**Externaliser l’automatisme sur un microcontrôleur**

**(Hardware In the Loop)**

# Introduction

## Intérêts pédagogiques

Bien qu’il soit possible de réaliser des automates avec CONNECT I/O pour la commande logique des systèmes simulés sous HOME I/O, nous verrons dans cette activité comment déléguer et externaliser la partie commande d’un automatisme vers un microcontrôleur externe et physique. Cette thématique de travaux regroupe plusieurs intérêts pédagogiques :

* Piloter un large éventail de systèmes d’une maison,
* Comprendre la conception de système embarqué, et de contrôleur logique,
* Utiliser des composants réels à interfacer avec la maison virtuelle,
* Etudier l’architecture de l’application : la partie commande, et la partie opérative,
* Câbler les flux d’informations entre les plateformes et représenter leurs types.

## Remarques préliminaires

Il est important de signaler que cette activité utilise l’ensemble des concepts présentés dans l’activité *:* ***Réaliser un module d’E/S Arduino pour créer une interface avec HOME I/O***. En effet, le module d’E/S agit comme une passerelle entre le domaine virtuel (représenté par la simulation numérique : HOME I/O, ou la commande logique : CONNECT I/O) et le domaine réel (représenté par l’ensemble du matériel physique) nécessaire à la réalisation de l’activité.

En revanche, si vous ne disposez pas de plateforme de développement - A4 Autoprog - il est possible de simuler le microcontrôleur PIC28X2 embarqué, grâce au logiciel PICAXE Editor 6. Dans ce cas, le PC est l’hôte de la partie opérative et de la partie commande. Voir Activité : ***Utiliser HOME I/O Avec PICAXE Editor 6***

# Prérequis

## Logiciels informatiques

La réalisation de cette activité, présuppose une version à jour et installée des logiciels suivants :

* **HOME I/O v1.2.0** : Logiciel de simulation des parties opératives d’une maison. L’application intègre 174 systèmes simulés d’une maison, qui sont autant d’axes de réflexion dans l’élaboration d’une commande adaptée,
* **CONNECT I/O v1.2.0** : Logiciel de programmation destiné à recevoir la commande destinée à la partie opérative (sous forme de blocs connectés grâce à sa zone de programmation graphique). De plus certain de ces blocs, jouent le rôle de « plugin » pour permettre l’interfaçage de HOME I/O avec d’autres applications ou matériels externes,
* **ARDUINO IDE 1.0.6** : Fournit l’interface de programmation et les drivers nécessaires à la communication entre le PC et la carte Arduino Uno. Nous utilisons la plateforme Arduino dans le cadre de l’utilisation d’un module d’E/S avec CONNECT I/O. Télécharger Arduino 1.0.6 [ici](http://arduino.cc/download.php?f=/arduino-1.0.6-windows.exe).
* **LOGICATOR IDE** : Environnement de développement Intégré et simple d’utilisation. Ce logiciel fournit une zone de programmation automate sous forme de diagramme de flux et permet de charger celui-ci dans des microcontrôleurs PICAXE. Télécharger Logicator [ici](http://www.rev-ed.co.uk/software/lgc001.exe).

## Firmware Arduino

Le programme nommé Analyseur syntaxique est destiné au microcontrôleur pour qu’il puisse agir en qualité de module d’E/S. De ce fait, il est chargé de lire les informations présentes sur ses ports d’entrée et les envoie à la simulation (HOME I/O). De la même manière il écrit sur ses ports de sortie les informations émanant de la simulation. L’échange d’informations entre le microcontrôleur et CONNECT I/O se fait par la liaison série USB via le Port COM256 (voir *Annexe 1 : fixer le port COM d’un périphérique* relatif à l’activité : ***Réaliser un module d’E/S Arduino***).

## Bibliothèque de classe

L’utilisation d’un module E/S interfacé avec HOME I/O nécessite une des deux bibliothèques de classes « DAQUino\*» (Data Acquisition Arduino) suivantes :

* DAQUinoDigit.dll : Représente la bibliothèque de classes nécessaire à CONNECT I/O pour utiliser le module E/S de type ***digital*** au sein de sa zone de programmation graphique.
* DAQUinoDigitAnalog.dll : Représente la bibliothèque de classes nécessaire à CONNECT I/O pour utiliser le module E/S de type ***digital et analogique*** au sein de sa zone de programmation graphique.

Charger ces bibliothèques dans le répertoire «plugin » du dossier d’installation CONNECT I/O, pour rendre actif les plugins au sein de CONNECT I/O. Pour cela faire glisser et déposer ces fichiers au sein du répertoire « Plugins » du dossier d’installation de CONNECT I/O (ex : *C:\Program Files (x86)\Real Games\Connect IO\Plugins\DAQuino*)

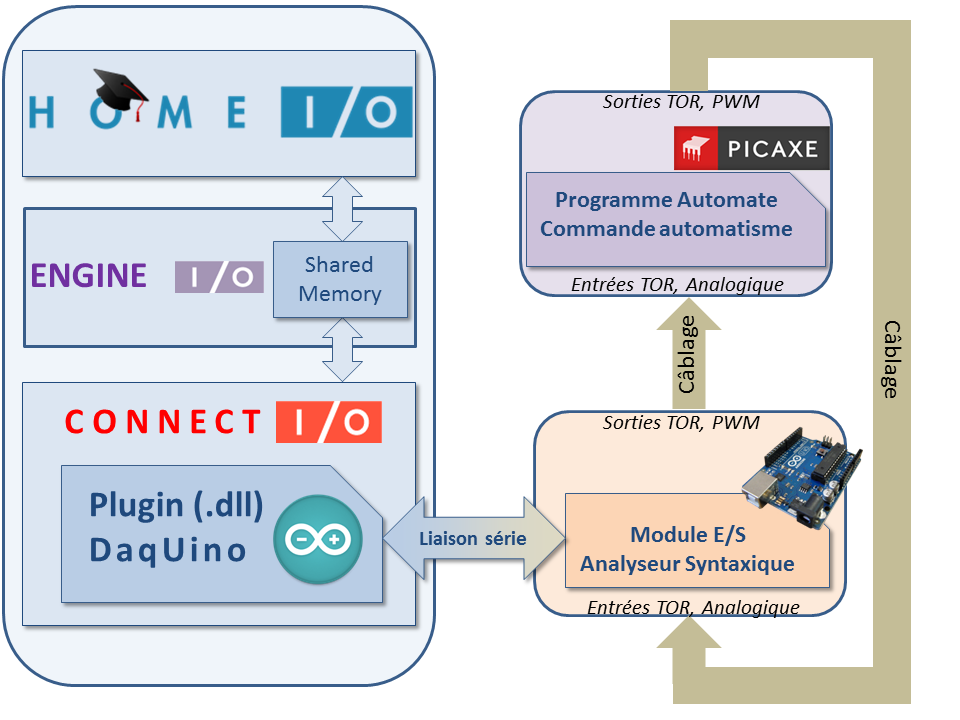
## Matériels

En plus des logiciels requis, il est nécessaire d’être équipé avec les éléments matériels suivants :

* Autoprog Uno : Dans cette activité, la carte Arduino UNO représente le module d’E/S physique intervenant dans l’acquisition des données impliqué dans cette interface. le microcontrôleur intégré à la carte exécute le programme « Analyseur Syntaxique ».
* Autoprog : Le microcontrôleur PIC28X2 intégré au boitier A4 représente la partie commande et exécute le programme automate.

# Présentation générale

## Architecture informatique



Dans cette expérience, l’architecture informatique présente 3 entités agissant de façon spécifique :

* L’ordinateur accueille la simulation virtuelle domotique HOME I/O, et son interface de programmation CONNECT I/O. La passerelle ENGINE I/O assure la communication des variables entre ces 2 logiciels complémentaires via l’écriture ou la lecture dans une zone mémoire partagée,
* Le module d’entrées sorties réalisé, via la plate-forme Arduino, agit comme une interface entre la simulation et la commande externalisée et embarquée au sein d’un microcontrôleur,
* Le microcontrôleur exécute l’automatisme lié à la commande du système simulé dans HOME I/O. Il met à jour ses données en sorties (principalement les commandes actionneurs) en fonction des donnée en entrée (principalement les informations capteurs).

Dans le cas où l’exercice induit l’utilisation de grandeurs analogiques, cette architecture nécessite l’utilisation de câble équipé de cellule filtrante dans le but transformer le signal PWM vers sa représentation analogique (voir annexe n°1 *: Transformation du signal PWM en grandeur Analogique*).

# Exemple applicatif

## Présentation du portail coulissant

### http://www.a4.fr/images/AP-APUno_categorie.jpgIntroduction

Dans un souci de facilité de mise en œuvre, nous utiliserons dans cet exemple les plates-formes de développements fourni par A4 :

* [AutoProg](http://www.a4.fr/systeme-autoprog_c774_1046.html) : boîtier à connecteur jack, embarquant un microcontrôleur PACAXE PIC28X2 à 8 Entrées digitales, 8 sorties digitales et 4 entrées Analogiques. Ce module agira en qualité d’automate dédié à la commande de l’application du portail coulissant.
* [AutoProg Uno](http://www.a4.fr/autoproguno_c2223.html): boîtier à connecteur jack, embarquant un microcontrôleur Arduino Uno à 16 Entrées/sorties digitales et 4 entrées Analogiques. Ce module interviendra dans le cadre de l’acquisition des E/S initiée par le PC pour communiquer les variables à CONNECT I/O.

En plus de réaliser le câblage désiré de manière simplifié grâce à la connectique jack, ces plates-formes sont accompagnées d’une interface de programmation ergonomique, assurant une prise en main rapide lors du développement de l’automate.

### Préparations préliminaires

* Placer les bibliothèques de classe DAQUino\*.dll et leurs fichier texte d’aide du même nom au sein du répertoire « Plugin » du dossier d’installation de CONNECT I/O en ayant préalablement créé un dossier pour accueillir les fichiers (ex : *C:\Program Files (x86)\Real Games\Connect IO\Plugins\DAQuino* ),
* Connecter la carte Arduino UNO au PC via la liaison USB,
* Attribuer le port COM 256 au périphérique Arduino Uno (voir *Annexe 1 : fixer le port COM d’un périphérique* relatif à l’activité : *Réaliser un module d’E/S Arduino*),
* Charger le programme « Parser.ino » dans le microcontrôleur intégré à l’Arduino Uno grâce à l’IDE Arduino.
* Connecter l’AutoProg au PC via la liaison USB.

## Manipulations

### Manipulation matérielle

Câbler les boîtiers AutoProg et AutoprogUno selon les correspondances suivantes :



### Manipulation logicielle

* Ouvrir le fichier “Portail coulissant 7-5.plf” (extrait du pack de programmes téléchargeable sur le site A4 :[*BE-APORT-COUL\_Programmes\_LOGICATOR\_060514.zip*](http://www.a4telechargement.fr/Portail_coulissant/BE-APORT-COUL_Programmes_LOGICATOR_060514.zip) ) à l’aide de Logicator.
* Cliquer sur  « Programme PIC » (ou Alt-F5) pour charger le programme automate du portail coulissant dans le microcontrôleur de l’Autoprog. Assurez-vous que le port COM relatif au PICAXE est le même que celui défini dans Logicator.
* Copier le fichier « PIXAXE\_HIL\_SlidingGate\_2015\_1\_6\_14\_47\_20.xml » dans le dossier de sauvegarde de HOME I/O (Généralement Documens/Home IO/Saves.
* Lancer HOME I/O et charger la sauvegarde "PIXAXE\_HIL\_SlidingGate\_2015\_1\_6\_14\_47\_20.xml".
* Ouvrir le fichier "PICAXE\_HIL\_SlidingGate.CONNECTIO" à l’aide de CONNECT I/O.
* Activer la source booléenne pour mettre en marche la communication.

## Tests

* Appuyer sur les touches digitales : #1 ou #2 de la télécommande. L’automate détecte l’ordre d’ouverture ou de fermeture seulement lorsqu’une des 2 touches est active.
* Remarquer le portail exécuter la commande ( ) générée par le microcontrôleur externe PIC28X2 en fonction des états des capteurs ( )

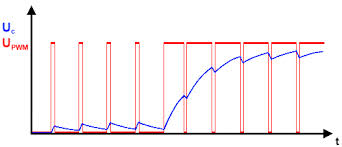
**Annexe 1**

**Transformation du signal PWM en grandeur Analogique**

# Présentation

Le signal PWM est une adaptation de la représentation binaire pour autoriser la codification d’une valeur analogique. Ce signal modulé en largeur d’impulsion induit une composante de rapport cyclique. Ce rapport détermine la valeur analogique que le signal PWM représente.

Dans le cadre de cette conversion, nous utiliserons un simple filtre de type Résistance Condensateur en vue de bloquer les fréquences hautes (celle de la modulation, ou du bruit) et de ne laisser passer seulement les variations de la valeur analogique.



Analogique

PWM

Cellule

RC

# Application

Signal

10µF

1kΩ

Masse

# Tester

* Charger le programme ci-dessous au sein de l’Autoprog Uno. Ce programme écrit la représentation PWM de la valeur 255 :

*void* ***setup****()*

*{*

*pinMode(9, OUTPUT);   // sets the pin as output*

*}*

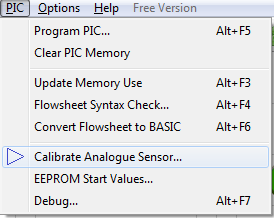
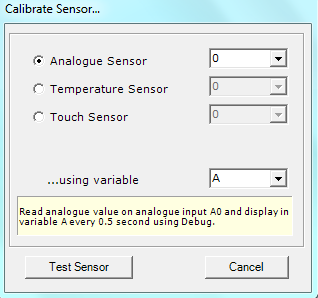
*void* ***loop****()*

*{*

*analogWrite(9, 255); //Ecrit 255 sur la broche 9*

*}*

* Câblez la sortie 9 de l’Autoprog Uno sur l’entrée Analogique A0 de l’Autoprog.
* Débranchez l’Autoprog Uno et branchez à la place l’Autoprog,
* Dans Logicator, cliquer sur « Calibrate Analogue Sensor » puis « Test Sensor »,

* Remarquez l’apparition de la grandeur continue, proche de 255.