

Formation : Arduino ↔ Matlab/Simulink

Commande d'un système thermique à l'aide de la carte ARDUINO UNO

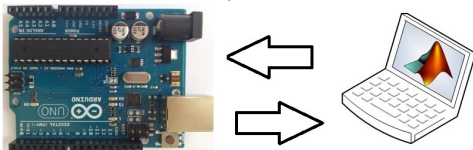
Hammamet 3/4 Mai 2014

CHELLY Nizar et CHARED Amine

Plan de l'exposé

Plan de l'exposé

Comment établir une communication série Arduino/Matlab ?

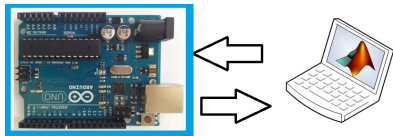


- **L'utilisation des fonctions Arduino/Matlab pour la communication série :**
 - 1 Pré-programmer la carte Arduino pour l'envoi et l'acquisition des données
 - 2 Exploiter les fonctions pour la communication série sous Matlab ou bien Simulink

Les fonctions permettant la communication série pour Arduino

Les fonctions pour la communication série :

- **Serial** : établir la communication série (via USB)
- **available()** : obtenir le nombre de bit disponible pour la lecture
- **read()** : permet la lecture des bits entrants
- **write()** : permet l'écriture des bits sur le port série



Les fonctions permettant la communication série pour Matlab/Simulink

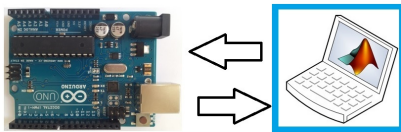
M-files :

- fscanf :
- fprintf :

Modèle Simulink :

- ***Instrument control toolbox***

- Serial configuration : configurer les paramètres du port série
- Serial Receive : Acquisition des données
- Serial Send : Envoie des données via le port série



Plan de l'exposé

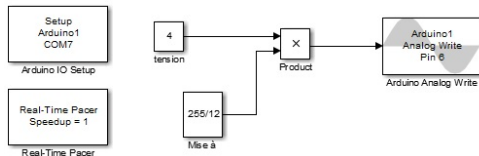
- **L'utilisation du package ArduinoIO :**

- 1 Pré-charger le programme 'adiosrv.pde' dans la carte Arduino (Analog and Digital Input and Output Server)
- 2 Exploiter la bibliothèque ArduinoIO Library sous Simulink ou bien Matlab

- Exemple d'exploitation sous Matlab :

- » `a=arduino('port')` → accès à la carte et aux commandes spécifiques d'arduino
- » `a.analogWrite(3,127);` → envoyer sur la pin 3 un signal PWM de rapport cyclique 127/255

- Exemple d'exploitation sous Simulink :



1 Pré-chargement du programme dans la carte Arduino :

- 1 Télécharger le package ArduinoIO
- 2 Décompresser vers "par exemple E :\arduinoio"
- 3 Ouvrir le dossier décompressé.
- 4 Aller vers : "ArduinoIO\pde\adiosrv" *
- 5 Charger le fichier *adiosrv.pde* vers le logiciel Arduino.
- 6 Televerser !

2 Installation du package ArduinoIO :

- 1 Lancer Matlab et placer vous sous "E :\arduinoio"
- 2 Exécuter la commande : install-arduino
- 3 Fermer et relancer Matlab puis Simulink
- 4 Dans les bibliothèques se trouvent maintenant *Arduino IO library*.

Les blocs nécessaires pour notre objectif d'asservissement :



- **Real-Time Pacer** : Ralentir le temps de simulation de sorte qu'il synchronise avec le temps réel écoulé.
- **Arduino IO Setup** : Pour configurer sur quel port la carte Arduino UNO est connectée.
- **Arduino Analog Read** : Pour configurer à partir de quel pin on va acquérir les données du capteur.
- **Arduino Analog Write** : Pour configurer à partir de quel pin on va envoyer la commande PWM vers l'actionneur.

Plan de l'exposé

Arduino Target :

- Cette solution consiste à utiliser la carte Arduino comme une cible.
 - Matlab compile le programme saisi depuis Simulink.
 - Transfert ce programme directement dans la carte Arduino.

Caractéristique :

- 6 entrées analogiques A0..A5.
- 1 seul CAN, durée de conversion $100 \mu s$.
- Résolution 10 bits
⇒ valeur numérique entre 0 et 1023.



Présentation du capteur :

- 4 pins : *Vcc*, *Trig*, *Echo*, *GND*
- Alimention 5V
- Gamme de mesure de distance entre 2cm et 4m.



Fonctionnement du capteur :

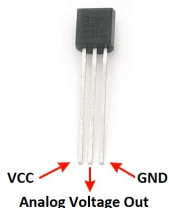
- Envoyer une impulsion niveau haut (+5v) pendant au moins 10μ sur la broche 'Trig' → déclenche la mesure.
- Si le module détecte un objet, la pin 'Echo', va fournir une impulsion (+5v) dont **la durée** est proportionnelle **à la distance**.

Présentation du capteur :

- Alimenter les pattes VCC et GND.
- Brancher la patte centrale à une entrée analogique.

Lecture du capteur :

- $10mV \rightarrow C^o$
- $1V \rightarrow 100C^o$
- La lecture analogique d'un signal de 0 à 5V étant codée de 0 à 1023.
- $Temp = Volt * (5/1023) * 100$
- *Volt* est entre 0 et 1023



Acquisition de température : LM35

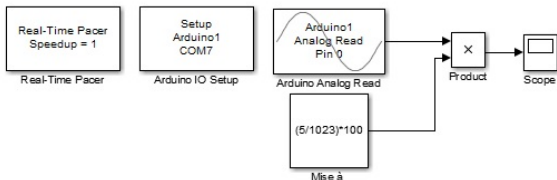
Exploitation du package ArduinoIO Library sous Simulink

1

Pré-chargement du programme *adiosrv.pde* sur la carte Arduino

2

Développement du modèle Simulink :



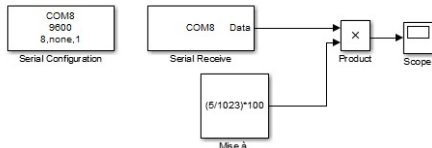
Acquisition de température : LM35

Exploitation de la bibliothèque Instrument control toolbox sous Simulink

1 Pré-programmation de la carte Arduino UNO :

```
int temp;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  temp = analogRead(A0); //lecture CAN (valeur entre 0 et 1023)
  Serial.write(temp); //envoi de la donnée via le port serie
  delay(1000); //delai de 1s avant nouvelle acquisition
}
```

2 Développement du modèle Simulink :



Plan de l'exposé

Présentation des sorties analogiques (mode PWM)

La carte Arduino Uno dispose :

- 6 sorties (3,5,6,9,10 et 11) qui peuvent être utilisées en mode PWM.

C'est quoi un signal PWM ?

- des signaux logiques binaires.
- de fréquence constante (500Hz).
- de rapport cyclique variable.



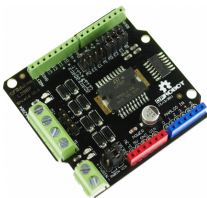
une tension continue ajustable entre 0V (rapport cyclique=0) et 5V (rapport cyclique=255).

$$V_{out} = V_s \times \frac{\tau_o}{\tau_c} ; \text{avec : } \tau_c = 2ms$$



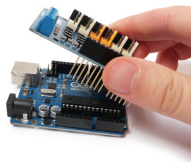
Plan de l'exposé

Utilisation du shield 2A Motor pour Arduino :

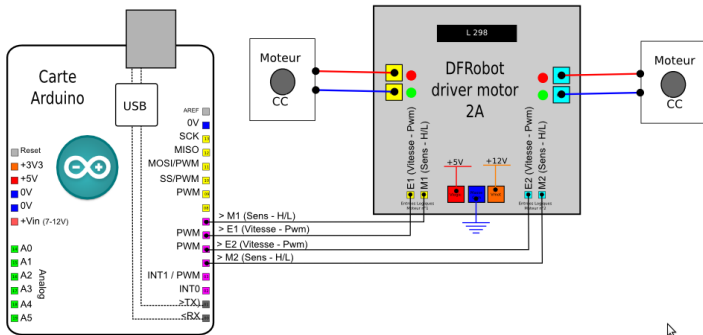


Avantage du shield 2A Motor :

- Ajouter directement sur la carte Arduino UNO
- Équipé du CI L298 qui permet la commande du moteur dans les deux sens



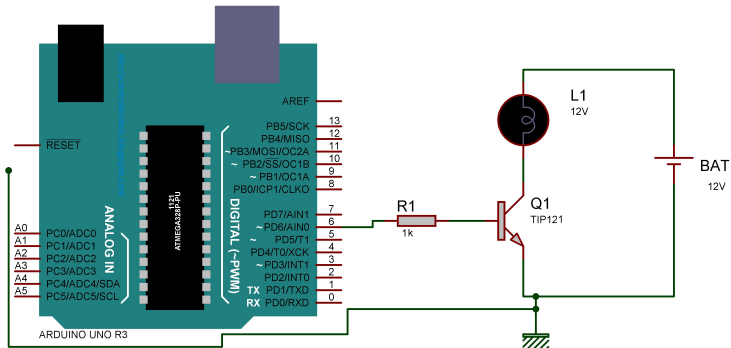
Branchement du shield 2A Motor



Plan de l'exposé

Commande de l'intensité lumineuse d'une Lampe

Branchement pour la commande de la lampe



1 Pré-programmation de la carte Arduino UNO

```
int cmd; //commande
void setup()
{
  Serial.begin(9600); //ouvre le port serie, fixe le d bit      9600 bauds
  pinMode(6,OUTPUT); //Configuration du pin 6 comme sortie
}
void loop()
{
  if (Serial.available()) // si des donn es entrantes sont presentes
  {
    cmd=Serial.read(); //lecture des donn es arriv es
    analogWrite(6,cmd); //Transfert de ces donn es sur la pin 6 pour g n rer le signal
  }
  delay(100); //delai de 100ms avant la nouvelle acquisition
}
```

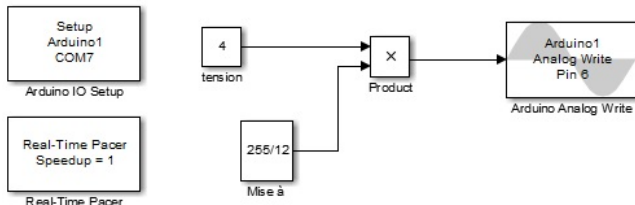
2 Développement du modèle Simulink



Commande de l'intensité lumineuse d'une Lampe

Exploitation de la bibliothèque ArduinoIO Library sous Simulink

- 1 Pré-chargement de adiosrv.pde sur la carte Arduino UNO
- 2 Développement du modèle Simulink



Réponse à un échelon : System Identification sous Matlab

Make Titles Informative.

Choix de la commande P,PI :PID Tuning sous Matlab

Make Titles Informative.

Commande du procédé thermique

