

Utiliser un Arduino comme boîtier d'E/S pour HOME I/O

Ramel Sébastien

Riera Bernard

Introduction

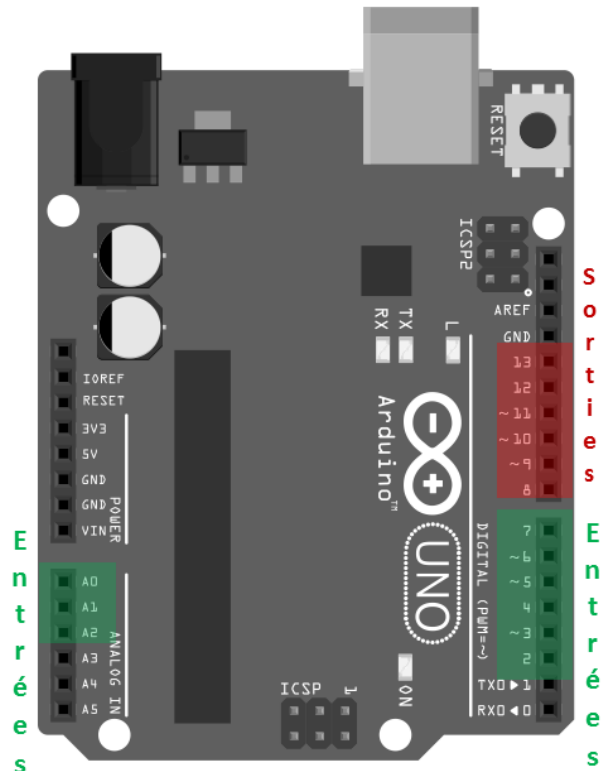
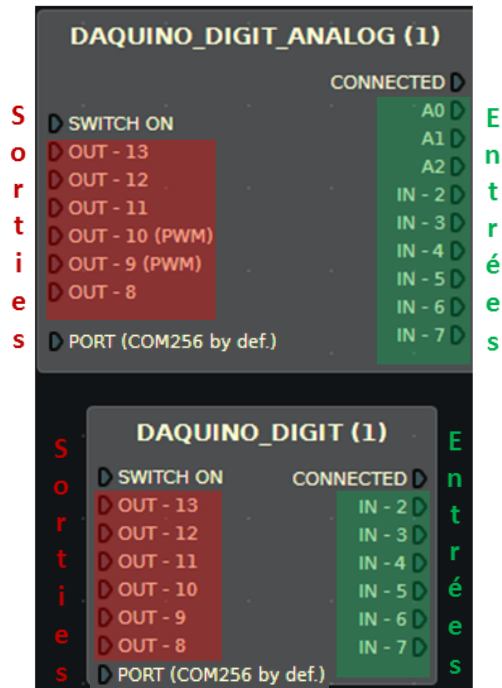
Objectifs

Ce document présente comment utiliser un Arduino comme boîtier USB d'Entrées/Sorties (E/S) « Tout ou Rien » (TOR) et analogiques. Il devient ainsi possible de disposer d'un DAQ USB E/S bon marché qui peut ainsi avantageusement remplacer un DAQ ADVANTECH 4750 ou 4704. Le principe a consisté à développer 2 plugins Arduino (DAQUINO_DIGIT (E/S TOR) et DAQUINO_DIGIT_ANALOG (E/S TOR et E analogiques)) pour CONNECT I/O. Ces plugins permettent la communication avec un Arduino UNO au travers la liaison USB et l'échange des valeurs d'E/S. Il devient alors possible de faire interagir le monde réel avec le monde virtuel de HOME I/O.

Fonctionnement

Le plugin DAQUino permet d'activer les broches de sorties TOR de l'Arduino et de récupérer la valeur des Entrées de l'Arduino. Les possibilités offertes sont donc nombreuses pour HOME I/O. On peut par exemple utiliser des capteurs réels avec une commande réalisée dans CONNECT I/O, ou encore utiliser des actionneurs virtuels, ou encore se connecter à une Partie Commande externe (microcontrôleur, Automate Programmable Industriel, ...). Les **entrées physiques** de la carte Arduino sont donc des **sorties du plugin** (voir *figure 1*). De la même façon, Les **entrées du plugin** sont les **sorties physiques** de la carte Arduino.

Plugins Arduino



Circuit Imprimé : Arduino Réel

Finalités

Grâce à ces plugins, il sera dès lors possible d'interfacer les parties opératives simulées de l'application HOME I/O avec :

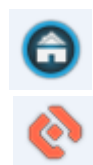
- Des capteurs ou des actionneurs réels (LEDs, boutons, interrupteurs, sondes de température, ...)
- Des automates externes pour externaliser la partie commande liée à un automatisme.

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** : 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.



Firmware Arduino

Le programme à mettre dans l'Arduino est un « Parser » (Analyseur syntaxique) qui permet de communiquer avec le PC au travers la liaison USB. De ce fait, il est chargé de lire les informations présentes sur ses ports d'entrée et les envoie à CONNECT I/O. De la même manière il écrit sur ses ports de sortie les informations envoyées par 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) « DAQuino* » (Data Acquisition Arduino) suivantes sont nécessaires :

- 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 ce faire glisser et déposer ces fichiers dans le répertoire DAQuino 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

Il est nécessaire de disposer également des éléments matériels suivants :

- Arduino UNO : Dans cette activité, la carte Arduino UNO représente le module d'E/S physique intervenant dans l'acquisition des données impliquée dans cette interface. le microcontrôleur intégré à la carte exécute le programme « Analyseur Syntaxique ».

Enfin, en vue d'être utilisé dans une application, quelques composants optionnels sont à prévoir en fonction de l'application :

- Des éléments capteur ou actionneur : dans le cadre de l'utilisation de la simulation de partie opérative d'une maison interfacée avec des composants réels, prévoir les capteurs qui entreront en compte dans la partie commande, ou prévoir les actionneurs qui recevront une commande.

- Un automate tiers : Utilisez un automate externe (un microcontrôleur par exemple) pour externaliser la commande de certains automatismes. (voir l'activité : *Externaliser la commande sur un microcontrôleur type PICAXE*)

Présentation générale

Architecture Informatique

L'architecture matérielle obéit au principe du modèle maître esclave et intègre donc 2 systèmes informatiques. Le protocole de communication entre ces 2 entités utilise le canal de Liaison série USB.

- L'ordinateur joue le rôle de maître et envoie des commandes et des requêtes à l'esclave que celui-ci exécute
- Le Microcontrôleur fonctionne en tant qu'esclave, il reçoit les commandes et requêtes du maître, les exécute et retourne le résultat (dans le cas d'une requête) par le même canal dont il a reçu la commande : la liaison série USB via un port COM.

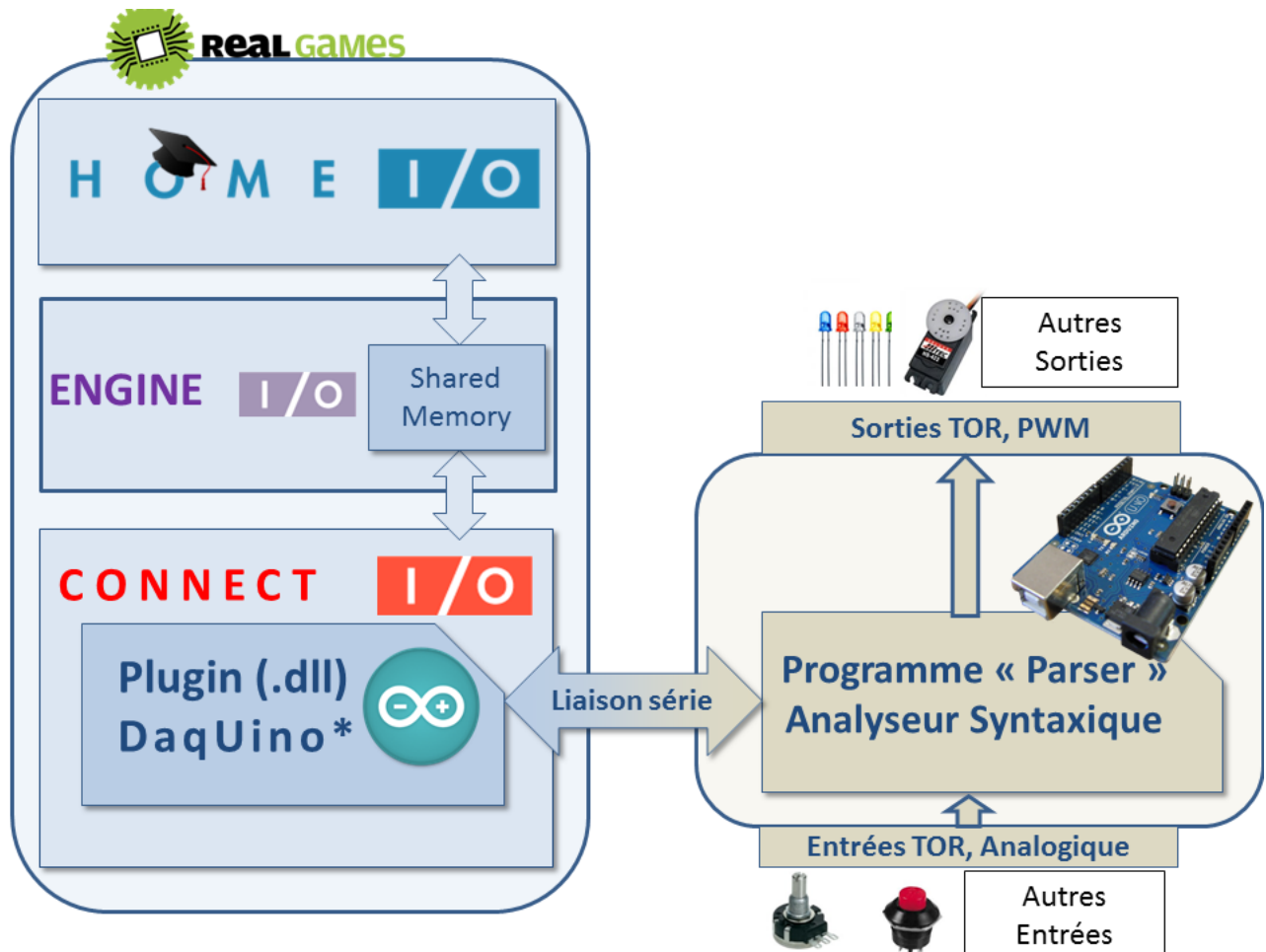
Architecture Logicielle

ENGINE I/O

L'ordinateur est le support d'exécution des applications HOME I/O et CONNECT I/O. De plus grâce à la technologie ENGINE I/O, une certaine partie de la mémoire attribuée à l'exécution de ces applications est partagée et commune entre elles. Ce principe de fonctionnement permet à ces deux applications d'échanger des informations.

Plugin DAQUino*.dll

Le module d'E/S à l'étude dans ce document s'utilise au sein du logiciel CONNECT I/O sous la forme d'un plugin. Il correspond à un fichier type **bibliothèque de classes (.dll)** qui initialise des attributs et implémente les méthodes pour la communication des informations à travers la liaison série avec l'interface E/S Arduino. Du point de vue du plugin - DAQUino* - sous CONNECT I/O, les informations présentes en sortie du bloc représentent les états courants des variables temps réel liées à HOME I/O alors que les informations en entrée représentent les grandeurs associées à des dispositifs matériels physiques (capteurs, actionneurs) interfacés avec la simulation via le module d'E/S.



Exemples applicatifs

Gradateur de lumière

Présentation

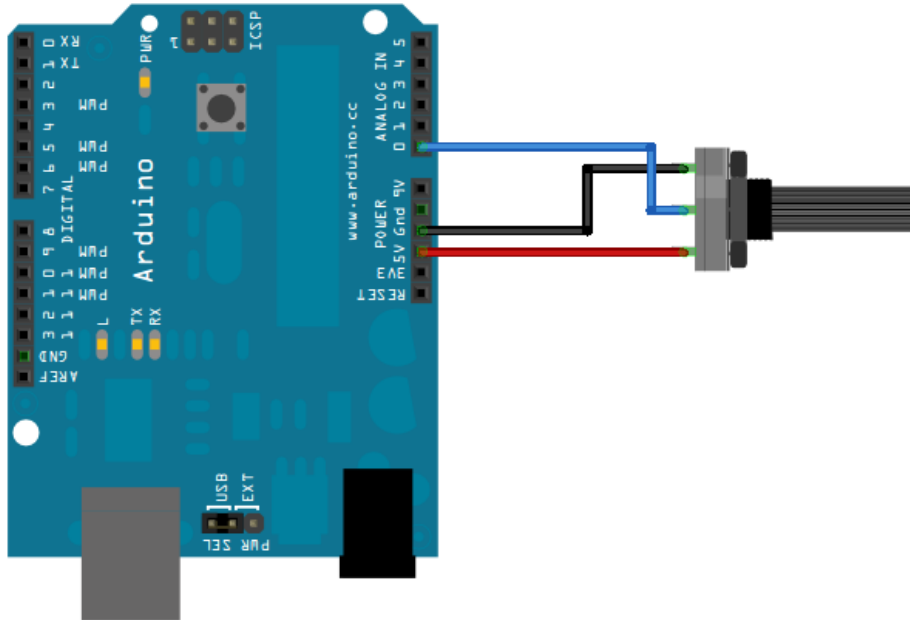
Le gradateur de lumière permet de commander une lumière de façon continue et dynamique. Dans ce cadre, il est nécessaire d'utiliser le plugin assurant le traitement des informations de type analogiques (DAQUinoDigitAnalog.dll). Cet exemple applicatif fait intervenir un dispositif matériel de type potentiomètre connecté à l'entrée analogique de la carte Arduino UNO afin de commander une lumière dont l'intensité lumineuse représente l'image de la position angulaire du potentiomètre.

Préparations préliminaires

- Placer la bibliothèques de classe DAQUinoDigitAnalog.dll au sein du répertoire « Plugin » du dossier d'installation de CONNECT I/O en ayant préalablement créer un dossier pour accueillir le fichier (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 au périphérique Arduino UNO,
- Téléverser le programme « Parser.ino » dans l'Arduino UNO grâce à l'IDE Arduino.

Manipulation Matériel

- Connecter le potentiomètre à la carte Arduino UNO, tel qu'illustré sur l'exemple ci-dessous :



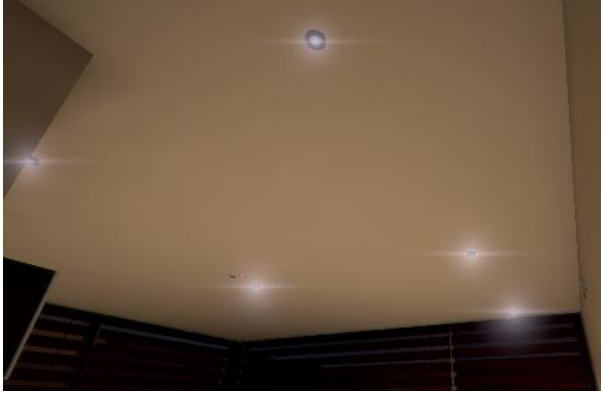
Manipulation logicielle

- Démarrer CONNECT I/O et charger le fichier « Arduino_GradateurLumiere.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 gradateur de lumière

Sous HOME I/O :

- Démarrer HOME I/O.
- Placez-vous dans la pièce dans laquelle vous avez attribué le mode « externe » (bleu) à la lumière,
- Tourner le potentiomètre et visualiser l'évolution lumineuse dans la pièce.



Interfacer avec un automate externe

Se référer à l'activité : Externaliser l'automatisme sur un Microcontrôleur type PICAXE

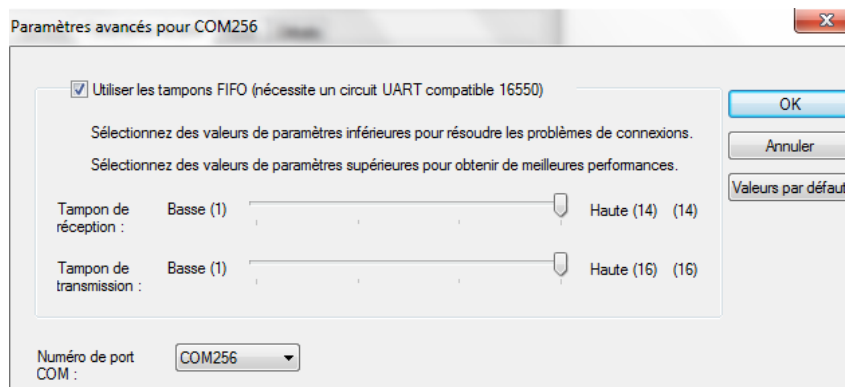
Annexe 1 Fixer le port COM d'un périphérique

Introduction

Pour effectuer cette manipulation, vous devez avoir préalablement installé les drivers Arduino encapsulé avec l'IDE. De plus, vous devez disposer des droits administrateurs pour mener cette configuration.

Fixer le port de communication avec Arduino

- Brancher la carte Arduino UNO au PC,
- Aller à, *Panneau de configuration\Matériel et audio\Périphériques et imprimantes*,
- Clic droit sur le périphérique « Arduino UNO », puis Clic sur « propriété »,
- Clic sur onglet « matériel » puis clic sur le bouton « propriété »
- Dans l'onglet « paramètre du port » cliquez sur « Avancé... »



- Enfin, attribuer le numéro de port à COM256 dans la liste déroulante « numéro de port COM »