

Matlab et Arduino

1 // Introduction

Le choix académique actuel en matière de logiciel de simulation multi-physique se porte actuellement sur Matlab.

Il convient alors de s'ouvrir aux possibilités actuelles que cette solution logicielle permet à ce jour en matière de communication avec l'extérieur d'autant que la situation évolue de jour en jour...

Il existe deux possibilités d'interfaçage d'arduino avec Matlab.

Table des matières

1 // Introduction.....	1
2 // ArduinoIO : Arduino devient un périphérique d'entrée sortie.....	2
2.1 / installation du package arduino_io.....	2
2.2 / La bibliothèque arduino IO.....	2
2.3 / Programmation du « Firmware » de l'arduino.....	3
2.4 / Un « PREMIER » exemple.....	4
2.5 / La démonstration en vidéo.....	4
3 // ArduinoTarget : Arduino est une cible du programme Simulink compilé.....	5
3.1 / Installation de la cible depuis Matlab 2012a.....	6
3.2 / La bibliothèque Arduino-target.....	7
3.3 / Un exemple2.....	7

2 // ArduinoIO : Arduino devient un périphérique d'entrée sortie

Cette solution consiste à utiliser la carte arduino comme une interface d'entrées/sorties

Elle consiste à

1. pré-charger un programme dans la carte Arduino afin que celle-ci fonctionne en serveur : Ce programme consiste à "écouter" les requêtes envoyées via la liaison série (USB) et de répondre à ces requêtes en renvoyant l'état d'une entrée ou en modifiant l'état d'une sortie. Ces mêmes entrées/sortie sont vues dans matlab comme des entrées logiques ou analogiques (utilisation du CAN) ou des sorties analogiques (mode PWM)

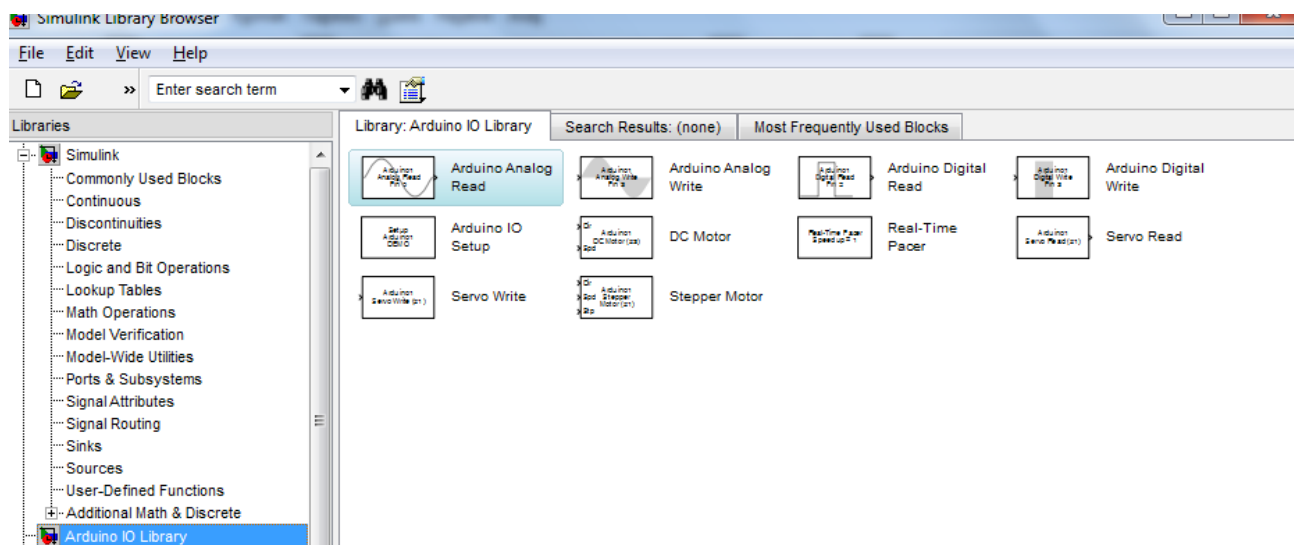
2.1 / installation du package arduino_io

- copier le [package](#) et le dézipper à la racine de votre disque par exemple dans [c:\arduinoio](#)

Remarque : toujours dans ce même répertoire, une notice d'installation appelée « readme.txt » peut se révéler utile.

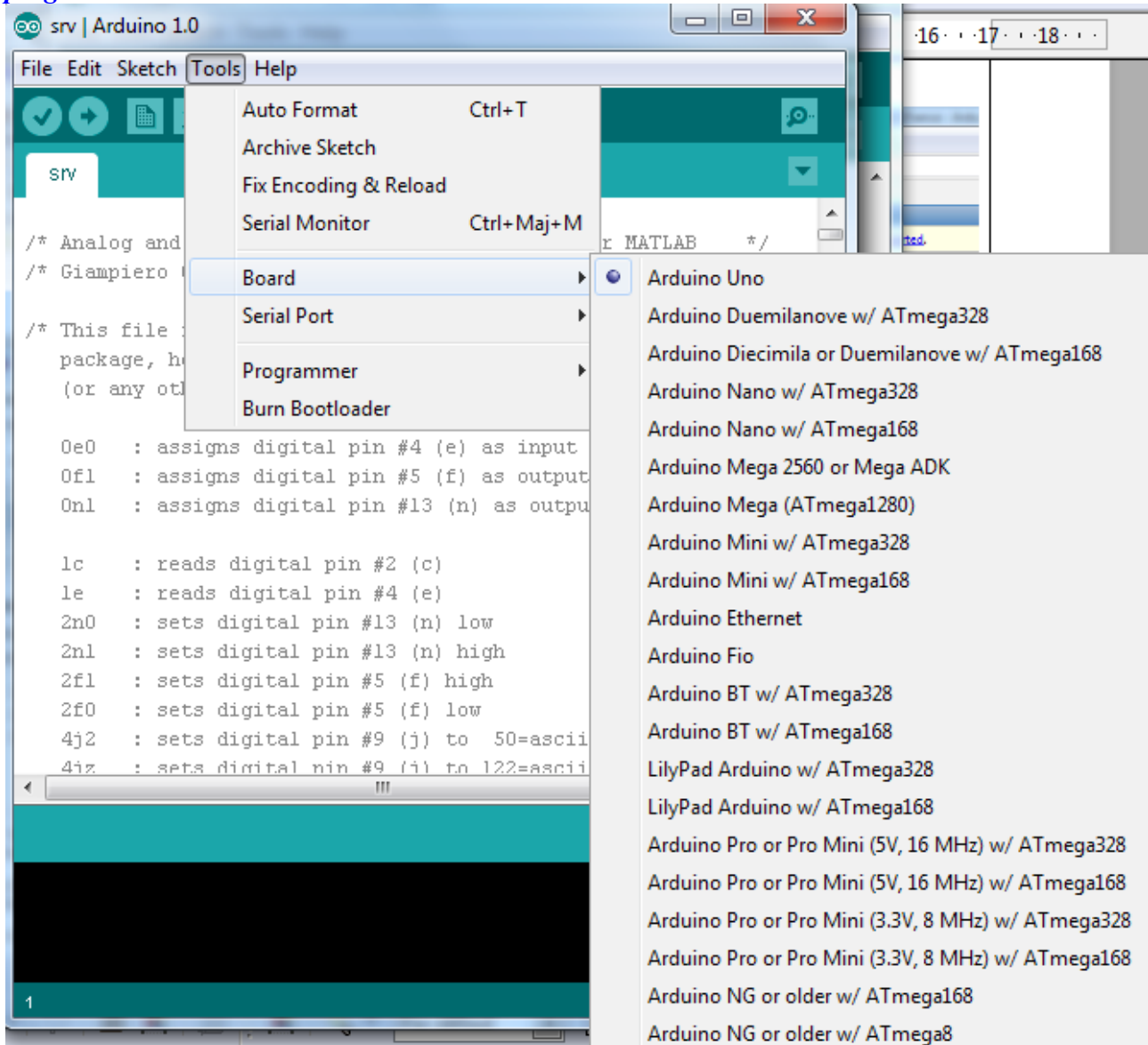
- Lancer Matlab2012 en tant qu'administrateur et se placer dans ce même répertoire
- exécuter la commande : install_arduino
- fermer et relancer Matlab puis simulink
- dans les bibliothèques se trouvent maintenant les blocs dans **arduino IO library**.

2.2 / La bibliothèque arduino IO

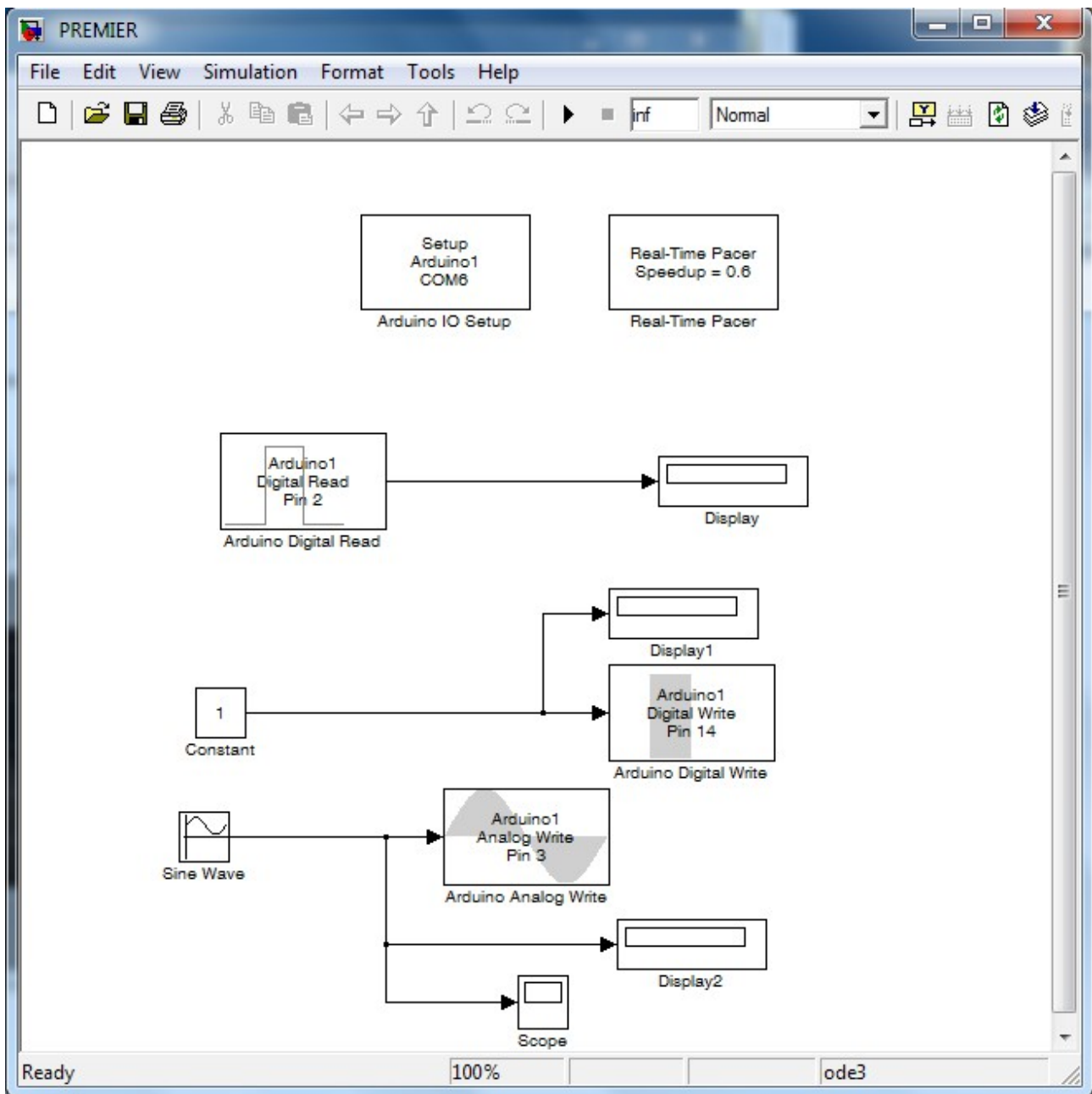


2.3 / Programmation du « Firmware » de l'arduino...

Attention : pour pouvoir utiliser l'ensemble, n'oubliez pas de programmer votre arduino avec le programme `srv.pde` (qui se trouve dans de sous répertoire « `pde` » de ce même et précédent répertoire « `c:\arduino` »). Bien évidemment, il vous faut l'IDE arduino pour executer cette programmation...



2.4 / Un « PREMIER » exemple



2.5 / La démonstration en vidéo...

Voir fichier vidéo joint...

3 // ArduinoTarget : Arduino est une cible du programme Simulink compilé

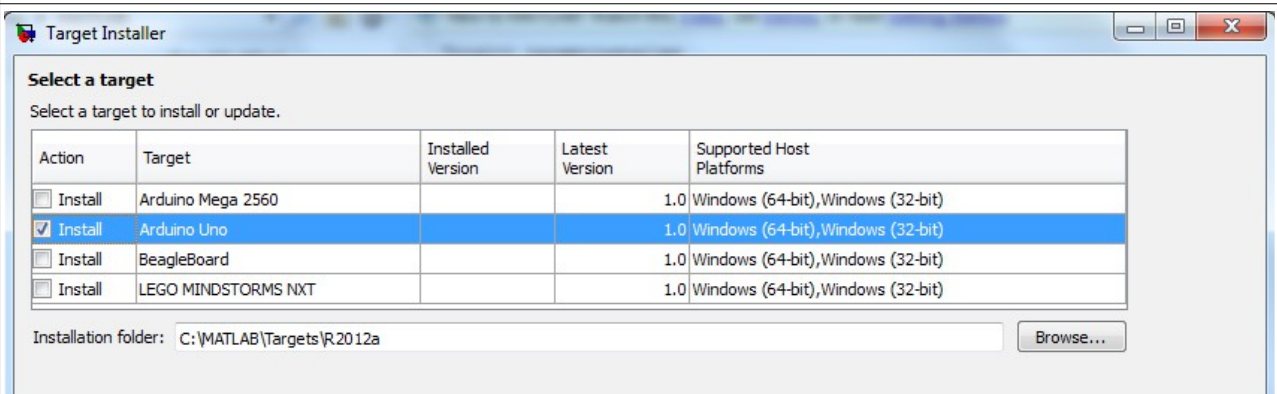
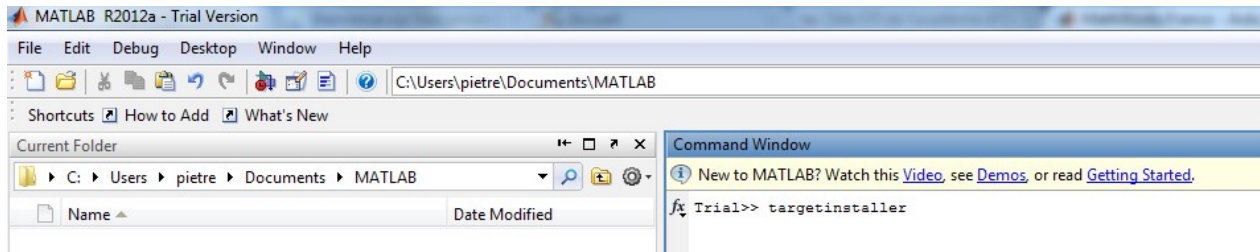
Cette solution consiste à utiliser la carte arduino comme une cible. Matlab compile le programme saisi depuis Matlab/Simulink et le transfère dans l'arduino de façon transparente.

Le programme est alors totalement autonome si on le souhaite. Il peut aussi converser avec la liaison série...

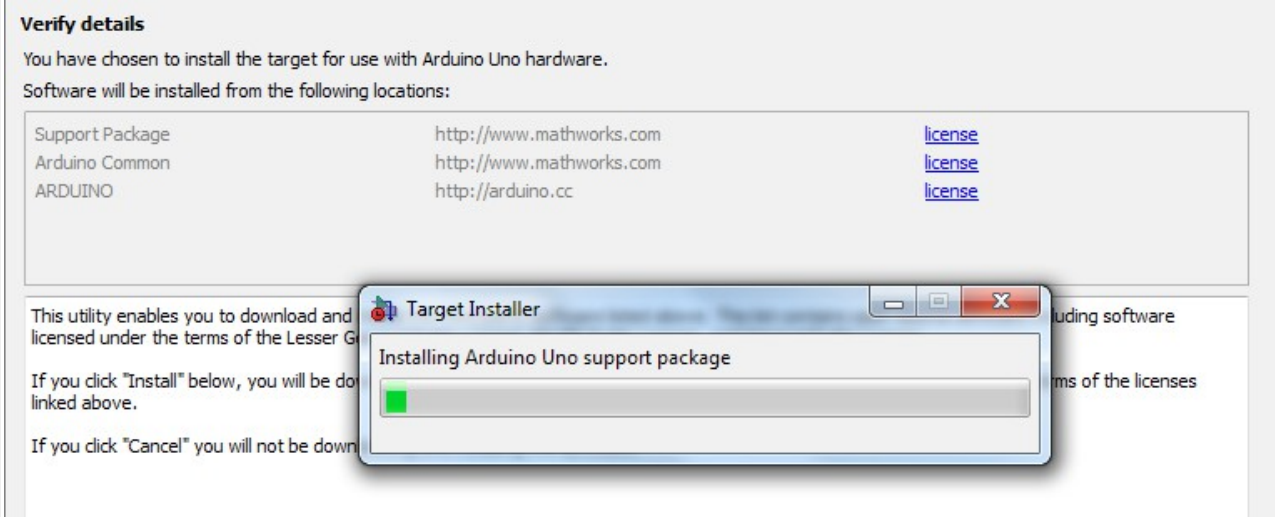
3.1 / Installation de la cible depuis Matlab 2012a

Remarque : Il est aussi possible d'installer sur Matlab 2011 mais c'est beaucoup moins simple...

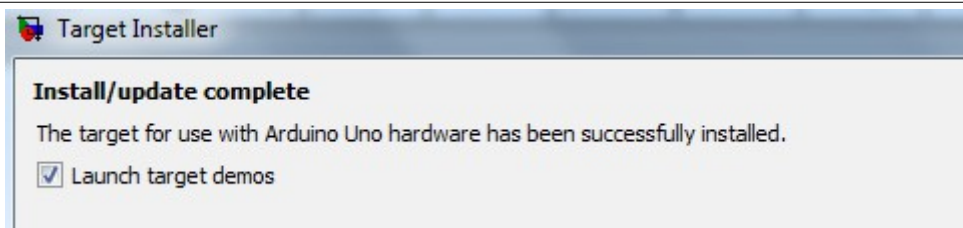
Depuis le prompt Matlab taper : `targetinstaller`



Choisir la cible...

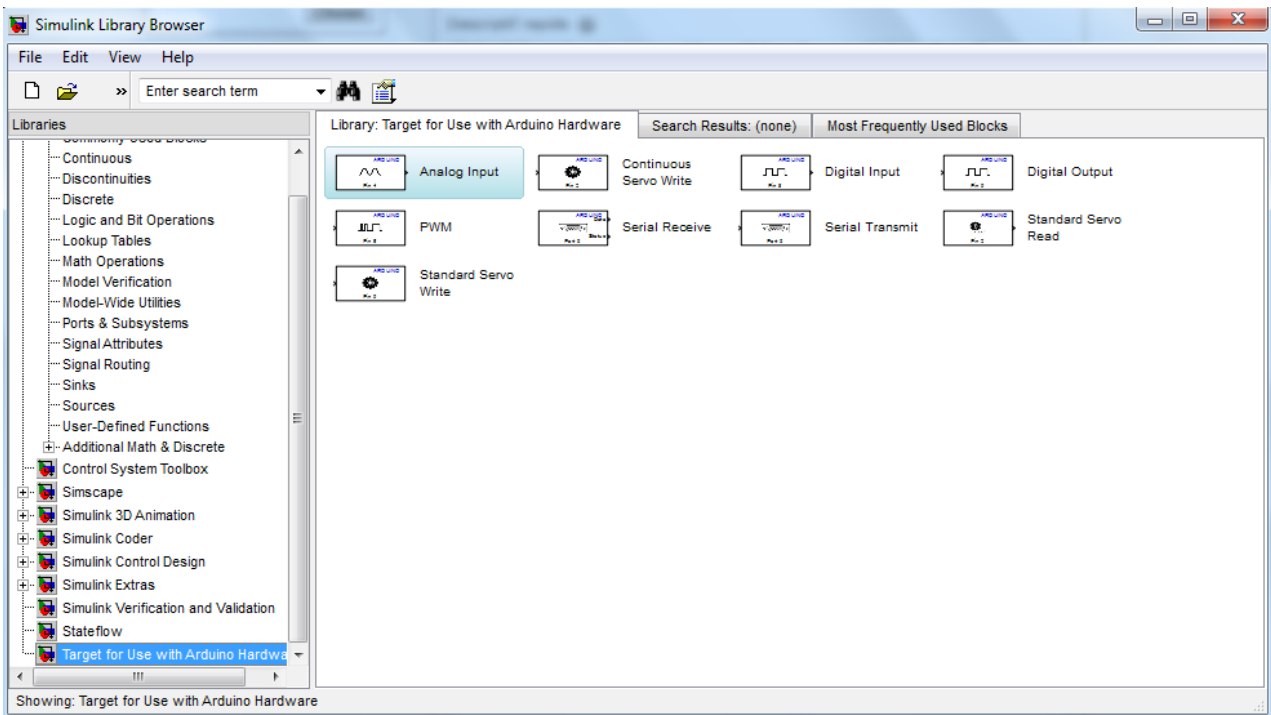


Attendre...



Et c'est tout !

3.2 / La bibliothèque Arduino-target



Il suffit alors d'utiliser les composants de la bibliothèque pour synthétiser ce que l'on souhaite réaliser...

3.3 / Un exemple2

Un exemple de réalisation avec un **arduinoMEGA 2560** doté d'une carte « **Motor Shield** » qui envoie un échelon de tension et qui réalise l'acquisition du courant moteur en temps réels. Le signal étant « parasité » un filtre de lissage permet d'avoir un signal « plus propre ».

