

## Utiliser un Arduino Uno pour piloter HOME I/O

*Riera Bernard*

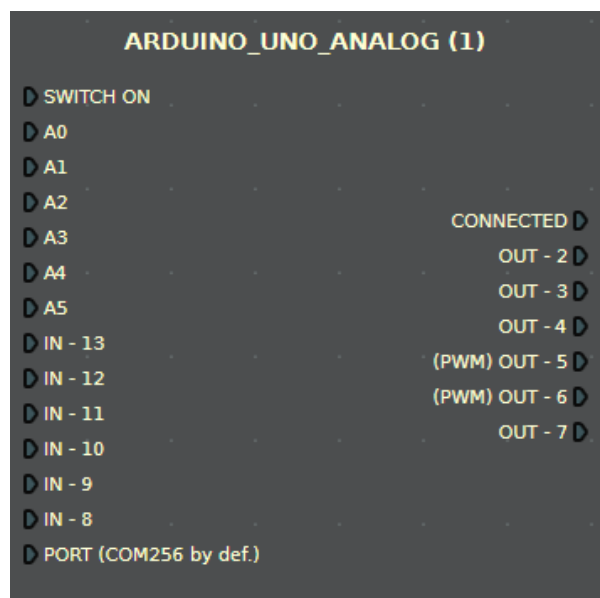
### Introduction

#### Objectifs

Ce document présente comment utiliser un Arduino Uno pour piloter HOME I/O au moyen de la liaison USB. Il devient ainsi possible de piloter HOME I/O sans avoir à disposer d'un DAQ ADVANTECH 4750 ou 4704. Le principe a consisté à développer 2 plugins Arduino pour CONNECT I/O : ARDUINO\_UNO\_DIGIT (E/S TOR) et ARDUINO\_UNO\_ANALOG (E/S TOR et E analogiques). Ces plugins permettent la communication avec un Arduino UNO au travers la liaison USB et l'échange des valeurs d'E/S.

#### Fonctionnement

Les plugins ARDUINO\_UNO\_DIGIT et ARDUINO\_UNO\_ANALOG permettent de connecter de façon logicielle des capteurs de HOME I/O aux entrées de l'Arduino Uno, et de récupérer l'état des sorties de l'Arduino pour activer des actionneurs de HOME I/O. Les E/S de l'Arduino Uno ne sont donc pas physiquement reliées aux E/S de HOME I/O. Pour programmer la commande, il suffit d'écrire son code à la fin du fichier Arduino\_DIGIT\_ANALOG.ino fourni et de le téléverser dans l'Arduino Uno.



#### Finalités

Grâce à ces 2 plugins, il devient possible de créer facilement des programmes pour l'Arduino Uno qui vont piloter HOME I/O.




Le plugin ARDUINO\_UNO\_DIGIT permet de gérer 6 entrées digitales (de IN-8 à IN-13) et 6 sorties digitales (de OUT-2 à OUT-7) de l'Arduino Uno.

Le plugin ARDUINO\_UNO\_ANALOG permet de gérer 6 entrées digitales (de IN-8 à IN-13), 6 entrées analogiques (de A0 à A5), 4 sorties digitales (de OUT-2, OUT3, OUT-4 et OUT-7) et 2 sorties analogiques en PWM (OUT-5 et OUT-6) de l'Arduino Uno.

## Prérequis

### Logiciels informatiques

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

- **HOME I/O v1.7** : Cette maison virtuelle intègre 174 objets (lumières, alarmes, chauffage, porte de garage, portail, ...) qu'il est possible de piloter. 
- **CONNECT I/O v1.2.7** : Automate Programmable logiciel qui s'interface très simplement avec HOME I/O et permet de faire la passerelle entre différentes technologies tierces logicielles ou matérielles au travers des plugins. La programmation se fait de façon graphique en reliant des blocs entre eux. 
- **ARDUINO IDE 1.8.16 ou supérieur** : Fournit l'interface de programmation et les drivers nécessaires à la communication entre le PC et la carte Arduino UNO. 

### Firmware Arduino

Le programme à mettre dans l'Arduino est le programme fourni **Parser\_Arduino\_DIGIT\_ANALOG.ino** dans lequel le code de commande de HOME I/O sera ajouté à la fin, en utilisant les E/S « réels » de l'arduino Uno. Ce programme contient d'une part le « Parser » (Analyseur syntaxique) qui permet de communiquer avec le PC au travers la liaison USB, et d'autre part le programme de commande. De ce fait, il est chargé de lire les informations présentes sur les ports d'entrée émulés de l'Arduino Uno par CONNECT I/O. De la même manière les données écrites sur les ports de sortie de l'Arduino Uno sont envoyées à CONNECT I/O. L'échange d'informations entre l'Arduino UNO et CONNECT I/O se fait par la liaison série USB via le Port COM. Par défaut le port COM256 est indiqué. Pour le modifier, il suffit de créer une source de type string en indiquant le port utilisé (COM5 par exemple).

### Bibliothèque de classe

Pour disposer des plugins Arduino dans CONNECT I/O, les deux dll (bibliothèques de classes) suivantes sont nécessaires :

- **Arduino\_UNO\_Digit.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.
- **Arduino\_UNO\_Digit\_Analog.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.

Il est nécessaire de charger ces bibliothèques dans le répertoire «plugins» du dossier d'installation CONNECT I/O, pour rendre actifs les plugins au sein de CONNECT I/O. Pour ce faire il suffit de glisser et déposer le dossier **Arduino\_UNO** au sein du répertoire « Plugins » du dossier d'installation de CONNECT I/O (ex : C:\Program Files (x86)\Real Games\Connect IO\Plugins\Arduino\_UNO)

## Matériels

Il est nécessaire de disposer d'un Arduino UNO.

## Exemple

### Présentation

Pour illustrer le fonctionnement de ces 2 plugins, le fichier **Arduino\_UNO\_ANALOG\_sample.CONNECTIO** est fourni. Celui-ci est à utiliser avec le programme exemple fourni dans le fichier **Parser\_Arduino\_DIGIT\_ANALOG.ino**.

### Préparations préliminaires

- Après l'avoir téléchargé et débloqué, placer le dossier **Arduino\_UNO** dans le répertoire « Plugins » du dossier d'installation de CONNECT I/O (ex : *C:\Program Files (x86)\Real Games\Connect IO\Plugins*) ;
- Connecter la carte Arduino UNO au PC via la liaison USB ;
- Attribuer le port COM au périphérique Arduino UNO ;
- Téléverser le programme « **Parser\_Arduino\_DIGIT\_ANALOG.ino** » dans l'Arduino UNO grâce à l'IDE Arduino.

### Manipulation logicielle

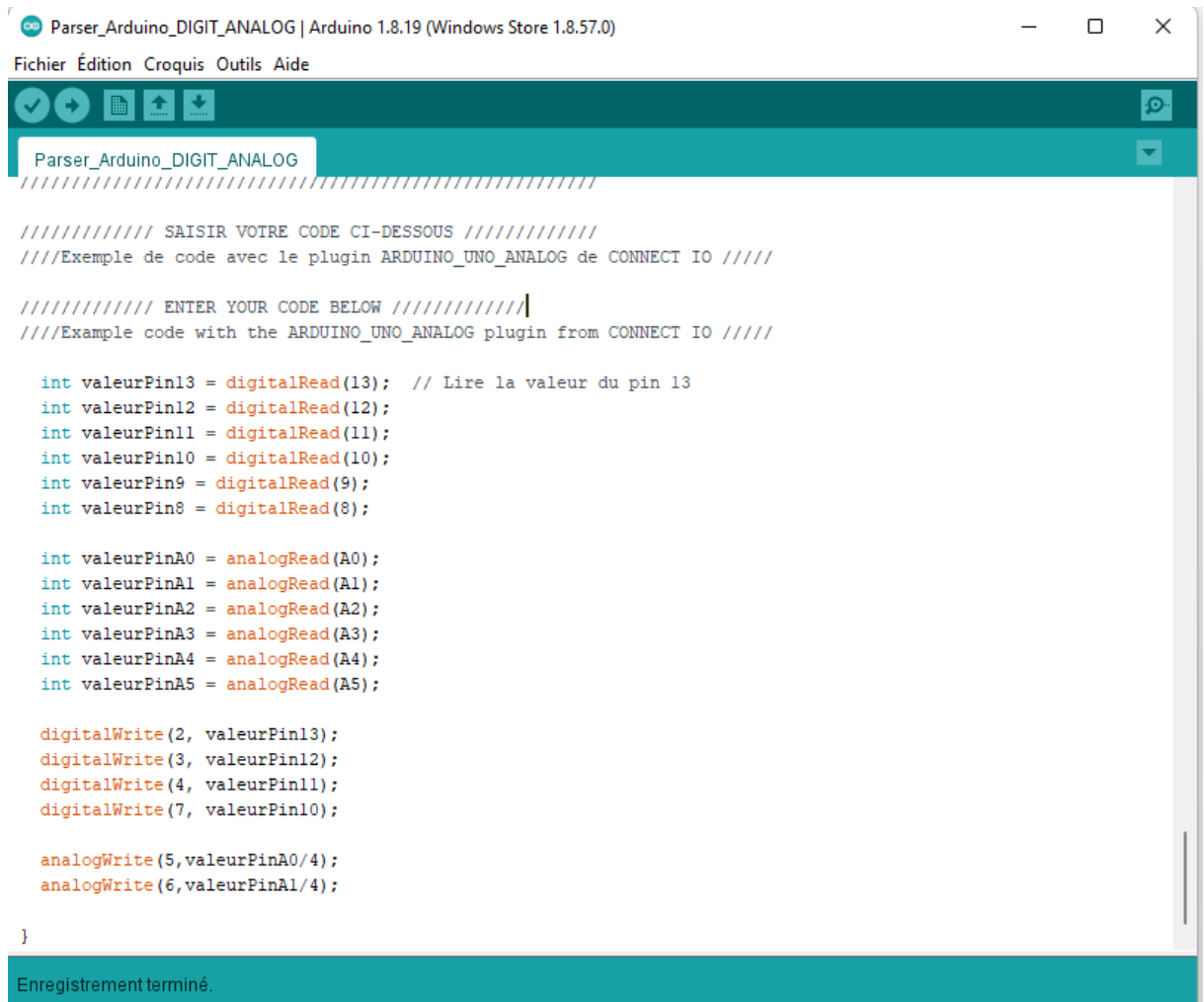
- Démarrer CONNECT I/O et charger le fichier « **Arduino\_UNO\_ANALOG\_sample.CONNECTIO** » ;
- Modifier le port, pour indiquer celui utilisé par votre Arduino UNO. ;
- Activer la source digitale connectée à l'entrée « SWITCH ON » du plugin afin de basculer celui-ci en mode marche ;
- Vérifier que la sortie « CONNECTED » du plugin soit bien activée. Si ce n'est pas le cas, vérifier le port COM utilisé par l'Arduino UNO.

### Tester le programme

Le programme proposé effectue les opérations suivantes :

- Recopie de l'entrée digitale IN-13 sur la sortie OUT2
- Recopie de l'entrée digitale IN-12 sur la sortie OUT3
- Recopie de l'entrée digitale IN-11 sur la sortie OUT4
- Recopie de l'entrée digitale IN-10 sur la sortie OUT7
- Recopie de l'entrée analogique A0 sur la sortie PWM OUT-5
- Recopie de l'entrée analogique A1 sur la sortie PWM OUT-6

On notera que le CAN des entrées analogiques sur l'Arduino UNO est sur 10 bits et retourne donc une valeur de 0 (0 v) à 1023 (5 v). En revanche les sorties PWM sont sur 8 bits, de 0 (0 v) à 255 (5 v). Une mise à l'échelle est donc réalisée en divisant par 4 les valeurs de A0 et A1.



```
Parser_Arduino_DIGIT_ANALOG | Arduino 1.8.19 (Windows Store 1.8.57.0)
Fichier Édition Croquis Outils Aide

Parser_Arduino_DIGIT_ANALOG

////////// SAISIR VOTRE CODE CI-DESSOUS //////////
////Exemple de code avec le plugin ARDUINO_UNO_ANALOG de CONNECT IO ////

////////// ENTER YOUR CODE BELOW //////////|
////Example code with the ARDUINO_UNO_ANALOG plugin from CONNECT IO ////

int valeurPin13 = digitalRead(13); // Lire la valeur du pin 13
int valeurPin12 = digitalRead(12);
int valeurPin11 = digitalRead(11);
int valeurPin10 = digitalRead(10);
int valeurPin9 = digitalRead(9);
int valeurPin8 = digitalRead(8);

int valeurPinA0 = analogRead(A0);
int valeurPinA1 = analogRead(A1);
int valeurPinA2 = analogRead(A2);
int valeurPinA3 = analogRead(A3);
int valeurPinA4 = analogRead(A4);
int valeurPinA5 = analogRead(A5);

digitalWrite(2, valeurPin13);
digitalWrite(3, valeurPin12);
digitalWrite(4, valeurPin11);
digitalWrite(7, valeurPin10);

analogWrite(5, valeurPinA0/4);
analogWrite(6, valeurPinA1/4);

}

Enregistrement terminé.
```

Pour vérifier le bon fonctionnement, il suffit d'activer les sources digitales de CONNECT I/O en entrées du plugin « ARDUINO\_UNO\_ANALOG » et de constater que les sorties OUT-2, OUT-3, OUT-4 et OUT-7 s'activent également.

Enfin, en modifiant les valeurs des entrées analogiques de A0 et A1, vous devez constater une recopie de A0 et A1, respectivement sur OUT-5 et OUT-6.

Vous pouvez maintenant piloter HOME I/O en connectant les capteurs et les actionneurs de HOME I/O qui vous intéressent.

A vous de jouer !