

Petit guide des possibilités d'utilisation du logiciel *Scratch* *en Technologie*

Présentation du logiciel.

Programmer les interfaces du laboratoire.



SCRATCH

Un logiciel à tout faire !

1. LA PRESENTATION GENERALE

Développé par le groupe de recherche Lifelong Kindergarten auprès du laboratoire Média du MIT, Scratch est un nouveau langage de programmation qui facilite la création d'histoires interactives, de dessins animés, de jeux, de compositions musicales, de simulations numériques ainsi que leurs partages sur le Web.

Scratch est un logiciel libre conçu pour initier les élèves, dès l'âge de 8 ans, à des concepts fondamentaux en mathématiques et en informatique. Il repose sur une approche ludique de l'algorithmique, pour les aider à créer, à raisonner et à coopérer. Il favorise également le partage sur le Web. A partir de 2007, le [site Web](#) a été ouvert afin de permettre à tous d'une part, de publier, donc de faire partager, ses projets sur le Web, et d'autre part d'apporter une aide à la mise en œuvre de [Scratch](#).



Qu'est ce que c'est ?

- Scratch est dynamique, il permet de modifier le code du programme en cours d'exécution. Orienté multimédia pour l'enseignement à l'univers informatique des enfants, il traite avec une grande facilité des concepts de base de la programmation comme les boucles, les tests, les affectations de variables, et surtout de la manipulation des objets, comme les sons et les vidéos.

Le principe

Des « briques » assemblées graphiquement constituent le programme.

- Scratch est visuel, tout le code est directement inscrit dans la langue maternelle de l'enfant (une vingtaine de langues européennes est disponible) sous forme de briques en couleurs (par exemple les contrôles en jaune, les variables en rouge, les mouvements en bleu).

- Scratch est libre et permet ainsi pour l'enseignant de diffuser sa pédagogie par une interactivité quasi-ludique des objets manipulés par ces briques logicielles. Le nom de Scratch fait référence à cet art de mélanger des sons grâce aux tables de mixage, comme à cette possibilité de réutiliser des objets.

Téléchargements

[Scratch en version 1.4](#)

[Scratch en version 2.0](#)

[Tutoriel de prise en main en français](#)

Nous allons exploiter les différentes possibilités d'interaction avec différentes interfaces utilisées au collège..

Sites incontournables

<http://www.cndp.fr/ecolenumerique/tous-les-numeros/boite-a-outices/apprendre-par-le-jeu/article/article/aide-memoire-recapitulatif-scratch.html>

<http://scratch.mit.edu/help/>

2. L'INTERFACE DE SCRATCH

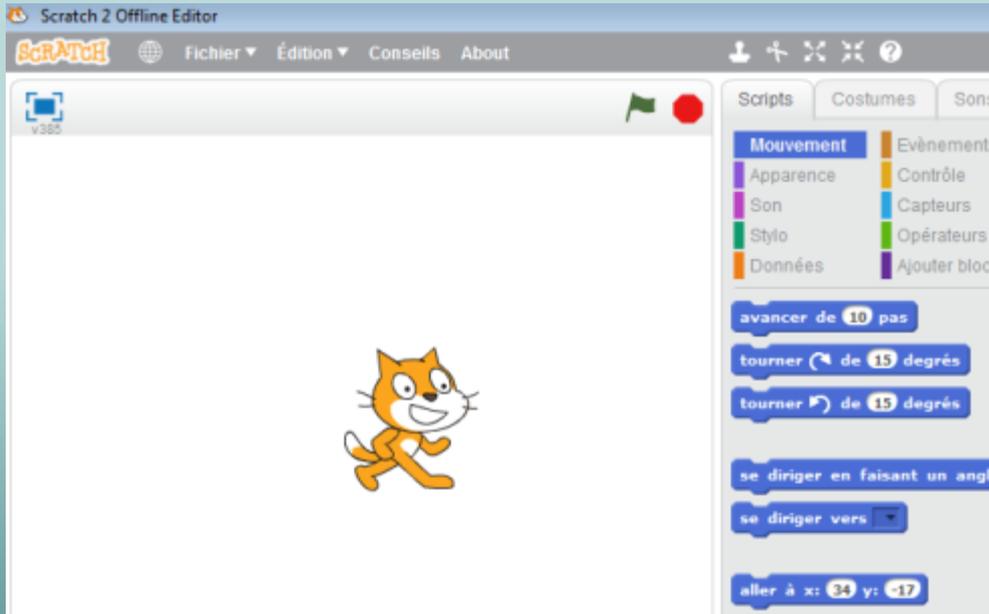
The image shows the Scratch 1.4 interface with several components labeled in French:

- PARTAGER SAUVEGARDER CHOISIR LE LANGAGE**: Buttons for sharing, saving, and language selection.
- Catégories de blocs**: A sidebar with categories like Motion, Looks, Sound, and Variables.
- PALETTE DES BLOCS**: A vertical list of programming blocks for sprites and objects.
- SPRITE PEUT ÊTRE TRADUIT PAR OBJET OU PAR LUTIN**: A note about the sprite palette.
- STYLES DE ROTATION**: Controls for rotation types for costumes.
- ONGLETS**: Tabs for editing scripts, costumes, or sounds.
- BARRE DE OUTILS**: A toolbar with icons for various actions.
- MODE DE VUE**: Controls to switch between small scene and full screen modes.
- MODE PRÉSENTATION**: A button to present the project in full screen.
- LE DRAPEAU VERT**: A green flag icon to activate scripts.
- LE SIGNE STOP**: A red stop sign icon to stop all scripts.
- LA SCÈNE**: The main stage area where the creation comes to life.
- L'AFFICHAGE DES COORDONNÉES DE LA SOURIS**: A feature to show the mouse cursor's position.
- BOUTONS POUR UN NOUVEAU LUTIN**: Buttons to create a new character or object.
- LISTE DES LUTINS OU OBJETS**: A list of sprites or objects with thumbnails.
- AIRE DES SCRIPTS**: The workspace where blocks are assembled into scripts.

Scratch 1.4 est un logiciel multi plateformes

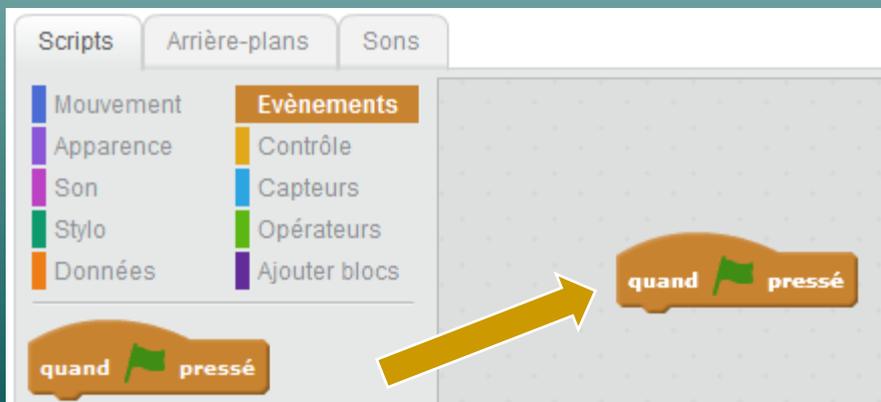
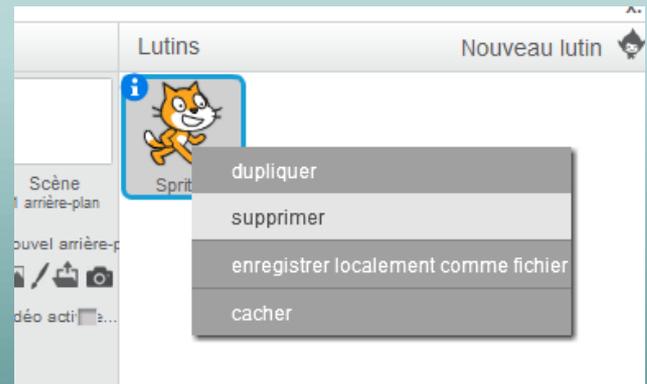
3. PREMIER PROGRAMME AVEC SCRATCH 2

La nouvelle version de Scratch inclut dans ses possibilités l'exploitation d'une caméra vidéo utilisée comme capteur. Nous allons réaliser un programme de détection de mouvement dans une zone précise. Les mouvements détectés vont déclencher une alarme. Ce programme peut être utilisé dans le cadre du thème confort et domotique, pour illustrer la fonction sécurité.



Quand Scratch 2.0 est lancé, passer le logiciel en français en cliquant sur l'icône représentant un globe terrestre.

Faites disparaître le chat.



Cliquer sur l'onglet « Scripts ». Choisir le menu « Evénements ». Faire glisser la commande dans la zone réservée aux programmes.

Le programme démarrera en cliquant sur le drapeau de couleur verte.

Attention, Scratch 2.0 fonctionne exclusivement sous Windows

3. PREMIER PROGRAMME AVEC SCRATCH 2

Dans un deuxième temps, nous allons configurer la caméra. Cliquer sur l'option « Capteurs ».



Faire glisser et poser les deux blocs ci contre. Les accrocher au bloc précédent.

Cliquer sur le bloc « activer la vidéo ». Cet essai est réalisé avec une caméra vidéo intégrée à l'ordinateur portable.

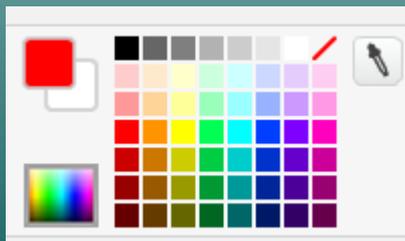


La vidéo est maintenant activée.



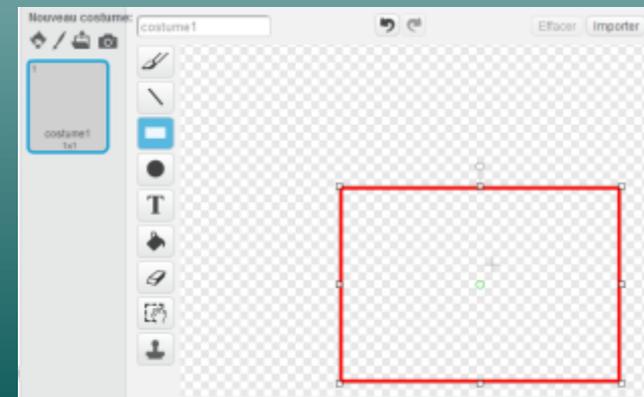
Nous allons créer une zone sensible sur l'écran.

Ce lutin, encore appelé « sprite » prendra la forme d'un rectangle de couleur rouge.



Choisir une couleur et dessiner un rectangle dans l'éditeur.

Cliquer sur le bouton « Vectoriser ».



3. PREMIER PROGRAMME AVEC SCRATCH 2



Cliquer à présent sur l'onglet « Script ». C'est normal qu'il soit vide car le script précédent est attaché à la scène. Faire glisser cette instruction. C'est le début du programme.



Créer maintenant une boucle à répétition infinie.



Créer maintenant un test.



La zone « sensible » est définie. Passer la main devant la caméra pour faire un essai.

Comment détecter le mouvement ?



Il faut créer un opérateur.



Préparer l'instruction de détection de mouvement

Détection d'un point en mouvement dans la zone c'est-à-dire sur le lutin.



Si le mouvement détecté dans la zone est supérieur à 70%

Le programme complet



Il ne reste plus qu'à lancer et tester le programme.

Dans ce cas, une alarme va être déclenchée sous forme d'un son joué.

3. PREMIER PROGRAMME AVEC SCRATCH 2

L'Aube est la proie de cambriolages nocturnes



Numéraire, bijoux, téléphones et ordinateurs portables : les cambrioleurs qui sévissent depuis quelques semaines dans l'Aube sont à la recherche d'argent et d'objets facilement revendables. Ils opèrent en fin d'après-midi, lorsque les maisons non éclairées trahissent l'absence de leurs occupants, ou en pleine nuit lorsque leurs victimes sont endormies. Ce scénario est celui qui inquiète le plus les forces de police.

Ce week-end, une quinzaine de cambriolages ont été commis dans des pavillons de l'agglomération troyenne, souvent après 22 h. Les malfaiteurs ont sévi à la Rivière-de-Corps, Saint-Parres-aux-Tertes et Saint-André-les-Vergers.

Article tiré de l'Est-Eclair, 2013

Problématique : Comment assurer la surveillance d'une habitation ?

Une des réponses possible évoque la vidéo surveillance.

Une autre réponse évoque la détection de mouvements autour des portes et des fenêtres.

Matériel nécessaire :

- Une webcam
- Des enceintes
- Le logiciel Scratch 2.0 installé.
- Un fichier son de sirène au format .wav.



Consignes :

Modifier le programme précédent pour permettre la surveillance des deux fenêtres.

Traitement du signal :
algorithme, organigramme,
programme.

1

Identifier les étapes d'un programme de commande représenté sous forme graphique.

2

Modifier la représentation du programme de commande d'un système pour répondre à un besoin particulier et valider le résultat obtenu.

La programmation d'un support automatique ne demande pas l'écriture de lignes de code. Elle doit être graphique si le support présente une interface qui le permet. Le système automatique doit être simple.

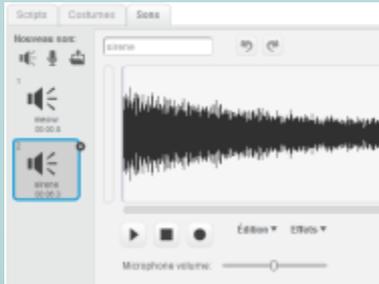
L'objectif est de comprendre de manière globale l'impact de la modification sur le fonctionnement du système.

**Compétences
visées**

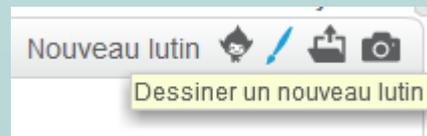
3. PREMIER PROGRAMME AVEC SCRATCH 2

Coups de pouce

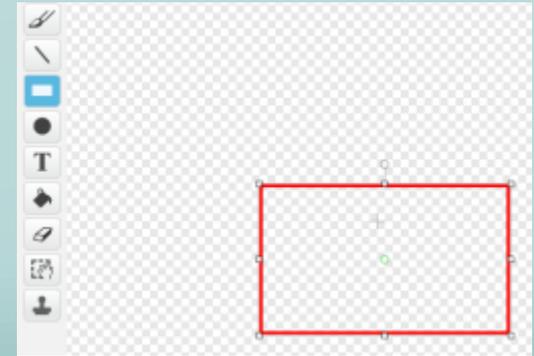
Aide donnée aux élèves sous formes de différentes procédures ou programmes.



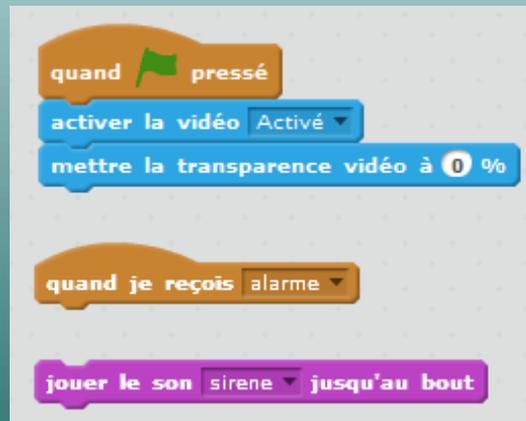
Pour charger un fichier son différent.



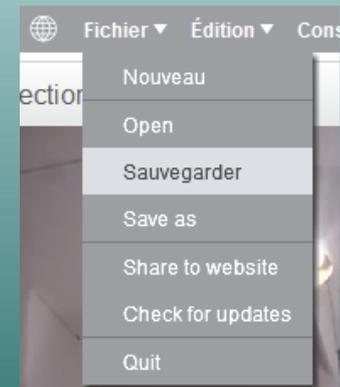
Pour définir une autre zone sensible



La commande « envoyer à tous » permet d'envoyer un message d'un programme à un autre.



La commande « quand je reçois » permet de déclencher un évènement en fonction du message reçu.



La commande « Sauvegarder » va vous permettre d'enregistrer tous vos essais.

3. PREMIER PROGRAMME AVEC SCRATCH 2



Dans un premier temps, il faut créer une ou plusieurs autres zones sensibles. Ces différentes zones doivent être ajustées en fonction de la pièce à surveiller.

Des « Sprites » ou lutins seront dessinés dans l'éditeur sous forme de rectangles.

Pour chacun des « Sprites », le programme ou script associé est toujours identique.



Le script suivant est à saisir pour tous les lutins.
Le message « alarme » est envoyé aux autres scripts



Ce script est associé à la scène 1. Si le message « alarme » est transmis, alors l'alarme va retentir.

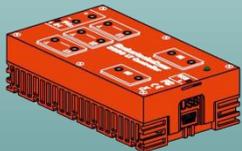
PROGRAMMER LES INTERFACES DU LABORATOIRE

Scratch est bien un outil universel permettant le pilotage et la programmation d'interfaces que l'on peut mettre en œuvre en 4^e ou 3^e.

Matériels souvent présents dans les laboratoires



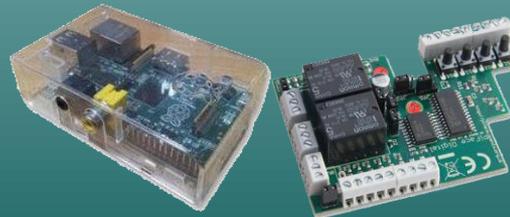
Lego TM
NXT
Enchanting
Windows



Fischertechnik TM
ROBO LT
CONTROLLER
Scratchfish
Windows



Fischertechnik TM
ROBO INTERFACE
Scratchfish
Windows



Raspberry PI
Interface Piface
Linux Raspbian
Scratch 1.4



Carte d'acquisition
Pico Board TM
Windows
-Scratch 1.4
Linux Raspbian
- Scratch 1.4



Matériels faisant leur entrée

Carte Arduino Uno TM
Interface de puissance
pour Arduino Uno.
Windows : S4A

Scratch 1.4 se décline donc en plusieurs « extensions ».

Légende

Marque
Matériel
Nom de l'extension
de Scratch
Plateforme

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE LA « BRIQUE » NXT

Enchanting est la première extension de Scratch que nous allons tester. C'est une version de Scratch 1.4 modifiée.



Toutes les informations sont disponibles ici : <http://wiki.scratch.mit.edu/wiki/Enchanting>.

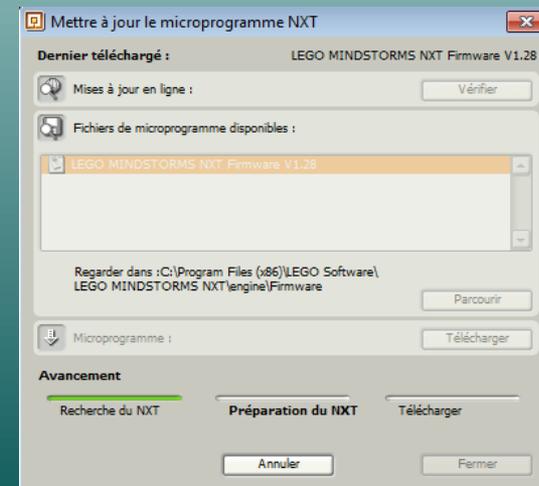
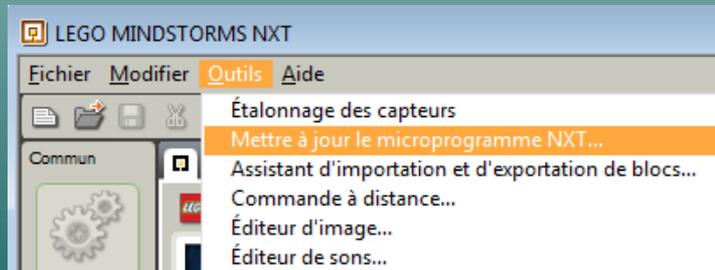
Pour télécharger la dernière version d' Enchanting, rendez-vous sur le site officiel : <http://enchanting.robotclub.ab.ca/tiki-index.php>

Sur ce site, une documentation sous forme de « cartes » au format .pdf en français permet une prise en main rapide. Pour en savoir plus : cliquer [ici](#).



PREPARATION DE LA « BRIQUE » NXT

Avant de pouvoir réaliser et tester des programmes avec Enchanting, nous devons « flasher » le microprogramme résidant dans la « brique » NXT. Cette opération n'est pas irrémédiable, on peut toujours revenir en arrière en utilisant le logiciel NXT Mindstorm pour télécharger le microprogramme d'origine.



Retour au « firmware » d'origine.

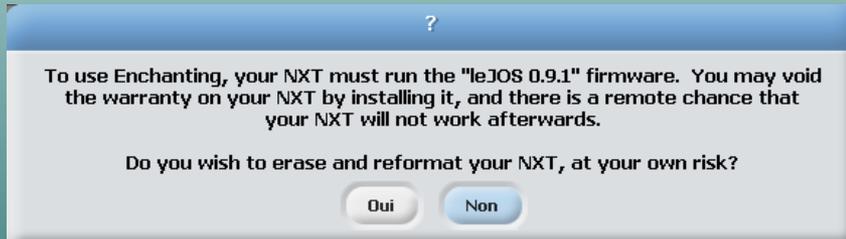
PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE LA « BRIQUE » NXT



Relier le micro contrôleur NXT à l'ordinateur à l'aide de son câble USB.

Dans le menu « Fichier », choisir la commande « Télécharger le micrologiciel... ».



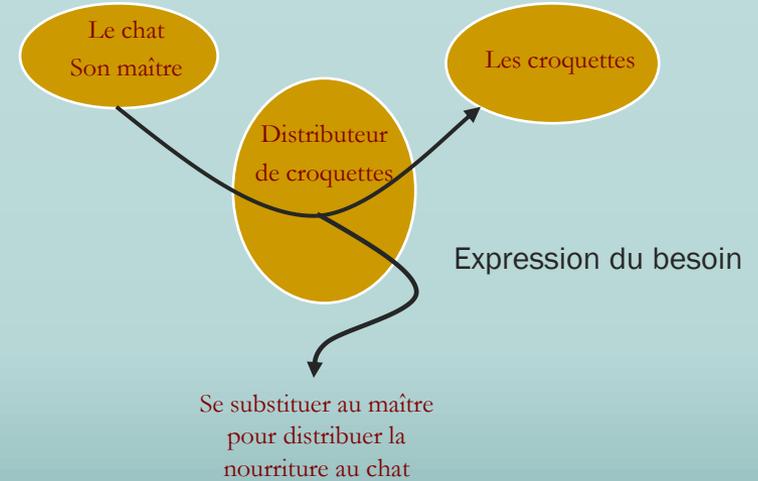
Cliquer sur « Oui » pour reformater le NXT. Une petite musique retentit en fin de téléchargement et l'opération est réussie !



Changer le « firmware » d'origine.

PROGRAMMATION DE LA « BRIQUE » NXT

REALISATION ET PROGRAMMATION SUR UNE MAQUETTE DE DISTRIBUTEUR DE CROQUETTES



Problème : comment partir en week-end tout en prenant soin de son chat ?

Extrait du cahier des charges fonctionnel

Fonctions de services	Critères	Niveaux	Tolérance
FP1 : Distribuer les croquettes pour chat pendant 48 heures.	Type de nourriture distribuée	Croquettes sèches	
	Quantité de nourriture distribuée par distribution simple	100g par jour pour un chat adulte	+ - 10%
	Réglage de la quantité distribuée	Aucun	
	Durée de la distribution	10 minutes	+ - 10 minutes
	Horaires de distribution	Définies par l'utilisateur	
	Type de distribution	- Programmé - Manuel	

PROGRAMMATION DE LA « BRIQUE » NXT

REALISATION ET PROGRAMMATION SUR UNE MAQUETTE DE DISTRIBUTEUR DE CROQUETTES

Extrait du cahier des charges fonctionnel (suite)

Fonctions de services	Critères	Niveaux	Tolérance
FC1 : Maintenir la nourriture à l'abris de l'air	Fermeture étanche du compartiment.		

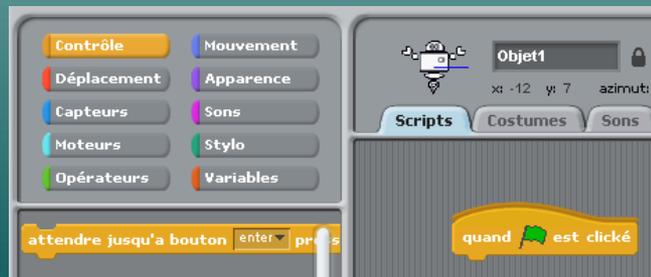
REALISATION DU PROGRAMME PAS A PAS

Configuration du moteur



Le servomoteur sera relié sur le port A.

Le début du programme :



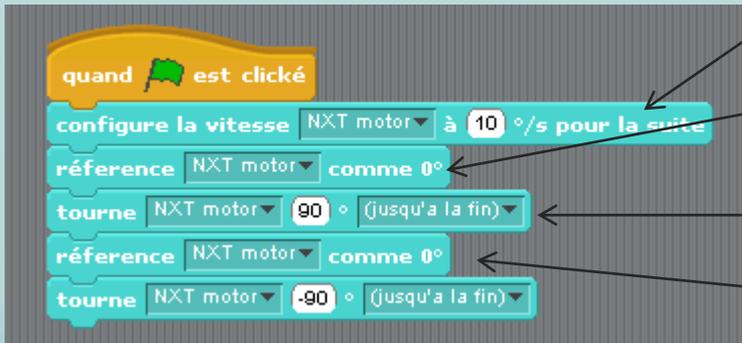
Le programme va démarrer si on clique sur le drapeau vert



PROGRAMMATION DE LA « BRIQUE » NXT

REALISATION DU PROGRAMME PAS A PAS

Configuration du moteur (suite).



La vitesse de rotation est fixée à 10° /s (très lent).

La référence est fixée par le compte-rendu du capteur angulaire du servomoteur. C'est le point de départ de la rotation.

Le moteur tourne de 90° et s'arrête.

La référence est fixée à nouveau et le moteur tourne de 90° dans le sens contraire. Retour à la position de départ.

Cette procédure peut être utilisée pour ouvrir et fermer le compartiment contenant les croquettes.



Configuration d'un capteur de contact

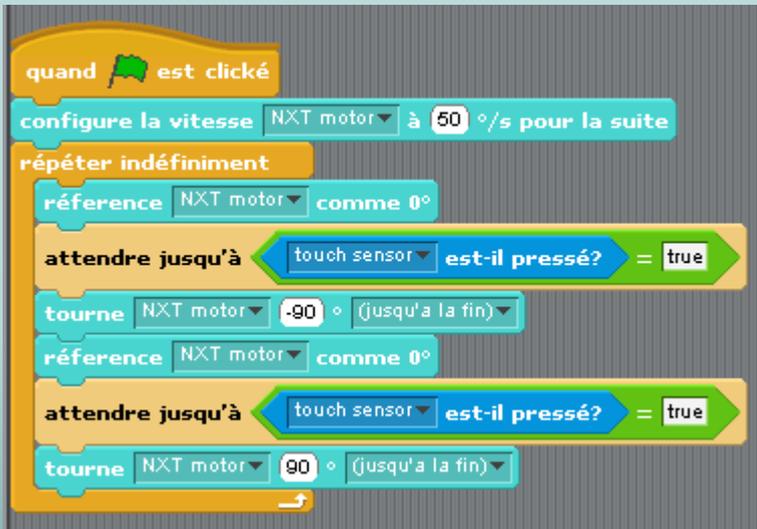


Dans l'éditeur, choisir un capteur de contact. Faire « glisser » le bloc sur le port 1. Il faudra relier le capteur sur l'entrée 1.



PROGRAMMATION DE LA « BRIQUE » NXT

REALISATION D'UN PROGRAMME PAS A PAS



Réalisons le programme suivant :

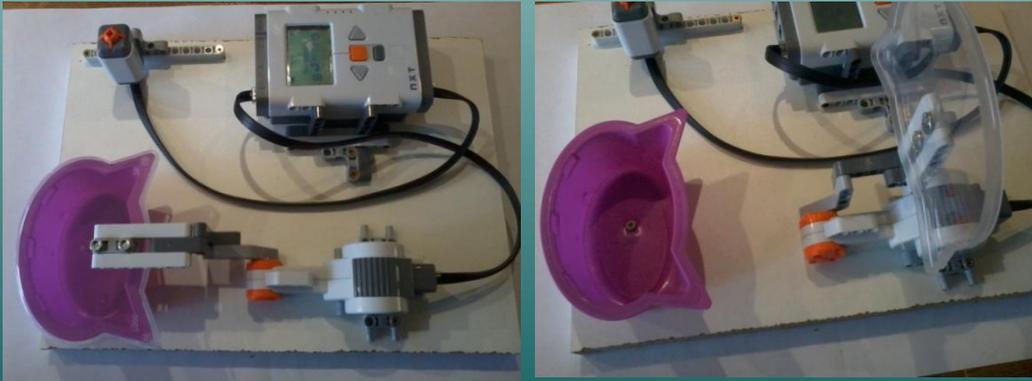
Une boucle est crée pour que les actions se répètent à l'infini.

Le programme attend un appui sur le bouton poussoir.

Dans ce cas là, le moteur tourne de 90° .

Un nouvel appui sur le bouton permet une rotation angulaire de 90° dans l'autre sens.

Ce programme peut être utilisée pour ouvrir et fermer le compartiment contenant les croquettes manuellement.



Ce prototype de distributeur de croquette peut servir de support pédagogique pour faire émerger la notion de condition de commande et de logique combinatoire de base : ET, OU, NON.

L'action sur le bouton poussoir permet de commander l'ouverture manuelle et la fermeture en douceur du distributeur de croquettes.

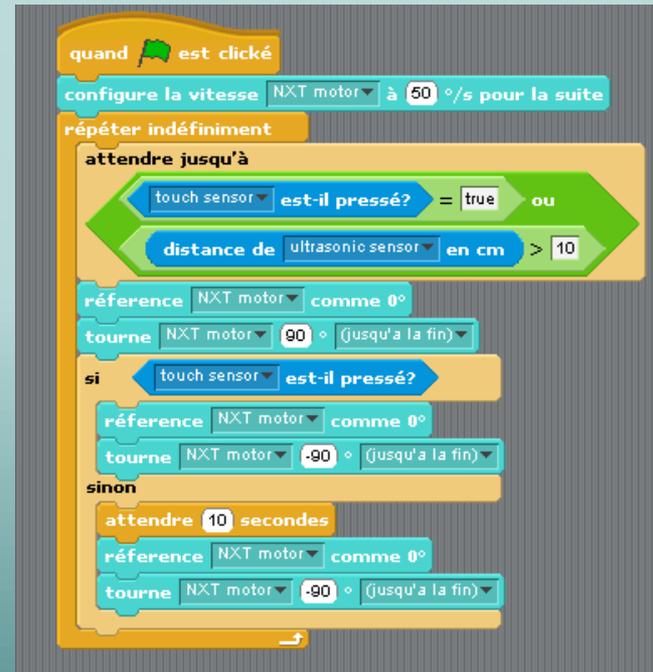
PROGRAMMATION DE LA « BRIQUE » NXT



Le capteur à ultrasons est utilisé pour mesurer la distance. Il faut le connecter sur l'entrée 2 du NXT.

Le bouton poussoir permet de commander en priorité l'ouverture ou la fermeture du récipient de croquettes.

Nous allons ajouter un capteur de distance à ultrasons. Si le chat s'approche à moins de 10 cm, le distributeur s'ouvre en douceur. Le distributeur se referme si aucune présence du chat n'est détectée pendant 10 secondes.



Commande d'un objet technique et logique combinatoire de base : ET, OU, NON.

2

Identifier une condition logique de commande.

On s'appuiera sur un objet pluri technique simple ou un système automatique simple. Il s'agit de montrer que la commande du dispositif peut être conditionnelle et que le comportement du système dépend d'informations captées et exploitées de façon logique.

Compétences visées

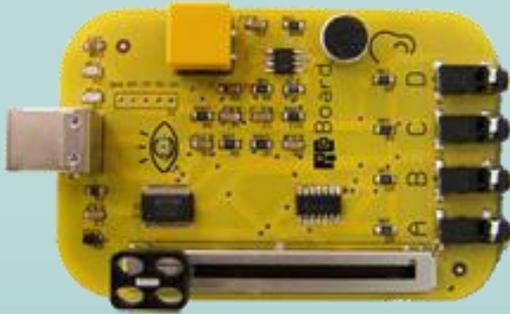
Le problème suivant peut être posé : A quelle condition s'ouvre le distributeur de croquettes ?

Les activités possibles :

- Tester le fonctionnement de la maquette et proposer une réponse sous forme d'un texte.
- Identifier les conditions directement dans le programme.

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE PICOBOARD



L'interface **Picoboard** permet d'acquérir des phénomènes physiques de toutes sortes.

- Mouvements
- Intensité sonore
- Intensité lumineuse
- Appuis
- Contacts
- Variation de résistivité.

Picoboard est disponible chez :

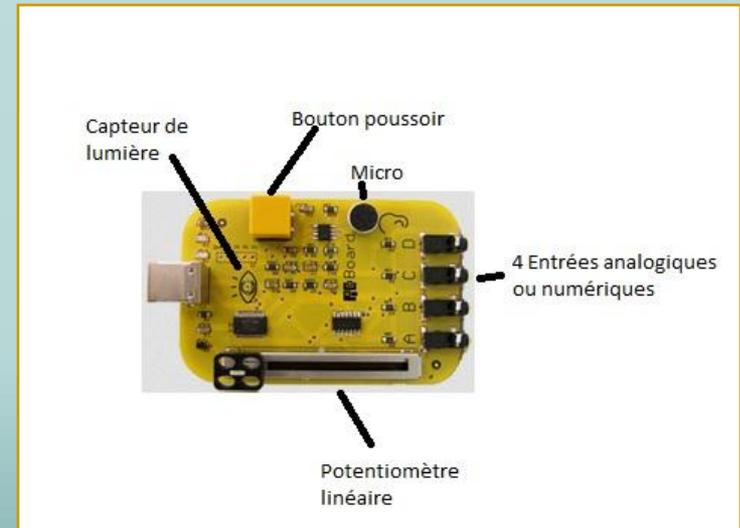
<https://www.sparkfun.com/products/10311>

Picoboard est aussi compatible avec Scratch 1.4 et Scratch 2.0 sous Windows.

Vous pouvez récupérer le driver à cette adresse :

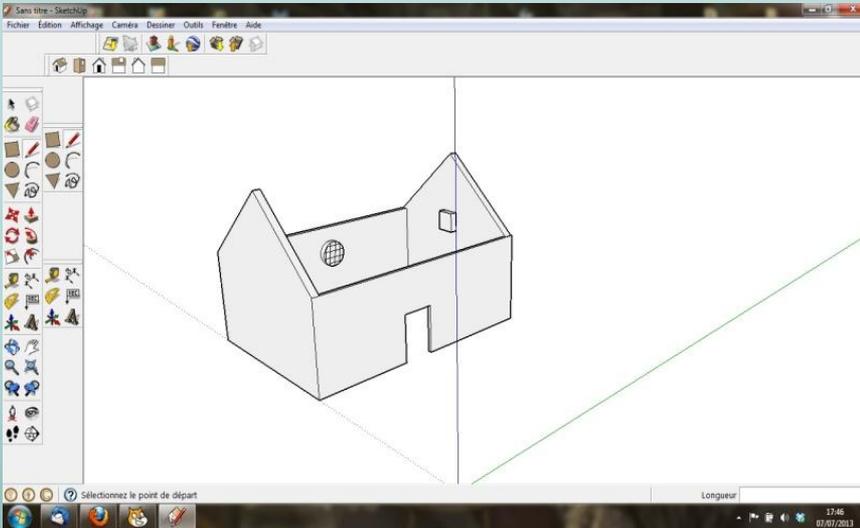
<http://www.picocricket.com/picoboardsetupUSB.html>

Les capteurs « tilt » et « distance » concernent l'interface des LEGO Education WeDo.

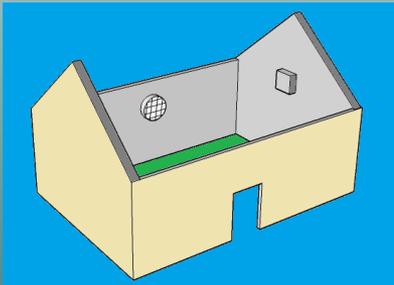


PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE PICOBOARD ET SIMULATION D'ÉCLAIRAGE AUTOMATIQUE



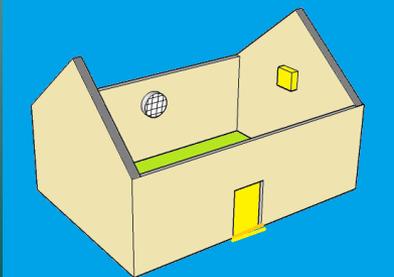
Scratch permet de réaliser des animations. Commençons par dessiner une maquette de maison avec Google Sketchup 8.



Exportons l'image et mettons là en forme (avec Paint par exemple) :

Le petit parallépipède rectangle représente une source de lumière (éteinte pour l'instant).

Enregistrons la au format .png sous le nom de : maison1.



Recommençons la même opération en simulant l'éclairage.

Enregistrons la au format .png sous le nom de : maison2.

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE PICOBOARD ET SIMULATION D'ÉCLAIRAGE AUTOMATIQUE

Dans Scratch 1.4, réalisons le programme suivant :

Cliquer sur le chat avec le bouton droit et faites le disparaître !

Importer un nouvel objet (icône dossier) : maison1.

Cliquer sur l'onglet : Costumes.

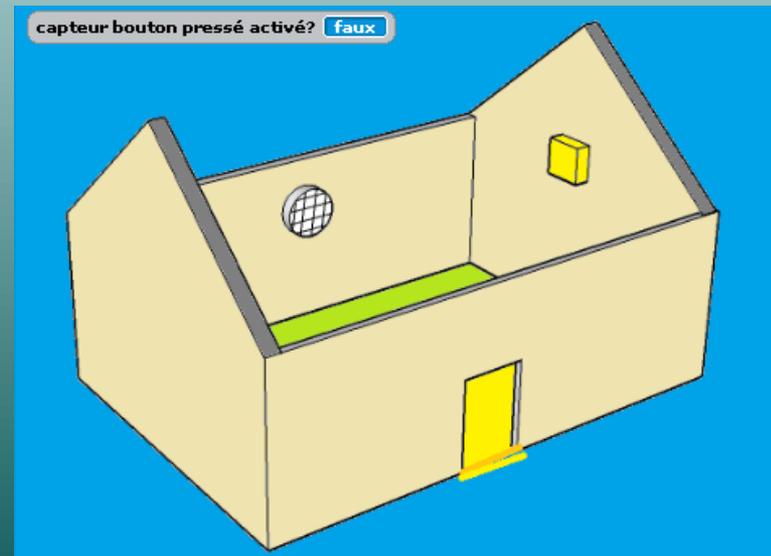
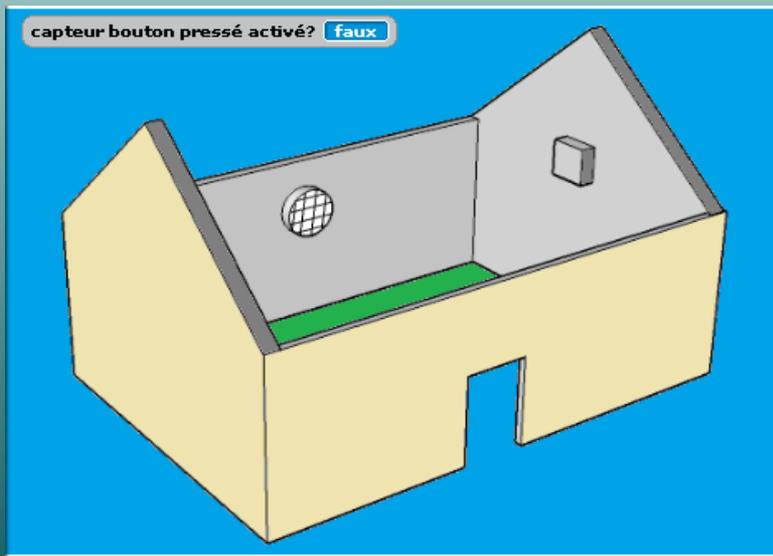
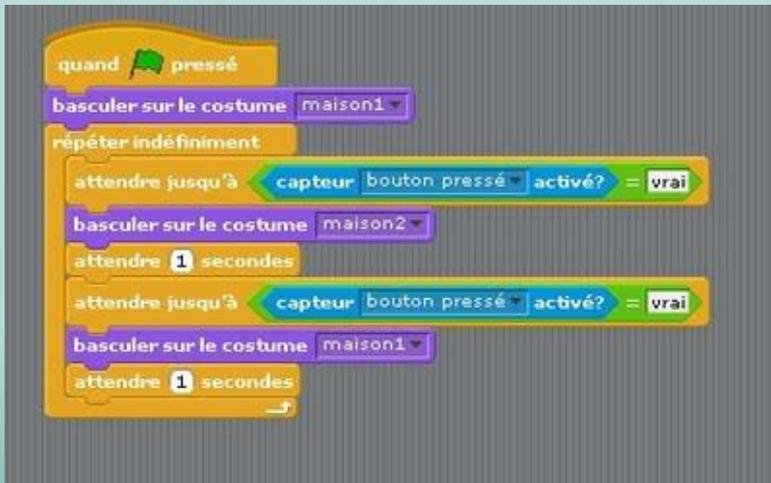
Cliquer sur le bouton : Importer et récupérer : maison 2.

Cliquer alors sur l'onglet : Scripts et saisir le "corps" du programme :

Pour lancer le programme, cliquer sur le drapeau vert.

En appuyant une fois sur le bouton de la Picoboard, la maison s'illumine.

En appuyant une nouvelle fois, la maison s'éteint.



PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE PICOBLOC ET SIMULATION D'ÉCLAIRAGE AUTOMATIQUE



Une première variante du programme : éclairage automatique en fonction de l'ambiance lumineuse extérieure (ci-contre).

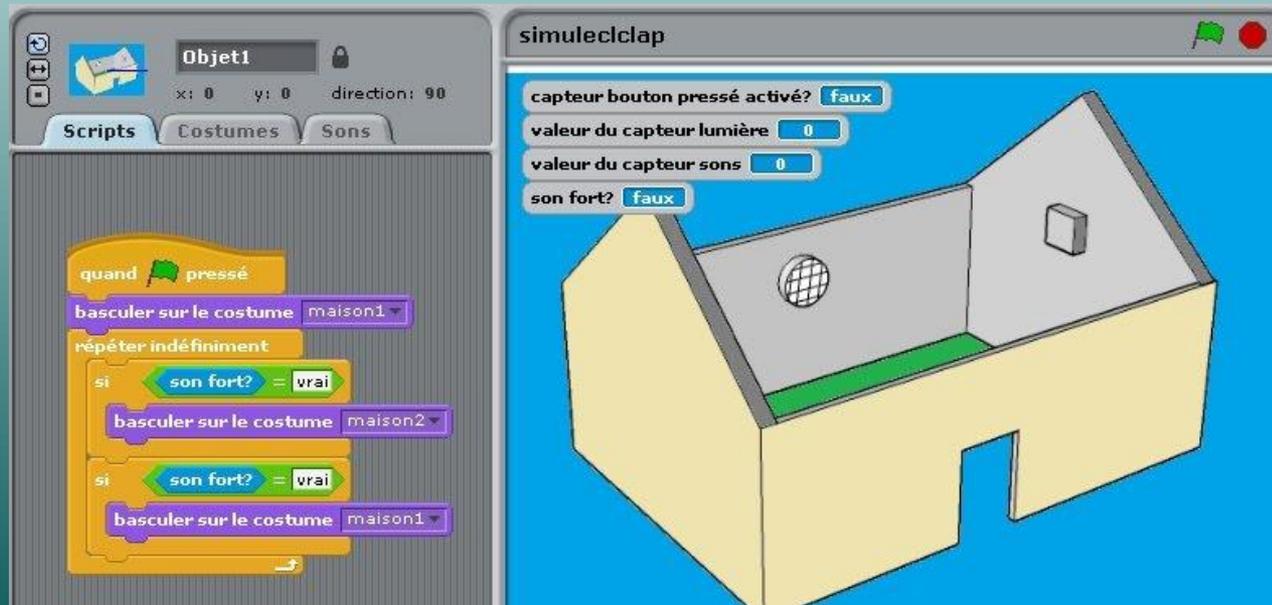
Le seuil de luminosité est fixé à 20%. Ce réglage est à effectuer en fonction de la luminosité ambiante.

capteur bouton pressé activé? faux

valeur du capteur lumière 2

Dans l'obscurité cette valeur est basse.

Une deuxième variante pour que l'éclairage obéisse aux sons (ci-dessous).



PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE LA CARTE ARDUINO « UNO ».

Une toute petite carte que l'on peut obtenir facilement. Cliquer [ici](#) pour en savoir plus.

Un câble USB est nécessaire pour l'alimenter et lui transférer des programmes.

Une version du logiciel Scratch 1.4 permet de la programmer "en temps réel".

Vous pouvez la récupérer à cette adresse : <http://s4a.cat/>

Vous aurez également besoin de l'éditeur de programme Arduino que l'on peut récupérer à cette adresse : <http://arduino.googlecode.com/files/arduino-1.0.5-windows.zip>

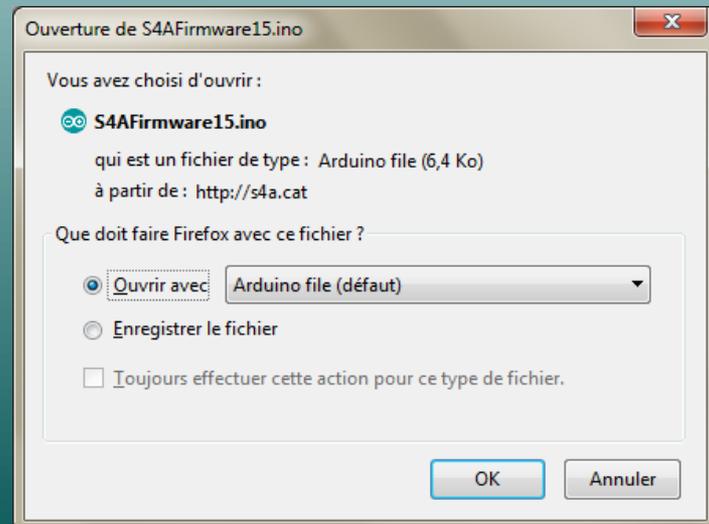
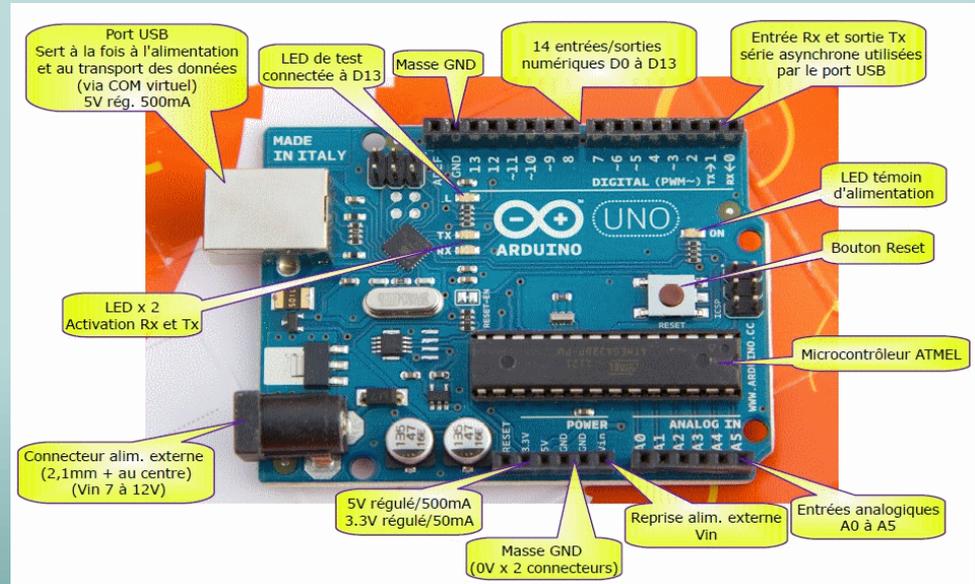
Pour terminer, il faut également récupérer le firmware à cette adresse et le transférer dans la carte avant de lancer Scratch (s4a).

<http://s4a.cat/downloads/S4AFirmware15.ino>

Récupérer ce fichier et le copier sur le bureau par exemple.

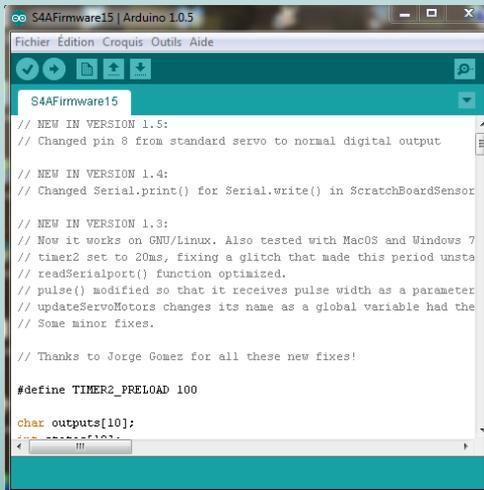
Lancer alors le logiciel Arduino et ouvrir le fichier.

Au démarrage, répondre OUI pour que le fichier soit mis au bon emplacement.



PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE LA CARTE ARDUINO « UNO ».



```
S4AFirmware15 | Arduino 1.0.5
Fichier Édition Croquis Outils Aide
S4AFirmware15
// NEW IN VERSION 1.5:
// Changed pin 8 from standard servo to normal digital output

// NEW IN VERSION 1.4:
// Changed Serial.print() for Serial.write() in ScratchBoardSensor

// NEW IN VERSION 1.3:
// Now it works on GNU/Linux. Also tested with MacOS and Windows 7
// timer2 set to 20ms, fixing a glitch that made this period unsta
// readSerialport() function optimized.
// pulse() modified so that it receives pulse width as a parameter
// updateServoMotors changes its name as a global variable had the
// Some minor fixes.

// Thanks to Jorge Gomez for all these new fixes!

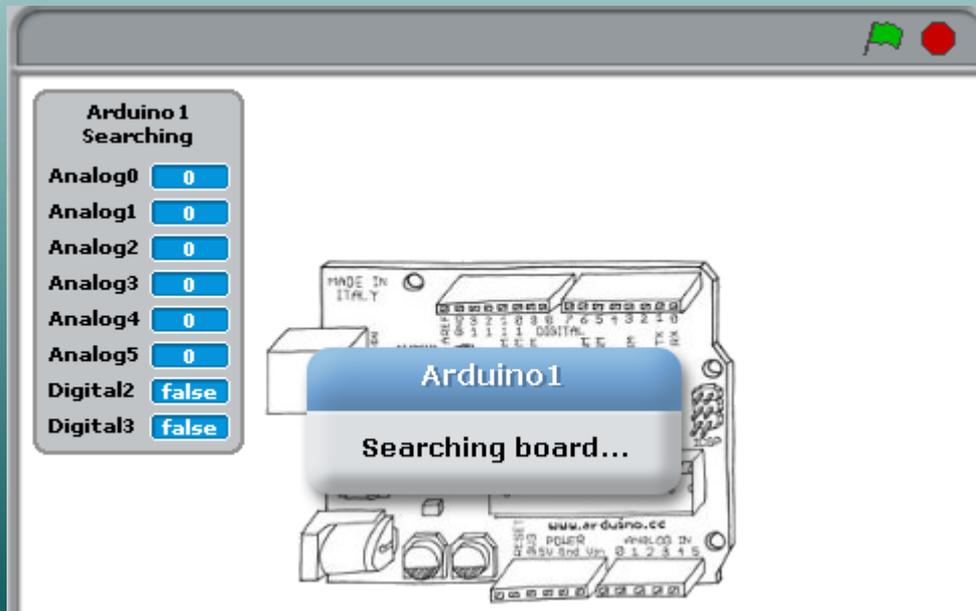
#define TIMER2_PRELOAD 100

char outputs[10];
```

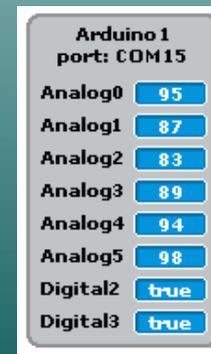
Brancher alors votre carte Arduino® et cliquer sur le bouton "Téléverser" dans le menu fichier.

Une fois le téléchargement vers la carte Arduino® effectué, le programme devient résident même si l'alimentation est coupée.

On peut alors lancer le logiciel Scratch pour Arduino®. Par contre, on ne peut pas transférer les programmes conçus sous Scratch pour qu'ils deviennent résidents. C'est un problème en cas de conception de robot autonome !.



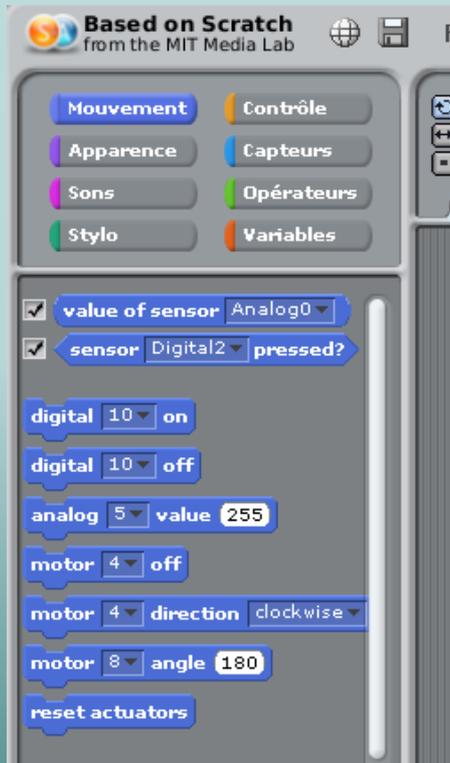
Si l'installation précédente a été réalisée correctement, au démarrage de S4A, le logiciel tente de se connecter à la carte.



Si vous connectez l'interface Arduino®, le message ci contre indique que la connexion est établie.

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE LA CARTE ARDUINO « UNO ».



Dans S4A, vous découvrez les commandes compatibles avec cette interface, en cliquant sur l'onglet « Mouvement ».



<http://www.velleman.co.uk/contents/en-uk/ka05.pdf>

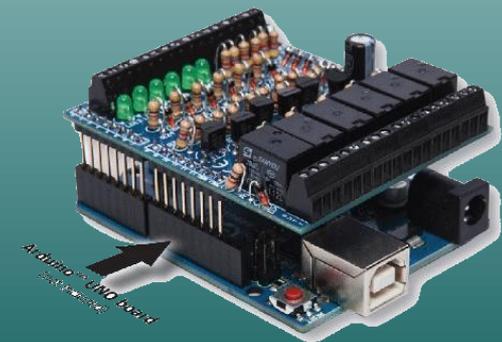
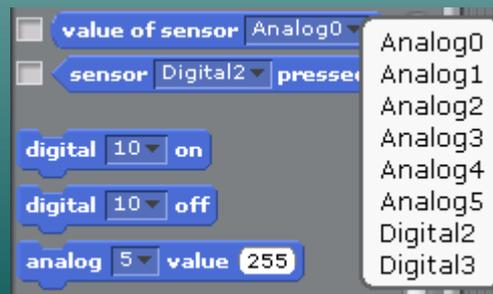
La carte Arduino « UNO » ne suffit pas en elle-même à piloter des actionneurs comme des lampes ou des moteurs électriques. Il faut lui adjoindre une carte de puissance comme celle disponible [ici](#).

Caractéristiques techniques

- 6 sorties sur relais
- 6 entrées analogiques
- 6 entrées numériques
- charge de 1A (max.) par sortie

Les instructions : digital n [on ou off] permettent le pilotage des sorties de l'interface.

Le compte rendu des entrées logiques ou analogiques se programme à partir de « value of sensor [Analog ou Digital] 0... ».



Cette carte se monte directement sur la carte Arduino.

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE ROBO COMPUTING PORTE DE GARAGE

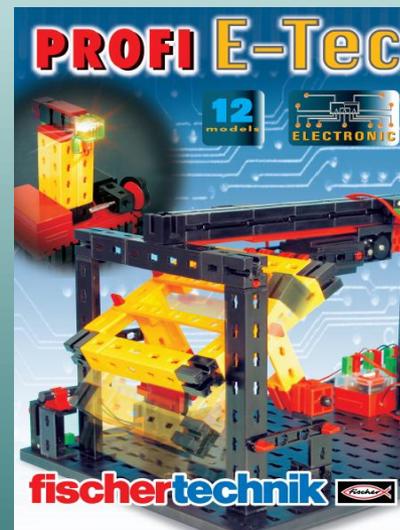
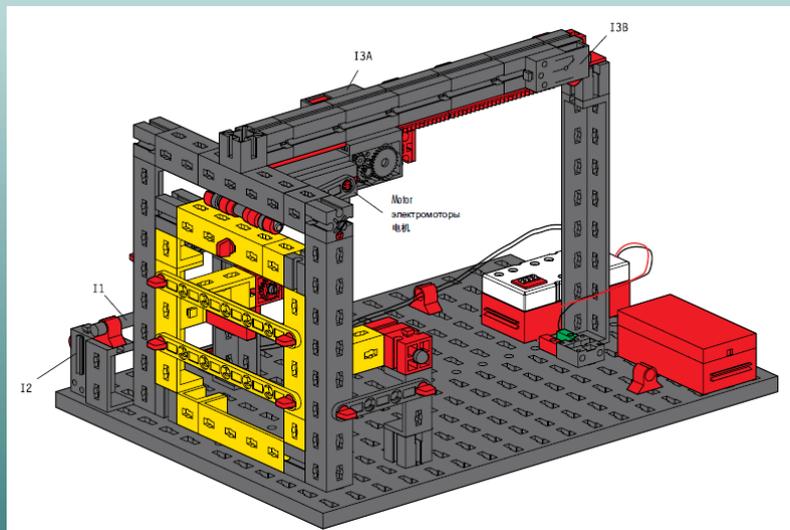
Quelques ressources en ligne :

http://ft-datenbank.de/web_document.php?id=46ab361a-2af2-4aea-aa81-a4c7f6c94dc0 [Notice de la boîte et différentes pièces]

http://ft-datenbank.de/web_document.php?id=6825aef8-e096-4f18-addf-979415bbb469 [Page 38 pour construire la maquette]

http://www.fischertechnik.de/en/PortalData/1/Resources/didactic/documents/activity-booklet/ProfiETech/E-Tech_E.pdf

[Tutoriel d'utilisation du module E-Tech (que nous n'utiliserons pas dans ce cas)]



INTERFACE
ROBO

http://ft-datenbank.de/web_document.php?id=b97581ed-4f74-4a83-8109-c98ef8cb520c [Documentation technique de l'interface ROBO]

<http://ft-datenbank.de/search.php?keyword=> [Lien pour retrouver une documentation technique de la marque au format PDF]

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE ROBO COMPUTING PORTE DE GARAGE

Pour installer ScratchFisch, respectez bien la procédure ci-dessous :

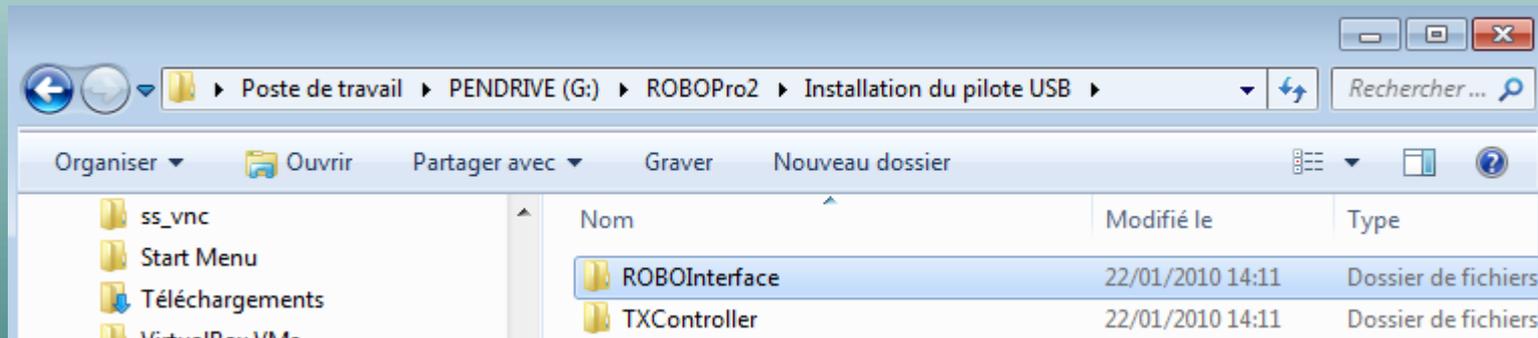
Votre ordinateur doit tourner sur Windows XP ou XPPro impérativement.

Si votre machine est munie d'un client pour permettre l'identification sur un réseau, désinstallez-le si possible.

Il faut commencer par installer le support JAVA sur l'ordinateur. Vous pouvez le télécharger à cette [adresse](#).

Il faut également installer le driver de l'interface ROBO. Vous le trouvez dans le CDROM du logiciel ROBO Pro :

Pour cela, connecter l'interface ROBO à l'ordinateur, attendre la boîte de dialogue relative à l'installation d'un nouveau matériel. Choisir l'installation manuelle en spécifiant l'emplacement.



Si vous ne disposez pas du CDROM, vous pouvez récupérer le driver à [cette adresse](#).

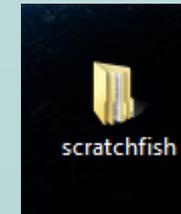
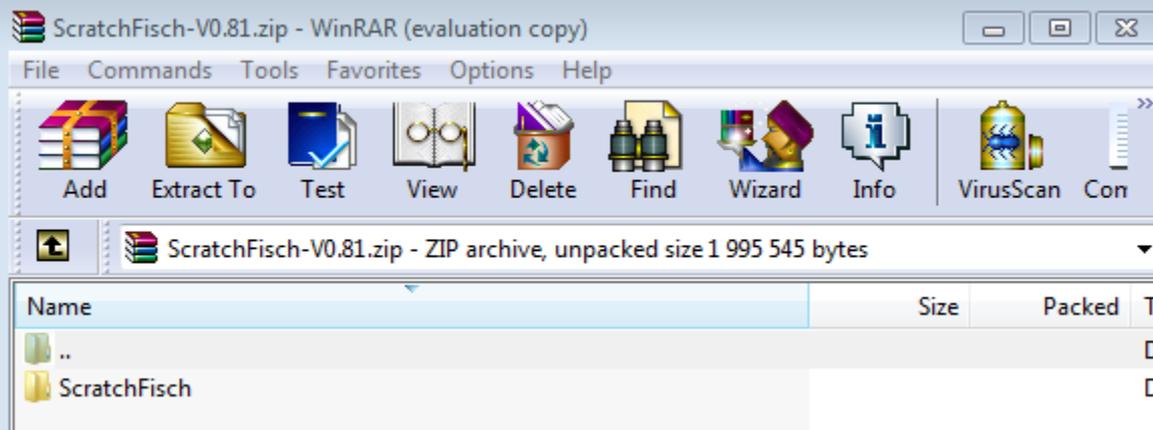
Maintenant, il faut vous rendre à l'adresse de ScratchFisch : <http://www.scratchfisch.org/download.html>



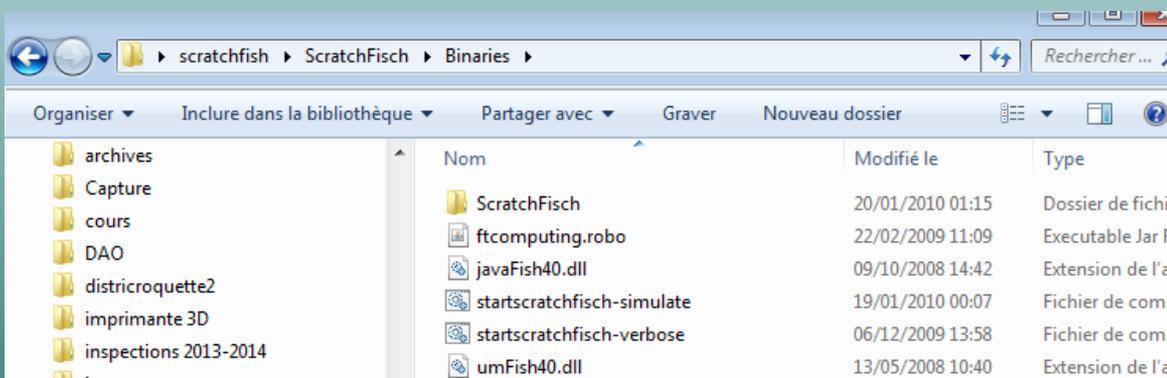
Choisir le deuxième lien correspondant à l'installation manuelle.

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE ROBO COMPUTING PORTE DE GARAGE



Glisser le répertoire ScratchFisch sur le bureau pour le décompresser.



Dans le dossier ScratchFisch, retrouver le sous répertoire Binaries.

Recopier son contenu dans :
C:\WINDOWS\SYSTEM32

Dans le répertoire ScratchFisch sur le bureau, lancer l'application Start-ScratchFisch.exe

```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\bricard\Bureau\ScratchFisch>java -cp ".\Binaries;.\Binaries\ftcomputing.rob.jar" ScratchFisch.ScratchFisch
ScratchFisch wurde gestartet.
Verbindung zu fischertechnik wurde aufgebaut.
```

Ce message doit s'afficher si la connexion avec l'interface ROBO est correcte.

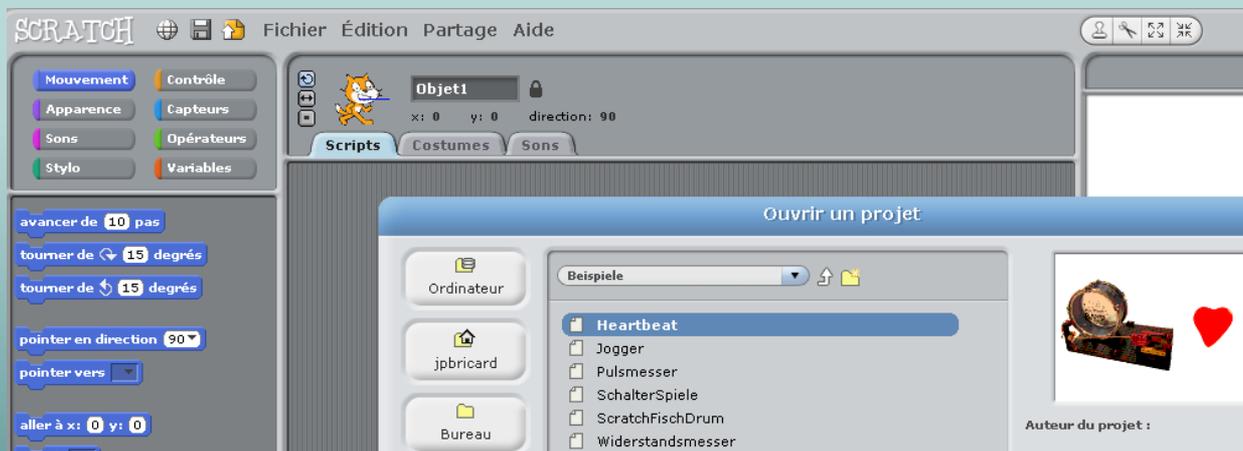
PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE ROBO COMPUTING PORTE DE GARAGE

Il faut maintenant récupérer Scratch 1.4 à l'adresse suivante : http://scratch.mit.edu/scratch_1.4/

Il ne reste qu'à installer et à lancer Scratch 1.4.

Pour que l'on puisse piloter l'interface ROBO, il faut encore créer un fichier de démarrage vide mais qui contient le protocole pour que le logiciel communique avec l'interface.



A partir de Scratch, ouvrir le fichier « Heartbeat » situé dans les exemples « Beispiele » du répertoire ScratchFisch.



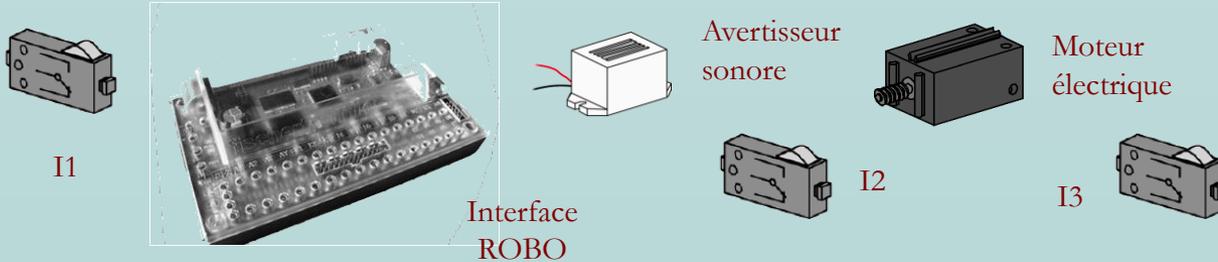
Créer un nouveau lutin en allant chercher le chat dans la bibliothèque, dans le répertoire « Animals ».

Supprimer les objets « Trommel » et « Hertz ».

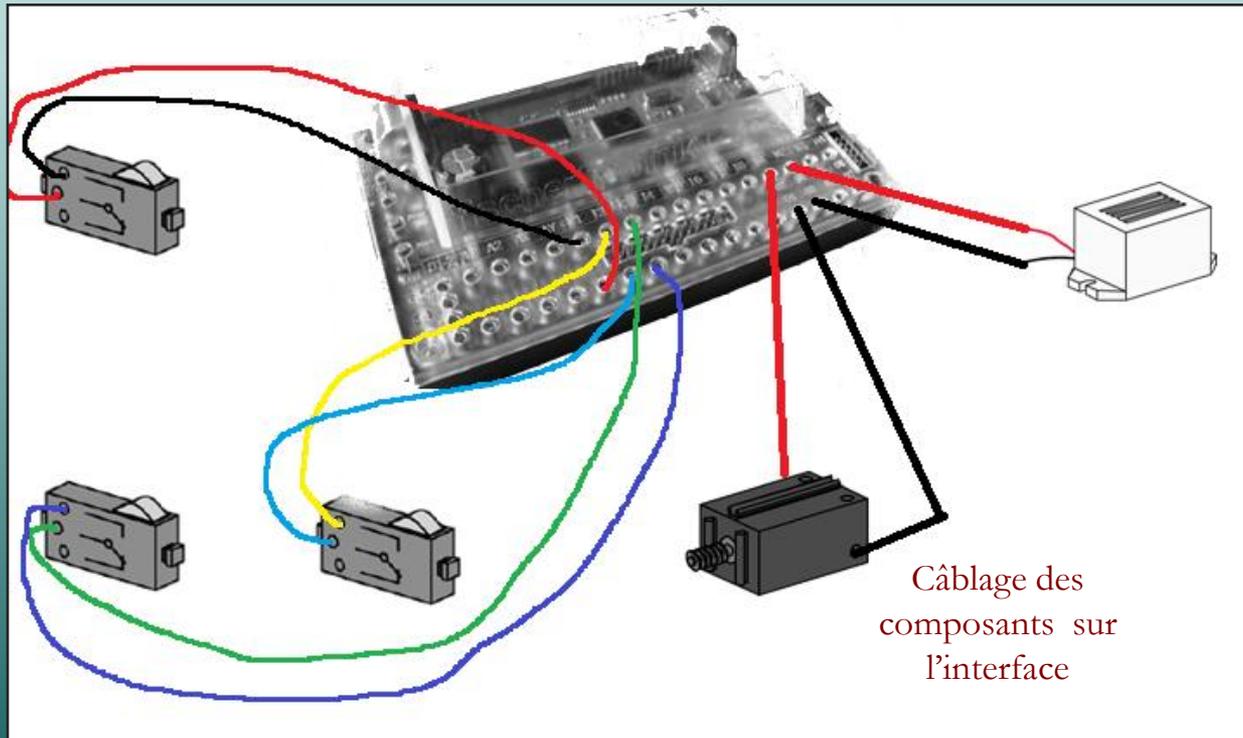
Enregistrer ce fichier « vide » sur le BUREAU avec le nom « Scratch pour ROBO ». C'est à partir de ce fichier que Scratch 1.4 sera lancé dorénavant.

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE ROBO COMPUTING PORTE DE GARAGE



Contacteur fin de course à relier sur I1 pour déclencher l'ouverture et la fermeture de la porte.



Contacteur fin de course à relier sur l'entrée I3 pour détecter la porte en position ouverte.

Contacteur fin de course à relier sur I2 pour détecter la porte en position fermée.

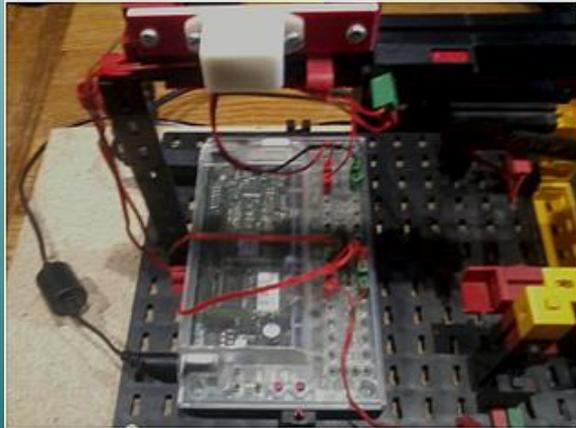
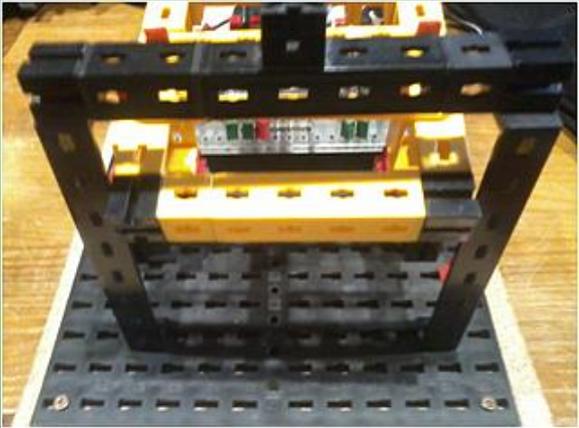
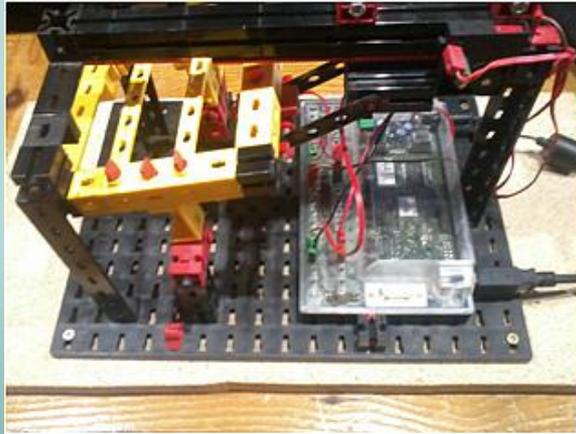
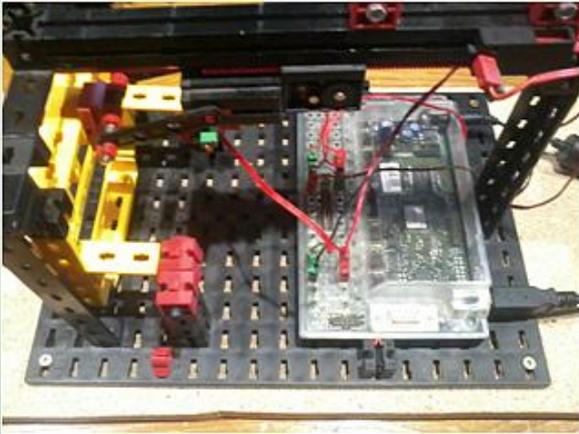
Le moteur est à relier sur la sortie M1.

L'avertisseur sonore est à relier sur la sortie M2.

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE ROBO COMPUTING PORTE DE GARAGE

Comment rendre l'ouverture et la fermeture de la porte moins dangereuse ?



Réponses possibles :

Déclencher un avertisseur sonore pendant l'ouverture et la fermeture de la porte.

Déclencher une lumière clignotante pendant l'ouverture et la fermeture de la porte.

Détecter une difficulté de fermeture ou d'ouverture de la porte.

Les programmes suivants permettent de comprendre comment déclencher deux actions simultanément.

La réalisation de la maquette peut être envisagée dans le cadre de la réalisation collective.

PROGRAMMER LES INTERFACES DU LABORATOIRE

PROGRAMMATION DE L'INTERFACE ROBO COMPUTING PORTE DE GARAGE

Programme initial

```
quand [drapeau vert] pressé
répéter indéfiniment
  attendre jusqu'à [valeur du capteur Eingang1] = 1
  envoyer à tous [Motor1.7]
  attendre jusqu'à [valeur du capteur Eingang3] = 1
  envoyer à tous [Motor1.0]
  attendre jusqu'à [valeur du capteur Eingang1] = 1
  envoyer à tous [Motor1.7]
  attendre jusqu'à [valeur du capteur Eingang2] = 1
  envoyer à tous [Motor1.0]
```

Le premier programme permet l'ouverture et la fermeture de la porte de garage à partir d'un appui sur le contacteur fin de course relié sur I1 « Eingang1 ».

La sortie Motor1.7 :

1 est le numéro de la sortie ici M1.

-7 et +7 correspondent à la quantité de courant envoyé au moteur ainsi que son sens de rotation. La vitesse de rotation du moteur peut varier de 0 à 8.

Le second programme permet de faire retentir une alarme pendant l'ouverture et la fermeture de la porte.

Il est nécessaire de créer une variable.

Programme modifié

```
quand [drapeau vert] pressé
à son attribuer 0
répéter indéfiniment
  attendre jusqu'à [valeur du capteur Eingang1] = 1
  envoyer à tous [Motor1.-7]
  à son attribuer 1
  attendre jusqu'à [valeur du capteur Eingang3] = 1
  envoyer à tous [Motor1.0]
  à son attribuer 0
  attendre jusqu'à [valeur du capteur Eingang1] = 1
  envoyer à tous [Motor1.7]
  à son attribuer 1
  attendre jusqu'à [valeur du capteur Eingang2] = 1
  envoyer à tous [Motor1.0]
  à son attribuer 0

quand [drapeau vert] pressé
  répéter indéfiniment si [son] = 1
    envoyer à tous [Motor2.7]
    attendre 0,5 secondes
    envoyer à tous [Motor2.0]
    attendre 0,5 secondes
```

Nouvelle variable
Supprimer une variable
 son
à son attribuer 0
changer son par 1
afficher la variable son
cacher la variable son
Créer une liste

La variable « son » quand elle prend la valeur 1, permet de déclencher l'avertisseur sonore par intermittence pendant l'ouverture ou la fermeture de la porte.