

Bibliothèque CDS5500 (En cours de rédaction)

Cible : Servomoteur AX12+

Langage : C++ (Arduino)

IDE : Visual studio 2013

Carte : Arduino Méga2560, Arduino Méga ADK

Caractéristiques et documentation du servomoteur

A compléter

Exemples

Nom	IDE	Description	Détails
RHEX2016V2	Visual Studio 2016 + Visual Micro	Commande de deux servomoteurs (position et rotation permanente) dans les deux sens.	
Test10Serial		Commande d'un moteur (rotation permanente). Changement de la vitesse de rotation en fonction de la position angulaire. Bouton Marche / Arrêt Transmission de la position avec la liaison série 2 pour sauvegarde dans un fichier .txt via le terminal de CVAVR.	Annexe 1
Test10SerialV2a Et b		Commande de deux moteurs (rotation permanente) Changement de la vitesse de rotation en fonction de la position angulaire. Bouton Marche / Arrêt Resynchronisation lors de chaque changement de vitesse. (V2a : solution avec des booléens, V2b : solution avec des compteurs)	Annexe1a
Test10SerialV3		Commande de trois moteurs (rotation permanente) Changement de la vitesse de rotation en fonction de la position angulaire. 2 Boutons : BP2 Marche / BP3 Arrêt (Clavier Digilent) Resynchronisation lors de chaque changement de vitesse.	Annexe1b
Test11Serial		Lecture de la table de contrôle d'un servomoteur et affichage dans le moniteur série Arduino.	Annexe 2





Constantes

Logiciel		Table de contrôle (Matériel) (EEPROM en gras, sinon RAM)			
Désignation	Nom de la constante	Add	Item	Access	Initial Value
Modèle de servomoteur	CTRAdd_MODEL_NUMBER_L	0(0x00)	Model Number(L)	RD	12(0x0C)
	CTRAdd_MODEL_NUMBER_H	1(0x01)	Model Number(H)	RD	0(0x00)
Version du micro logiciel	CTRAdd_VERSION	2(0x02)	Version of firmware	RD	?
Identifiant	CTRAdd_ID	3(0x03)	ID	RD,WR	1(0x01)
Débit binaire	CTRAdd_BAUD_RATE	4(0x04)	Baud Rate	RD,WR	1(0x01)
Temporisation avant transmission de la réponse	CTRAdd_RETURN_DELAY_TIME	5(0x05)	Return Delay Time	RD,WR	250(0xFA)
Valeur minimum de l'angle à atteindre	CTRAdd_CW_ANGLE_LIMIT_L	6(0x06)	CW Angle Limit(L)	RD,WR	0(0x00)
	CTRAdd_CW_ANGLE_LIMIT_H	7(0x07)	CW Angle Limit(H)	RD,WR	0(0x00)
Valeur maximum de l'angle à atteindre	CTRAdd_CCW_ANGLE_LIMIT_L	8(0x08)	CCW Angle Limit(L)	RD,WR	255(0xFF)
	CTRAdd_CCW_ANGLE_LIMIT_H	9(0x09)	CCW Angle Limit(H)	RD,WR	3(0x03)
		10(0x0A)	Reserved	-	0(0x00)
Température max	CTRAdd_LIMIT_TEMPERATURE	11(0x0B)	The Highest Limit Temperature	RD,WR	85(0x55)
Tension min	CTRAdd_DOWN_LIMIT_VOLTAGE	12(0x0C)	The Lowest Limit Voltage	RD,WR	60(0x3C)
Tension max	CTRAdd_UP_LIMIT_VOLTAGE	13(0x0D)	The Highest Limit Voltage	RD,WR	190(0xBE)
Couple max	CTRAdd_MAX_TORQUE_L	14(0x0E)	Max Torque(L)	RD,WR	255(0xFF)
	CTRAdd_MAX_TORQUE_H	15(0x0F)	Max Torque(H)	RD,WR	3(0x03)
Condition de renvoi d'un Status Packet	CTRAdd_STATUS_RETURN_LEVEL	16(0x10)	Status Return Level	RD,WR	2(0x02)
Led clignote si erreur	CTRAdd_ALARM_LED	17(0x11)	Alarm Led	RD,WR	4(0x04)
Couple moteur désactivé si erreur	CTRAdd_ALARM_SHUTDOWN	18(0x12)	Alarm Shutdown	RD,WR	4(0x04)
	CTRAdd_OPERATING_MODE	19(0x13)	Reserved	RD,WR	0(0x00)
	CTRAdd_DOWN_CALIBRATION_L	20(0x14)	Down Calibration(L)	RD	?
	CTRAdd_DOWN_CALIBRATION_H	21(0x15)	Down Calibration(H)	RD	?
	CTRAdd_UP_CALIBRATION_L	22(0x16)	UP Calibration(L)	RD	?
	CTRAdd_UP_CALIBRATION_H	23(0x17)	UP Calibration(H)	RD	?
	CTRAdd_TORQUE_ENABLE	24(0x18)	Torque Enable	RD,WR	0(0x00)
Etat de la led	CTRAdd_LED	25(0x19)	LED	RD,WR	0(0x00)
	CTRAdd_CW_COMPLIANCE_MARGIN	26(0x1A)	CW Compliance Margin	RD,WR	0(0x00)
	CTRAdd_CCW_COMPLIANCE_MARGIN	27(0x1B)	CCW Compliance Margin	RD,WR	0(0x00)
	CTRAdd_CW_COMPLIANCE_SLOPE	28(0x1C)	CW Compliance Slope	RD,WR	32(0x20)
	CTRAdd_CCW_COMPLIANCE_SLOPE	29(0x1D)	CCW Compliance Slope	RD,WR	32(0x20)
Position à atteindre	CTRAdd_GOAL_POSITION_L	30(0x1E)	Goal Position(L)	RD,WR	[Adress36]val
	CTRAdd_GOAL_POSITION_H	31(0x1F)	Goal Position(H)	RD,WR	[Adress37]val
Vitesse souhaitée	CTRAdd_MOVING_SPEED_L	32(0x20)	Moving Speed(L)	RD,WR	0(0x00)
	CTRAdd_MOVING_SPEED_H	33(0x21)	Moving Speed(H)	RD,WR	0(0x00)
Couple max	CTRAdd_TORQUE_LIMIT_L	34(0x22)	Torque Limit(L)	RD,WR	[Adress14]val
	CTRAdd_TORQUE_LIMIT_H	35(0x23)	Torque Limit(H)	RD,WR	[Adress15]val
Position angulaire	CTRAdd_PRESENT_POSITION_L	36(0x24)	Present Position (L)	RD	?
	CTRAdd_PRESENT_POSITION_H	37(0x25)	Present Position (H)	RD	?
Vitesse	CTRAdd_PRESENT_SPEED_L	38(0x26)	Present Speed(L)	RD	?
	CTRAdd_PRESENT_SPEED_H	39(0x27)	Present Speed(H)	RD	?
Charge	CTRAdd_PRESENT_LOAD_L	40(0x28)	Present Load(L)	RD	?
	CTRAdd_PRESENT_LOAD_H	41(0x29)	Present Load(H)	RD	?
Tension	CTRAdd_PRESENT_VOLTAGE	42(0x2A)	Present Voltage	RD	?
Température	CTRAdd_PRESENT_TEMPERATURE	43(0x2B)	Present Temperature	RD	?
Ecriture terminée	CTRAdd_REGISTERED_INSTRUCTION	44(0x2C)	Registered Instruction	RD,WR	0(0x00)
		45(0x2D)	Reserved	-	0(0x00)
Servomoteur entraîné par la charge	CTRAdd_MOVING	46(0x2E)	Moving	RD	0(0x00)
Verrouillage de la table	CTRAdd_LOCK	47(0x2F)	Lock	RD,WR	0(0x00)
Courant minimum pendant l'opération	CTRAdd_PUNCH_L	48(0x30)	Punch(L)	RD,WR	32(0x20)
	CTRAdd_PUNCH_H	49(0x31)	Punch(H)	RD,WR	0(0x00)

Constructeur

Syntaxe	Description
<code>CDS5500(byte ID);</code>	<p>Construit un objet CDS5500 avec le numéro ID.</p> <p>ID : Identifiant du servomoteur (0 à 253).</p>

Méthodes publiques

Syntaxe	Description
 <code>void DisableEndlessTurn();</code>	Activation du mode "commande en position" avec les valeurs par défaut.
 <code>void EnableEndlessTurn();</code>	Activation du mode "rotation continue".
 <code>unsigned int GetDataInStatusPacket(byte numberOfBytes);</code>	<p>Renvoie la donnée contenue dans un <i>Status Packet</i> sous la forme d'un entier positif codé sur 16 bits..</p> <p>numberOfBytes : nombre d'octets à extraire (1 ou 2)</p> <p>Remarques: Cette méthode doit être précédée de <i>WriteInstructionPacketforReading</i>. si numberOfBytes = 2 la méthode retourne LowBytes + 255*HighBytes</p>
 <code>unsigned int GetPosition();</code>	<p>Renvoie la position angulaire du servomoteur sous la forme d'un entier positif codé sur 16 bits.</p> <p>Celui-ci est :</p> <ul style="list-style-type: none"> - soit représentatif de la position angulaire du servomoteur (lue dans la table de contrôle). - Entre 0° et 300° $0(0x000) \leq \text{position} \leq 1023(0x03FF)$ - Entre 301° et 359° : position = 0 - soit représentatif d'une erreur de transmission : <ul style="list-style-type: none"> - En-tête erronée = 4094, - Pas de réception = 4095 (Rx_time > 400µs) <p><u>Conditions de fonctionnement</u> Baud Rate = 1Mb/s (par défaut) Return Delay Time : par défaut</p>
 <code>boolean Ping();</code>	Renvoie true si le servomoteur est présent dans le réseau, false sinon.
 <code>void Reset();</code>	Initialise la table de contrôle du servomoteur avec les valeurs par défaut. Le nouvel ID est égal à 1.
 <code>void SetDataInControlTable(byte CTRRegisterAdd, byte valeur);</code> <code>void SetDataInControlTable(byte CTRRegisterAdd, byte valeur1, byte valeur2);</code> <code>void SetDataInControlTable(byte CTRRegisterAdd, byte valeur1, byte valeur2, byte valeur3, byte valeur4);</code>	<p>Ecriture de l'octet valeur, à l'adresse CTRRegisterAdd, dans la table de contrôle du servomoteur.</p> <p>CTRRegisterAdd : Adresse du registre dans la table de contrôle. Peut être remplacée par son nom. (Commence par CTR)</p> <p>valeur : octet à placer dans la table.</p>
 <code>Void SetInstructionPacketforReading(byte CTRRegisterAdd, byte numberOfBytes);</code>	<p>Envoie une trame pour lire une valeur, codée sur 1 ou deux octets, dans la table de contrôle du servomoteur.</p> <p>CTRRegisterAdd : Adresse du registre dans la table de contrôle. Peut être remplacée par son nom. (Commence par CTR)</p> <p>numberOfBytes : un ou deux octets</p>
 <code>void SetLed(boolean action);</code>	<p>Comande de la led du servomoteur.</p> <p>Eclairée si action = true. Eteinte si action = false ;</p>
 <code>void SetPosition(unsigned int position, unsigned int velocity);</code>	<p>Déplace l'axe du servomoteur de sa position actuelle à la position spécifiée par <i>position</i> avec la vitesse <i>velocity</i>.</p> <p>position: position à atteindre. 0°(0x00) à 300°(0x3ff)</p> <p>velocity: vitesse angulaire. 14RPM(0x3ff)</p> <p>Remarque : La vitesse angulaire maximum dépend de la tension d'alimentation. velocity = 0 permet de régler la vitesse angulaire à sa valeur maximum pour la tension appliquée. (pas de contrôle de la vitesse)</p>
 <code>void Speed(unsigned int velocityWithDirection);</code>	<p>Règle le sens et la fréquence de rotation du servomoteur.</p> <p>velocityWithDirection: sens et vitesse. Vitesse maxi = 114RPM</p> <ul style="list-style-type: none"> - sens AH si $0(0x0000) \leq \text{velocityWithDirections} \leq 1023(0x3FF)$. - sens H si $1024(0x400) \leq \text{velocityWithDirections} \leq 2045(0x7FF)$

Annexe 1 : Test10Serial

Matériel

- Carte Arduino 2560
- Carte Tinkerkit Mega
- Interface Rb-Dfr-142
- Alimentation continue 9V 1A
- Servomoteur Dynamixel AX12+
- Clavier 4 touches Digilent Pmod Button Header
- ModuleUSB/RS232 Digilent PMODRS232
- Adaptateur USB/RS232

Logiciel

Algorithme Prog_princ

```
// variables
position : entier positif codé sur 16 bits initialisé à 0

début
    lire(position)
    si (position ≤ 1023) alors début
        Exécuter Graphe d'état
        Transmettre la position
    sinon rien
    fin si
    Attendre(n) // Règle la période d'échantillonnage n en millisecondes
fin
```

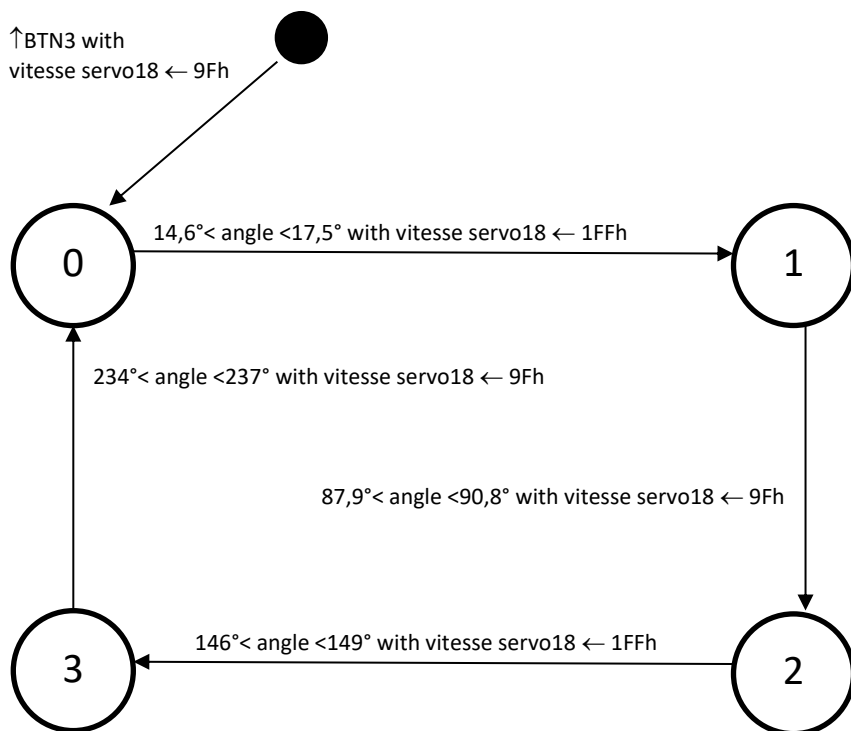
Algorithme SP_Interruption

```
// variables
EtatID18 : booléen initialisé à faux

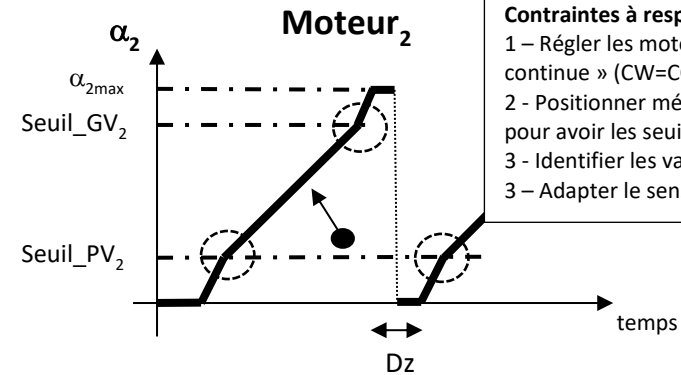
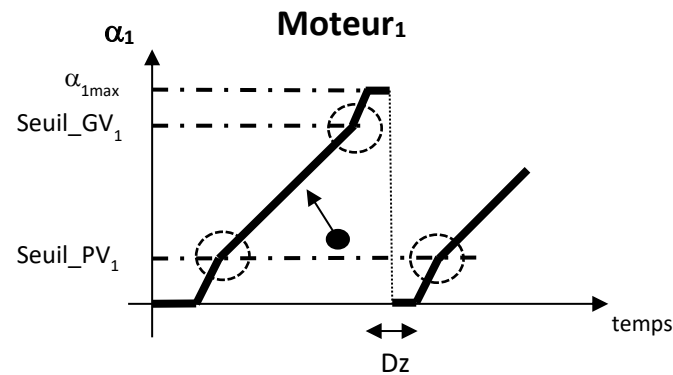
début
    si (EtatID18 = faux)
        alors début
            activer servomoteur
            EtatID18 ← vrai
        fin
    sinon début
        désactiver servomoteur
        EtatID18 ← faux
    fin
fin
fin
```

Note : le sous-programme d'interruption est exécuté sur le ↑ de BTN3 (bouton-poussoir du clavier 4 touches)

Graphe d'états



Annexe 1a : Test10SerialV2a et b (Commande de deux moteurs avec resynchronisation sur chaque changement de vitesse)



Contraintes à respecter

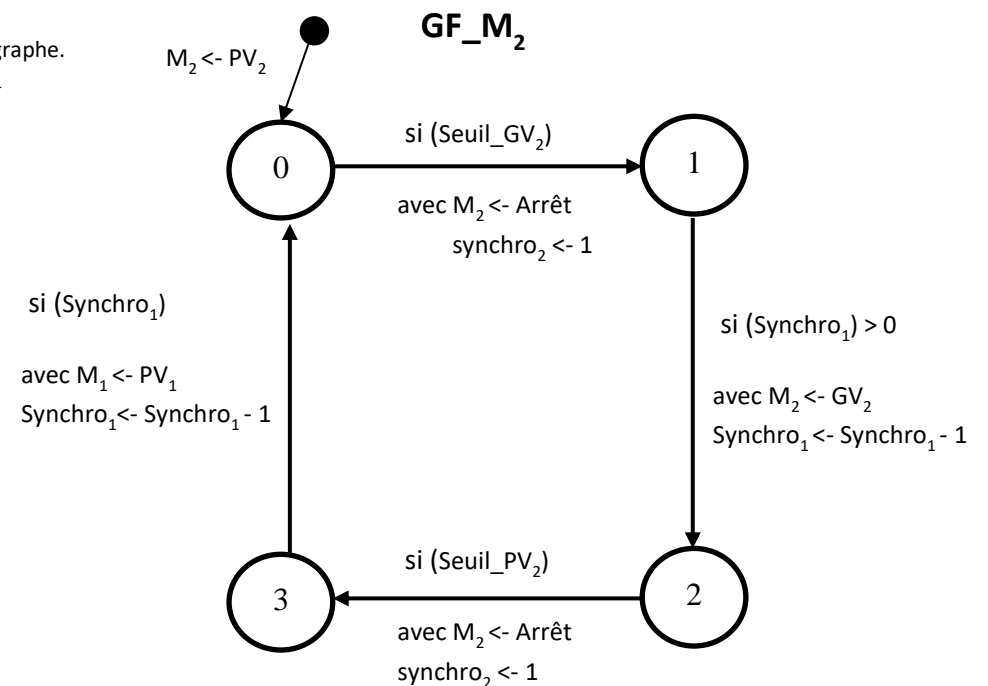
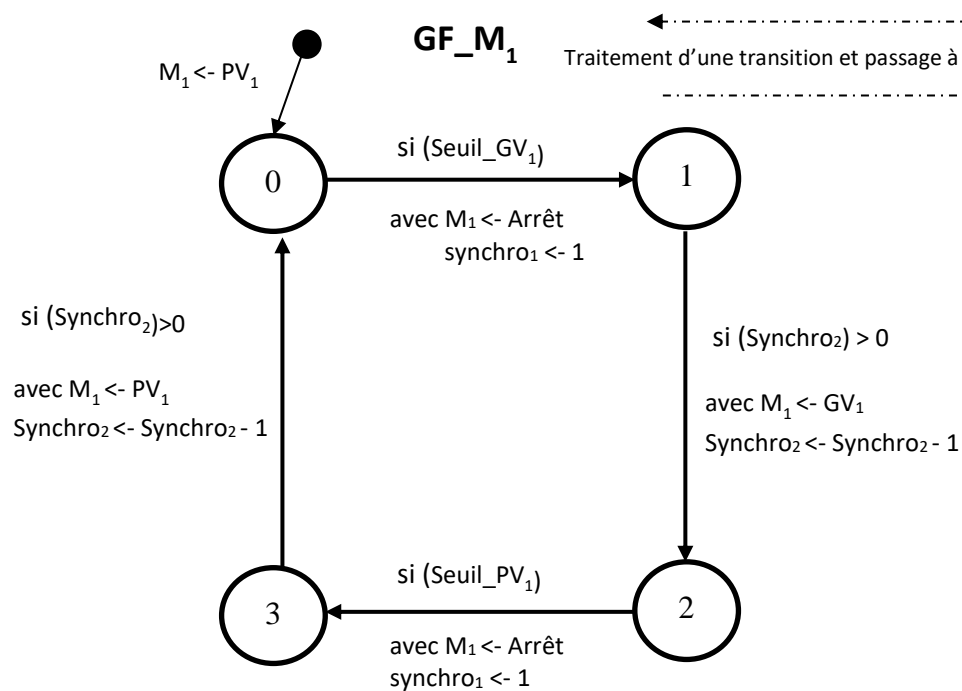
- 1 – Régler les moteurs en mode « rotation continue » (CW=CCW=0).
- 2 - Positionner mécaniquement les moteurs pour avoir les seuils hors de la DZ.
- 3 - Identifier les valeurs des seuils.
- 3 – Adapter le sens de rotation.

α_x : angle du moteur x

PV_x : petite vitesse du moteur x

GV_x : grande vitesse du moteur x

Dz : pas de lecture de la position



Annexe 1b : Test10SerialV3 (Commande de trois moteurs avec resynchronisation sur chaque changement de vitesse)

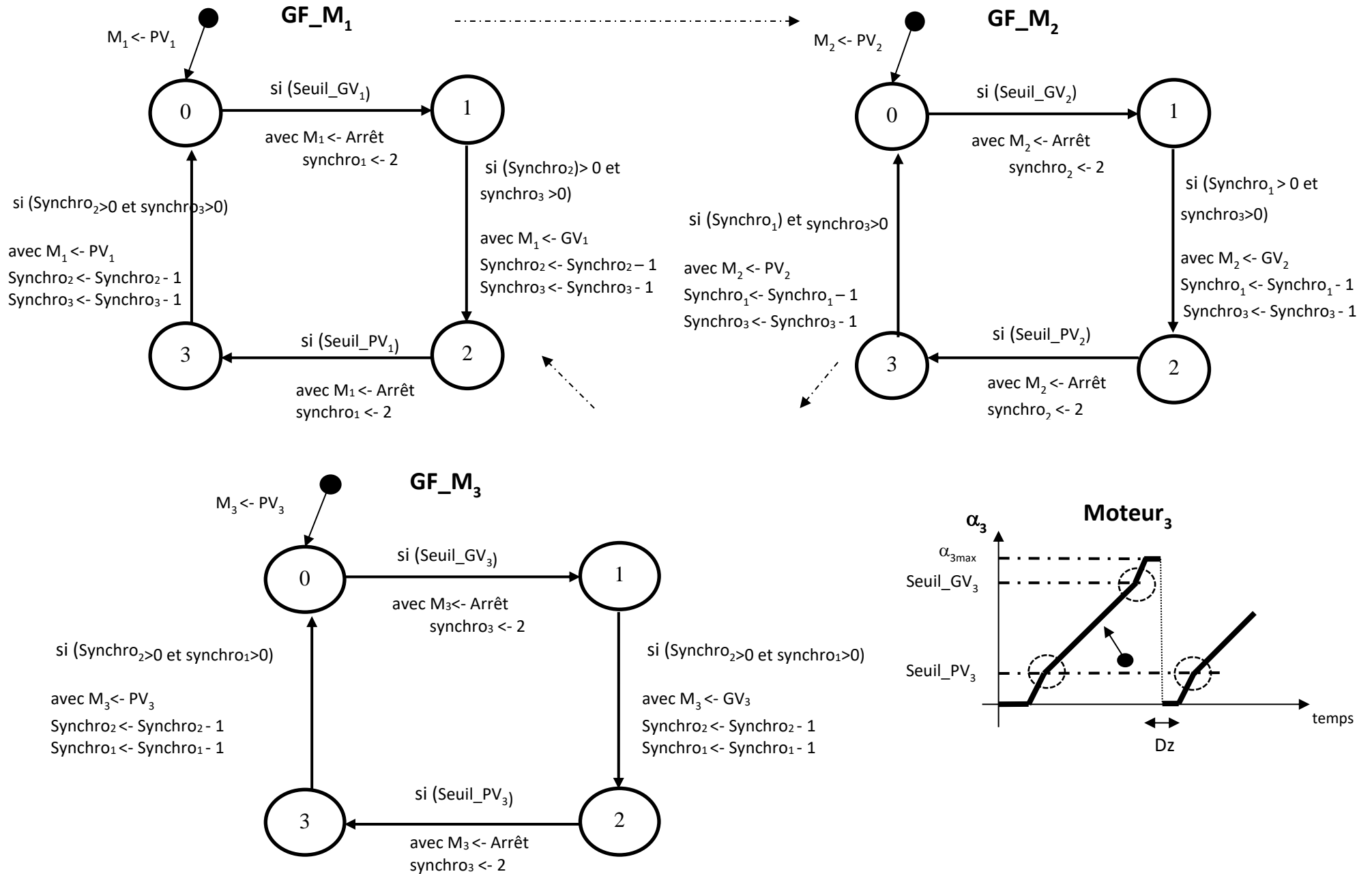


Tableau des paramètres de réglage

	M1	M2	M3	M4	M5	M6
ID						
SeuilPV						
SeuilGV						
PV						
GV						
$100* PV - GV /1023$						

Annexe 2 : Exemple d’affichage du contenu d’une table de contrôle

Lecture de l’EEPROM dans la table de contrôle

```
-----  
EEPROM  
-----  
  
Model=12  
Version=24  
ID=18  
Baud Rate=1  
Return Delay Time=2micro seconde  
CW Angle Limit=0deg  
CCW Angle Limit=0deg  
Temperature max=70deg  
Tension min=6V  
Tension max=14V  
Couple max=1020  
Status Return Level=2  
Alarm Led=36  
Alarm Shutdown=36  
-----
```

Lecture de la RAM dans la table de contrôle

```
-----  
RAM  
-----  
  
Torque Enable=0  
LED=0  
Goal Position=1020  
Moving Speed=0  
Torque Limit=1020  
Position=1020  
Speed=0  
Load=0  
Voltage=9V  
Temperature=34deg  
Registered Instruction=0  
Moving=0  
Lock=0  
Punch=32  
-----
```