

Use an Arduino UNO as Input/Output Module to create an interface with HOME I/O

Ramel Sébastien & Riera Bernard

Introduction

Objectives

The work presented in this document, leads to the realization of a binary and analog data acquisition module that remain – in a virtual way - within CONNECT I/O and HOME I/O or - in a real context - with the use of physical sensors and actuators connected to the module. This physical input, output module, is realized via the platform (hardware and software) Arduino and it acts as an interface to communicate with HOME I/O with the use of a plugin in CONNECT I/O.

Operating Principle

The “DAQUino” plugin used in CONNECT I/O is not the virtual equivalent of the real component Arduino. Indeed the inputs and outputs of the real component (the Arduino UNO board) are the opposite of the virtual component (plugin), so that it can transmit the information between the two areas (real and virtual). Typically, to pass information from the real to the virtual context, we connect real sensors in the **physical inputs** of the Arduino UNO, however we retrieve these data in the virtual area via the **virtual outputs** of the plugin (see *figure 1*)

Purposes



Thanks to this entity, it will therefore be possible to interface the operative part, simulated in HOME I/O with:

- Real Sensors or actuators (LEDs, buttons, switches, sensors of temperature...),
- External controllers to outsource the control part related to an automation system.

Prerequisites

Computer software

The achievement of this activity presupposes an installed version of the following software:

- **HOME I/O v1.7.0:** Software which provides simulation of operative part of a house. This application incorporates 174 simulated systems of a classical house, which are all reflection axes in the development of an appropriate command, 
- **CONNECT I/O v1.2:** Programming software intended to receive the command part of the simulated operative part (represented as blocks connected within a graphical area). Some of these blocks, act as plugin useful for interfacing HOME I/O and CONNECT I/O with other applications or external devices.
- **ARDUINO IDE 1.8.16:** provides the programming interface and the drivers required for communicating between the PC and the Arduino Uno board. We use Arduino platform to perform data acquisition within CONNECT I/O. 

Firmware Arduino

Program named "parser.ino" is executed within the microcontroller of Arduino UNO PCB, so that it could act as Inputs/Outputs module. The program is responsible for reading the information on its physical input ports and sends them to simulation (HOME I/O). In the same way, it writes on its output ports, information which come from the simulation. The exchange of information between the Arduino UNO and CONNECT I/O is done by the USB serial link via the COM port. By default the COM256 port is indicated. To modify it, you just have to create a string source indicating the port used (COM5 for example).

Class library

The use of an I/O module interfaced with HOME I/O requires one of two following class libraries "DAQUino *":

- "DAQUinoDigit.dll": Represents the class library needed by CONNECT I/O to use **digital** inputs/outputs module within its graphical programming area.
- "DAQUinoDigitAnalog.dll": Represents the class library needed by CONNECT I/O to use **digital and analogue** inputs/outputs module within its graphical programming area.

Drag and drop these libraries in the CONNECT I/O installation folder named: "plugin" to make those plugins available within CONNECT I/O. (ex:C:\Program Files (x 86) \Real Games\Connect IO\Plugins\DAQuino).

Materials

In addition to the required software, it is necessary to be equipped with the following hardware:

- Printed Circuit Arduino Uno board: In this activity, the Arduino UNO board represents the physical I/O module used in data acquisition involved in this interface. The onboard microcontroller executes the "Parser.ino" program.

Finally, to be used in an application, some optional components are required, depending on the desired application:

- Sensor or actuator elements: in the context of using the simulation of operating part of a House interfaced with real components. Provide sensors that will be used in control part, or provide actuators that will receive a command.
- A third-party PLC: use an external controller (such as a microcontroller) to outsource some automation control. (see activity: *outsource the command on a type PICAXE microcontroller*)

Overview

Computing architecture

The architecture follows the master slave model principle within 2 computer systems. The communication protocol between these 2 entities uses the serial USB channel.

- The computer acts as the master and sends commands and queries to the slave, the Arduino PCB
- The microcontroller works as a slave, it receives commands and queries from the master, executes them and returns the result (in the case of a request) in the same channel from which it received the request : the serial USB via the COM Port.

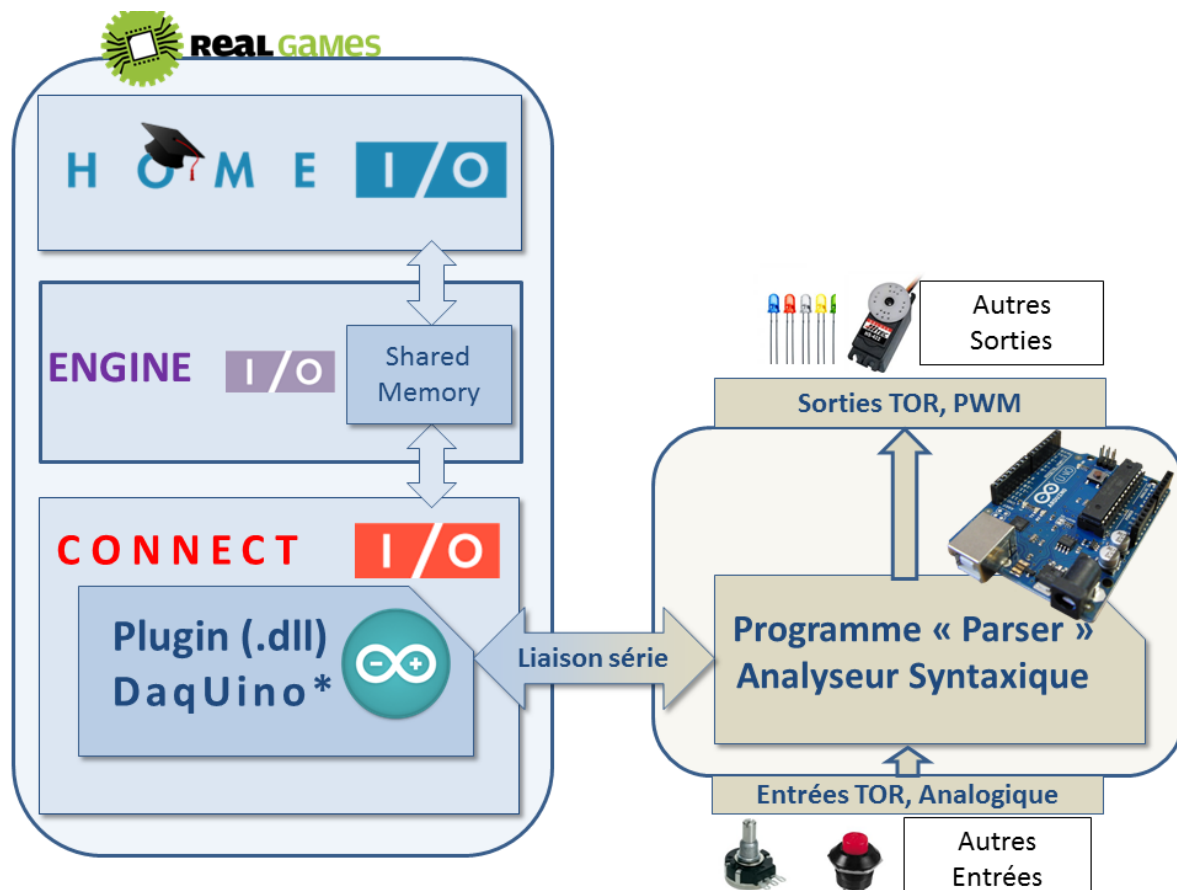
Software architecture

ENGINE I/O

The computer is the runtime support for the execution of HOME I/O and CONNECT I/O applications. In addition, the technology ENGINE I/O, a part of the memory allocated to run these 2 applications is shared and common between them. This operating principle allows these complementary applications to exchange information.

Plugin DAQUino*.dll

The IO module is used within the software CONNECT I/O via a plugin. It corresponds to a **class library (.dll)** that initializes attributes and implements methods for communication between the master and the slave via serial connection. From the point of view of the plugin - DAQUino* - within CONNECT I/O, the information present in the outputs of this block represents current states of real time variables of HOME I/O, however inputs information represents quantities associated with physical hardware devices (sensors, actuators) interfaced with simulation via the IO module.



Application examples

Dimmer

Presentation

The dimmer controls a light in a continuous and dynamic way. In this context, it is necessary to use the "DAQUino" plugin related to the manipulation of analog data ("DAQUinoDigitAnalog.dll"). This

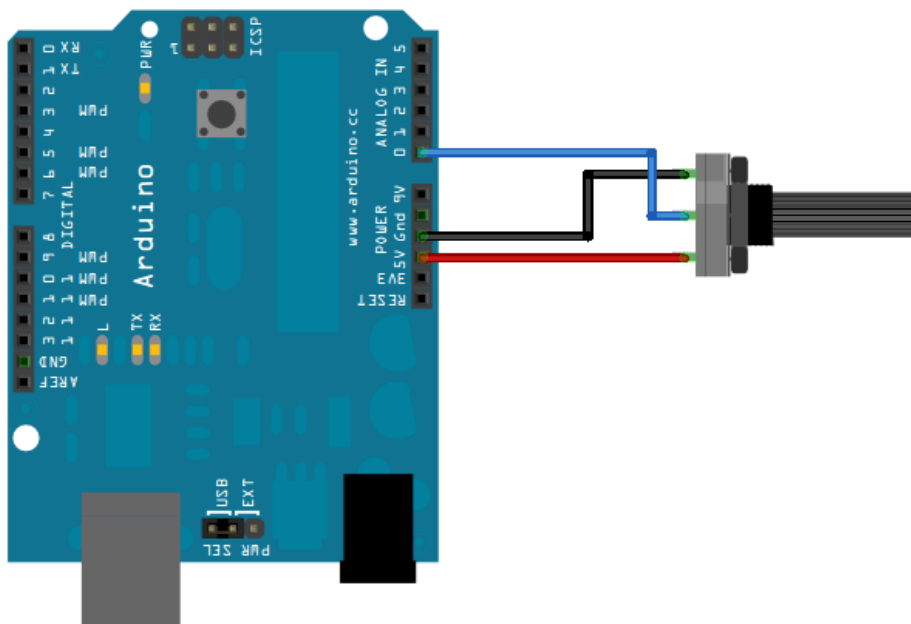
application example requires a potentiometer connected to an analog input of the Arduino UNO board to drive light whose intensity represents the image of the angular position of the potentiometer.

Preliminary preparations

- Put the DAQUinoDigitAnalog.dll class library in the directory "Plugin" from CONNECT I/O installation folder by having previously create a folder to receive the file (ex: *C:\Program Files (x86)\Real Games\Connect IO\Plugins\DAQuino*),
- Connect the Arduino Uno to the PC via the USB connection, card
- Set the Arduino UNO device COM port index to : 256 (see *Annex 1: set a constant COM port index of a device*),
- Upload the program «Parser.ino» in the microcontroller integrated inside the Arduino UNO board via the Arduino IDE.

Material handling

Connect the potentiometer to the Arduino UNO BOARD, as shown in the following example:



Software handling

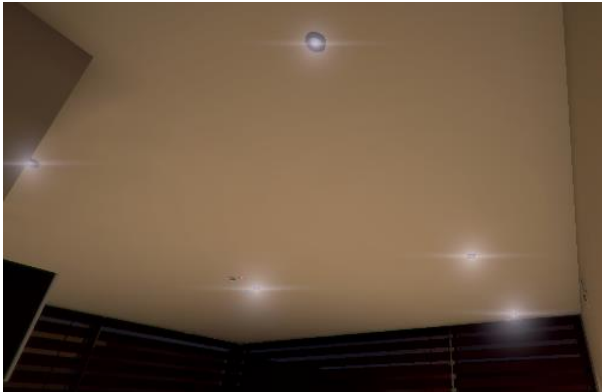
- Start CONNECT I/O and load the file " Arduino_GradateurLumiere.CONNECTIO ".
- Change the port, to indicate the one used by your Arduino UNO.
- Activate the digital source connected to the "SWITCH ON" input of the plugin to switch it on.
- Check that the "CONNECTED" output of the plugin is activated. If it is not, check the COM port used by the Arduino UNO.

Test the light dimmer

Within HOME I/O:

- Place you in the room in which you assigned a light actuators in external mode.

- Turn the potentiometer and visualize the evolution of the light.



Interface with an external PLC

Refer to the activity: *Outsource the automation on a type PICAXE microcontroller.*