

Ce premier cahier va vous permettre de comprendre le choix de la carte arduino, et des cartes « shield » Easycon 1 et 2, pilotées par le logiciel Organigram pour automatiser vos maquettes.

On va ensuite initialiser l'interface arduino afin qu'elles soit utilisable, *sinon l'interface ne sera pas reconnue par le logiciel ! MAIS cette préparation n'est à faire qu'une seule fois...*

### Sommaire :

|  |      |
|--|------|
| Introduction aux cartes arduino                                  | P.2  |
| Cartes « shield » Easycon1 et Easycon 2                          | P.3  |
| Assemblage des cartes  | P.4  |
| <b>Préparation</b> : programmation/initialisation de l'interface | P.5  |
| Installation et enregistrement du logiciel Organigram            | P.7  |
| Configuration de la mémoire externe                              | P.8  |
| Configuration et gestion d'un afficheur LCD                      | P.9  |
| Configuration de l'horloge temps réel                            | P.12 |
| Alimentation d'une interface arduino                             | P.13 |
| Comprendre le câblage des connecteurs jack                       | P.14 |
| Connecter le potentiomètre                                       | P.15 |
| Connecter des moteurs  | P.17 |
| Utiliser la télécommande   | P.18 |
| Connecter un servomoteur   | P.19 |
| <b>PRECAUTIONS</b>   | P.21 |

# INTRODUCTION AUX INTERFACES ARDUINO



## Qu'est-ce qu'Arduino ?

C'est une plate-forme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino peut être utilisé pour développer des objets interactifs, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles. Les projets Arduino peuvent être autonomes, ou bien ils peuvent communiquer avec des logiciels tournant sur votre ordinateur (tels que Flash, Processing ou MaxMSP). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré-assemblées ; le logiciel de développement open-source peut être téléchargé gratuitement.

Le langage de programmation Arduino est une implémentation de Wiring, une plate-forme de développement similaire, qui est basée sur l'environnement multimédia de programmation Processing.

## Pourquoi Arduino :

→ pas cher : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plate-formes. La moins chère des versions du module Arduino peut être assemblée à la main, et même les cartes Arduino pré-assemblées coûtent moins de 25 €uros (microcontrôleur inclus...) !!!

→ multi-plateforme : le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.

→ un environnement de programmation clair et simple : l'environnement de programmation Arduino (= le logiciel Arduino) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puisse en tirer profit également.

→ logiciel Open Source et extensible : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés.

→ matériel Open source et extensible : les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, etc... Les schémas des modules sont publiés sous une licence Creative Commons, et les concepteurs de circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dans le but de comprendre comment elle fonctionne et pour économiser de l'argent.

**Mais pour nos besoins en technologie avec nos élèves, la connectique et la programmation restent compliquées et cela nécessiterait d'y passer trop de temps.**

# CARTES D'EXTENSION « SHIELD »

## « Shield » Easycon 1

La carte EASYCON1 permet de connecter facilement 20 entrées-sorties à l'interface Arduino Mega. Les embases jacks stéréo 2,5mm servent à la connectique avec les cordons et les cartes I/O présentées. Extensible selon vos besoins, grâce à ces connecteurs stackables, et l'adjonction d'une carte EASYCON2 (18 entrées supplémentaires).

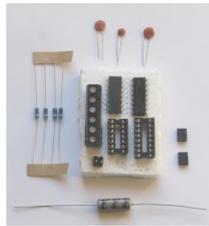
Deux kits optionnels de composants permettront d'étendre les capacités du shield en lui ajoutant :

- une commande de deux moteurs courant continu (contrôle de la vitesse et de la direction).
- une ou deux "banks" de mémoire EEPROM externe pour le stockage des projets autonomes.

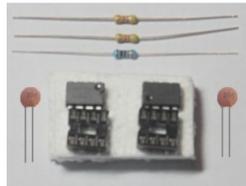
Carte à souder soi-même



Kit commande de 2 moteurs



Kit mémoire



OU

+

=

Carte montée

A souder sur la carte

Vous pouvez choisir de l'acheter montée, ou à monter (prix différents...)

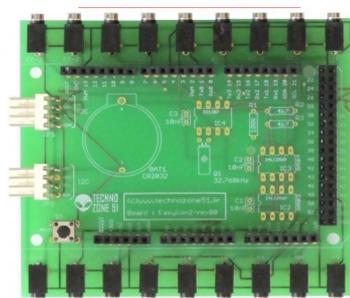
## « Shield » Easycon 2

La carte, ou shield, EASYCON2 permet de connecter facilement 18 entrées-sorties à l'interface Arduino Mega. Les embases jacks stéréo 2.5mm servent à la connectique avec les cordons et les cartes I/O présentées. Extensible selon vos besoins, grâce à ces connecteurs stackables, et l'adjonction d'une carte EASYCON1 (20 entrées supplémentaires).

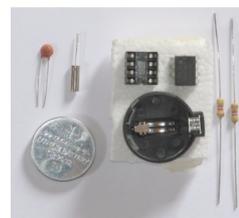
Deux kits optionnels de composants permettront d'étendre les capacités du shield en lui ajoutant :

- une horloge temps réel sauvegardée par pile qui donne accès à la gestion horaire.
- une ou deux "banks" de mémoire EEPROM externes pour le stockage des projets autonomes.

En kit ou monté

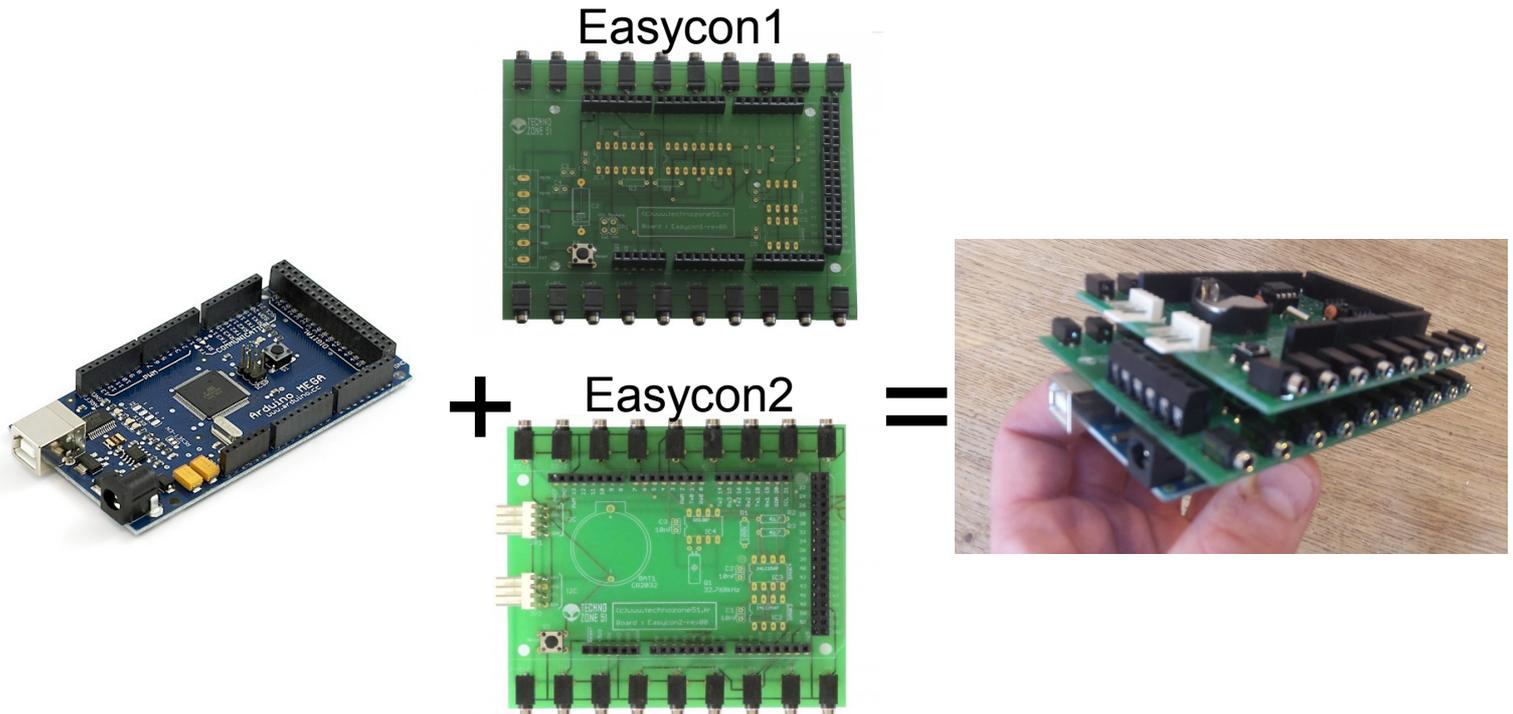


+



A souder soi-même

# ASSEMBLAGE DES CARTES



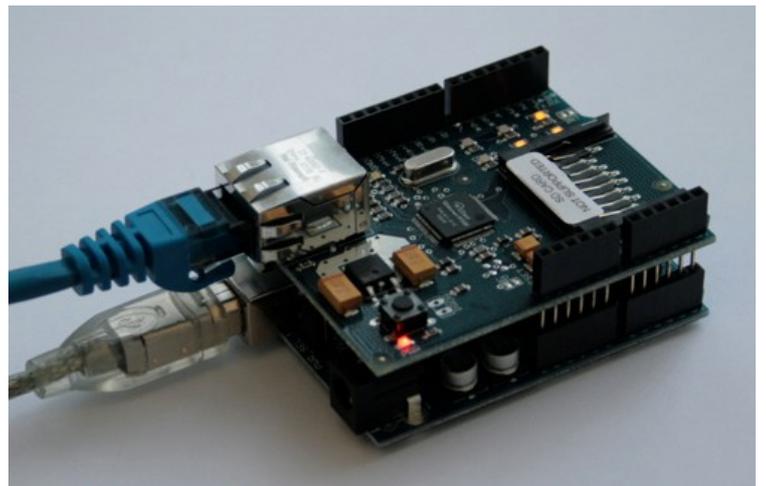
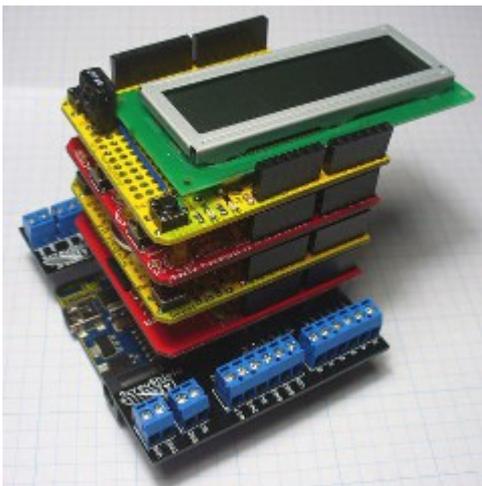
Interface Arduino  
Mega  
*Programmée avec  
l'interpréteur de  
commande*

Shield pour Arduino  
Mega :  
Easycon1 +  
Easycon2

On assemble les  
trois cartes :  
INTERFACE

*Mais vous pouvez bien sûr n'utiliser que l'interface arduino, ou bien la carte arduino avec seulement la carte Easycon1 ou la carte Easycon 2.*

Les cartes additionnelles pour l'interface arduino sont nommées « **shield** » (pas encore de traduction française), les cartes easycon respectent le brochage de ces shields et vous permettent donc d'en empiler autant que vous le voulez pour étendre les possibilités de votre interface.



# PROGRAMMATION / INITIALISATION DE L'INTERFACE

## Pré-requis :

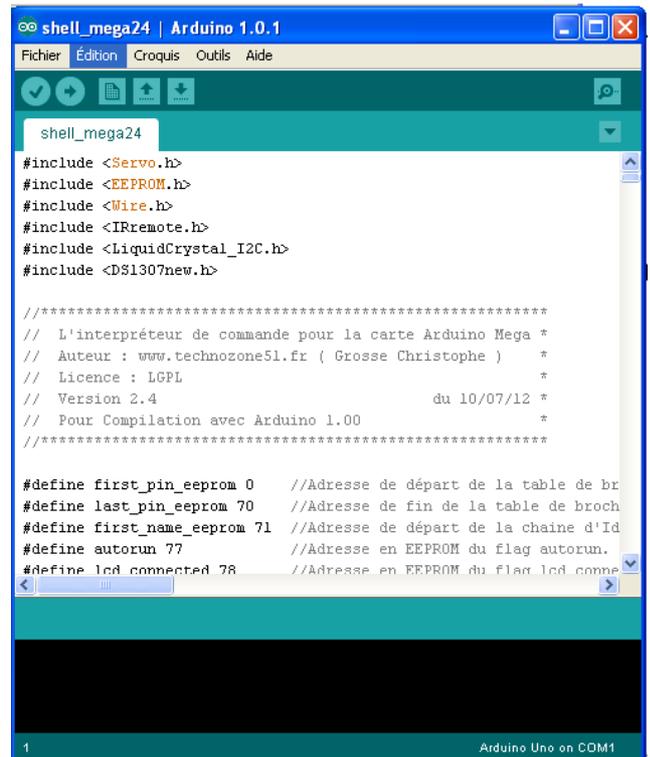
- avoir téléchargé et dézippé le logiciel Arduino
- la platine est reconnue par windows grâce au pilote (voir dossier '**Arduino-1.0.1\drivers\**') et un périphérique '**Port COM**' est rajouté avec le nom '**Arduino MEGA 2560**'

1) Lancez l'environnement de programmation Arduino en cliquant sur le fichier 'arduino.exe'

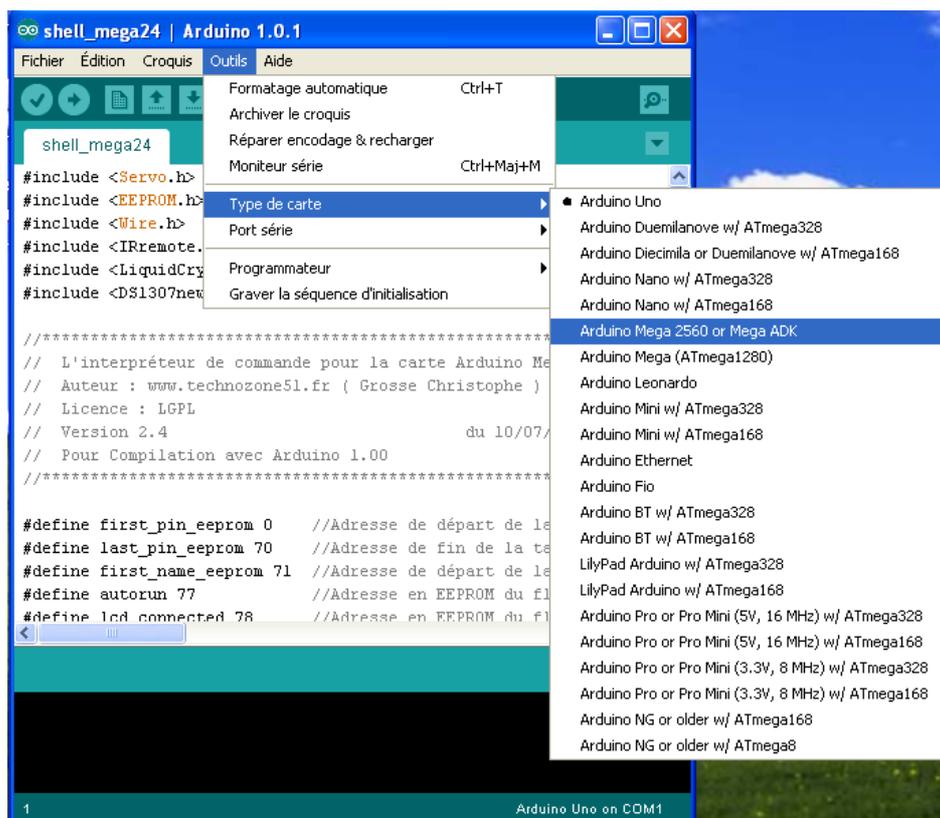


2) Cliquez sur « Fichier » puis « Ouvrir » puis sélectionnez SHELL\_MEGA24.INO que vous aurez téléchargé et dézippé depuis :

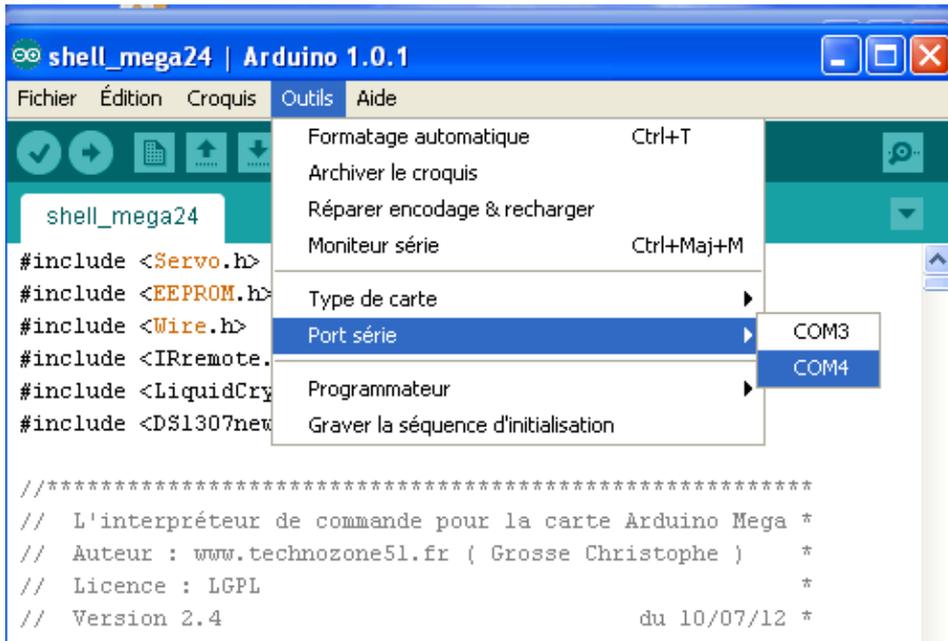
[http://www.techno-zone-51.fr/attachment.php?id\\_attachment=21](http://www.techno-zone-51.fr/attachment.php?id_attachment=21)



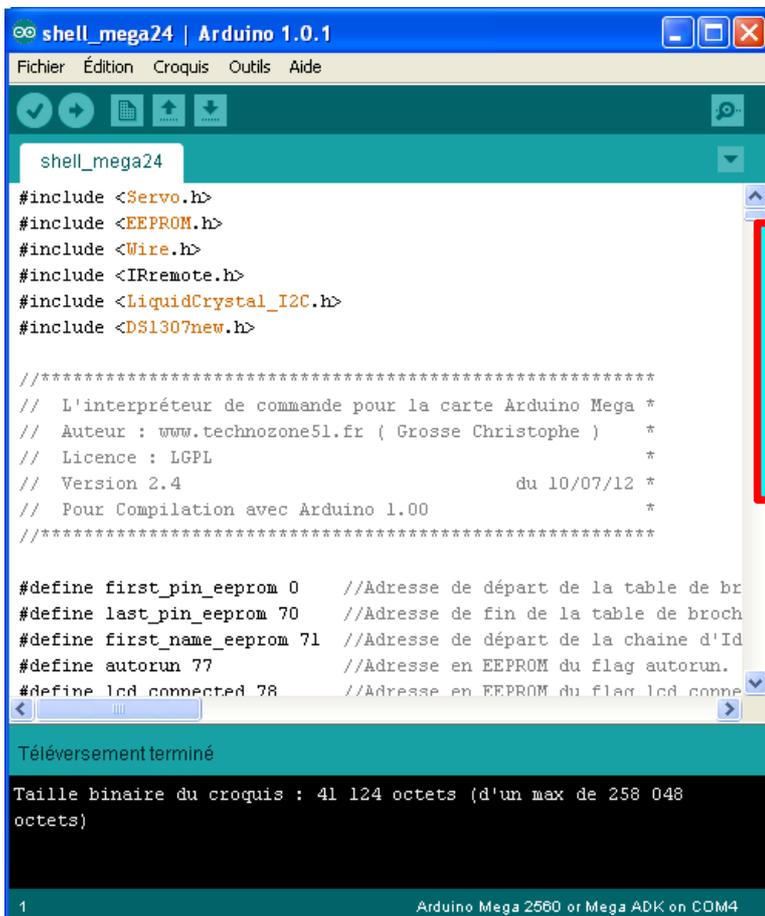
3) Cliquez sur « Outils » puis « Type de carte » puis sélectionnez la carte dont vous disposez.



4) Cliquez sur « Outils » puis « Port Série» puis sélectionnez le port COM sur lequel est connectée votre carte (si plusieurs port COM vous sont proposés, il faudra les tester un à un jusqu'à trouver lequel fonctionne ou vérifier dans le 'Gestionnaire de périphériques').



Cliquer sur l'icône  pour téléverser le programme dans la carte. Le programme est tout d'abord compilé puis il est transféré dans la carte. Normalement le message « Done uploading » indique que tout s'est bien passé. En cas de message d'erreur, retentez l'upload en ayant pris soin de choisir un autre port COM...



**VOTRE CARTE EST  
MAINTENANT PRETE POUR  
RECEVOIR LES SHIELD  
EASYCON**

# INSTALLATION ET ENREGISTREMENT DU LOGICIEL ORGANIGRAM

1) Pour le télécharger, cliquer sur le lien ci-dessous

<http://www.techno-zone-51.fr/logiciels/20-organigram.html>

ORGANIGRAM permet de commander vos cartes arduino en dessinant un organigramme. Particulièrement adapté pour automatiser facilement des maquettes pédagogiques, le logiciel est ouvert et permet de choisir soi même la syntaxe présente dans l'organigramme.

**Pré-requis** pour faire fonctionner ORGANIGRAM :

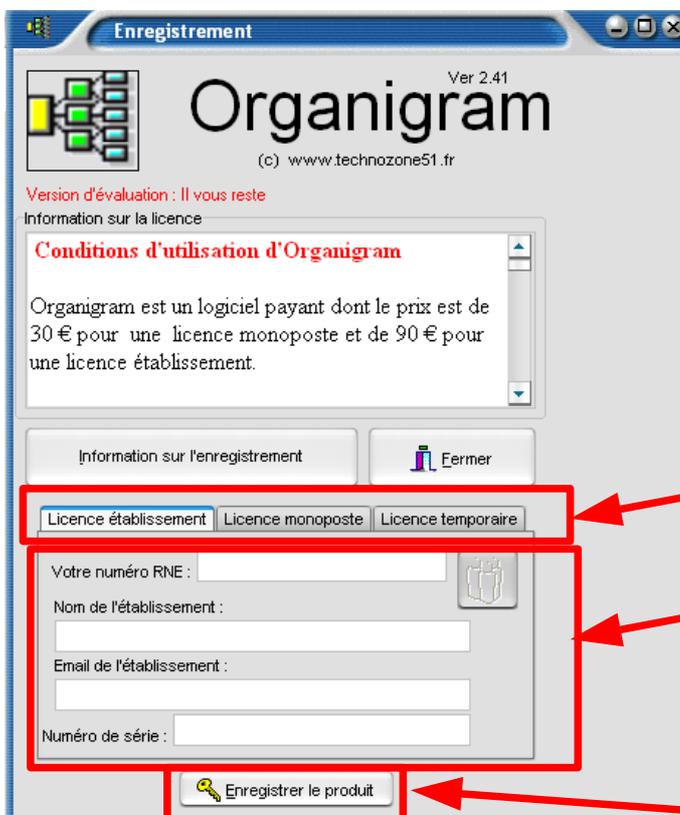
- une carte Arduino Uno ou Arduino Mega initialisée (voir pages 5 & 6)

2) Un fichier setup Organigram s'est chargé : double-cliquer dessus et suivre les instructions pour installer le logiciel.



3) Quand le logiciel est installé, un dossier Organigram est créé dans **C:\Program Files\**.

4) Lancer le logiciel et cliquer sur **Option** puis **Enregistrement** (si vous avez acheté le logiciel) :

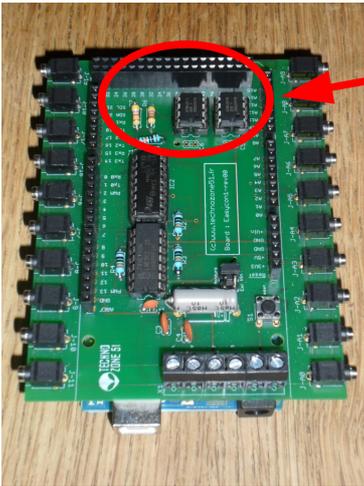


5) Sélectionner le type de licence

6) Saisir les données reçues lors de votre commande.

7) Cliquer sur **Enregistrer le produit**.

# CONFIGURATION DE L'EXTENSION DE MEMOIRE EXTERNE



Les cartes d'interface EASYCON1 et EASYCON2 peuvent être équipées d'un kit de mémoires externes afin d'augmenter les capacités de stockage de vos organigrammes en mode autonome (*sans PC*). Les organigrammes stockés dans ces slots de mémoire pourront donc s'exécuter sans l'aide de l'ordinateur et en parallèle (*voir page 3*).

1) En mode expert : cliquer sur « Options », puis « Configuration Slots mémoire/LCD/Horloge temps réel ».



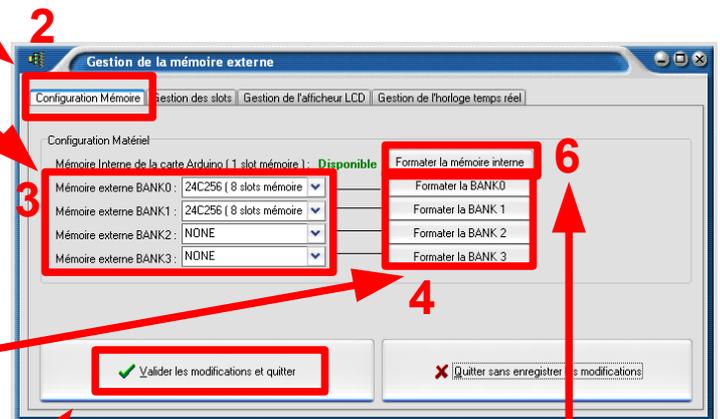
2) Cliquer sur « Configuration Mémoire ».

3) Pour chacune des BANK mémoire, sélectionner le composant que vous avez installé. *Dans l'exemple ci-contre, une mémoire 24C256 a été installée sur l'emplacement BANK0 et une 24C256 sur l'emplacement BANK1 de la carte EASYCON1. Les emplacements vides sont laissés sur NONE.*

4) Cliquez ensuite sur les boutons « Formater la Bank X » en face des composants mémoire installés afin de vider proprement et de formater ces mémoires nouvellement installées.

5) Puis cliquez sur « Valider les modifications et quitter ».

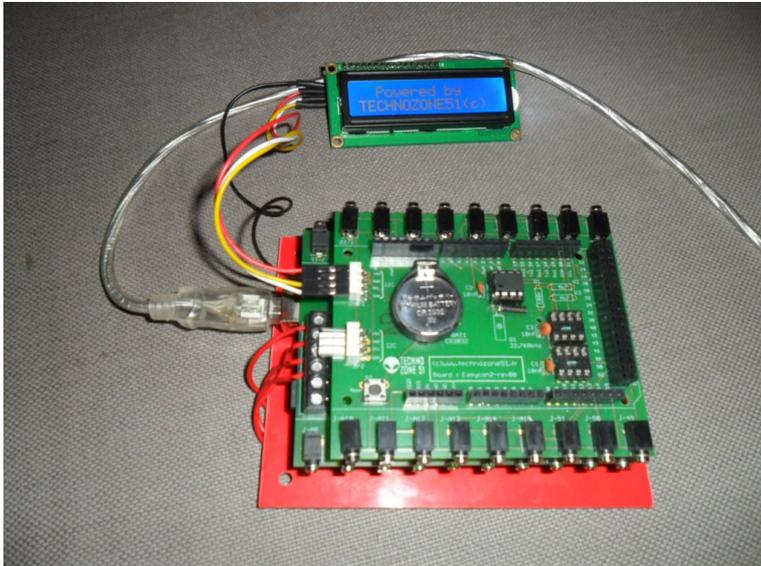
6) Vous pouvez aussi utiliser le slot mémoire interne de la carte Arduino. (Slot0) et le formater.



# CONFIGURATION D'UN AFFICHEUR LCD SUR BUS I2C

Organigram permet de gérer tout afficheur LCD Alphanumérique à interface I2C.

**ATTENTION ! COUPER L'ALIMENTATION DE LA CARTE ARDUINO AVANT DE BRANCHER VOTRE AFFICHEUR LCD**

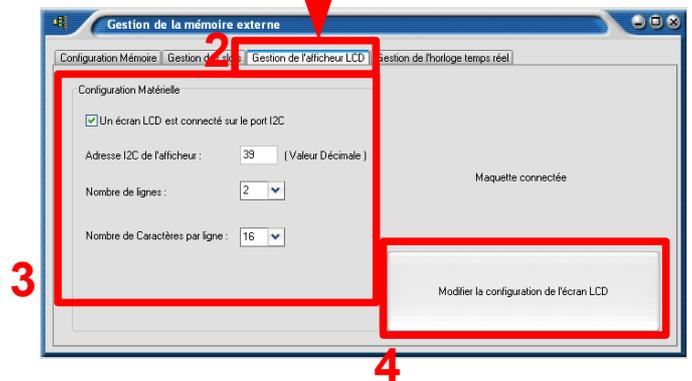


Le fil rouge se place sur le 5V du connecteur I2C

1) En mode expert : cliquer sur « Options », puis « Configuration Slots mémoire/LCD/Horloge temps réel ».



2) Cliquer sur « Gestion de l'afficheur LCD ».



3) Cocher la case « Un écran LCD est connecté sur le port I2C » puis renseigner les champs suivants :

Adresse I2C de l'afficheur : l'adresse I2C du composant (*par défaut 39*)

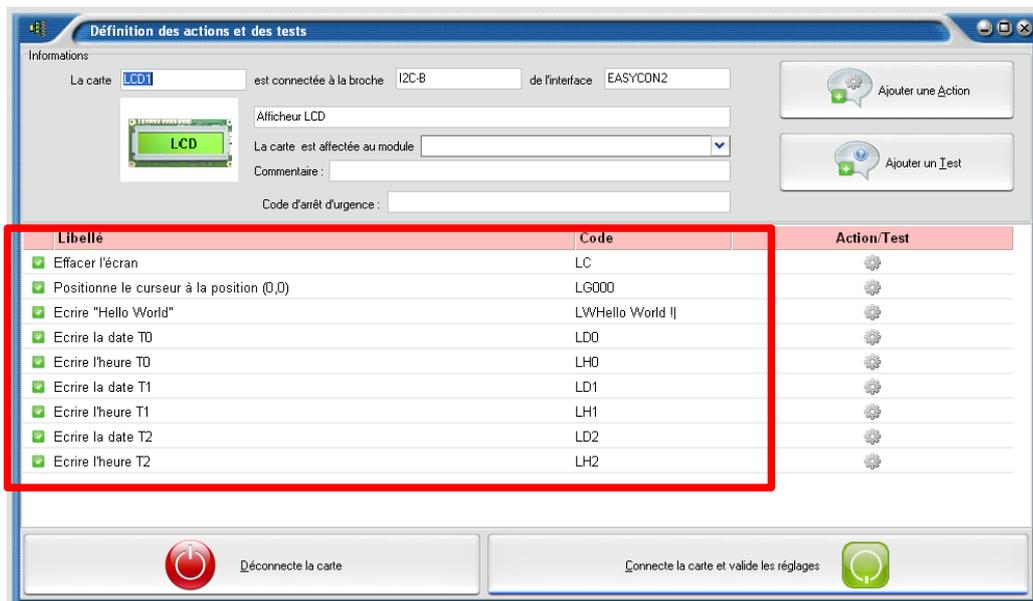
Nombre de lignes : indiquez le nombre de lignes (*1, 2 ou 4*)

Nombre de caractères par ligne : indiquez le nombre de caractères par ligne (*16 ou 20*)

4) Cliquez sur le bouton « Modifier la configuration de l'écran LCD » pour mettre à jour la carte Arduino. Désormais celle-ci gère l'écran LCD.



5) Les actions par défaut de l'écran LCD :



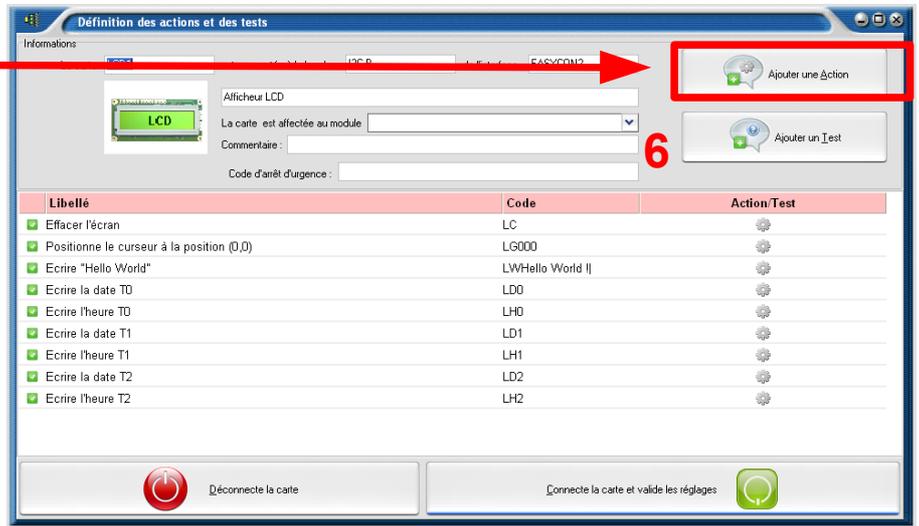
Mais on peut en créer d'autres. Voici un tableau pour comprendre les codes actions :

| Format de la commande | Signification   | Exemples  |
|-----------------------|---|---|
| <b>LC</b>             | Efface l'écran LCD  | <b>LC</b>   |
| <b>LGccr</b>          | La commande <LG> Positionne le curseur à la position (cc,r) ou cc est le numéro de colonne sur deux chiffre et r le numéro de ligne   | <b>LG000</b> : Positionne le curseur à la position (0,0)<br><br><b>LG091</b> : Positionne le curseur à la position (9,1)<br><br><b>LG153</b> : Positionne le curseur à la position (15,3) |
| <b>LWsss...sss </b>   | Ecrire la chaîne de caractères ssss...sss à partir de la position courante du curseur. La fin de la chaîne de caractères est marqué par le caractère   (pipe)   | <b>LWHello World </b> : écrit Hello World sur l'écran à partir de la position courante du curseur.  |
| <b>LVvvvccr</b>       | Ecrire la valeur vvv à la position (cc,r). Avant d'écrire la valeur vvv, les 3 caractères à partir de la position d'affichage (cc,r) sont effacés afin que des affichages successifs d'une valeur ne se chevauchent pas ! | <b>LV254101</b> : écrit la valeur 254 à la position (10,1)<br><br><b>LV005020</b> : écrire la valeur 5 à la position (10,1) (Deux espaces blancs sont également écrits !)                 |
| <b>LV#Pccr</b>        | Ecrire la valeur contenu dans la variable P à la position (cc,r)  | <b>LV#P001</b> : écrit la valeur contenue dans la variable P à la position (0,1)  |
| <b>LDh</b>            | Ecrire la date contenue dans la mémoire horaire h à partir de la position courante du curseur. h est compris entre 0 et 7   | <b>LD0</b> : Ecrit la date contenue dans la variable horaire n°0 à la position courante du curseur. Ecrit par exemple <b>23/03/12</b>   |
| <b>LHh</b>            | Ecrire l'heure contenue dans la mémoire horaire h à partir de la position courante du curseur. h est compris entre 0 et 7   | <b>LD0</b> : Ecrit l'heure contenue dans la variable horaire n°0 à la position courante du curseur. Ecrit par exemple <b>10:32:05</b>   |
| <b>LTvvvccr</b>       | Ecrire la valeur de la température en degré à convertir à la position (cc,r)  | <b>LT174000</b> : écrit la température correspondante à 174 à la position (0,0) de l'écran  |
| <b>LT#V c c r</b>     | Ecrire la valeur de la température en degré contenue dans la variable V à la position (cc,r)  | <b>LT#T001</b> : écrit la température comprise dans la variable T à la position (0,1) de l'écran  |

Voici un exemple : on écrit la date sur la ligne 0 et la température sur la ligne 1



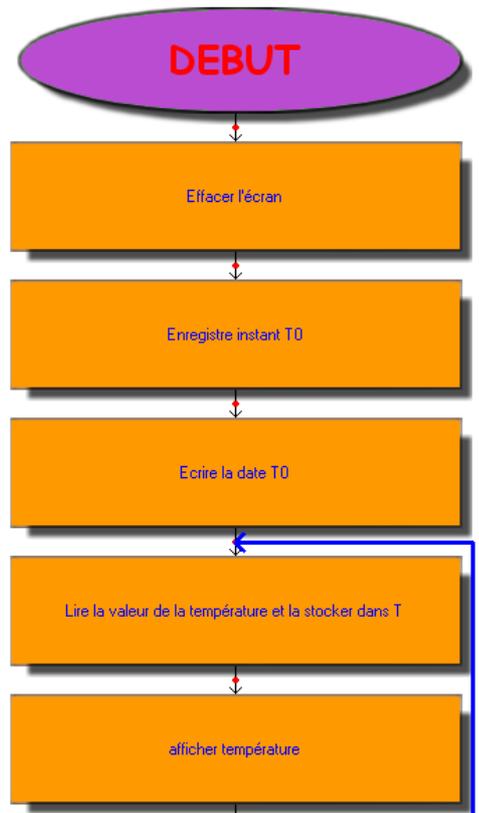
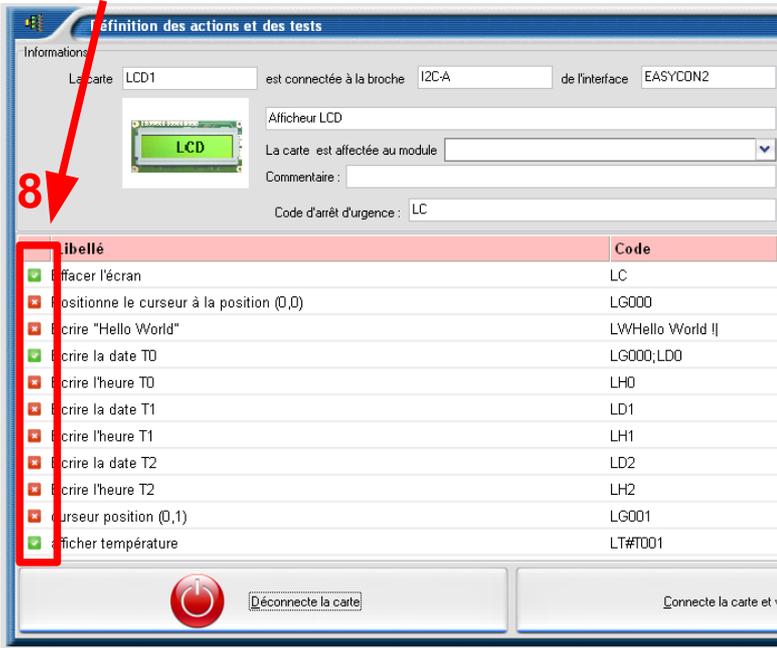
6) Cliquer sur « Ajouter une action »



7) Cliquer sur le libellé et le modifier :

| Libellé   | Code                 |
|---|----------------------|
| <input checked="" type="checkbox"/> Effacer l'écran                           | LC                   |
| <input checked="" type="checkbox"/> Positionne le curseur à la position (0,0) | LG000                |
| <input checked="" type="checkbox"/> Ecrire "Hello World"                      | LWHello World !!     |
| <input checked="" type="checkbox"/> Ecrire la date T0                         | LD0                  |
| <input checked="" type="checkbox"/> Ecrire l'heure T0                         | LH0                  |
| <input checked="" type="checkbox"/> Ecrire la date T1                         | LD1                  |
| <input checked="" type="checkbox"/> Ecrire l'heure T1                         | LH1                  |
| <input checked="" type="checkbox"/> Ecrire la date T2                         | LD2                  |
| <input checked="" type="checkbox"/> Ecrire l'heure T2                         | LH2                  |
| <input type="checkbox"/> (Editer le libellé de votre action ici)              | (Editer le code ici) |

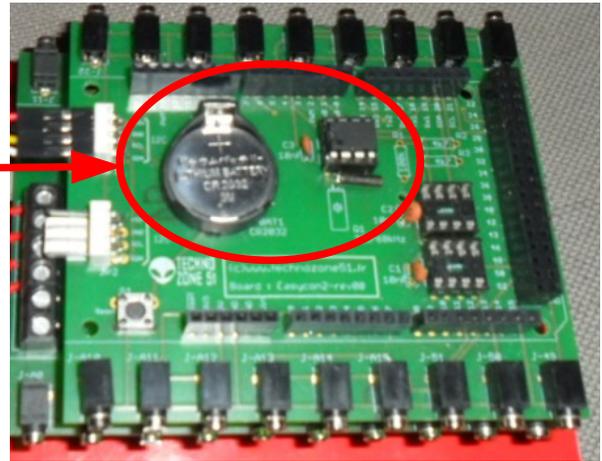
8) N'oublier pas d'activer les actions pour qu'elles apparaissent ou non lors de l'édition des organigrammes :



Exemple de ce que peut donner l'organigramme pour afficher la date et la température :

# CONFIGURATION DE L'HORLOGE TEMPS REEL

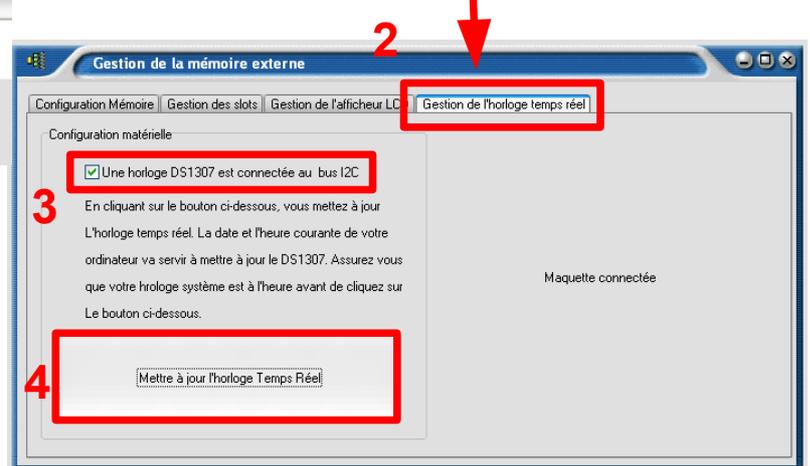
Organigram gère l'horloge temps réel à base de DS1307 qui est disponible en option sur la carte EASYCON2 (voir page 3). Cette option donne la possibilité de faire de la « gestion horaire » dans vos organigrammes.



1) En mode expert : cliquer sur « Options », puis « Configuration Slots mémoire/LCD/Horloge temps réel ».



2) Cliquer sur « Gestion de l'horloge temps réel ».



3) Il vous suffit de cocher la case « Une horloge DS1307 est connectée au bus I2C ».

4) cliquer sur le bouton « Mettre à jour l'horloge Temps Réel » pour que l'horloge présente dans le DS1307 soit mise à l'heure (à partir de l'heure système de votre ordinateur) et démarre.

Cette mise à l'heure n'est à effectuer que la première fois ou si la dérive de l'horloge devient trop importante. Assurez vous que l'horloge système (date et heure) de votre ordinateur soit correcte avant de cliquer sur le bouton « Mettre à jour l'horloge temps réel ».

# ALIMENTATION D'UNE INTERFACE ARDUINO

La carte Arduino Mega 2560 peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA), soit à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2,1mm positif au centre dans le connecteur de la carte.



La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte est entre 7V et 12V.



OU



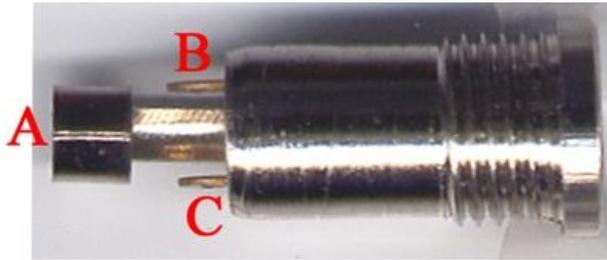
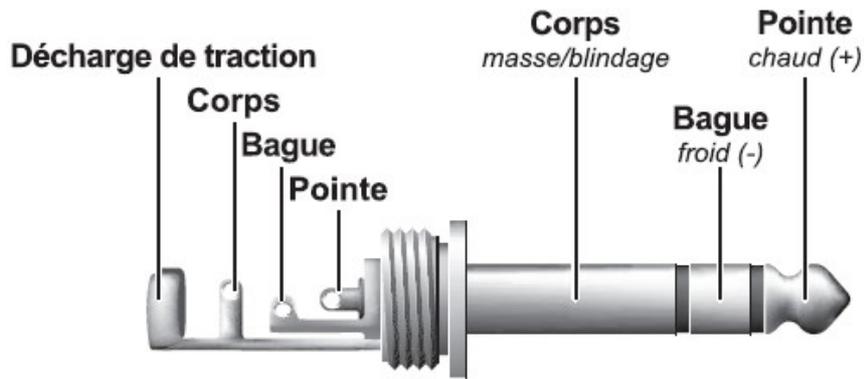
+



Très utile pour rendre autonome la carte (→ robot) !

*Malgré le connecteur pour pile 9V, il vaut mieux connecter un bloc de piles de 1,5V (6 à 8) pouvant fournir une puissance plus importante (de 2000 à 3000 mAh).*

# CABLAGE CONNECTEUR JACK



A : Tresse : 3  
 B : Rouge : 2  
 C : Blanc : 1



*Pour être sûr, il faut vérifier avec un ohmètre.*

## Pour le voltmètre qui va servir à analyser le signal



**Corps : noir**  
**Pointe : rouge**

*Ici, on choisit de changer les couleurs pour que les élèves aient les codes couleur habituels : noir et rouge.*



Soit on ressort le fer à souder, soit on achète des adaptateurs car les dédoubleurs pour jack 2,5 n'existent pas...



# CONNECTER LE POTENTIOMETRE

Cette carte d'entrée-sortie se connecte rapidement sur une entrée analogique :

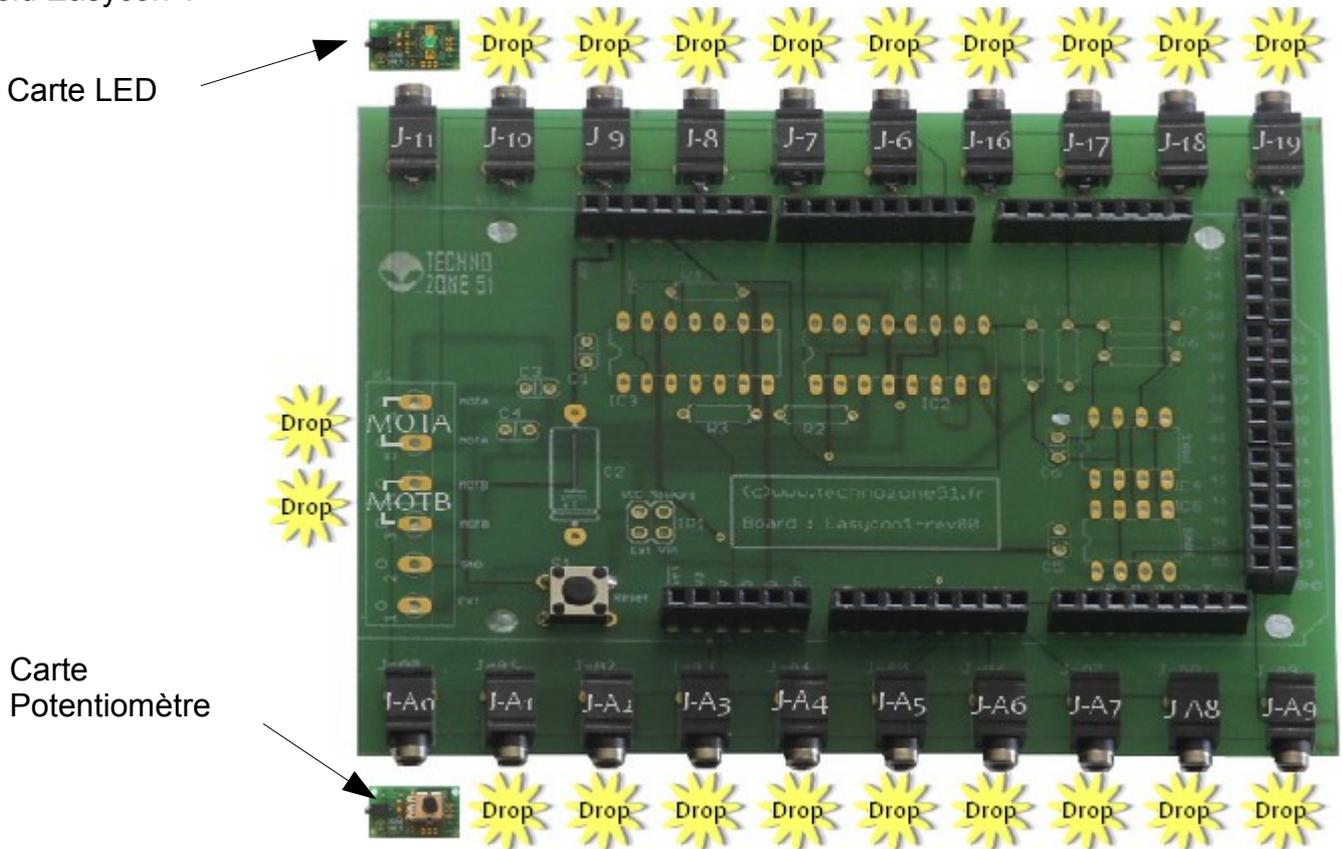


- JA0 à JA9 sur le shield Easycon 1
- J49 à J51 sur le shield Easycon 2

Ce capteur analogique va utiliser une variable S relative à la position du potentiomètre.

Un cas particulier d'utilisation pour ce potentiomètre : faire varier l'intensité lumineuse d'une LED.

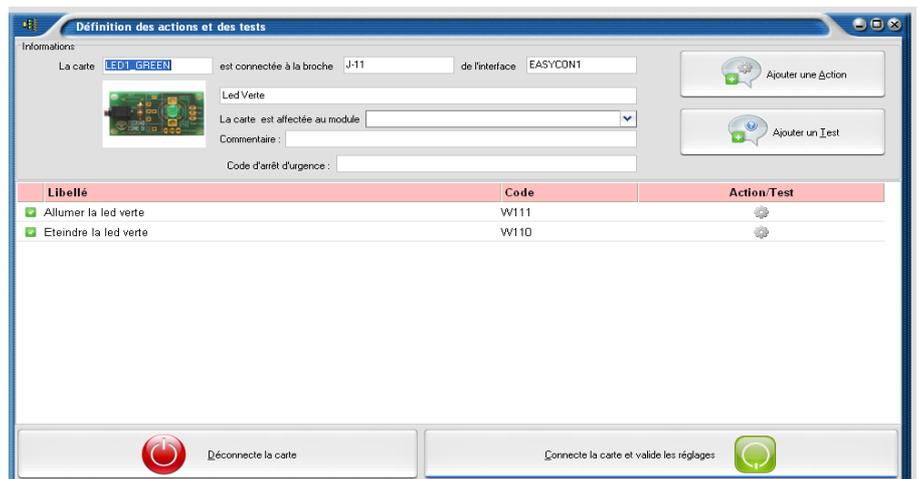
Dans ce cas, il faudra connecter la Carte LED sur une sortie de puissance de J6 à J11 sur le shield Easycon 1



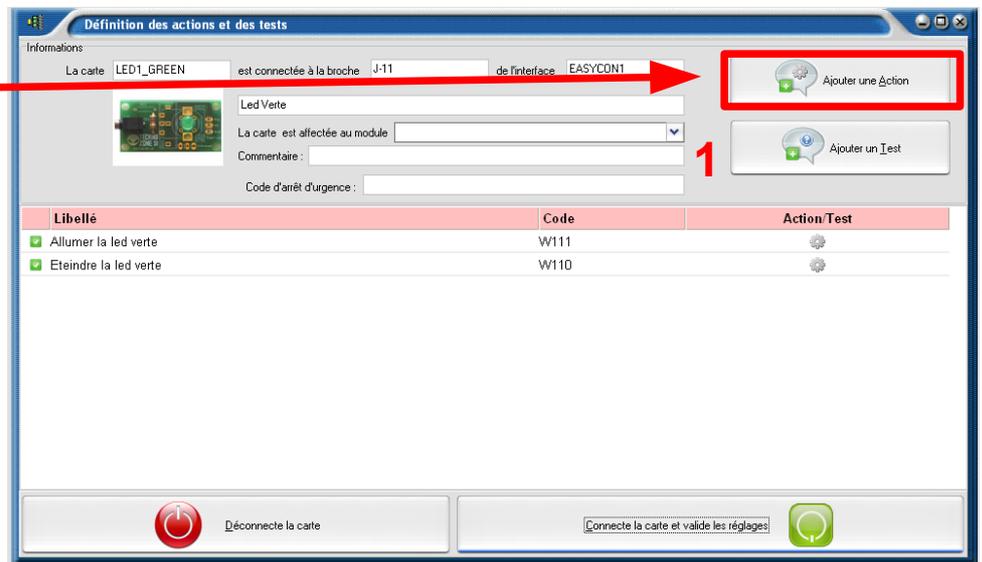
Par défaut, les actions disponibles d'une Carte LED sont:

- Allumer la LED verte
- Éteindre la LED verte

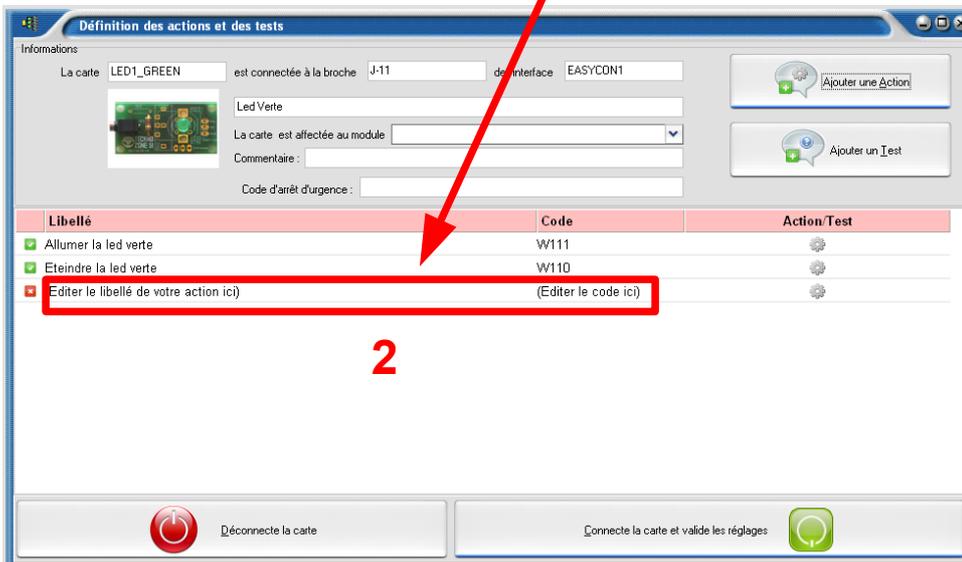
Il va donc falloir ajouter une action qui va prendre en compte la variable S du potentiomètre relative à sa position.



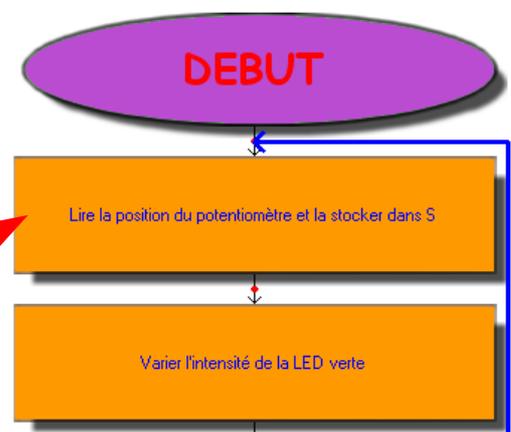
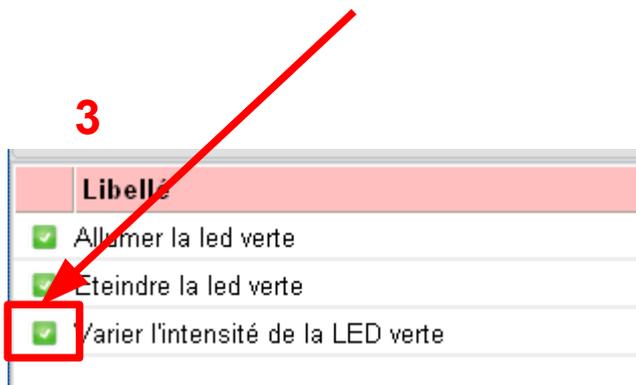
1) Cliquer sur **Ajouter une action.**



2) Cliquer sur le libellé et le modifier en rajoutant le code action M11#S (le 11 représente le numéro de la broche).



3) N'oubliez pas d'activer l'action pour qu'elle apparaisse lors de l'édition des organigrammes.



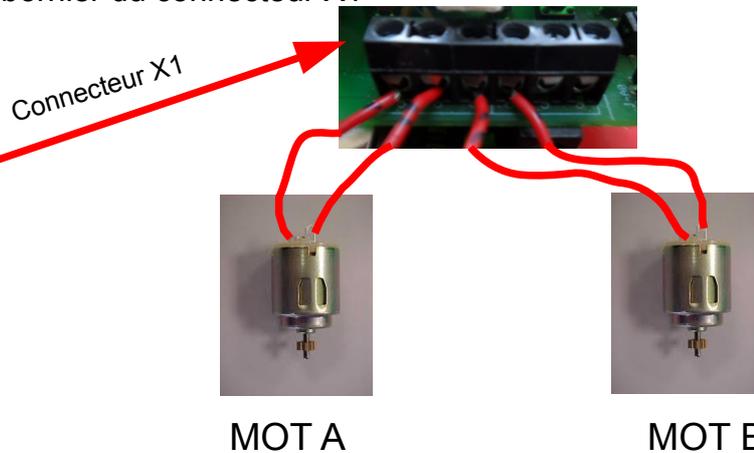
Exemple de ce que peut donner l'organigramme pour faire varier l'intensité lumineuse d'une LED commandée par un potentiomètre.

# CONNECTER LES MOTEURS



Pour pouvoir commander simplement deux moteurs, vous pouvez acheter et ajouter le kit de commande de deux moteurs CC pour EASYCON1 (voir page 3 du cahier 0).

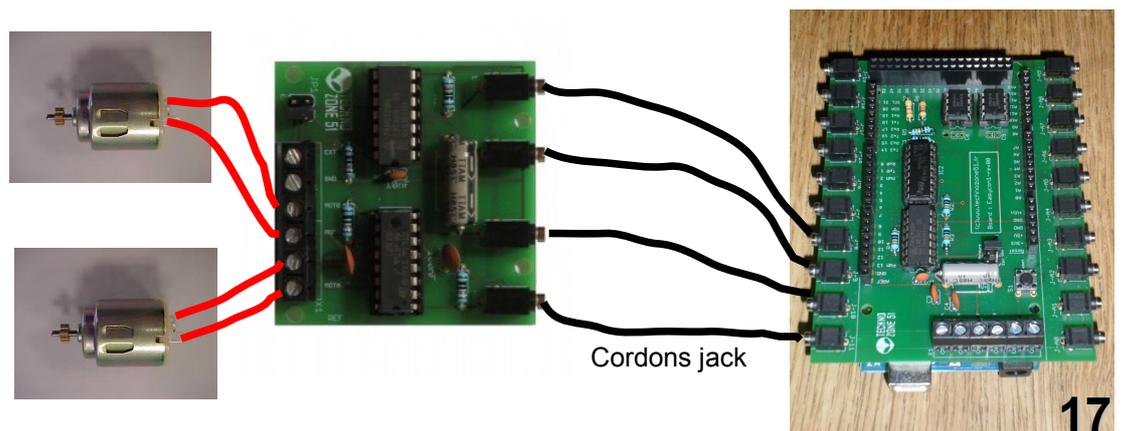
Ensuite vous pouvez simplement brancher en direct les moteurs sur le bornier du connecteur X1



On peut sélectionner différentes sources pour l'alimentation des deux moteurs à l'aide du Header 2x2 broches référencé JP1 et de deux straps amovibles selon le tableau ci-dessous :

| Position des straps amovibles | Source d'alimentation sélectionnée   |
|-------------------------------|--|
|                               | Si aucun strap n'est positionné, les moteurs ne sont pas alimentés et ne pourront pas tourner. Aucune tension n'est présente entre les broches <i>Ext</i> et <i>Gnd</i> du connecteur <b>X1</b>  |
|                               | Les moteurs sont alimentés par la tension <i>Vin</i> présente sur l'entrée alimentation de la carte Arduino Mega. Aucune tension n'est présente entre <i>Ext</i> et <i>Gnd</i> du connecteur <b>X1</b>   |
|                               | Les moteurs sont alimentés par la tension <i>Vin</i> présente sur l'entrée alimentation de la carte Arduino Mega. Cette tension <i>Vin</i> est également disponible entre <i>Ext</i> et <i>Gnd</i> du connecteur <b>X1</b> pour alimenter d'autres montages si nécessaire. |
|                               | Les moteurs sont alimentés par une source externe qu'il faudra connecter entre les broches <i>Ext</i> (+) et <i>Gnd</i> (-) du connecteur <b>X1</b> .  |

On peut ensuite connecter jusqu'à 6 autres moteurs supplémentaires en utilisant des cartes « Commande de deux moteurs cc » :

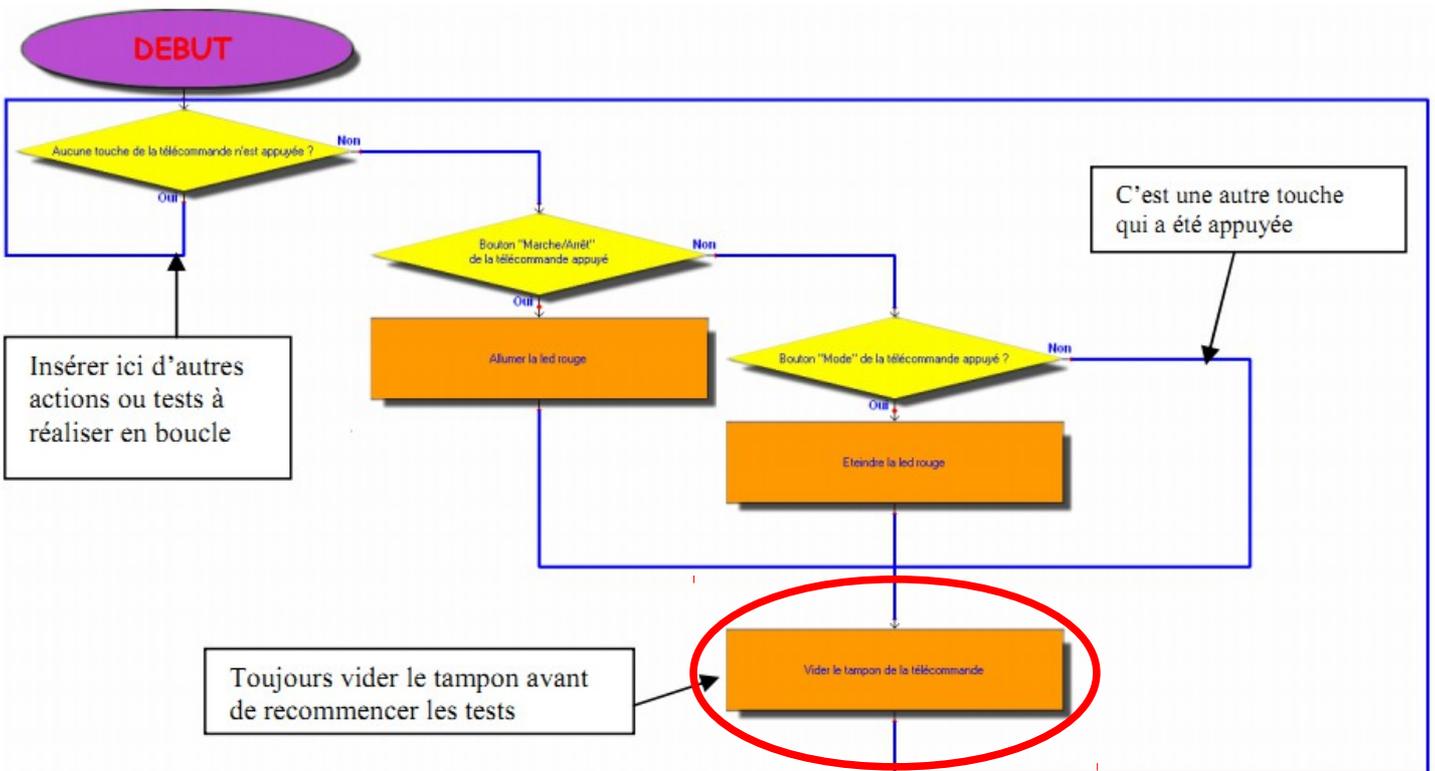


# LA TELECOMMANDE



L'édition d'un organigramme contenant une télécommande nécessite d'avoir une certaine logique.

Voici ci-dessous un exemple d'organigramme :

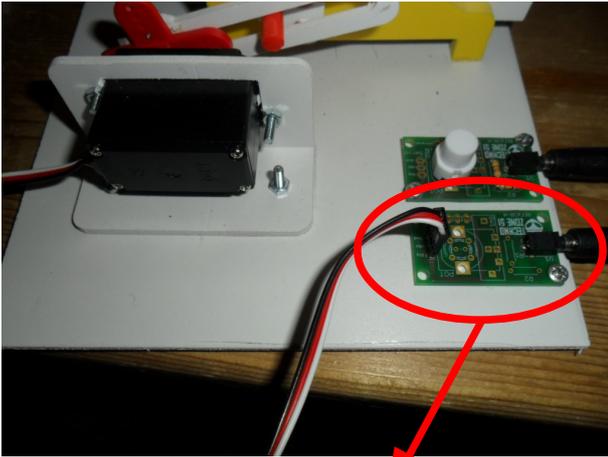


**On commence par vérifier qu'aucune touche n'a été appuyée.**

Si aucune touche de la télécommande n'a été appuyée, on reboucle vers le début du programme (on peut également insérer ici d'autres actions à réaliser en boucle).

Si une touche de la télécommande a été appuyée, on teste s'il s'agit de la touche « marche/arrêt » ou de la touche « mode » ou d'une autre. Dans tous les cas, il faut impérativement vider le tampon de la télécommande avant de revenir sur la boucle principale car un test ne vide pas le tampon. Si l'on oublie de vider le tampon de la télécommande, le premier test «Aucune touche de la télécommande n'est appuyée » sera toujours négatif créant une boucle infinie.

# CARTES SORTIE SERVO-MOTEUR

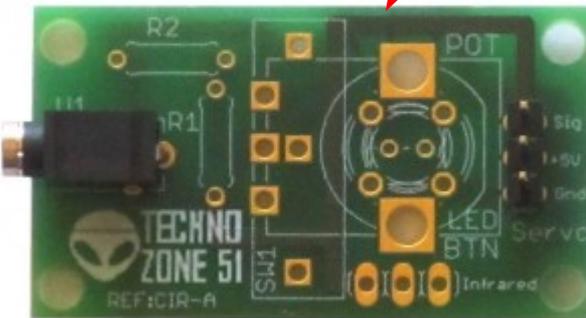


+5V Masse  
Connecter une alimentation externe ( 5 V - 500 ma )

La carte actionneur "sortie Servo-Moteur alimentation séparée" permet d'ajouter une sortie pour Servo Moteur de modélisme. Cette carte d'entrée-sortie se connecte rapidement sur une sortie servo de la carte EASYCON1.

L'alimentation du servo-moteur doit être fournie par une alimentation séparée de 5V régulée et stabilisée (6V maximum sous peine de destruction des servo-moteurs standards).

Chaque servo-moteur standard consomme environ 500mA. Pour commander 4 servo-moteurs standards, il vous faudra donc une alimentation de 5V 2A.



La carte actionneur "sortie Servo-Moteur" permet d'ajouter une sortie pour Servo Moteur de modélisme. L'alimentation du servo-moteur étant directement prise du 5V de la carte EASYCON1, **il est conseillé de ne connecter que deux servo-moteurs.**

Lorsque vous connectez une carte servo-moteur sur l'easycon, il y a 4 actions par défaut :

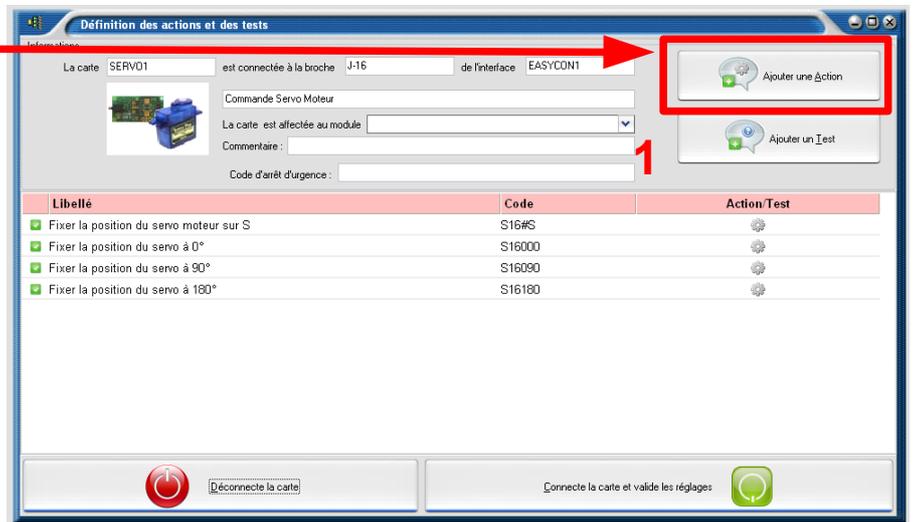
Rotation du servo-moteur en fonction d'un potentiomètre.

Rotation du servo-moteur en fonction d'angle précis.

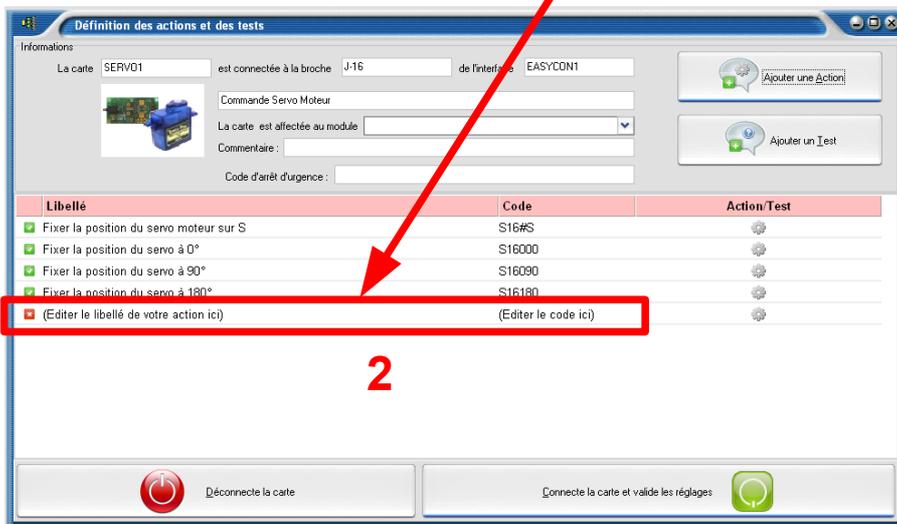
| Libelle   | Code   | Action/Test |
|---|--------|-------------|
| <input checked="" type="checkbox"/> Fixer la position du servo moteur sur S | S16#S  | ⚙️          |
| <input checked="" type="checkbox"/> Fixer la position du servo à 0°         | S16000 | ⚙️          |
| <input checked="" type="checkbox"/> Fixer la position du servo à 90°        | S16090 | ⚙️          |
| <input checked="" type="checkbox"/> Fixer la position du servo à 180°       | S16180 | ⚙️          |

Il est possible bien sûr d'ajouter d'autres actions dont les angles sont différents.

1) Cliquer sur **Ajouter une action**



2) Cliquer sur le libellé et le modifier en rajoutant le **code** action Sbbaaa (bb représente le numéro de la broche et aaa représente l'angle en degrés).

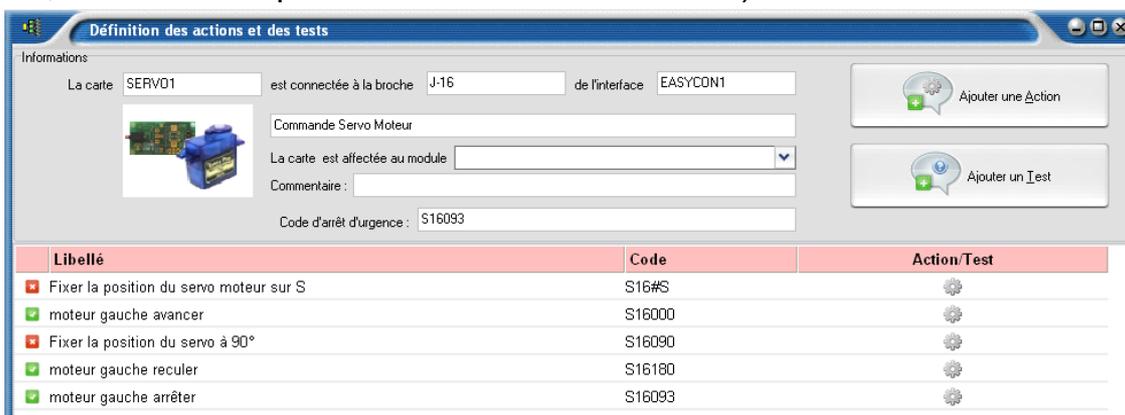


3) N'oubliez pas d'activer l'action pour qu'elle apparaisse lors de l'édition des organigrammes.



## Cas particulier des servomoteurs à rotation continue

L'angle que vous allez préciser dans vos actions va permettre de fixer la vitesse du servomoteur (avec 0 étant la pleine vitesse dans un sens, 180 étant la pleine vitesse dans l'autre sens, et une valeur proche de 90 entraînant l'arrêt).



# PRECAUTIONS

## 1) Attention au branchement et débranchement « à chaud » des cordons jacks.

Ne **jamais** manipuler des cordons jacks (les brancher ou débrancher des cartes) lorsque la carte est alimentée (par USB ou alimentation externe). Cela va occasionner des **courts circuits** ! Si vous voulez retirer ou ajouter des cordons avec des cartes :

- débrancher l'alimentation électrique de votre interface
- déconnecter USB avec votre ordinateur
- brancher ou débrancher votre cordon jack.
- connecter l'USB
- brancher l'alimentation électrique
- sur le logiciel Organigram, il faut vous reconnecter à la maquette en cliquant sur la *roue dentée*

dentée



2) Lors de la mise en fonctionnement de la carte Arduino, les broches de celle-ci sont automatiquement mises à 1, d'où la mise en route des moteurs qui y sont connectés jusqu'à ce que le programme SHELL\_MEGA démarre réellement. En effet il y a un temps de latence d'environ 1s dû au démarrage/initialisation de la carte arduino.

Durant tout ce temps, les broches restent à 1 ! Ce n'est que lorsque SHELL\_MEGA démarre qu'elles sont initialisées à 0 (les moteurs s'arrêtent).

*Pour pallier à ce problème, il faut ajouter un interrupteur à chaque moteur.*

**Avant chaque mise en fonctionnement de la carte arduino, il faut couper les moteurs.**

Un interrupteur permet de couper rapidement le moteur :

