

Tutoriel Automation B&R

SHOW LASER

Tutoriel réalisé par :
GUILLEUX Benjamin
CHAPERON Jérémie
CHOUTEAU Lucie

Projet encadré par :
M. LAGRANGE



Année 2013-2014

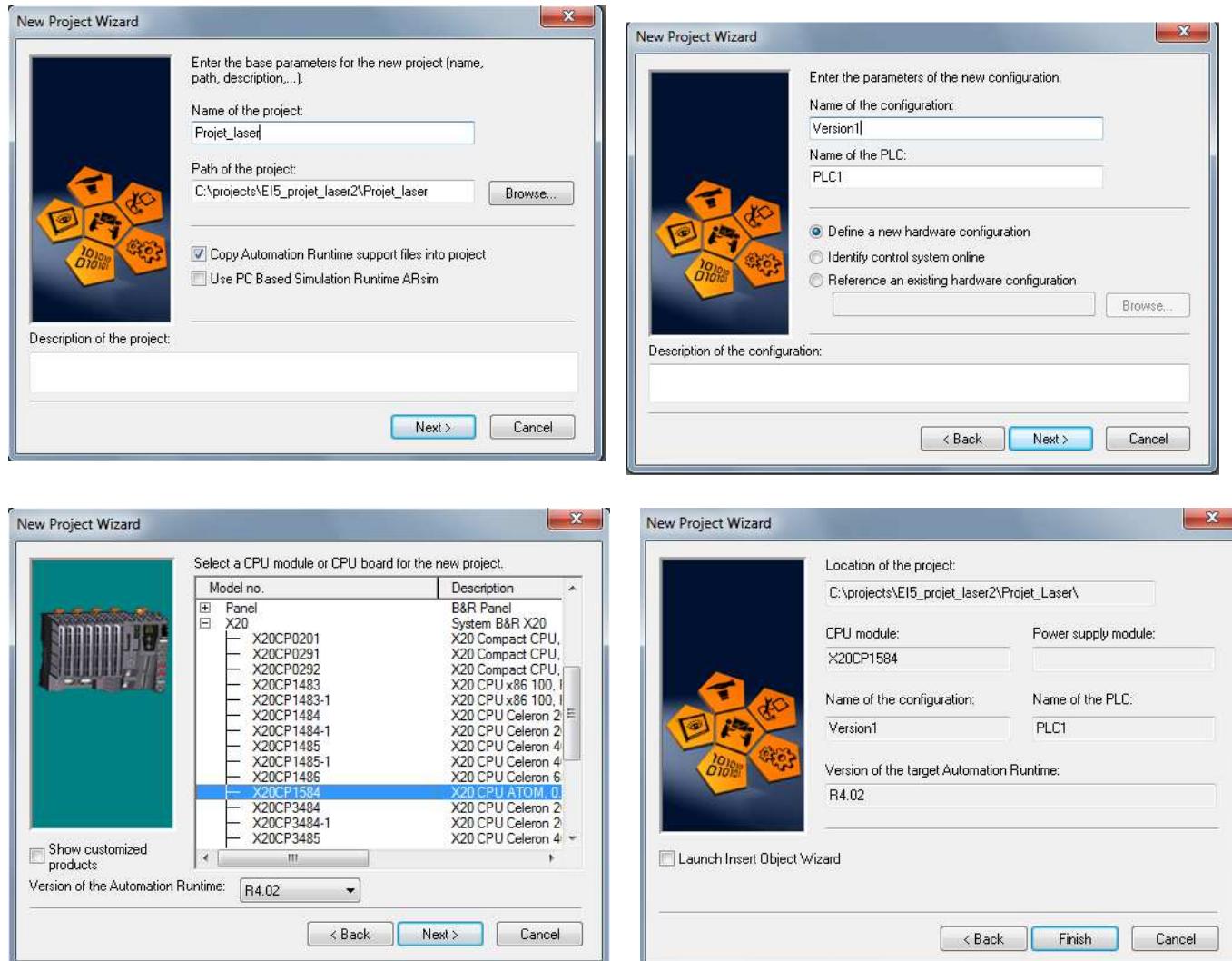
ISTIA – Université d'Angers

PREMIER PAS AVEC AUTOMATION STUDIO.....	3
<i>Création du projet.....</i>	<i>3</i>
<i>Mise à jour de la bibliothèque.....</i>	<i>4</i>
<i>Réalisation du câblage</i>	<i>4</i>
<i>Configuration de l'adresse IP</i>	<i>5</i>
<i>Connexion entre la CPU et l'ordinateur</i>	<i>6</i>
<i>Déclaration des équipements (moteurs)</i>	<i>7</i>
<i>Création de la compact flash</i>	<i>8</i>
<i>Enlever le mode simulation</i>	<i>9</i>
1^{er} DÉMARRAGE.....	10
<i>Enlevez les défauts</i>	<i>10</i>
<i>Autotuning</i>	<i>11</i>
<i>Sauvez dans la compact flash.....</i>	<i>11</i>
PROGRAMME	12
<i>Création d'un programme</i>	<i>12</i>
<i>Mouvement sans modèle géométrique.....</i>	<i>13</i>
<i>Profil de cam</i>	<i>15</i>
<i>Création d'une fonction</i>	<i>18</i>

PREMIER PAS AVEC AUTOMATION STUDIO

Création du projet

- *Création d'un nouveau projet*



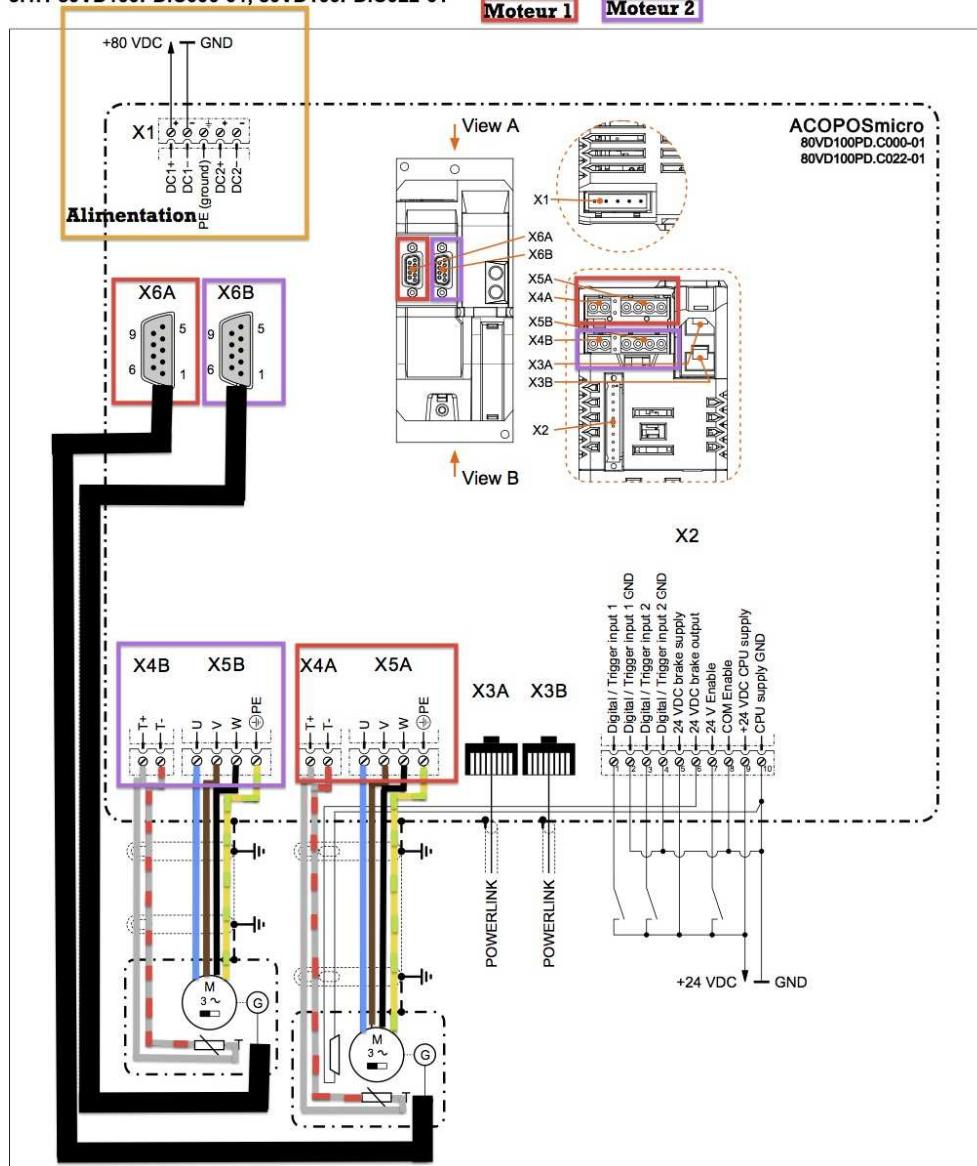
- Choisissez le X20CP1584, qui est la référence de l'automate.
- Si vous ne trouvez pas votre automate, exécutez le point : "mise à jour de la bibliothèque".
- Puis recommencez.

Mise à jour de la bibliothèque

- Allez dans Tools/Upgrade
- Installez la librairie la plus récente correspondant à votre CPU.

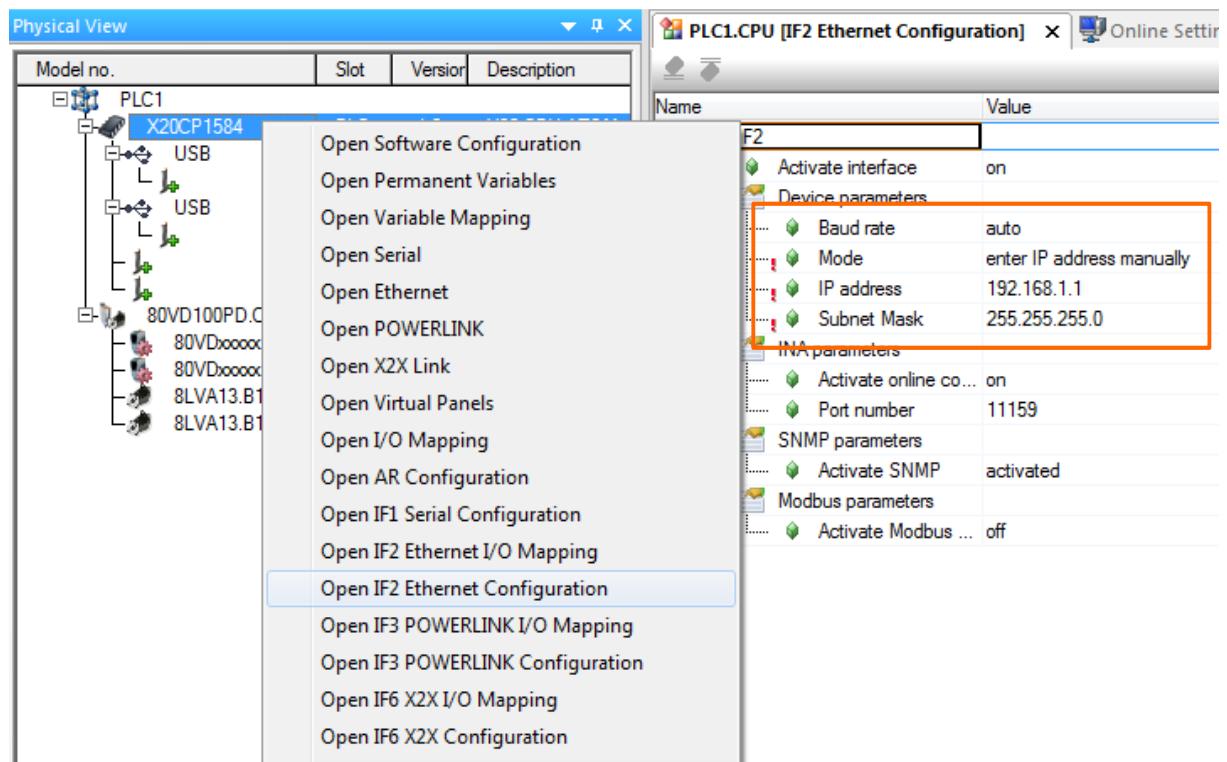
Réalisation du câblage

3.1.1 80VD100PD.C000-01, 80VD100PD.C022-01



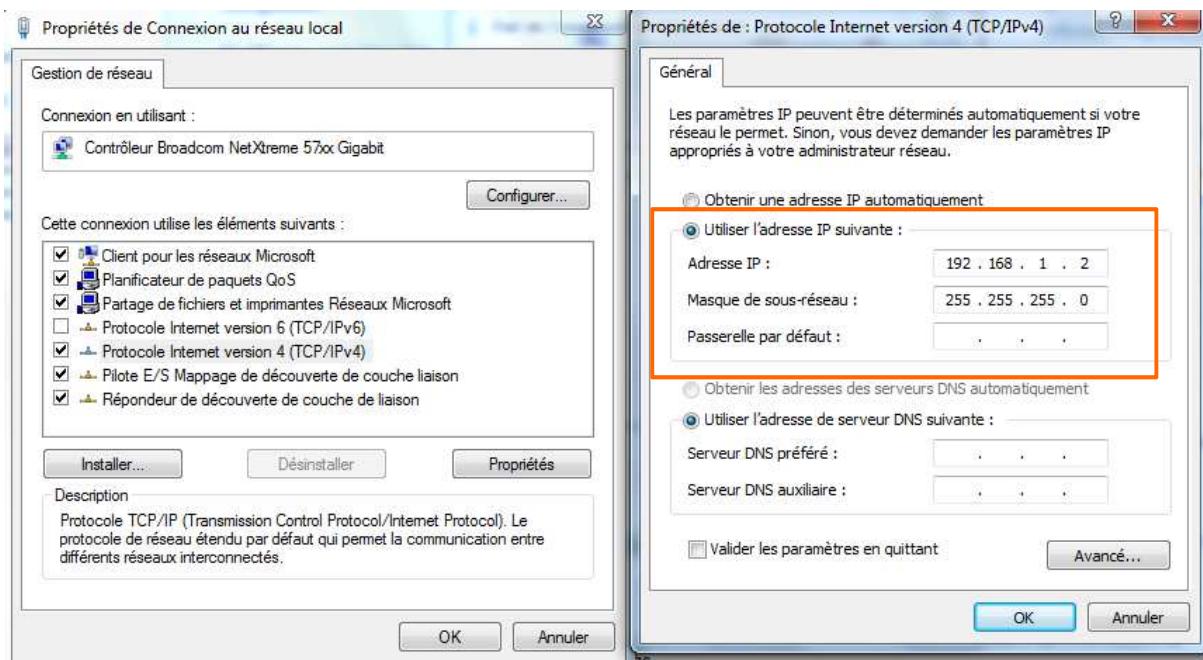
Configuration de l'adresse IP

- Allez dans *PhysicalView/ Open IF2 Ethernet Configuration*.
- Renseignez les adresses IP.

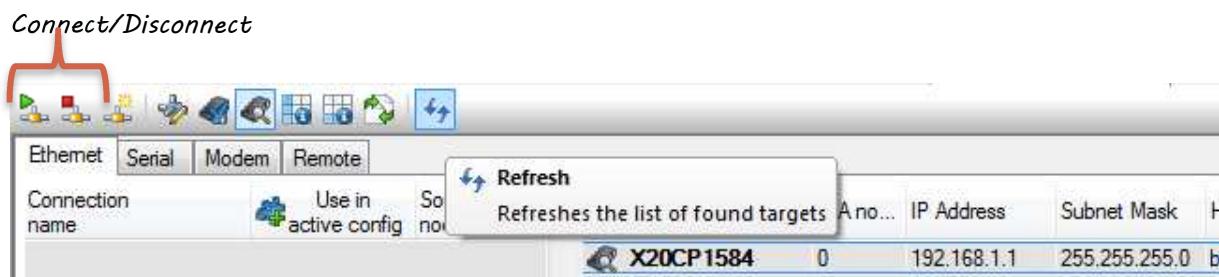


Connexion entre la CPU et l'ordinateur

- Indiquez une adresse IP à votre ordinateur :



- Sur Automation studio, allez dans Online/Setting
- Connectez-vous à la carte.
- Réalisez un « refresh » si celle-ci n'apparaît pas.



Déclaration des équipements (moteurs)

- Dans Physical View, ouvrez Open POWERLINK

Physical View

Model no.	Slot	Version	Description
PLC1		1.0.2.0	X20 CPU HAT
X20CPU1524			Open Software Configuration Open Permanent Variables Open Variable Mapping Open Serial Open Ethernet
80VD	IF3		Open POWERLINK Open X2X Link Open Virtual Panels Open I/O Mapping Open AR Configuration Open IF1 Serial Configuration Open IF2 Ethernet I/O Mapping Open IF3 Ethernet Configuration Open IF3 POWERLINK I/O Mapping Open IF3 POWERLINK Configuration Open IF6 X2X I/O Mapping Open IF6 X2X Configuration

PLC1.CPU [POWERLINK]

Slave Module	Slave Backplane	Connection	Description
IF3 80VD100PC	ST1	ACOPOSmicro Servo, 2x 8 A, 2x EnDat	

- Clic droit sur IF3 Insert
- Choisir la référence du variateur

Select controller module

Model no.	Description
ACOPOSmulti	B&R ACOPOSmulti
8B0Poooooooooooo	ACOPOSmulti power supply unit
8BVooooooooDSxxxx	ACOPOSmulti inverter unit w/o
8BVooooooooDxxxxx	ACOPOSmulti inverter unit, 2
8BVooooooooSxxxxx	ACOPOSmulti inverter unit w/o
8BVooooooooSxxxxx	ACOPOSmulti inverter unit, 1
8BVooooooooooooxx	ACOPOSmulti power supply unit
8CVE28000HC00.00-1	Motion POWERLINK fuse module
8CVbooooS1xxSxxxxx	ACOPOSmulti65 inverter unit
80VD100PD.1xxSxxxxx	B&R ACOPOSmulti
80VD100PD.C000-01	ACOPOSmicro Servo, 2x 8 A
80VD100PS.C022-01	ACOPOSmicro Servo, 2x 8 A
80VD100PS.C02X-01	ACOPOSmicro Servo, 1x 8 A
ACOPOSinverter	B&R ACOPOSinverter
8B4oooooooo.01P-1	ACOPOSinverter P84
X20	System B&R X20
X20BC0083	X20 Bus Controller POWERLINK
X20RC1083	X20 RIO Controller POWERLINK

Insert module Replace module

Select Drive Usage

Global Settings

Continue with axis definition

CPU runs synchronized to network (recommended)

Power Supply

400 V Supply Voltage 80 VDC Bus Voltage

AC powered DC powered

Single phase operation

Module Parameter

Module:	ACPμDoubleE1
System:	ACOPOSmicro
Type:	POWERLINK controller
Model number:	80VD100PD.C000-01
Module address:	none
Backplane number:	none
Address:	BP
Version:	1.0.2.0

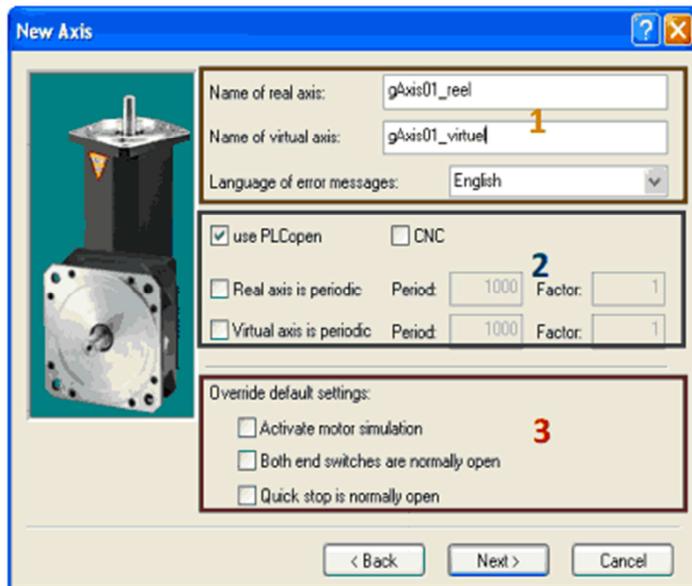
Enter node number: **2**

Must correspond with dial switch setting on the module.

Enter optional hardware module name:

< Back Next > Cancel

Le « 2 » correspond aux nombres de moteurs déclarés.



Configuration : défini le nom de l'axe réel et d'un axe virtuel du moteur.

Pour un axe réel, on a un axe virtuel utilisé pour synchroniser les axes des différents moteurs entre eux.

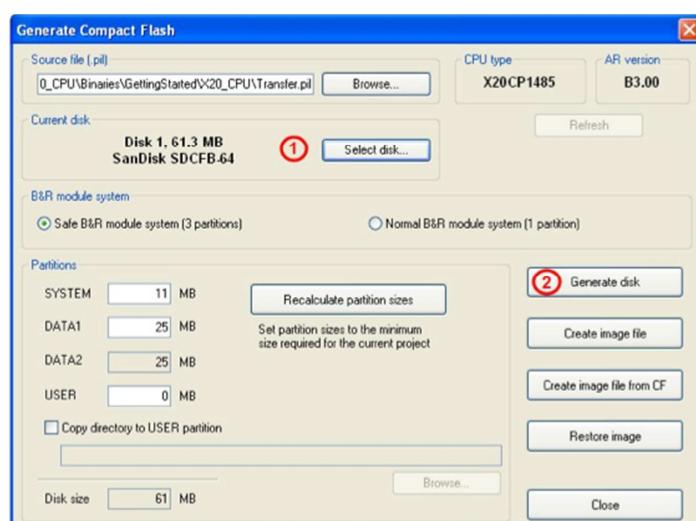
Paramètres : PLC Open : standard utilisé par les contrôleurs d'axe.

CNC : commande numérique.

On peut définir si les axes ont un fonctionnement périodiques ou non.

Override default settings : permet de définir un moteur en simulation.

Création de la compact flash

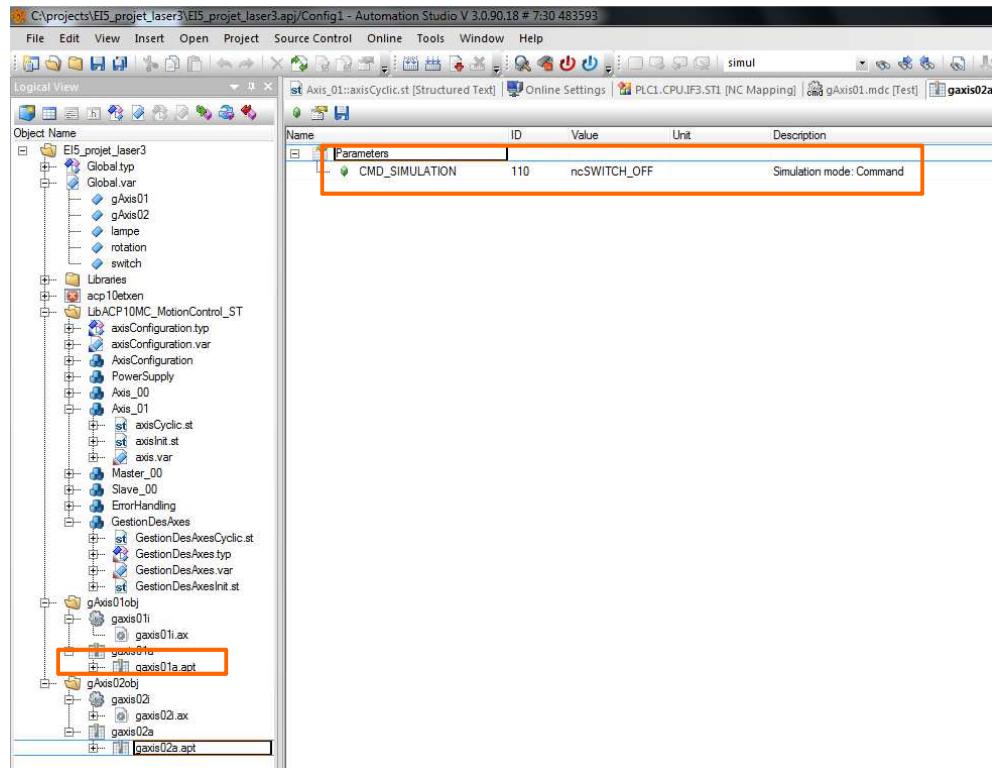


- Branchez le lecteur de carte qui permet de flasher la carte sur l'ordinateur.

- Tools/Create compact flash.
- Sélectionnez le disque (1).
- Générez le disque (2).
- Enlevez la carte et l'insérez dans la CPU.

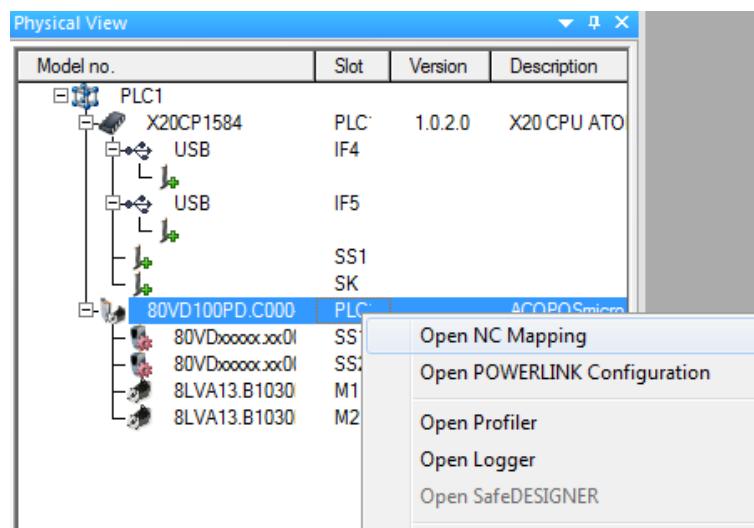
Enlever le mode simulation

- Allez dans `gaxis01obj / gaxis01a / gaxis01a.apt`.
- Ajoutez `CMD_SIMULATION` et mettre sa valeur à `false`.

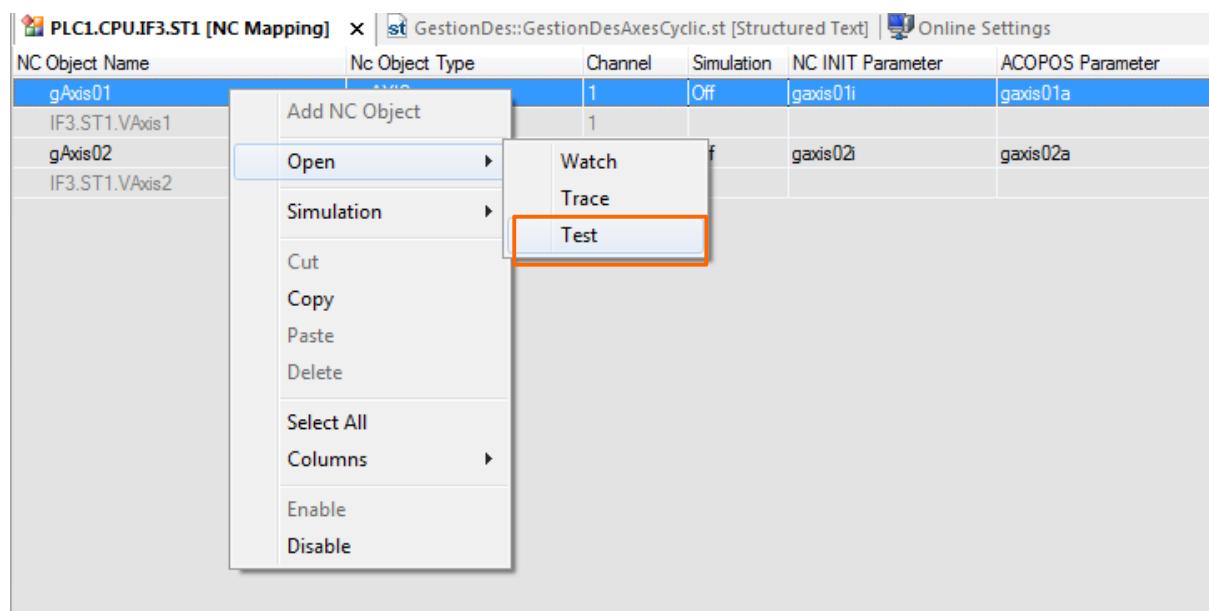


1er DÉMARRAGE

Enlevez les défauts

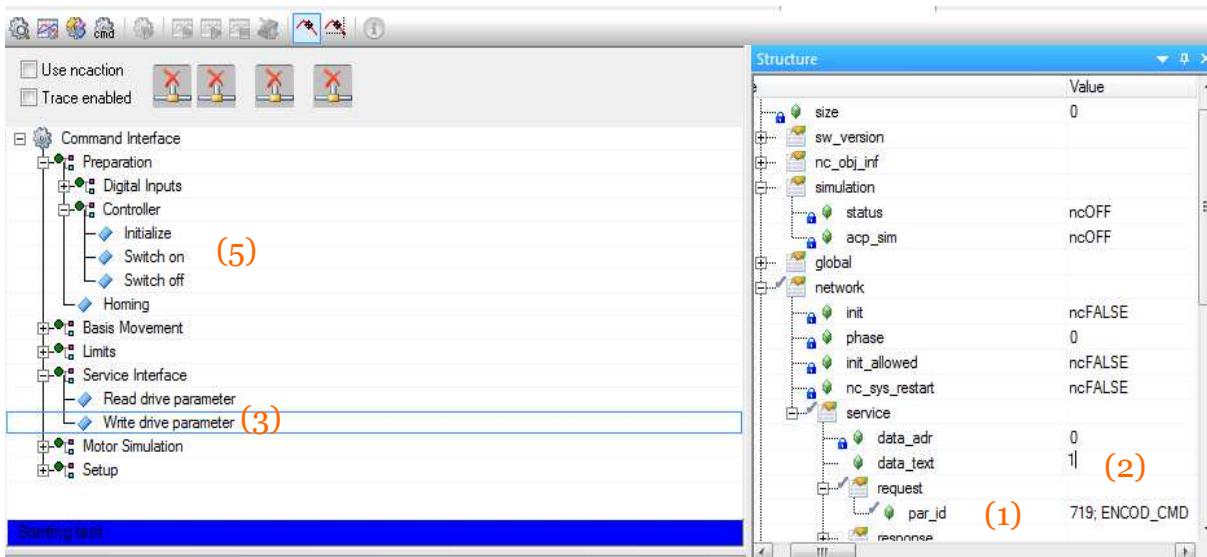


- Au premier démarrage, il faut acquitter plusieurs défauts:
- Allez dans physical view.
- Clic droit sur 80VD100PD.C000
- Open NC Mapping.



- Choisir Exclusive mode





- Dans la partie de droite, network/service/request/par-id/ cherchez le paramètre 719 (1).
- Dans data-text, mettre la valeur 1,(2).
- Puis double clic sur write drive parameters à gauche(3).
- Acquittez les défauts (à l'aide du feu vert).
- Allumez le moteur avec le bouton controller/ Switch on et éteignez le rapidement (5).

Auto-tuning

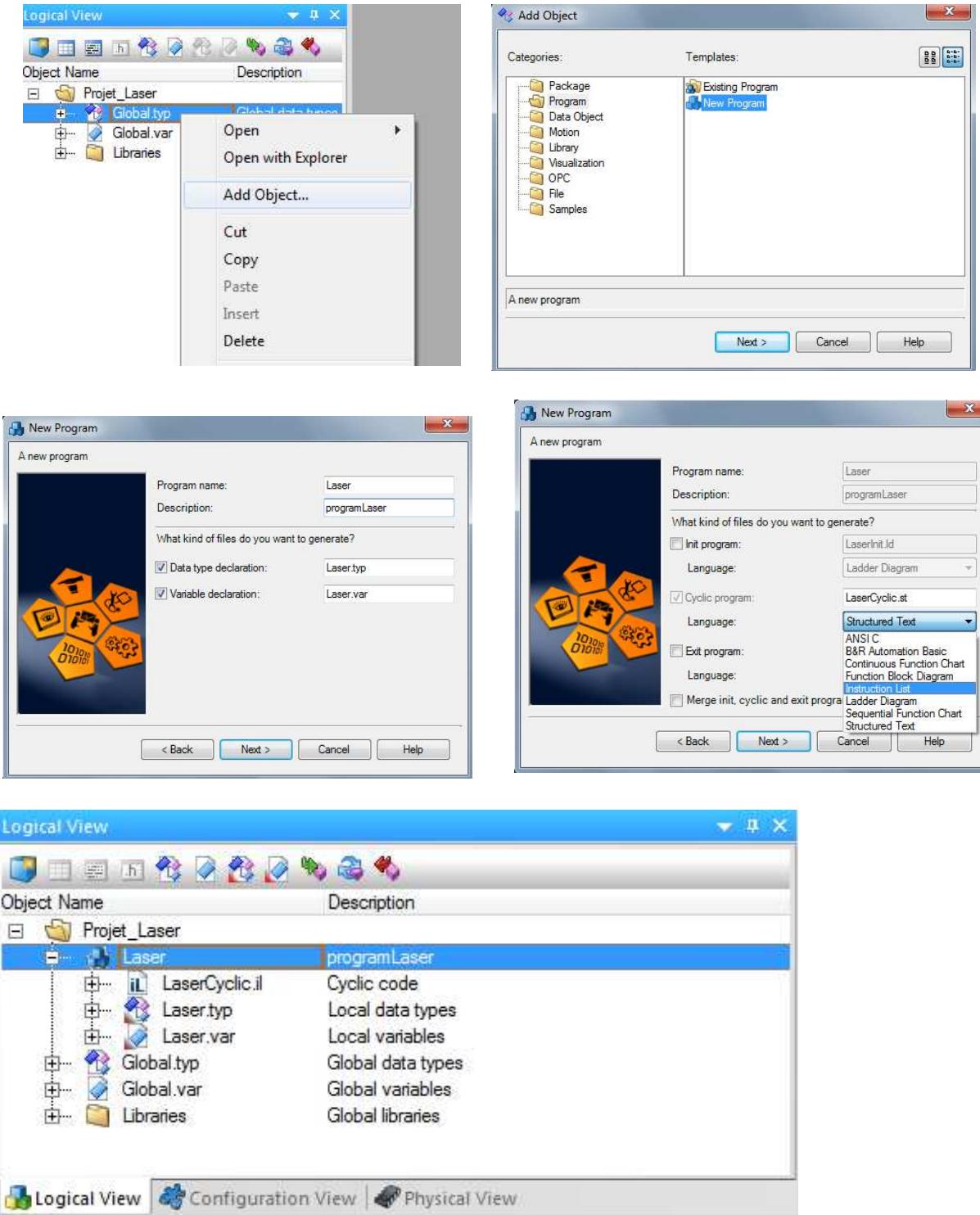
- ➔ Setup/controller auto-tuning/ start tuning.
- ➔ Écran de droite/ setup/ controller/parameter/mode mettre ncSPEED et ne pas modifier les autres paramètres.
- ➔ Écran de gauche start tuning (il doit y a voir trois bruits)
- ➔ Écran de droite / setup/ controller/parameter/mode mettre ncPOSITION
- ➔ Écran de gauche start tuning (il doit y a voir un petit bruit)
- ➔ Sauvegardez et transférez dans la compact flash.

Sauvez dans la compact flash

- Tools / create compact Flasch
- Generate disk
- Yes

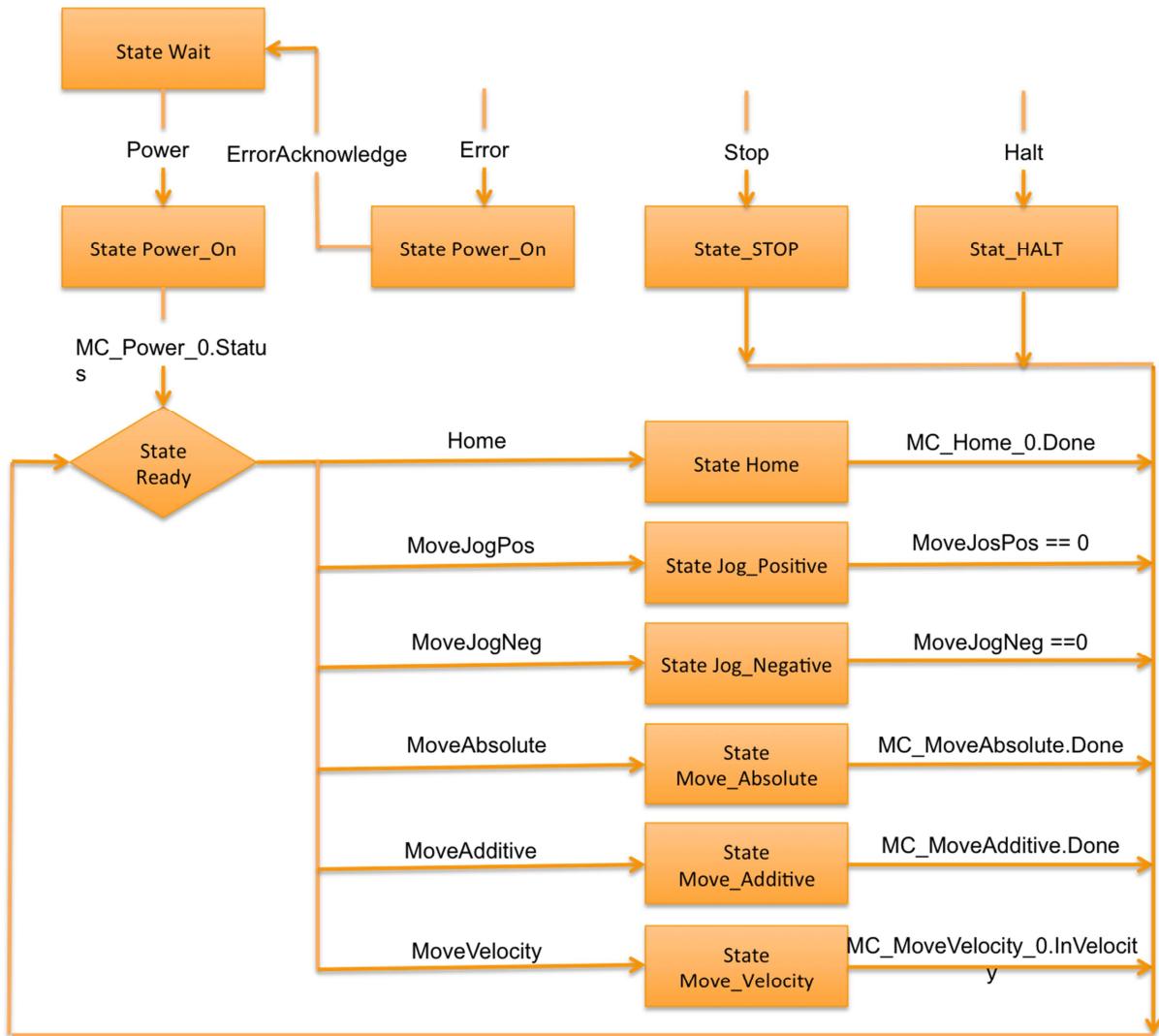
PROGRAMME

Création d'un programme



Mouvement sans modèle géométrique

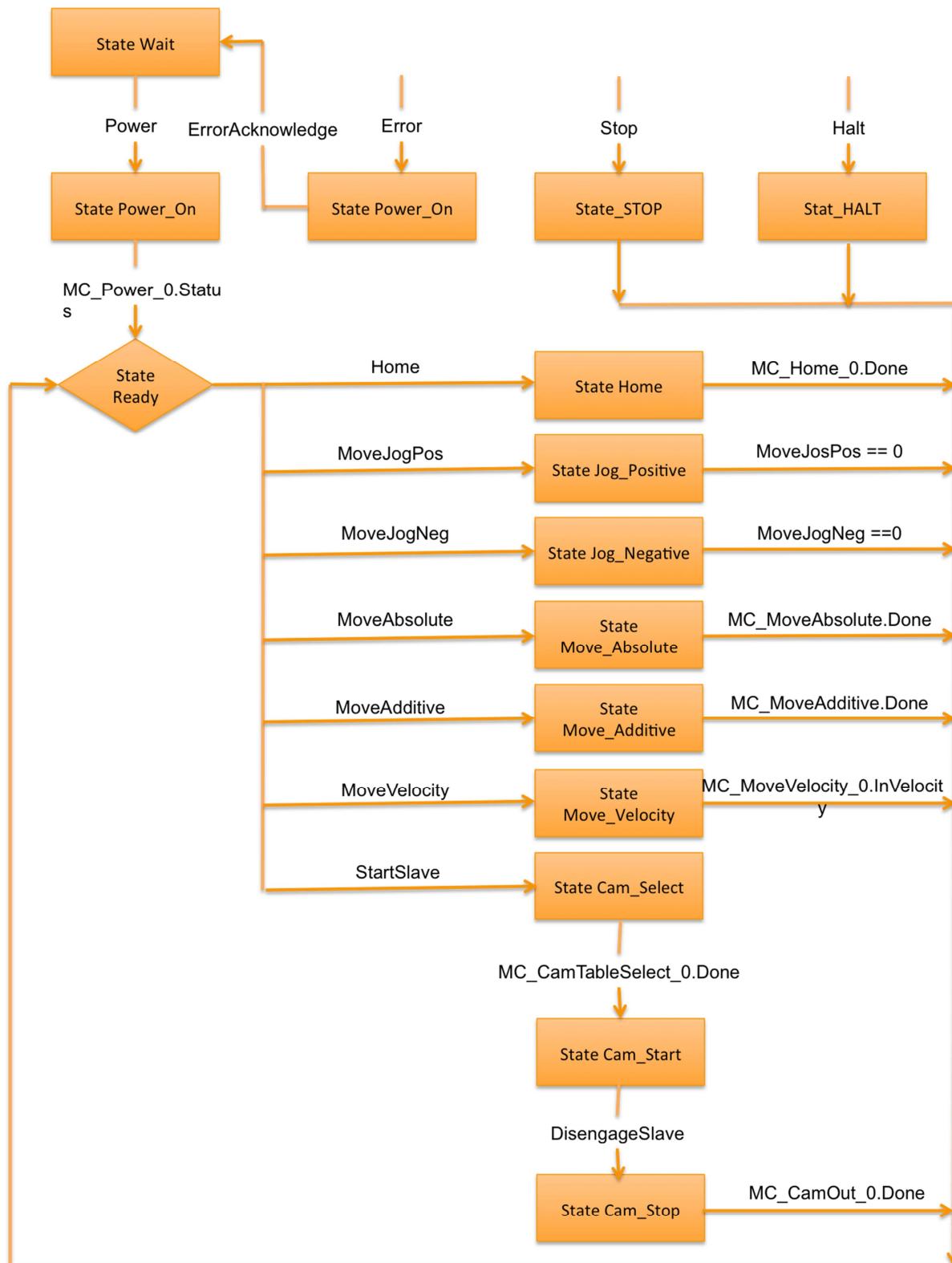
Schéma de principe



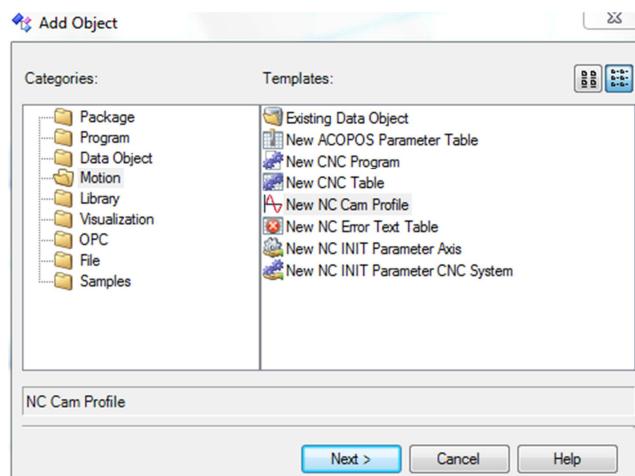
Command.Power	Démarrage du moteur
Command.Home	Effectuer la procédure de prise d'origine pour l'axe
Command.MoveJogPos	Mouvement dans les positifs aussi longtemps que la variable de commande est active
Command.MoveJogNeg	Mouvement dans les négatifs aussi longtemps que la variable de commande soit active
Command.MoveAbsolute	lancer un mouvement avec une position définie
Command.MoveAdditive	Lancer un mouvement avec un trajet défini
Command.MoveVelocity	Lancer un mouvement avec une vitesse définie
Command.Halt	Arrêter un mouvement actif
Command.Stop	Arrêter un mouvement actif et éviter un nouveau mouvement de départ
Command.ErrorAcknowledge	Acquitter une erreur sur un axe

Profil de came

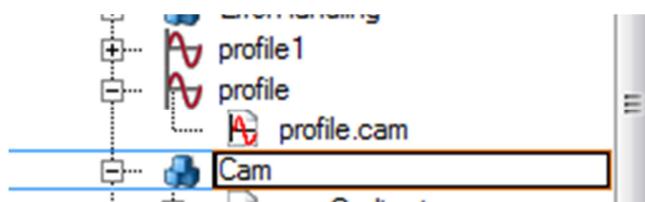
Schéma de principe



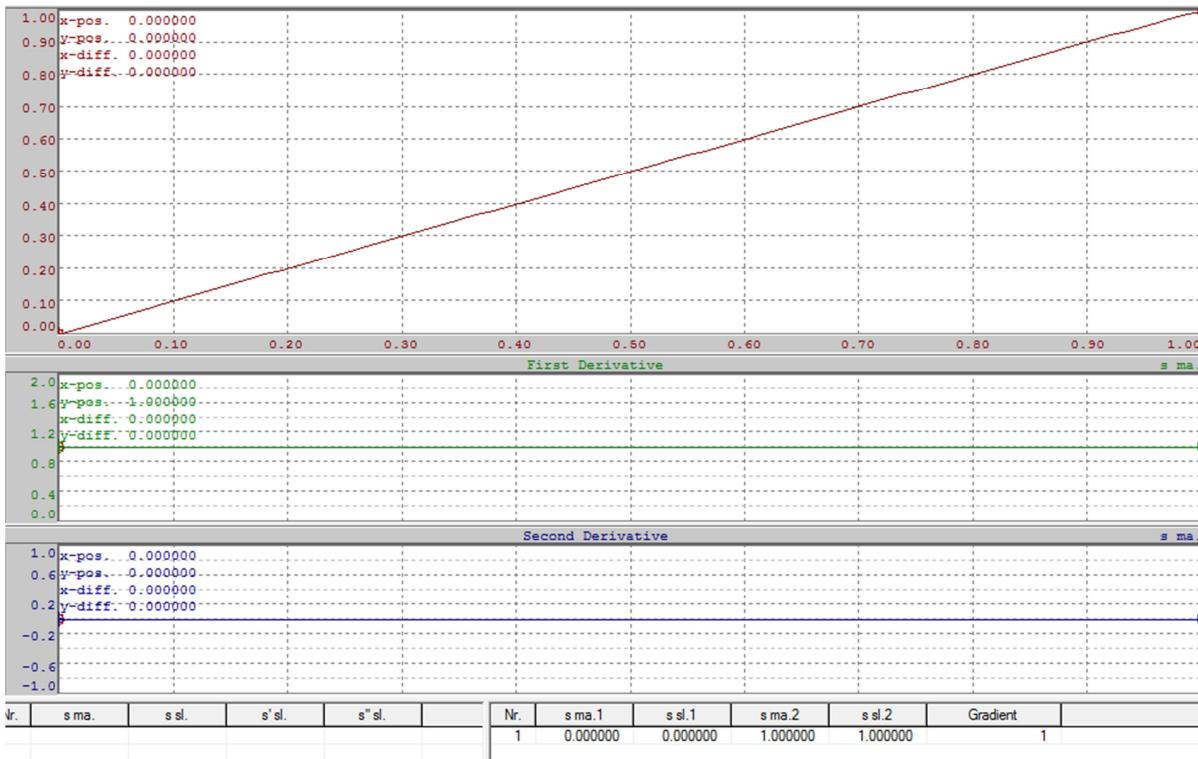
Command.Power	Démarrage du moteur
Command.Home	Effectuer la procédure de prise d'origine pour l'axe
Command.StartSlave	Démarrer le lien entre l'axe maître et l'axe esclave à l'aide d'un profil de cam
Command.DisengageSlave	Arrêter le lien entre l'axe maître et l'axe esclave
Command.MoveJogPos	Mouvement dans les positifs aussi longtemps que la variable de commande est active
Command.MoveJogNeg	Mouvement dans les négatifs aussi longtemps que la variable de commande est active
Command.MoveAbsolute	lancer un mouvement avec une position définie
Command.MoveAdditive	Lancer un mouvement avec un trajet défini
Command.MoveVelocity	Lancer un mouvement avec une vitesse définie
Command.Halt	Arrêter un mouvement actif
Command.Stop	Arrêter un mouvement actif et éviter un nouveau mouvement de départ
Command.ErrorAcknowledge	Acquitter une erreur sur un axe



- *Création d'un profil de cam*
- *Clic droit sur le fichier*
- *Add object*

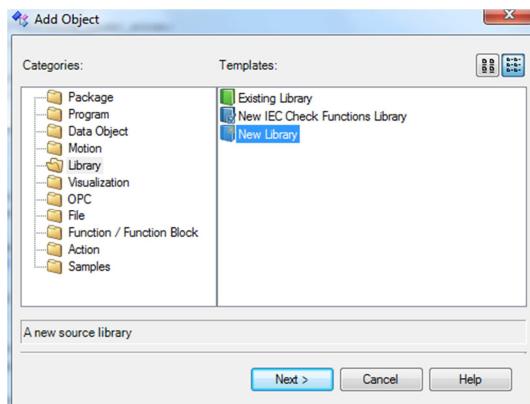


- *Dans notre cas on a créé deux profils pour réaliser deux diagonales.*



Ceci est la courbe d'un profil de came qui nous permet de réaliser une diagonale.
 L'abscisse correspond à la déclaration du maître et l'ordonnée correspond à la déclaration de l'esclave.
 Par exemple lorsqu'on indique au maître d'aller en 0,50 l'esclave veut se déplacer également de 0,50.

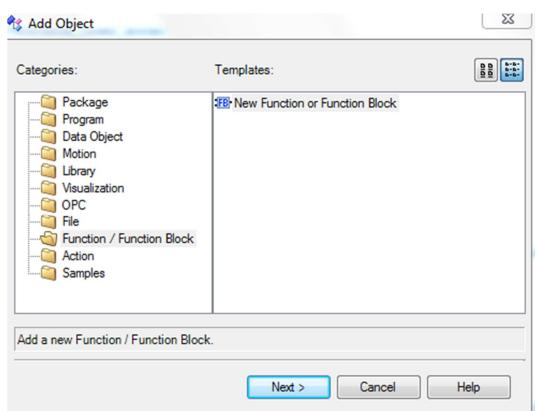
Création d'une fonction



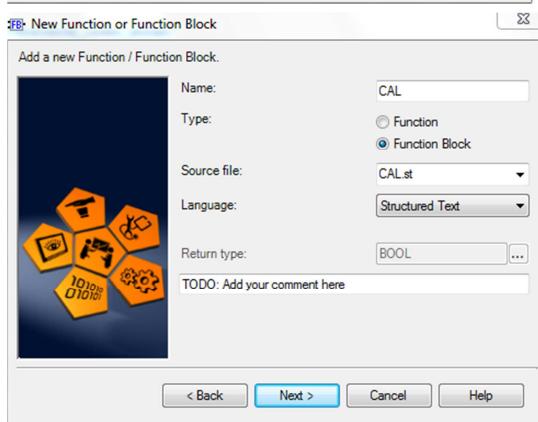
- *Création d'une librairie*
- *Clic droit sur le fichier*
- *Add object*



- *Choisissez le mode IEC library, ce qui permet par la suite d'écrire avec le langage ST.*
- *Un nouveau dossier Malibrain s'est créé*
- *Clic droit sur le fichier Malibrain*
- *Add object*



- *Création d'un bloc fonctionnel*



- *On indique que le bloc fonctionnel sera en langage ST*

•	Z	REAL	<input type="checkbox"/>	VAR_INPUT
•	Y	REAL	<input type="checkbox"/>	VAR_INPUT
•	Distance_laser_ecran	REAL	<input type="checkbox"/>	VAR_INPUT
•	Distance_y	REAL	<input checked="" type="checkbox"/>	VAR_OUTPUT
•	Distance_z	REAL	<input checked="" type="checkbox"/>	VAR_OUTPUT
•	B	REAL	<input type="checkbox"/>	VAR
•	Teta1	REAL	<input type="checkbox"/>	VAR
•	Teta2	REAL	<input type="checkbox"/>	VAR
•	A	REAL	<input type="checkbox"/>	VAR
•	Adeux	REAL	<input type="checkbox"/>	VAR
•	Bdeux	REAL	<input type="checkbox"/>	VAR
•	Ydeux	REAL	<input type="checkbox"/>	VAR
•	Yquatre	REAL	<input type="checkbox"/>	VAR
•	Zdeux	REAL	<input type="checkbox"/>	VAR
•	Zquatre	REAL	<input type="checkbox"/>	VAR

- On déclare les variables dont nous avons besoin
- On écrit le code du bloc fonctionnel
- Pour utiliser le bloc fonctionnel CALCUL il faut:
- Allez dans les variables globales
- Déclarez une variable CALCUL_O de type CALCUL

```

FUNCTION_BLOCK CALCUL
  A := 2*Distance_laser_ecran;
  Adeux:= EXPT(A,2);

  Zquatre := EXPT(Z,4);
  Zdeux:= EXPT(Z,2);

  Yquatre:= EXPT(Y,4);
  Ydeux := EXPT(Y,2);

  IF (Z=0) THEN
    Teta2 := 0;
  ELSE
    Teta2 := ATAN( (A/(2*Zdeux)) * ( SQRT( ((4*Zquatre)/Adeux) + Zdeux ) - (SQRT(Zdeux)) ) );
  END_IF;

  B := A / ( (COS(Teta2)*COS(Teta2)) - (SIN(Teta2)*SIN(Teta2)) );
  Bdeux := EXPT(B,2);

  IF (Y=0) THEN
    Teta1 := 0;
  ELSE
    Teta1 := ATAN( (B/(2*Ydeux)) * ( SQRT( ((4*Yquatre)/Adeux) + Ydeux ) - (SQRT(Ydeux)) ) );
  END_IF;

  Distance_y := Teta1 * 1000 / (2 * 3.14159265358979);
  Distance_z := Teta2 * 1000 / (2 * 3.14159265358979);

END_FUNCTION_BLOCK

```

- Dans le programme, écrire:
- `CALCUL_O(Z := A_Faire_Z, Y := A_Faire_Y, Distance_Laser_Ecran := (500));`