Using an Arduino Uno to control HOME I/O

Riera Bernard

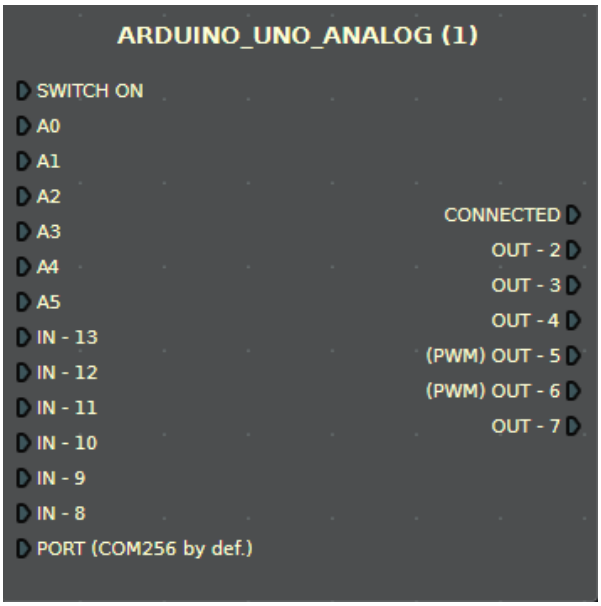
Introduction

Objectives

This document shows how to use an Arduino Uno to control HOME I/O via USB. This makes it possible to control HOME I/O without the need for an ADVANTECH 4750 or 4704 DAQ. The idea was to develop 2 Arduino plugins for CONNECT I/O: ARDUINO_UNO_DIGIT (digital I/O) and ARDUINO_UNO_ANALOG (digital and analog I/O). These plugins enable communication with an Arduino UNO via USB, and the exchange of I/O values.

Operating principle

The ARDUINO_UNO_DIGIT and ARDUINO_UNO_ANALOG plugins enable software connection of HOME I/O sensors to Arduino Uno inputs, and retrieval of the status of Arduino outputs to activate HOME I/O actuators. The Arduino Uno I/Os are not physically connected to the HOME I/Os. To program the command, simply write its code at the end of the Arduino_UNO_DIGIT_ANALOG.ino file supplied and upload it to the Arduino Uno.



Purpose

Thanks to these 2 plugins, you can easily create programs for the Arduino Uno to control HOME I/O.

The ARDUINO_UNO_DIGIT plugin manages 6 digital inputs (from IN-8 to IN-13) and 6 digital outputs (from OUT-2 to OUT-7) on the Arduino Uno.

The ARDUINO_UNO_ANALOG plugin manages 6 digital inputs (from IN-8 to IN-13), 6 analog inputs (from A0 to A5), 4 digital outputs (from OUT-2, OUT3, OUT-4 and OUT-7) and 2 analog PWM outputs (OUT-5 and OUT-6) of the Arduino Uno.

Prerequisites

Computer software

To carry out this activity, you need to have an up-to-date and installed version of the following software:

- **HOME I/O v1.7:** This virtual house integrates 174 objects (lights, alarms, heating, garage door, gate, etc.) that can be controlled.
- **CONNECT I/O v1.2.7:** Programmable Logic Controller (PLC) software that interfaces seamlessly with HOME I/O, enabling you to bridge the gap between different third-party software or hardware technologies via plug-ins. Programming is done graphically by linking blocks together.
- **ARDUINO IDE 1.8.16 or higher:** Provides the programming interface and drivers needed for communication between the PC and the Arduino UNO board.

Firmware Arduino

The program to be installed in the Arduino is the supplied **Arduino_UNO_DIGIT_ANALOG.ino**, to which the HOME I/O control code will be added at the end, using the "real" I/O of the Arduino Uno. This program contains both the Parser, which communicates with the PC via USB, and the control program. As such, it is responsible for reading the information present on the Arduino Uno's emulated input ports via CONNECT I/O. Similarly, data written to the Arduino Uno's output ports is sent to CONNECT I/O. Information is exchanged between the Arduino UNO and CONNECT I/O via the USB serial link, using the COM port. The default port is COM256. To modify it, simply create a string source indicating the port used (COM5, for example).

Class library

To use the Arduino plug-ins in CONNECT I/O, the following two dlls (class libraries) are required:

- **Arduino_UNO_Digit.dll:** Represents the class library required by CONNECT I/O to use the digital I/O module within its graphical programming area.
- **Arduino_UNO_Digit_Analog.dll:** Represents the class library required by CONNECT I/O to use the digital and analog I/O modules within its graphical programming area.

These libraries must be loaded into the "plugins" directory of the CONNECT I/O installation folder, to make the plugins active within CONNECT I/O. To do this, simply drag and drop the **Arduino_UNO** folder into the "Plugins" directory of the CONNECT I/O installation folder (e.g.: *C:\Program Files (x86)\Real Games\Connect IO\Plugins\Arduino_UNO*).

Hardware

An Arduino UNO is required.

Example

Presentation

To illustrate how these 2 plugins work, the **Arduino_UNO_ANALOG_sample.CONNECTIO** file is provided. This is to be used with the example program provided in the **Arduino_UNO_DIGIT_ANALOG.ino** file.

Preliminary preparations

- Once downloaded and unlocked, place the **Arduino_UNO** folder in the "Plugins" directory of the CONNECT I/O installation folder (e.g.: *C:\Program Files (x86)\Real Games\Connect IO\Plugins*) ;
- Connect the Arduino UNO board to the PC via USB;
- Assign the COM port to the Arduino UNO device;
- Download the "**Arduino_UNO_DIGIT_ANALOG.ino**" program to the Arduino UNO using the Arduino IDE.

Software handling

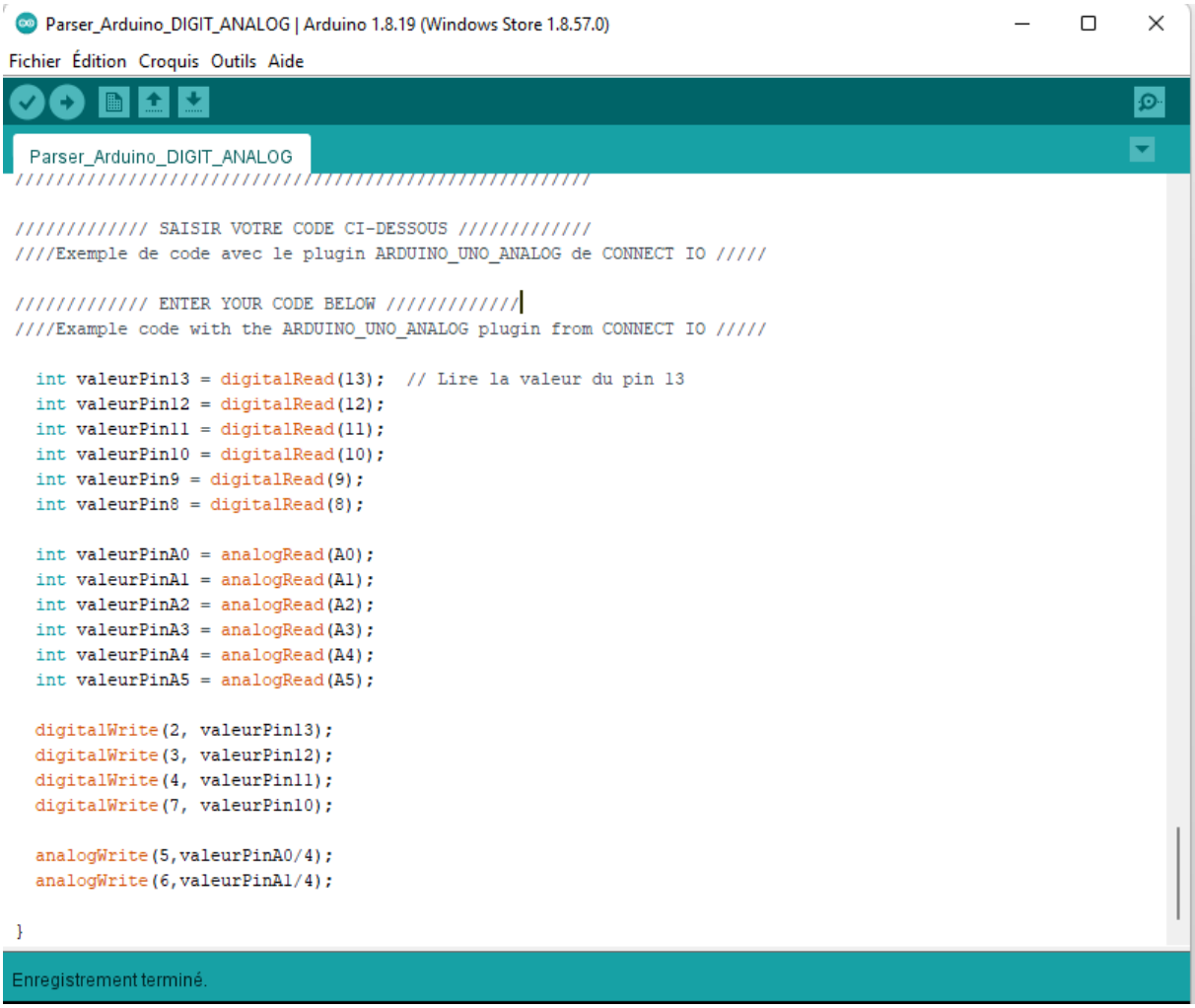
- Start CONNECT I/O and load the "**Arduino_UNO_ANALOG_sample.CONNECTIO**" file;
- Change the port to that used by your Arduino UNO;
- Activate the digital source connected to the plugin's "SWITCH ON" input to switch the plugin on;
- Check that the plugin's "CONNECTED" output is activated. If not, check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

Testing the program

The proposed program performs the following operations:

- -Copy digital input IN-13 to output OUT2
- Copy digital input IN-12 to output OUT3
- Copy digital input IN-11 to output OUT4
- Copy digital input IN-10 to output OUT7
- Copying analog input A0 to PWM output OUT-5
- Copying analog input A1 to PWM output OUT-6

Note that the analog-to-digital converter on the Arduino UNO is 10-bit, and therefore returns a value from 0 (0 v) to 1023 (5 v). PWM outputs, on the other hand, are on 8 bits, from 0 (0 v) to 255 (5 v). Scaling is therefore performed by dividing the values of A0 and A1 by 4.



To verify correct operation, simply activate the digital sources of CONNECT I/O as inputs to the "ARDUINO_UNO_ANALOG" plugin and note that outputs OUT-2, OUT-3, OUT-4 and OUT-7 are also activated.
Finally, by modifying the analog input values of A0 and A1, you should see A0 andA1 copied to OUT-5 and OUT-6 respectively.

You can now control HOME I/O by connecting the HOME I/O sensors and actuators that interest you.

Now it's your turn!