



HAL
open science

HOME I/O : du collège à l'enseignement supérieur

David Violeau, Bernard Riera

► **To cite this version:**

David Violeau, Bernard Riera. HOME I/O : du collège à l'enseignement supérieur. TDC-Réseau Canopé, 2018, pp.62-66. hal-02977470

HAL Id: hal-02977470

<https://hal.science/hal-02977470v1>

Submitted on 3 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HOME I/O : du collège à l'enseignement supérieur

DAVID VIOLEAU*, BERNARD RIERA**

Intégration d'une connexion avec Scratch, nouveau module pour communiquer avec n'importe quel éditeur Python, HOME I/O s'enrichit pour répondre aux évolutions des programmes. Voici des exemples d'exploitations pédagogiques, du collège à l'enseignement supérieur.

Nous avons eu l'occasion de consacrer deux articles au logiciel HOME I/O (*Technologie* n° 193 et n° 194), dont voici un rappel des points forts.

HOME I/O est un logiciel pédagogique 3D qui simule en temps réel une maison moderne. Il est le fruit d'un partenariat fécond entre le Crestic (Centre de recherche en sciences et techniques de l'information et de la communication) de l'université de Reims Champagne-Ardenne et la société Real Games. L'utilisateur se déplace dans une maison virtuelle sur le modèle des jeux vidéo FPS (*first person shooter*) et peut interagir avec les objets pilotables de la maison (interrupteurs, lumières, volets, radiateurs...) ou avec les paramètres environnementaux (température extérieure, taux d'humidité, géolocalisation...).

HOME I/O facilite la démarche d'investigation des élèves de collège autour du domaine d'application : « confort et domotique ». La récente connexion avec Scratch permet de programmer la maison pour la rendre automatique. HOME I/O est donc bien plus qu'une simulation, c'est un jeu sérieux, qui offre aux élèves une immersion et une expérience inédite motivante.

HOME I/O associé à CONNECT I/O trouve toute sa place en lycée dans la voie générale (S option SI), technologique (STI2D, en particulier les spécialités « Énergies et environnement » et « Systèmes d'information et numérique ») et professionnelle. Il peut être utilisé dans les établissements d'enseignement supérieur dans de nombreux domaines scientifiques et technologiques : automatique, informatique industrielle, réseaux, bases de données...

HOME I/O est à la fois une source de données et un système dynamique temps réel à piloter.

MOTS-CLÉS

programmation, automatisme, réseau, pédagogie

Interactions avec HOME I/O

Chaque objet de la maison peut être piloté au moyen de trois modes différents : câblé (sans domotique, il s'agit d'une installation électrique « classique » où l'ensemble des appareils sont câblés et fonctionnels), console ou domotique « simple » (une console virtuelle intégrée à la maison peut être programmée à la façon d'une box domotique) et externe (où l'utilisateur peut piloter la maison avec différents langages ou technologies selon le niveau d'enseignement considéré en gérant les entrées/sorties (I/O) de chaque objet).

Le mode externe offre de nombreuses possibilités pédagogiques : en collège, une bibliothèque Scratch permet d'accéder aux éléments de la maison ; en lycée, on pourra utiliser AppInventor pour développer une application Android afin de piloter la maison ; en CPGE, un module Python permet de commander les éléments de la maison et également d'illustrer les aspects de régulation et de programmation ; au niveau licence, l'utilisation de CONNECT I/O ou du SDK permet de s'interfacer avec tout type de contrôleur matériel (automate programmable industriel, micro-contrôleur...) ou logiciel (Matlab, LabView...).

Nous allons découvrir au travers des différents langages de programmation l'utilisation que l'on peut faire de HOME I/O.

Utilisation en collège

Pour découvrir les possibilités de HOME I/O et son adaptation au collège, nous allons prendre l'exemple de l'automatisation de volets.

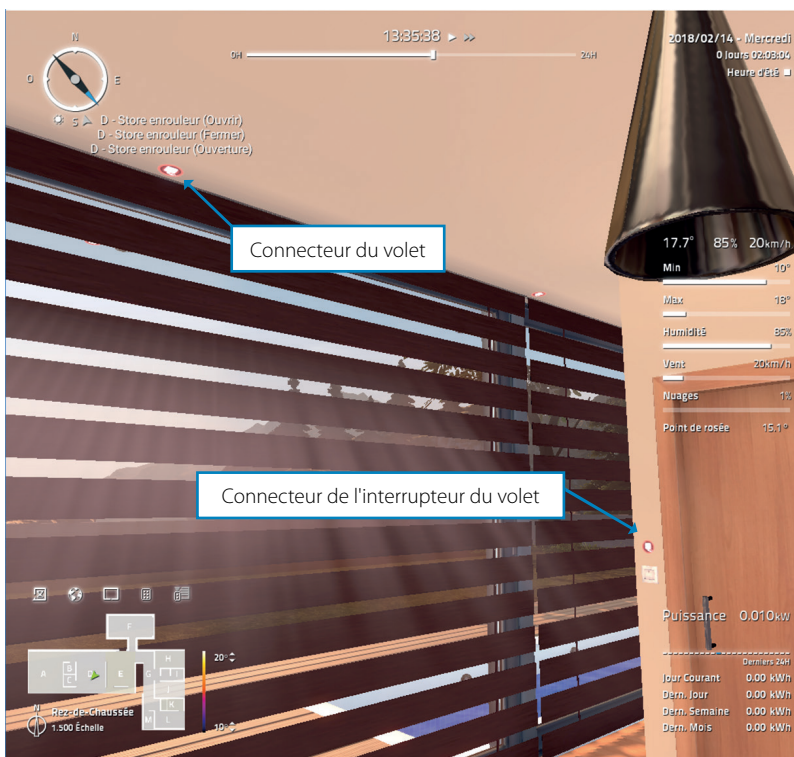
Après s'être déplacé dans la pièce D **1**, on peut repérer l'interrupteur qui commande le volet roulant de la pièce. L'actionneur du volet roulant est intégré dans le mur, seul le connecteur est visible au-dessus du volet.

En gardant le mode câblé, on peut appuyer sur la partie haute de l'interrupteur pour faire monter le volet. Un appui sur la partie basse stoppe le volet. Un nouvel appui le fait descendre et inversement pour la montée.

On passe en mode pilotage externe en cliquant sur le connecteur rouge au-dessus de l'interrupteur haut-bas d'ouverture du volet, ainsi que sur le

* Professeur agrégé de SI, enseignant au lycée Janson-de-Sailly, Paris.

** Professeur des universités, Université de Reims Champagne-Ardenne, directeur du CReSTIC, EA 3804.



1 Volet à automatiser de la pièce D

connecteur au-dessus du volet (actionneur). Ces éléments passent verts puis bleus. Le mode qui nous intéresse est le mode externe bleu.

Un fichier Scratch vierge est disponible sur le site teachathomeio.com et contient les blocs de commande permettant d'interagir avec HOME I/O.

Sous Scratch, on peut faire glisser le bloc « Stopper volet » depuis la zone « Ajouter blocs » vers la zone de programmation 2a. En modifiant les paramètres de ce bloc pour ouvrir le volet 1 de la pièce D et en cliquant sur ce bloc, on visualise directement le résultat d'ouverture du volet dans HOME I/O. On peut également fermer ou stopper le volet.

Pour récupérer dans le logiciel l'information d'appui sur l'interrupteur, on dispose du bloc « Interrupteur volet » 2b. Après paramétrage du bloc, en double-cliquant sur celui-ci, on observe la valeur 0 si la partie haute de l'interrupteur est non appuyée ou 1 si elle l'est, ce qui correspond à une information logique.



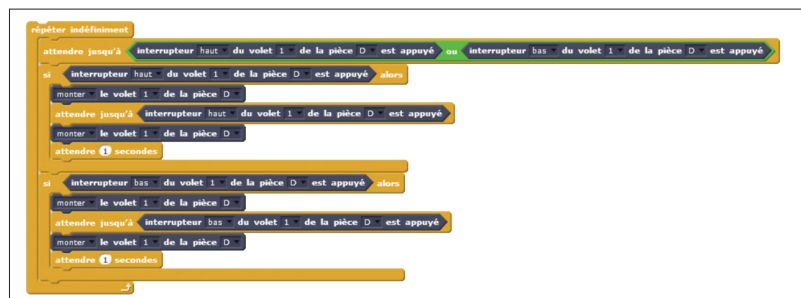
2 Blocs de programmation pour le volet



3 Exemple de programme simple

Différents programmes utilisant ces blocs simples peuvent être réalisés et testés directement. Ils permettent entre autres d'illustrer la notion de répétition ou les notions de conditions « si/sinon ». On peut demander aux élèves de réaliser des programmes très simples 3. Pour éviter des blocages de communication entre Scratch et HOME I/O, une attente courte a été ajoutée sur le programme simple. On peut également proposer aux élèves de modifier un programme 4 qui comporte des fautes pour respecter une spécification. C'est une méthode pédagogique intéressante pour l'apprentissage de l'informatique.

Pour améliorer le comportement du volet, on peut modifier les conditions de manière à ce qu'il soit stoppé lorsqu'il atteint ses positions haute et basse



4 Exemple d'un programme comportant des fautes



5



6 Blocs pour améliorer l'automatisation du volet

(pour ne pas faire forcer le moteur). Ceci est réalisé grâce au bloc 2c qui donne la position du volet (entre 0 V fermé et 10 V ouvert).

Pour automatiser encore plus le volet, il est possible d'utiliser une télécommande ; comme pour les autres éléments, on passe en mode programmation avec Scratch en appuyant sur le bouton pour le faire passer du vert au bleu 5.

Le bouton 1 de la télécommande doit permettre, par appuis successifs, de lancer l'ouverture, de stopper le volet et de le fermer également. La difficulté qui apparaît est d'associer plusieurs actions à un même bouton en fonction de ce qui s'est passé avant. On remarque qu'il est important de parler de notion de front ou d'appui sur le bouton plutôt que de la valeur du bouton lui-même. On gère la notion de front en mettant une attente après une action exécutée 6a.

On peut également automatiser l'ouverture du volet en fonction de l'heure en utilisant des conditions et les blocs 6b, par exemple fermeture automatique à 18 h

et ouverture automatique à 7 h. La possibilité d'accélérer le temps est particulièrement utile pour tester le comportement du programme. Cette gestion en fonction de l'heure n'est pas optimale, car le jour se lève ou se couche à une heure différente selon la date.

Nous pouvons alors utiliser un détecteur de luminosité pour choisir quand ouvrir ou fermer le volet. Le capteur délivre une information analogique qui peut être visualisée à l'aide d'une variable dont la valeur est automatiquement mise à jour et varie en fonction de l'ouverture du volet et de l'heure 6c.

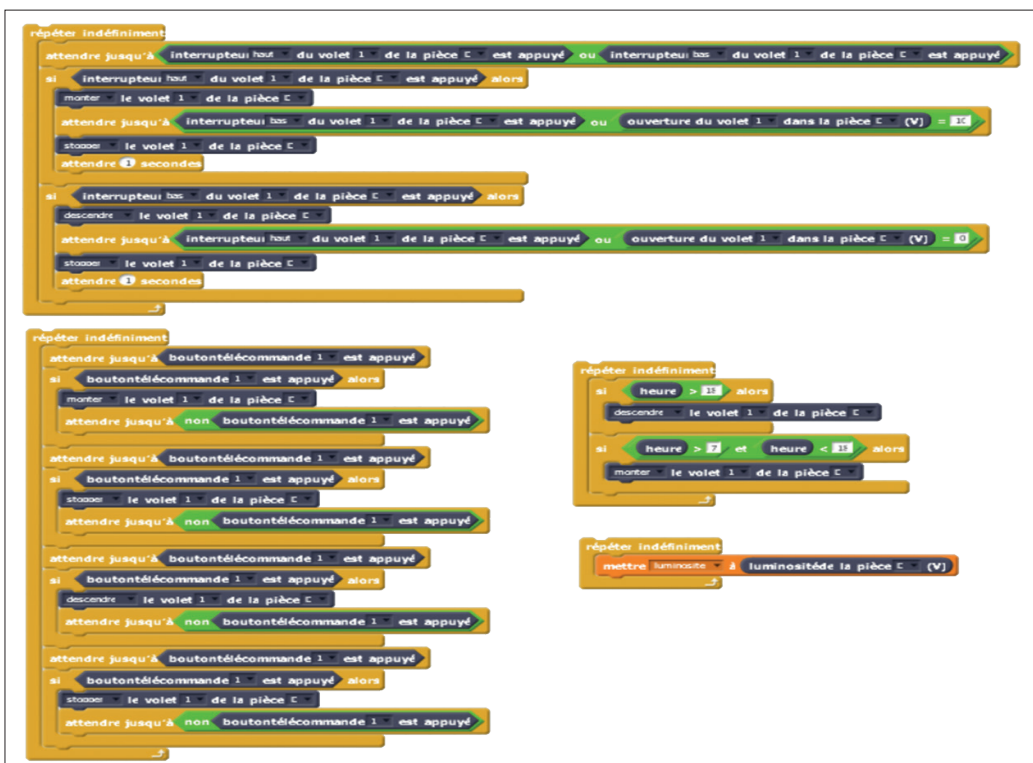
Les programmes 7 correspondent au fonctionnement complet automatisé du volet.

Utilisation au lycée

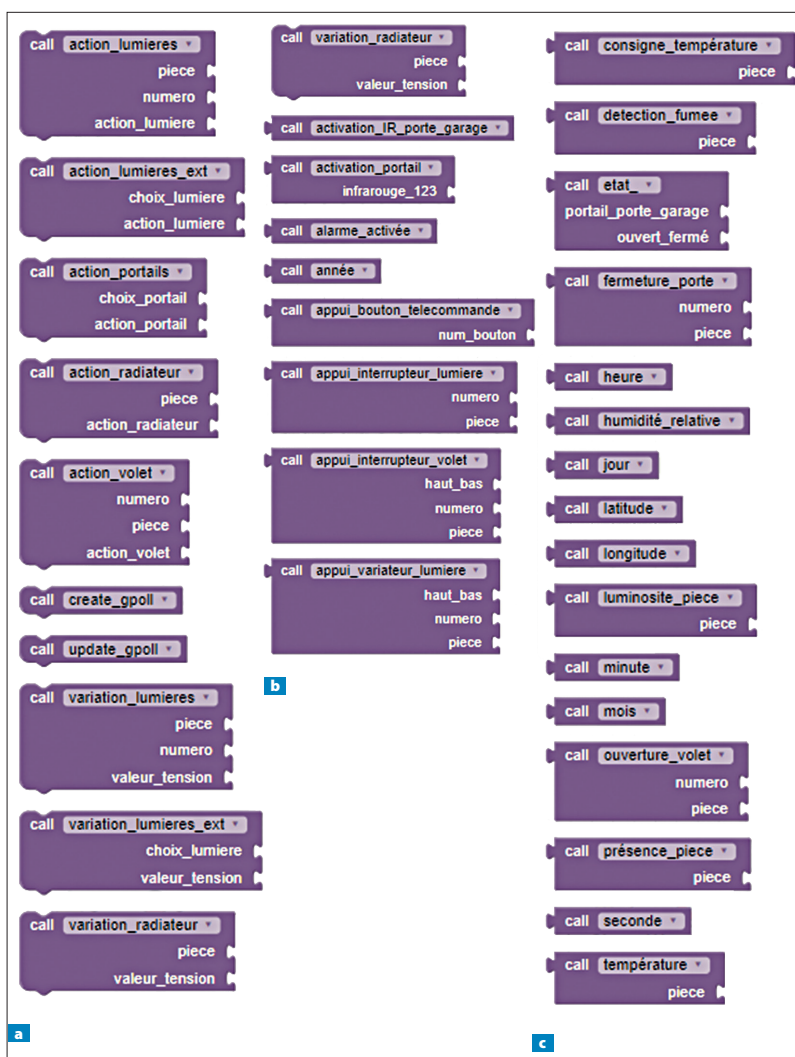
Au collège, il peut être envisagé de réaliser une application pour smartphone, mais c'est surtout en STI2D (spécialité SIN) où cette compétence prend tout son sens. L'intégration d'un serveur à HOME I/O permet de le coupler directement à un smartphone à l'aide de l'application AppInventor. La seule condition d'utilisation est d'avoir un réseau Wifi et accès à Internet. Il n'y a aucune installation à faire sur les PC.

Sur une tablette, on installe le logiciel MIT Companion AI2 qui permet de tester un programme.

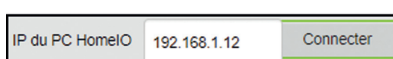
Sur le PC, on se connecte à AppInventor en utilisant un compte Google pour accéder à ses programmes automatiquement sauvegardés. Un fichier contenant les blocs de communication avec HOME I/O est disponible sur le site teachathomeio.com dans les



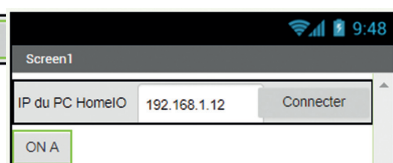
7 Programmes du fonctionnement complet automatisé du volet



8 Blocs de communication sur AppInventor



9 Barre de connexion AppInventor

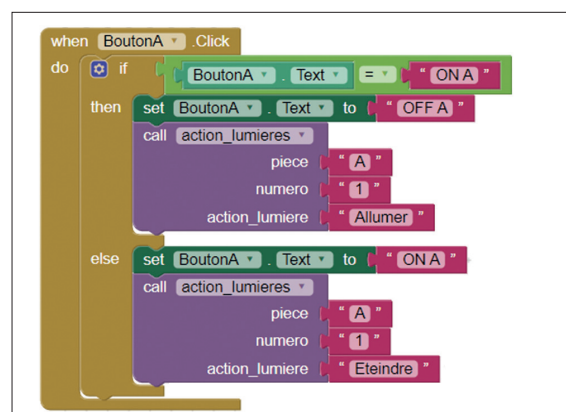


10 Interface de l'application sur AppInventor

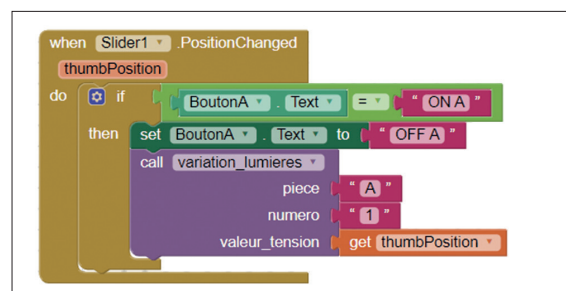
ressources associées à cet article. Il doit être importé et modifié en fonction du programme souhaité.

Les blocs de communication 8 sont répartis en deux : les blocs permettant d'agir avec les lumières, les volets, etc. ; les blocs permettant d'obtenir les valeurs des capteurs de la maison (ouverture de porte, état bouton-poussoir...) ou des données extérieures (température, vent, date...).

Sur AppInventor, dans la partie « Design » de HOME I/O, il est indispensable de mettre une barre de connexion 9. On rentre ici l'adresse IP du PC sur lequel tourne HOME I/O (on pourrait ainsi interagir avec plusieurs smartphones sur une seule maison virtuelle !). Pour déterminer l'adresse IP du PC sur



11 Programmation du BoutonA sur AppInventor



12 Programmation du Slider sur AppInventor

lequel est lancé HOME I/O, il suffit de taper « ipconfig » dans un éditeur DOS. Les blocs de commande associés à cette connexion sont déjà proposés dans le fichier exemple.

On peut ensuite faire glisser un bouton que l'on nommera BoutonA avec le texte « ON A » dans la zone « Design » de l'application 10.

Dans la zone « Programmation » ou « Blocks », on fait glisser le bloc permettant d'allumer ou éteindre une lampe d'une pièce et on le paramètre pour la lampe 1 de la pièce A (depuis la palette « Procédure »). Un appui sur le BoutonA 11 changera le texte en « OFF A » et allumera la pièce, un nouvel appui remettra le texte « ON A » et éteindra la lumière. On peut également jouer sur la couleur de fond du bouton en fonction de l'état.

Pour tester, on clique sur Connect/AI Companion et on scanne dans l'application AI Companion 2 le code-barres pour tester avec le smartphone. Il ne faut pas oublier de passer la lampe de la pièce A en mode externe dans HOME I/O.

On peut ainsi ajouter autant de boutons que l'on souhaite et allumer toutes les pièces de cette manière.

Pour découvrir l'objet Slider (curseur), on peut associer au BoutonA un curseur qui permettra de changer l'intensité lumineuse. Après avoir mis en place et en forme le Slider dans la zone « Design », on ajoute dans la partie « Programmation » 12, la fonction VariationLumiere qui permet d'envoyer une commande de 0 à 10 V pour modifier cette intensité

```

from HomeIO import *

maison=HomeIO()

tic=time.time()
while(time.time()-tic<10):
    intF1="F-Interrupteur Lumière 1"
    intF2="F-Interrupteur Lumière 2"
    print(maison.get(intF1),maison.get(intF2))

```

13 Code permettant de lire les valeurs des interrupteurs 1 et 2

```

tic=time.time()
while(time.time()-tic<10):
    etat=maison.get("F-Interrupteur Lumière 1")
    print(etat)
    if (etat==True):
        maison.set("F-Lumières 1",1)
    elif (etat==False):
        maison.set("F-Lumières 1",0)

```

14 Code pour utiliser un interrupteur bistable

```

tic=time.time()
while(time.time()-tic<10):
    etat=maison.get("F-Interrupteur Lumière 1")
    print(etat)
    if (etat==True):
        maison.set("F-Lumières 1",1)
    elif (etat==False):
        maison.set("F-Lumières 1",0)

```

15 Code de changement d'état de l'interrupteur

lumineuse en fonction de la position du Slider. On peut ajouter une condition permettant de ne pouvoir faire varier l'intensité que si le BoutonA est sur ON.

Une application typique de cette interaction smartphone-HOME I/O est la gestion d'une intrusion. En utilisant les informations des capteurs de présence dans chaque pièce, on enclenche l'alarme et on informe l'utilisateur sur son smartphone d'une présence dans une pièce donnée.

Utilisation en CPGE

En CPGE, la programmation en Python et la notion de diagrammes d'états sont apparues lors de la dernière réforme.

HOME I/O offre depuis le départ la possibilité de programmer en Python en utilisant VisualStudio et le langage IronPython. Un module Python a cependant été développé et permet d'accéder au serveur interne à HOME I/O depuis n'importe quel éditeur Python.

Dans HOME I/O, on passe en mode externe les éléments souhaités. Après avoir importé le module HOME I/O (fichier HOME I/O.py), on se connecte à la maison en tapant : « maison=Home I/O() ». Pour lire les données des entrées de HOME I/O ou les données externes, il suffit de taper « get(«Nom de l'entrée dans HOME I/O») ».

Par exemple, le code 13 permet de lire les valeurs des interrupteurs 1 et 2 de la pièce F et de les afficher dans la console Python pendant 10 secondes.

Pour interagir avec des sorties de HOME I/O (lumières, volets...), on tape « set(«nom de la sortie»,valeur) ». L'argument « valeur » peut être un booléen (*true/false*), un réel (tension entre 0 et 10 pour certaines sorties) ou une chaîne de caractères (« monter », « descendre », « stopper », « allumer », « éteindre », « fermer », « ouvrir », les mots-clés que l'on retrouve dans l'application Scratch ou AppInventor).

La programmation en Python permet de comprendre la difficulté à gérer des états différents, notamment dans le cas de l'utilisation d'interrupteur monostable.

Le code 14 est satisfaisant pour un interrupteur bistable, mais ne convient pas pour un interrupteur monostable.

Pour allumer une lumière avec un interrupteur monostable, on utilise une variable qui permet de savoir dans quel état se trouve l'interrupteur. Il faut alors détecter le changement d'état de l'interrupteur et non pas la valeur 15.

Une application typique de l'interaction smartphone-HOME I/O est la gestion d'une intrusion

Conclusion

L'environnement virtuel HOME I/O permet donc d'être géré de différentes manières en utilisant des langages ou logiciels de programmation spécifiques à un niveau d'étude donné. Il est ainsi simple de faire passer des notions élémentaires ou plus complexes d'informatique pilotant un système complexe dans le domaine de la domotique.

Le logiciel CONNECT I/O et le SDK fournis avec HOME I/O permettent d'aller encore plus loin pour l'enseignement au niveau post-bac, en offrant la possibilité d'interfacer HOME I/O avec tout type de contrôleur matériel (automate programmable industriel, microcontrôleur...) ou logiciel (Matlab, LabView...). Plusieurs applications à destination des étudiants en BTS, IUT, licence et master sont proposées à ce sujet (voir « En ligne »). Real Games développe également le logiciel FACTORY I/O qui permet de concevoir ou d'utiliser une usine virtuelle. Des solutions de programmation en Scratch, AppInventor, Python ou Yakindu pourraient être envisagées pour mener des études de systèmes encore plus complexes. ■

EN LIGNE

Le logiciel HOME I/O, des activités pédagogiques et des tutoriels sont téléchargeables sur : <https://teachathomeio.com/>

Tous les liens sur <http://eduscol.education.fr/sti/revue-technologie>