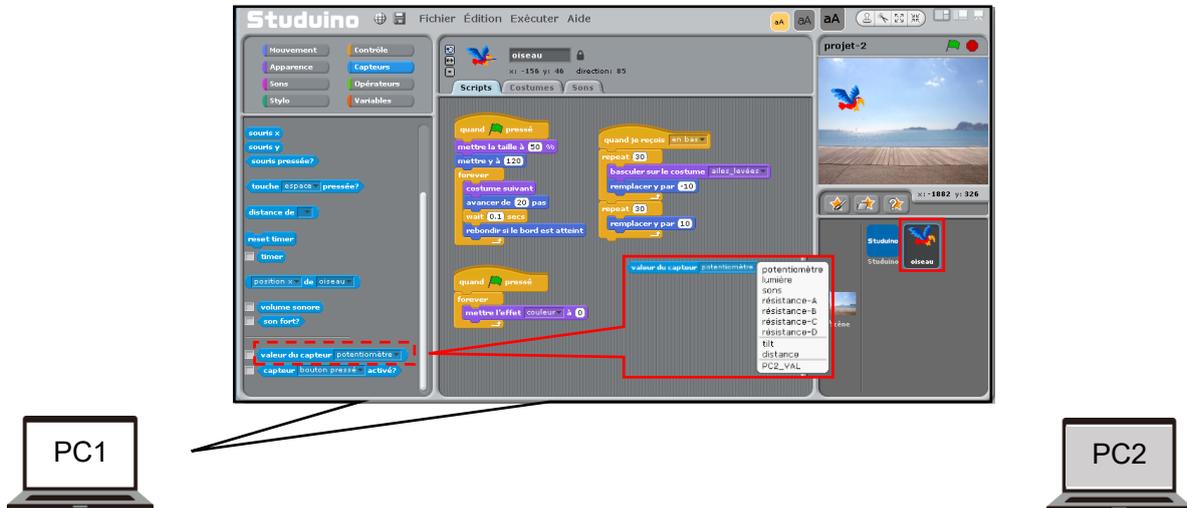
Il est aussi possible de lier votre Studuino aux objets sur un réseau en partageant des variables entre des PC. Utilisez votre objet Studuino du PC2 pour créer une variable. Cliquez sur le bouton **Variable** (①), puis sur **Nouvelle variable** (②).



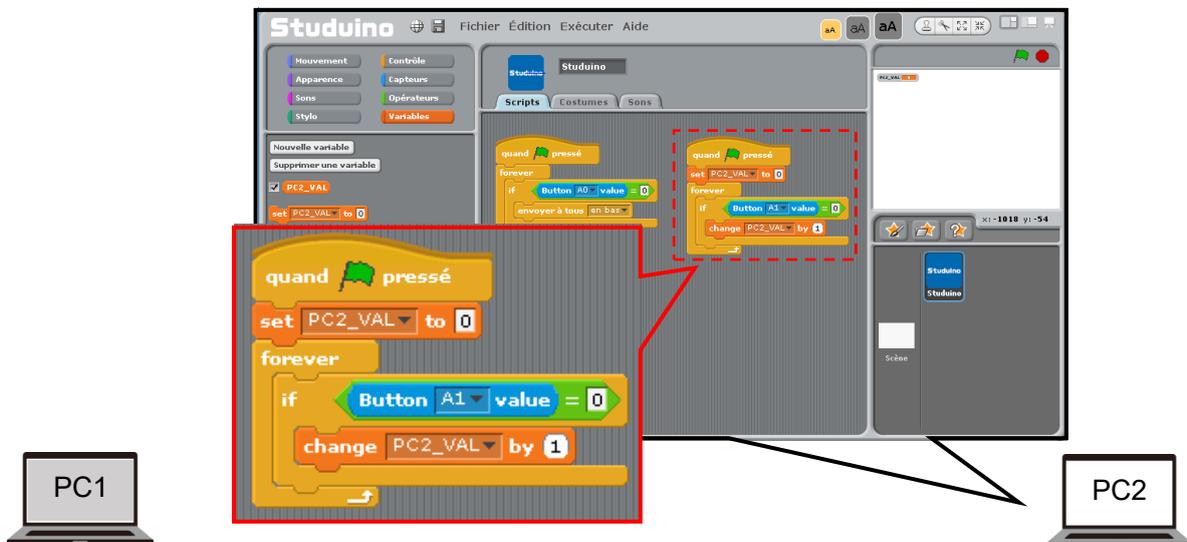
Nommez la variable **PC2_VAL** (①) et sélectionnez **Pour tous les objets** (②). Puis cliquez sur **OK** (③).



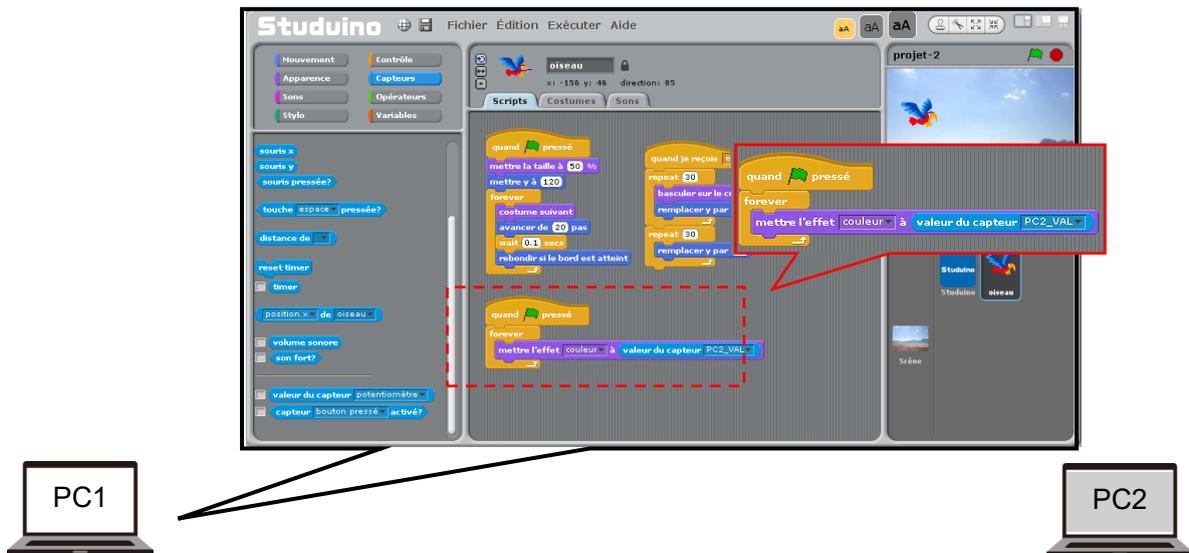
Vous pourrez voir la valeur de la variable sur le **PC2** en utilisant le bloc de l'icône **oiseau** valeur du capteur potentiomètre sur le **PC1**. Cliquez sur l'icône **oiseau** sur le **PC1**. Cliquez ensuite sur ▼ dans le bloc valeur du capteur potentiomètre et sélectionnez la variable **PC2_VAL** que vous avez créée sur le **PC2**.



Reproduisez le programme ci-dessous dans le champ de script du **PC2**.

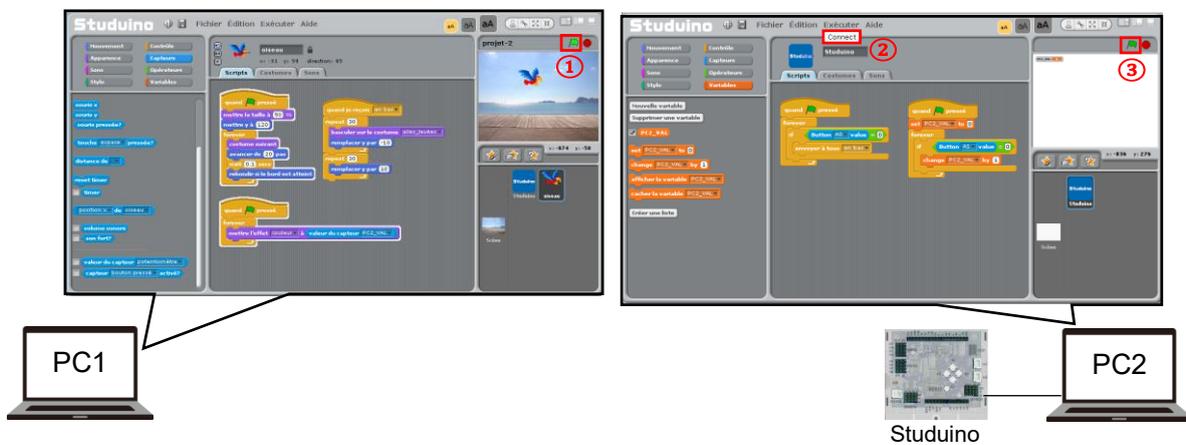


Reproduisez le programme ci-dessous dans le champ de script du **PC1**.



Suivez les étapes ci-dessous pour exécuter le programme :

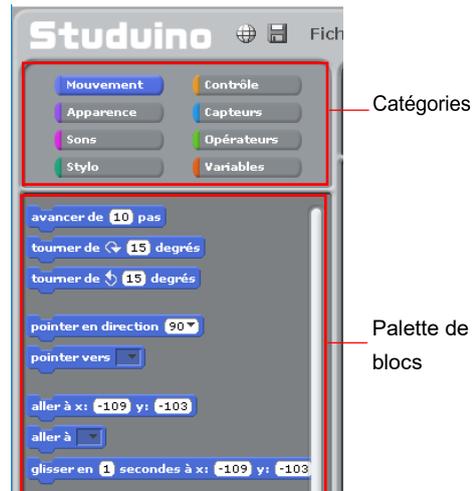
1. Cliquez sur le **Drapeau vert** (①) sur le **PC1** : l'oiseau volera à travers la scène.
2. Utilisez un câble USB pour connecter le **Studino** au **PC2**.
3. Cliquez sur **Connecter** dans le menu Exécuter du **PC2** (②).
4. Cliquez sur le **Drapeau vert** (③) dès que le Studino est connecté au **PC2**.



Quand vous appuierez sur le bouton **A1** du Studino, l'oiseau sur le PC1 changera de couleur. Le Studino connecté au PC2 contrôle l'oiseau sur le PC1. Cette façon d'utiliser les meshes et les variables vous permet d'utiliser un Studino pour contrôler des objets sur un réseau.

6.3. Blocs de l'objet Studuino

Pour programmer les scènes, les objets ou l'objet Studuino, vous devrez utiliser les blocs dans la palette de blocs. Ces blocs sont rangés dans les catégories : **Mouvement**, **Contrôle**, **Apparence**, **Capteurs**, **Sons**, **Opérateurs**, **Stylo** et **Variables**. Cliquez sur l'une des catégories pour accéder à leurs blocs. Voir ci-dessous une description des blocs que vous utiliserez pour l'objet Studuino. Lisez la documentation de Scratch ou visitez le wiki de Scratch ([https://wiki.scratch.mit.edu/wiki/Blocks_\(1.4\)](https://wiki.scratch.mit.edu/wiki/Blocks_(1.4))) pour en apprendre plus sur les blocs de la scène et des objets (appelés aussi lutins).



Catégorie	Les actions des blocs
Mouvement	Blocs qui contrôlent les moteurs, les servomoteurs, les buzzers et les LED
Contrôle	Contrôler les éléments de programmation basiques comme les conditions, les fonctions et les processus d'attente
Capteurs	Observer les valeurs des capteurs
Opérateurs	Contrôler les opérateurs arithmétiques et logiques basiques
Variables	Créer et contrôler des variables et des listes
Apparence	Les blocs d'Apparence, de Sons et de Stylo ne peuvent pas être utilisés avec l'objet Studuino.
Sons	
Stylo	

¹ [https://wiki.scratch.mit.edu/wiki/Blocks_\(1.4\)](https://wiki.scratch.mit.edu/wiki/Blocks_(1.4))

6.4. En savoir plus sur les blocs

Référez-vous au 4.2.1 pour en apprendre plus sur la programmation avec des blocs.

Cette partie explique les différents types de blocs qui se trouvent dans la palette de blocs.

□ Blocs de la palette Mouvement

Les blocs de la palette Mouvement sont utilisés pour contrôler les éléments de votre robot. Pour en savoir plus sur ces blocs, référez-vous à la partie **4.2.1 Blocs de la palette Mouvement**.

□ Blocs de la palette Contrôle

Les blocs de la palette Contrôle permettent de contrôler le déroulement de votre programme.

Bloc	Description
	Cliquez sur le drapeau vert  pour exécuter tous les blocs qui lui sont connectés.
	Cliquez sur la touche du clavier paramétrée en ① pour exécuter tous les blocs qui sont connectés à ce bloc.
	La réception du message paramétrée en ① exécutera tous les blocs connectés à ce bloc.
	Envoie le message paramétré en ①.
	Envoie le message paramétré en ① et attend que le message soit reçu.
	Empêche les blocs qui lui sont connectés de s'exécuter.
	Empêche tous les blocs de s'exécuter.

Pour en savoir plus sur ces blocs, référez-vous à la partie **4.2.1 Blocs de la palette Contrôle**.

□ Blocs de la palette Capteurs

Les blocs de la palette Capteurs sont utilisés pour récupérer les valeurs des capteurs et la saisie externe provenant d'un clavier.

Bloc	Description
	Détecte le mouvement de la souris le long de l'axe X de la scène.
	Détecte le mouvement de la souris le long de l'axe Y de la scène.
	Détecte si le bouton gauche de la souris a été cliqué. C'est vrai si le bouton est cliqué et faux s'il ne l'est pas.
	Détecte si la touche paramétrée en ① a été pressée. Ce sera vrai si la touche est pressée et faux si elle ne l'est pas.

En savoir plus sur ces blocs dans la partie **4.2.1 Blocs de la palette Capteurs**.

□ Blocs de la palette Opérateurs

La palette Opérateurs contient les blocs qui réalisent des opérations mathématiques avec les valeurs que vous entrez.

Bloc	Description
	Joint les chaînes de caractères que vous entrez en ① et ②. Ce bloc renvoie la chaîne "Hello world".
	Renvoie la lettre de la chaîne de caractère paramétrée en ② qui se trouve à la position paramétrée en ①. Ce bloc renverrait le caractère "W".
	Observe le nombre de caractères d'une chaîne de caractères entrée en ①. Ce bloc renverrait 4.
	Réalise l'opération en ① avec la valeur entrée en ②. Vous pouvez choisir entre des valeurs absolues, des racines carrées, des fonctions trigonométriques, des logarithmes et des exposants.

Pour en savoir plus sur ces blocs, référez-vous à la partie **4.2.1 Blocs de la palette Opérateurs**.

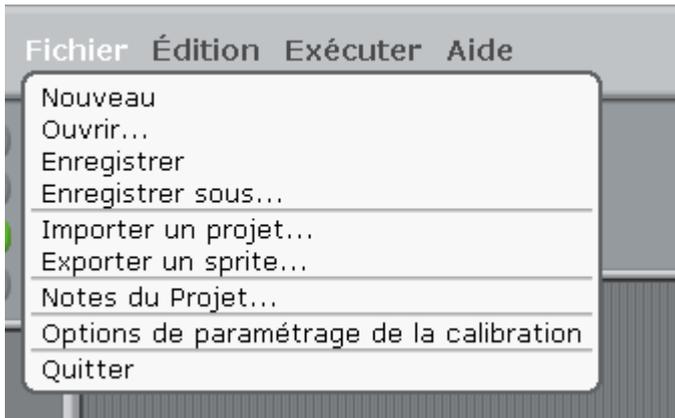
□ Blocs de la palette Variables

La palette Variables te permet de créer des variables et des listes. Pour en savoir plus sur la création de variables et de listes, référez-vous à la partie **4.2.1 Blocs de la palette Variable**.

6.5. Le menu principal

□ Le menu Fichier

Vous pouvez utiliser le menu Fichier pour sauvegarder et charger des projets.



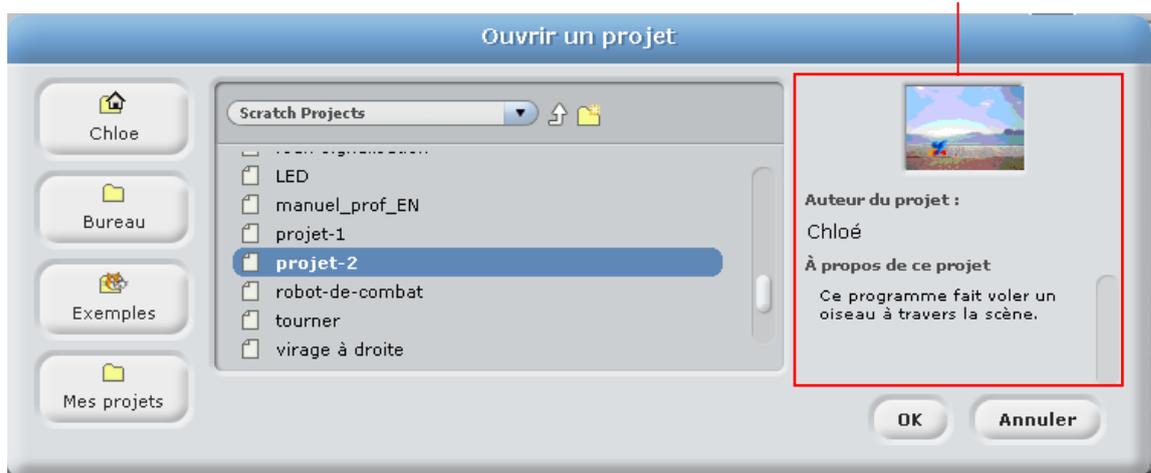
- **Nouveau**

Démarrer un nouveau projet.

- **Ouvrir...**

Dans la fenêtre Ouvrir un projet, choisir un projet à ouvrir. Vous y verrez la capture d'écran, le nom du projet et du créateur du projet sélectionné dans la fenêtre. Vous pouvez aussi ouvrir les fichiers **.bpd** créés dans l'environnement de programmation par bloc et les fichiers **.sb** créés dans Scratch 1.4 en plus des projets créés dans ce logiciel.

Aperçu du détail du projet.

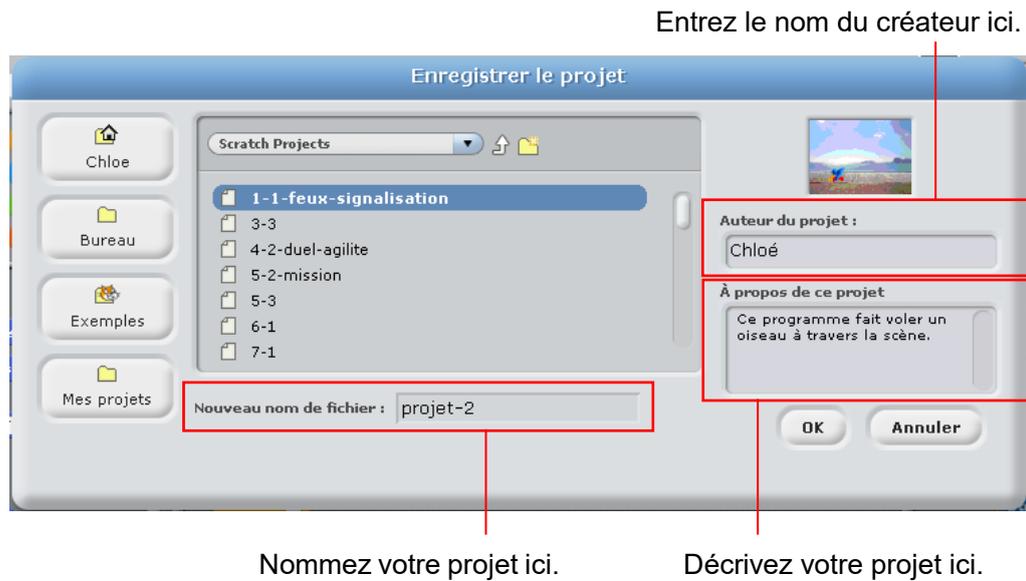


- **Enregistrer**

Enregistre le projet en cours. Une fenêtre **Enregistrer le projet** apparaît quand vous sauvegardez le fichier pour la première fois.

- **Enregistrer sous...**

Enregistre le programme sous un nom spécifique. Une fenêtre **Enregistrer le projet** s'ouvrira. Nommez votre projet dans le champ **Nouveau nom de fichier**. Vous pouvez aussi nommer le créateur du projet et donner une description.

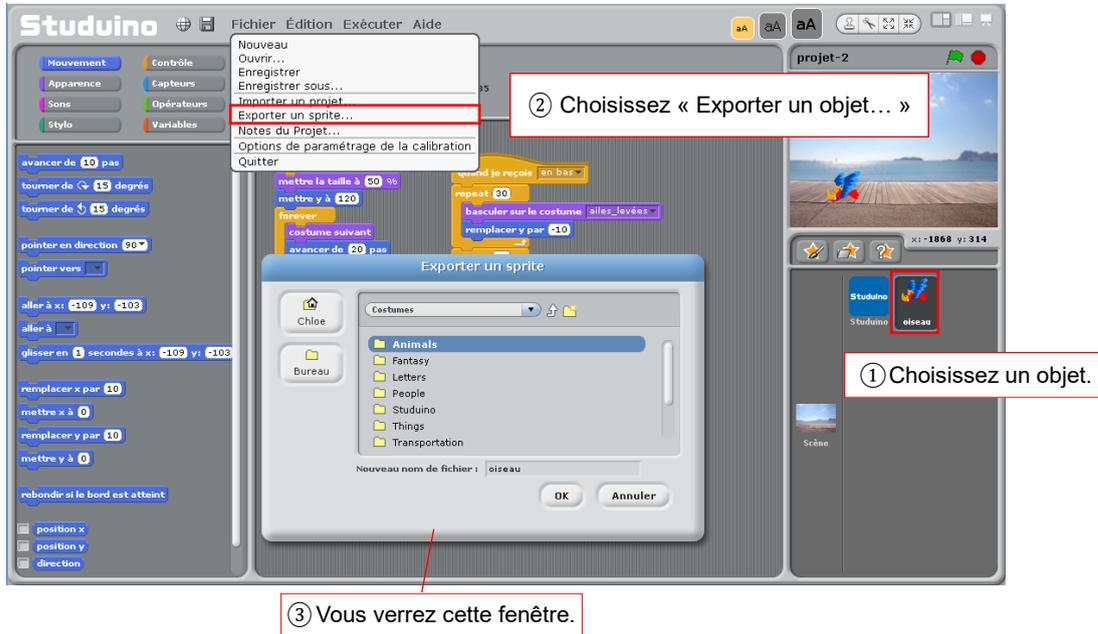


- **Importer un projet...**

Chargez les scripts du fichier d'un précédent projet. Référez-vous au **Menu Fichier** dans la partie **6.5 Le menu principal** pour plus de détails. Vous pouvez également charger les fichiers **.bpd** créés dans l'environnement de programmation par blocs et les fichiers **.sb** créés dans Scratch 1.4 en plus des scripts créés dans ce logiciel.

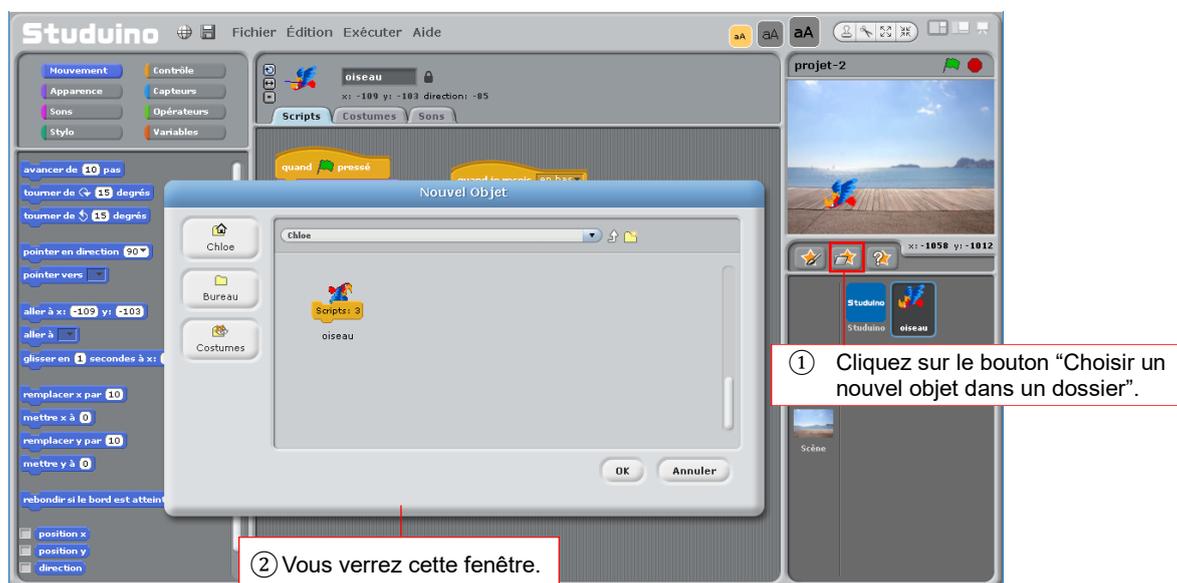
- **Exporter un objet...**

Exporte un objet de votre choix. Choisissez l'icône de l'objet que vous voulez exporter (①). Puis cliquez sur **Exporter un objet...**(②) depuis le menu **Fichier**. Une fenêtre **Exporter un objet...** apparaîtra (③). Entrez le nom de votre objet dans le champ **Nouveau nom de fichier**.



Vous pouvez exporter les objets habituels, mais aussi les objets Studuino. Tous les objets Studuino que vous exporterez seront enregistrés avec l'extension **.studuinosprite**. Tous les objets habituels que vous exporterez seront enregistrés avec l'extension **.sprite**.

Vous pouvez charger n'importe quel objet que vous avez exporté en cliquant sur le bouton **Choisir un nouvel objet dans un dossier**. Vous pouvez aussi charger n'importe quel objet fait sur Scratch. Quand vous chargez un objet Studuino, le programme de cet objet sera ajouté à l'objet Studuino déjà présent dans votre projet.



- **Notes du projet...**

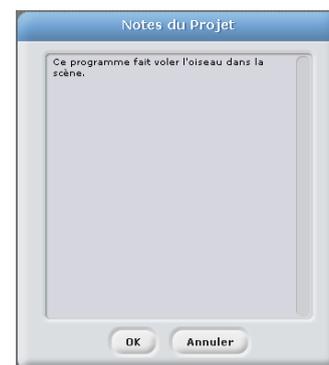
Vous permet d'ajouter ou d'éditer des notes sur votre projet (dans la fenêtre montrée ci-contre). Ce que vous y notez apparaîtra dans la section "À propos de ce projet" dans la fenêtre de sauvegarde du projet.

- **Options de paramétrage de la calibration**

Choisissez où sauvegarder le paramétrage de calibration du moteur. Pour en savoir plus, référez-vous au **Menu Fichier** dans la partie **6.5 Le menu principal**.

- **Quitter**

Fermer le logiciel.



- **Le menu Édition**

Le menu Édition permet de modifier votre programme.



- **Restaurer**

Restaure un bloc qui a été supprimé.

- **Démarrer/arrêter le pas-à-pas**

Démarre ou arrête le mode pas-à-pas. Pour plus de détails, référez-vous au **Menu Édition** dans la partie **6.5 Le menu principal**.

- **Définir le mode pas-à-pas...**

Change la vitesse à laquelle se déroule le pas-à-pas. Pour plus de détails, référez-vous au **Menu Édition** dans la partie **6.5 Le menu principal**.

- **Calibration du moteur...**

Sert à ajuster les angles de vos servomoteurs ou la vitesse de vos moteurs et à enregistrer les paramètres dans un fichier. Pour plus de détails, référez-vous au **Menu Édition** dans la partie **6.5 Le menu principal**.

- **Paramètres des ports**

Permettent de paramétrer les éléments connectés à votre Studuino dans l'environnement de programmation par icône. Pour plus de détails, référez-vous au **Menu Édition** dans la partie **6.5 Le menu principal**.

- **Héberger le mesh**

Démarre l'hébergement d'un réseau.

- **Rejoindre le mesh**

Se connecter à un réseau sur un autre PC.

- **Montrer l'adresse IP**

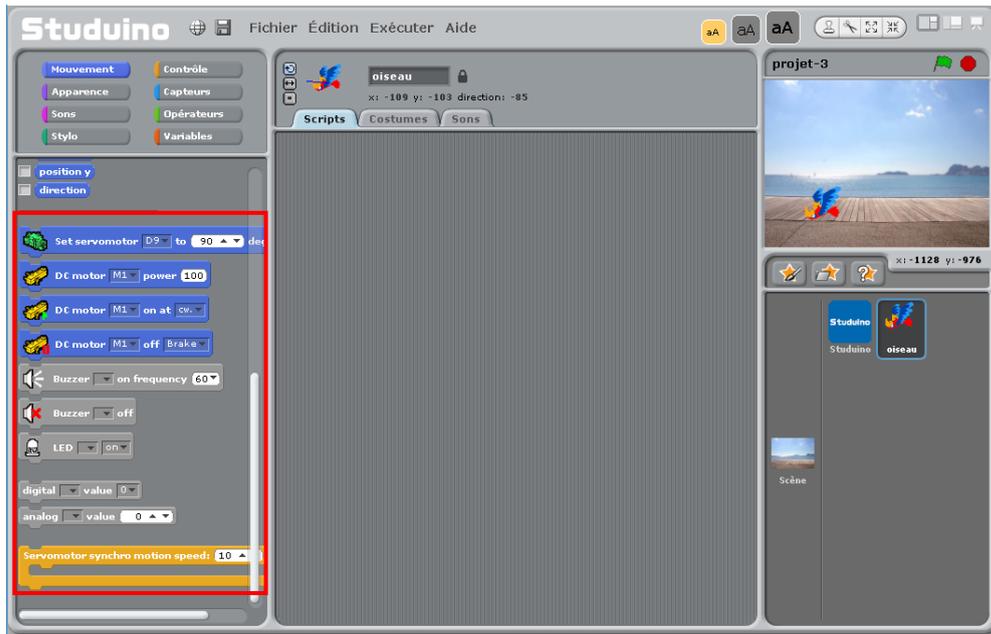
Vous le verrez seulement une fois connecté à un réseau. Il montre l'adresse IP de votre PC.

- **Arrêter l'hébergement du mesh**

Vous le verrez seulement une fois connecté à un réseau. Il déconnecte votre PC du réseau.

- **Montrer/Cacher les blocs Studuino**

Montrer ou cacher les blocs pour les servomoteurs, les moteurs, les buzzers et les LED dans la palette Mouvement de l'objet.



Comme montré ci-dessous, vous pouvez aussi utiliser les blocs **valeur du capteur potentiomètre** et **valeur du capteur potentiomètre** pour récupérer les valeurs des capteurs connectés à votre Studuino.

valeur du capteur potentiomètre	<ul style="list-style-type: none"> [A4/A5] Studuino acceleration (X) [A4/A5] Studuino acceleration (Y) [A4/A5] Studuino acceleration (Z) [A6] Studuino IR Photoreflector [A7] Studuino light potentiomètre lumière sons résistance-A résistance-B résistance-C résistance-D tilt distance
--	---

capteur bouton pressé	<ul style="list-style-type: none"> [A0] Studuino button pressed [A1] Studuino button pressed [A2] Studuino button pressed [A3] Studuino button pressed bouton pressé A connecté B connecté C connecté D connecté
------------------------------	---

□ **Le menu Exécuter**

Le menu Exécuter vous permet d'établir la liaison avec votre Studuino pendant l'élaboration de votre programme.



● **Connecter/Déconnecter**

Permet de communiquer avec votre Studuino et de contrôler ses éléments en temps réel. Activer ou désactiver cette fonctionnalité en cliquant sur **Connecter** ou **Déconnecter** dans le menu Exécuter. Assurez-vous que votre PC et votre Studuino sont connectés avant de choisir Connecter.

Lors de la connexion, vous verrez le message ci-dessous.



Ne déconnectez pas votre Studuino du PC durant le transfert du programme.

□ **Menu Aide**

Le menu Aide contient des informations sur l'environnement de programmation par blocs.



● **L'environnement de programmation par blocs Studuino**

À propos de l'environnement de programmation par blocs.

● **FAQ**

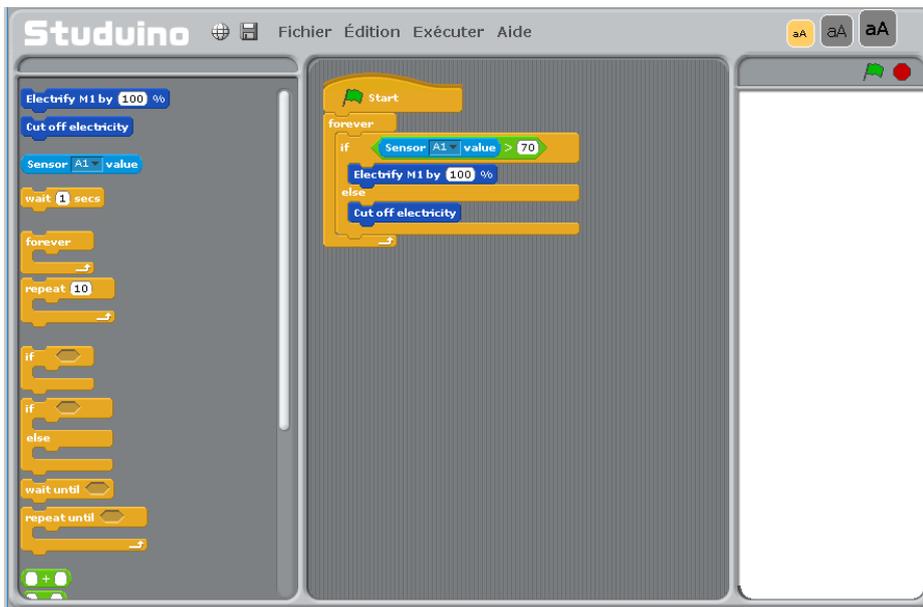
Visitez la page FAQ sur le site web de Studuino.

6.6. Menus contextuels

Référez-vous à la partie **4.6** du manuel.

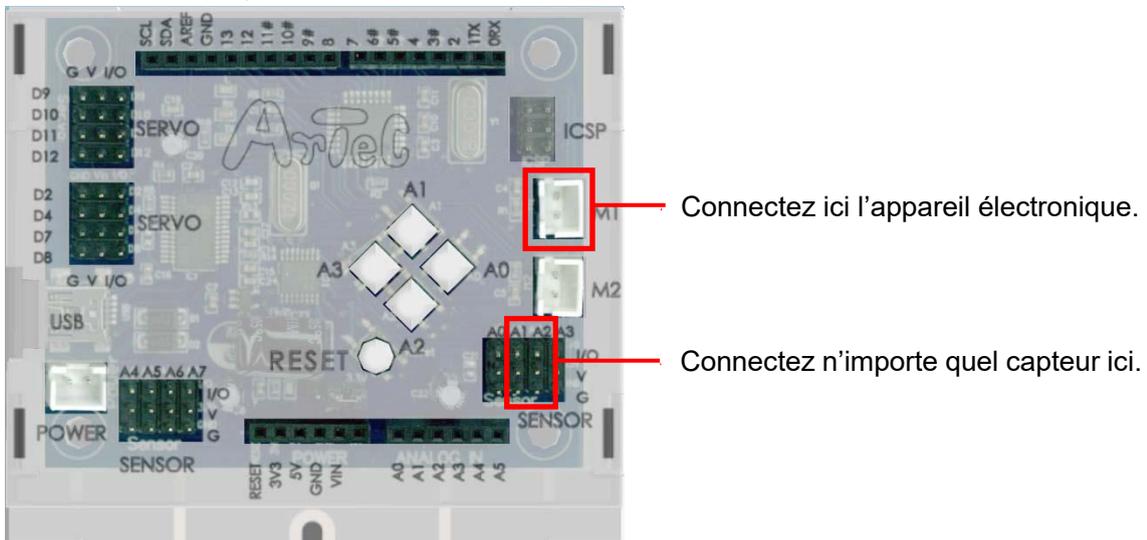
7. Expériences

7.1. Aperçu et fonctionnalités

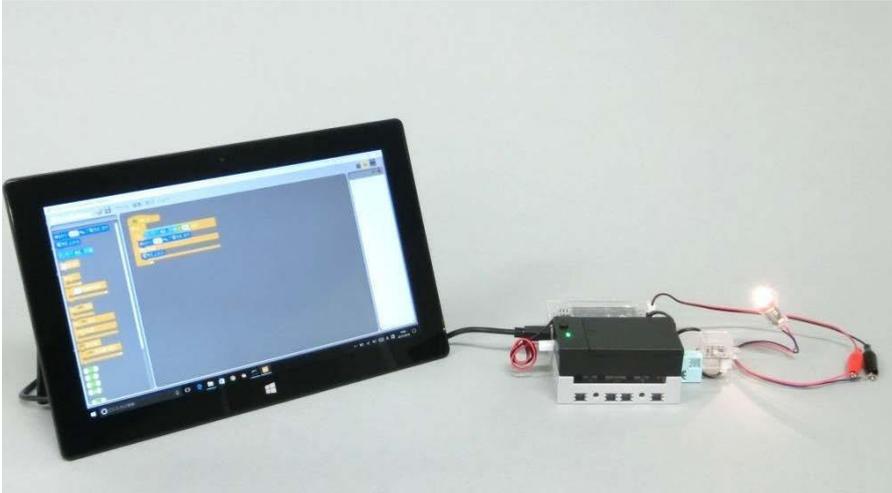


La partie Expériences intégrée dans l'environnement de programmation par blocs vous permet de connecter un fil conducteur de test (produit 151097, fil conducteur de test Alligator pour Studuino) sur le connecteur M1 de votre Studuino et d'utiliser les valeurs de vos capteurs pour changer la quantité d'électricité qui alimente les appareils (exemple : mini ampoules électriques qu'on trouve dans le commerce, moteurs, buzzers ou LED).

Cette version du logiciel utilise seulement les connecteurs A1, A2 et M1 du Studuino.



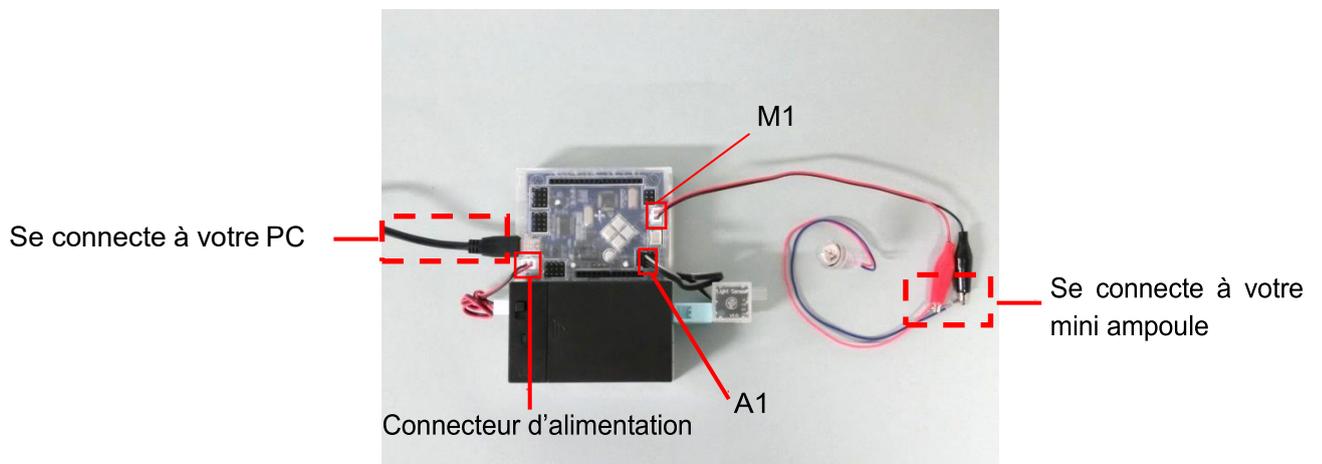
7.2. Programmation



Nous allons programmer ici un capteur de lumière qui allume et éteint une mini ampoule électrique.

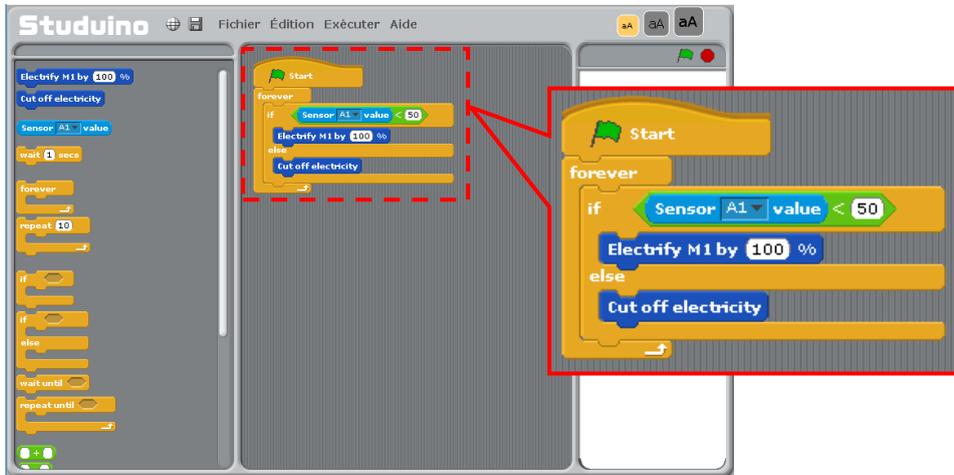
7.2.1. Paramétrage de Studuino

Branchez votre capteur de lumière sur le connecteur A1 de votre Studuino. Branchez le fil conducteur de test sur M1, connectez-le à votre mini ampoule et branchez votre batterie sur le connecteur d'alimentation. Connectez votre Studuino à votre PC avec un câble USB.

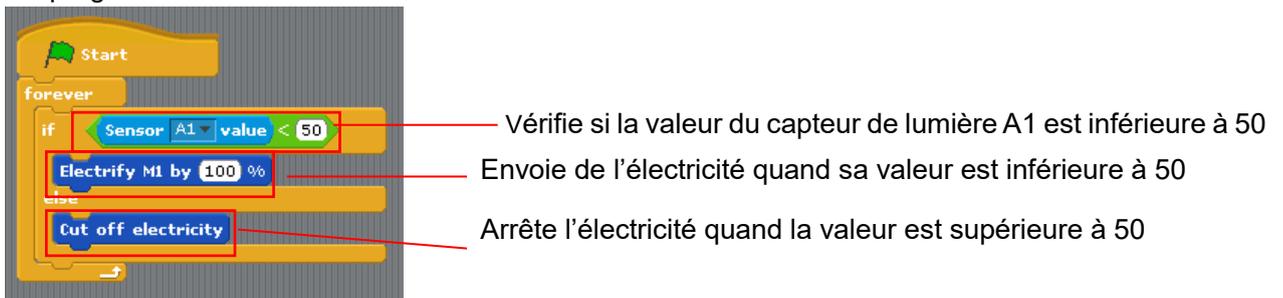


7.2.2. Programmation

Reproduisez le programme ci-dessous dans le champ de script.



Ce programme fera les actions suivantes :



7.2.3. Observer le programme en mode test

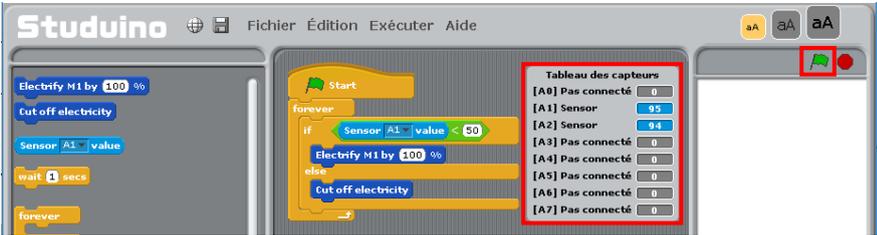
Avec le mode test, vous pouvez voir comment fonctionne votre programme en temps réel. Cliquez sur le menu **Exécuter** (1), puis sur **Activer le test** (2) pour démarrer le mode test.



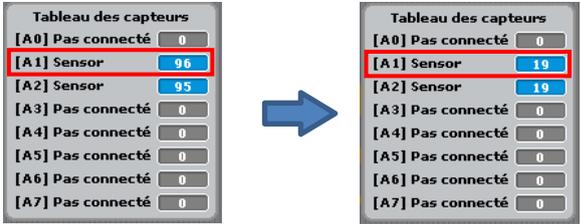
Le message ci-dessous apparaîtra quand vous ouvrirez le mode test.



Le message disparaît et le **tableau des capteurs** apparaît (1). Cliquez sur le **Drapeau vert** (2) et allumez la batterie. Les blocs en cours d'exécution sont surlignés en jaune.



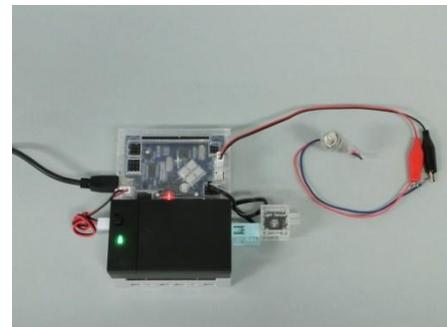
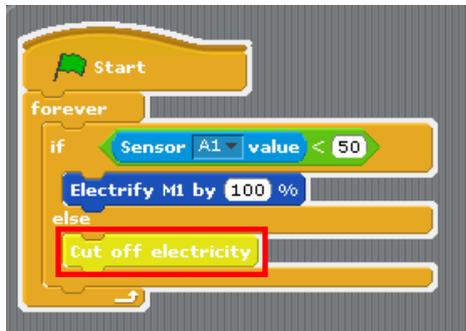
Le tableau des capteurs montre les valeurs de tous les capteurs connectés au Studuino. Couvrez de votre main le capteur de lumière branché sur A1 pour bloquer la lumière. La valeur du capteur [A1] devrait être plus petite.



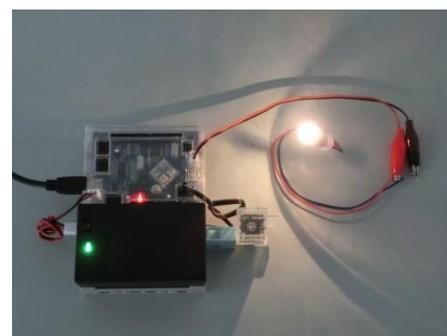
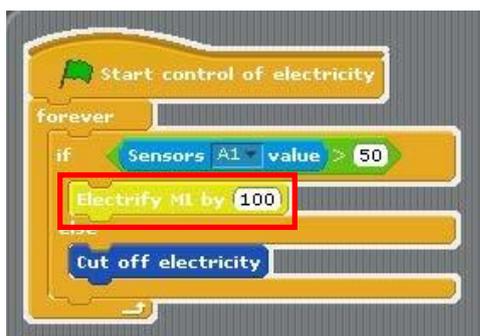
Dans une pièce éclairée

Capteur couvert avec la main

Le programme qui s'exécute en mode test lancera l'exécution du bloc **Cut off electricity** qui coupe l'électricité de M1 et éteint la mini ampoule quand la valeur du capteur de lumière est supérieure à 50 dans une pièce éclairée.

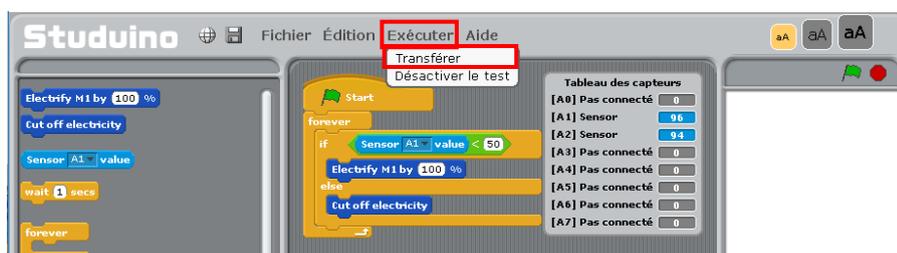


Quand la pièce s'assombrit et que la valeur du capteur de lumière tombe en-dessous de 50, le programme exécutera le bloc **Electrify M1 by 100 %** pour envoyer de l'électricité à M1 et allumer l'ampoule.



7.2.4. Transférer un programme

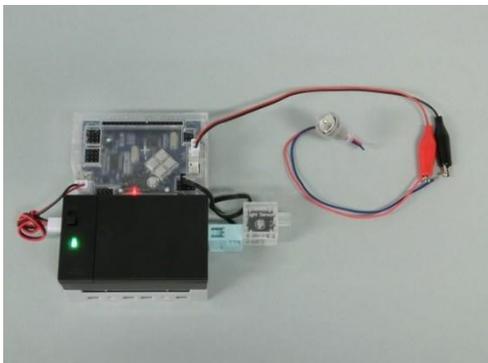
Dès que le programme fonctionne, vous pouvez le transférer sur votre Studuino et l'exécuter sans avoir besoin du câble USB. Éteignez votre batterie. Cliquez sur le menu **Exécuter** (1), puis sur **Transférer** (2).



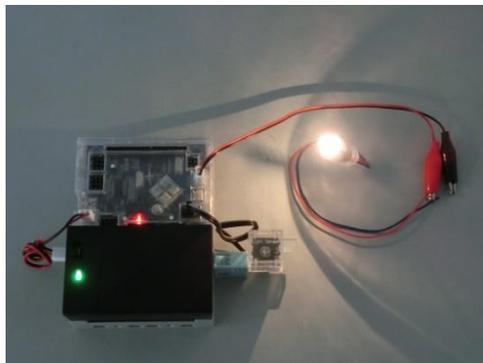
Le message ci-dessous apparaît au cours du transfert de votre programme.



Dès que le message disparaît, débranchez le câble USB du Studuino et allumez la batterie. Tout comme en mode test, votre ampoule restera éteinte tant que la pièce sera lumineuse et s'allumera quand la pièce s'assombriera.

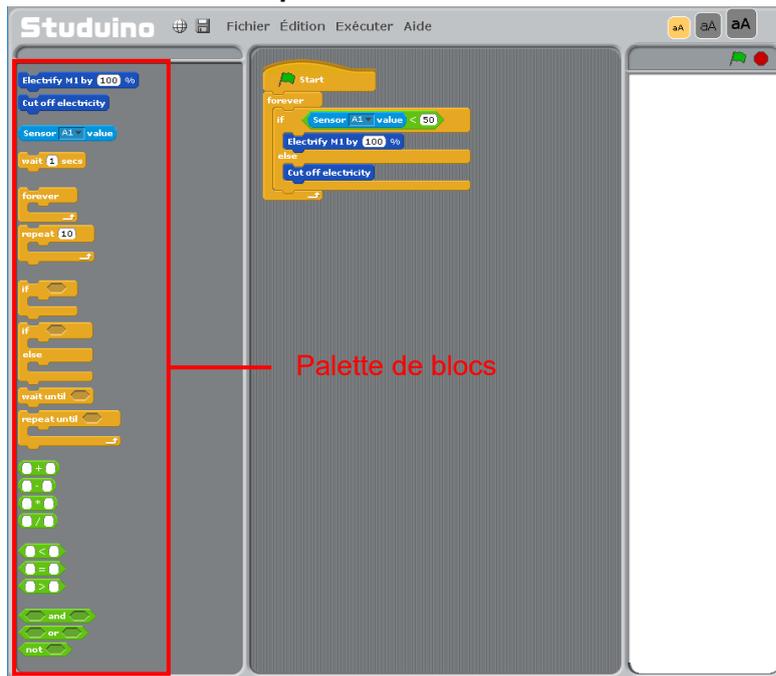


Quand la pièce est lumineuse



Quand la pièce est sombre

7.3. En savoir plus sur les blocs



Dans cette version du logiciel, tous les blocs dont vous aurez besoin sont réunis dans une seule palette. Consultez la partie ci-dessous pour en savoir plus sur les blocs qui n'existent que dans cette version. Référez-vous à la partie 4.2.1 pour en savoir plus sur les autres blocs.

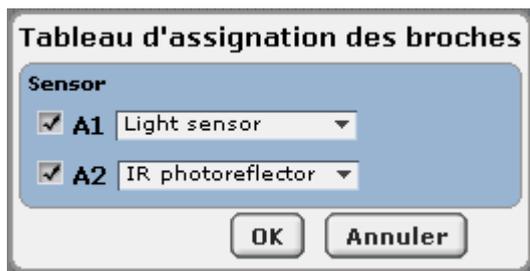
Bloc	Description
<p>①</p>	Envoie la quantité d'électricité indiquée en ① au connecteur M1.
	Coupe l'électricité envoyée à M1.
<p>①</p>	Observe la valeur du capteur connectée sur ①.

7.4. Le menu principal

Consultez cette partie pour en savoir plus sur les menus du logiciel.
Référez-vous à la partie 4.5 pour en savoir plus sur les autres menus.

- **Menu Éditer**
- **Paramètres des ports...**

Les paramètres des ports permettent d'indiquer les capteurs connectés au Studuino. Les paramètres des ports ouvriront une fenêtre « Tableau d'assignation des broches ». Vous pouvez paramétrer un capteur de lumière, un photoréflexeur IR ou un capteur de température.



7.5. Le menu contextuel

Référez-vous à la partie 6.6 pour en savoir plus.

7.6. Pour les tablettes Windows

Depuis la version 1.4.0.0 de l'environnement de programmation par blocs Studuino, les fonctionnalités suivantes ont été ajoutées ou modifiées pour fonctionner avec les tablettes Windows.

Clavier tactile pour le mode tablette Windows 10

L'utilisation de Windows 10 en mode tablette fera apparaître automatiquement le clavier tactile sur l'écran dès qu'une saisie de texte est requise.

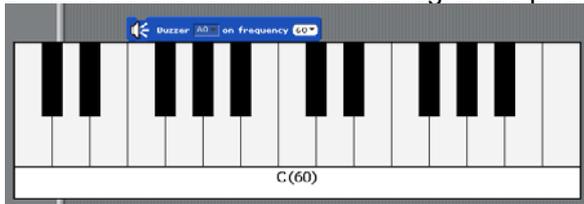
Affichage du pavé numérique pour les valeurs numériques

Le logiciel fera apparaître un pavé numérique au lieu d'un clavier tactile lors de la saisie de valeurs numériques dans un bloc.



Un clavier musical plus large

La taille du clavier musical a été agrandie pour faciliter la sélection des notes du buzzer.



Les tailles de police

L'utilisateur peut désormais choisir entre une taille de police petite, moyenne ou grande. Quand une taille de police plus grande est choisie sur une tablette Windows, la taille des blocs s'agrandit également pour qu'ils soient plus faciles à manipuler.



7.7. Studuino 2.0 (version complète)

- **Pièces optionnelles (version complète de Studuino, partie Environnement de programmation par icônes seulement)**

Cochez cette option si vous souhaitez utiliser des gyroscopes, des capteurs de couleur, des capteurs ultrasons, des capteurs de température et des modules Bluetooth. Rendez-vous sur la page Téléchargements et support pour consulter les manuels des pièces optionnelles et apprendre à les utiliser ces pièces.

- **Enregistrer les signaux infrarouges (version complète de Studuino, partie Environnement de programmation par icônes seulement)**

Une fenêtre d'enregistrement s'ouvre dans laquelle vous pouvez enregistrer les signaux infrarouges. Vous devrez cocher les pièces optionnelles et le récepteur infrarouge dans les paramètres des ports.