

**Note d'information :**

Programmes TRIO de paramétrage et démarrage CDE « pilotage CAN Open ».

Mots clés : TRIO CDE CAN OPEN

Date : 19/03/2009

Auteur : MG

Table des matières

1 Schéma d'implantation.....	2
2 Caractéristiques	2
2.1 Commande numérique TRIO:.....	2
2.2 Variateur CDE:.....	2
3 La connexion du bus CAN.....	3
3.1 Côté carte CAN TRIO P293:.....	3
3.2 Côté variateur CDE:.....	3
4 Programme de configuration de la carte CAN TRIO.....	4
4.1 Les variables de configuration.....	4
4.2 Contrôle du nombre de cartes CAN.....	4
4.3 Paramétrage des axes.....	5
4.4 Configuration des buffers CAN de la carte TRIO.....	6
4.5 Configuration axe par axe Mapping des données à échanger.....	7
4.5.1 Configuration axe	7
4.5.2 Paramètre ATYPE.....	8
4.5.3 Mapping des PDO.....	8
Mapping PDO1 RX:.....	8
Paramétrage « Transmission type » & « InhibitTime »:.....	9
Mapping PDO1 TX:.....	9
Activation mode « cyclic » du PDO1:.....	10
Temps de cycle CAN:.....	10
5 Commande et état des variateurs par CANopen.....	11
5.1 Paramètre DRIVE ENABLE:.....	11
5.2 Mot de Commande variateur:.....	11
5.2 Mot d'Etat variateur:.....	12
5.4 Dévalidation variateur:.....	12
5.5 Acquiescement défaut variateur:.....	12
5.6 Séquence de mise sous couple variateur:.....	13

1 Schéma d'implantation



2 Caractéristiques

2.1 Commande numérique TRIO:

Pour utiliser le programme, il faut obligatoirement utiliser une **MC206X** ou une **MC224** avec une version de Firmware minimum **1.67** .

Pour toute nouvelle application, il est préférable d'utiliser la carte CAN réf P293. Les cartes P290 peuvent être utilisées mais nécessitent une configuration différente des buffers CAN. (voir avec le service technique de Transtechnik).

2.2 Variateur CDE:

Il faut utiliser la carte métier PCB 1 « Position, déplacement et cde via bus option ». Pour la configuration du CDE, veuillez utiliser la documentation « CDE – DSP 402 ».

3 La connexion du bus CAN

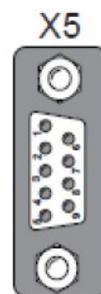
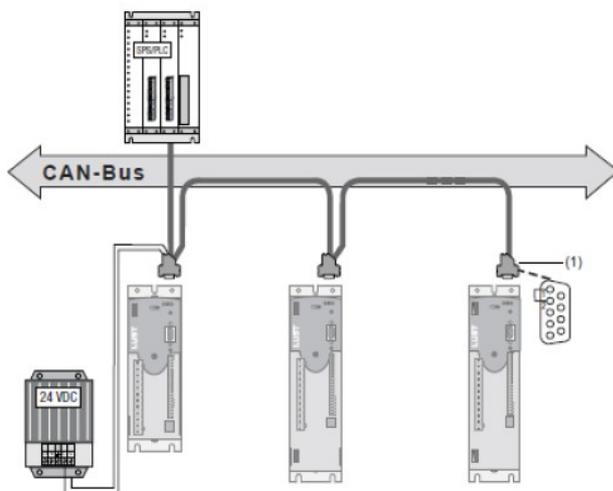
3.1 Côté carte CAN TRIO P293:

Trio Product Code P293



Pin	Use
1	no connection
2	CAN_L
3	CAN_GND
4	no connection
5	Shield
6	no connection
7	CAN_H
8	120 Ohm terminating resistor
9	no connection

3.2 Côté variateur CDE:



Pin	Fonction
1	À relier avec la pin 2 pour activer la résistance de terminaison du bus
2	CAN_LOW
3	CAN_GND
6	CAN_GND
7	CAN_HIGH
9	CAN_+24V

REMARQUE : le 24V apporté par le connecteur X5 est obligatoire. Il permet d'alimenter le module interne CANopen de manière indépendante du reste de l'électronique du variateur.

4 Programme de configuration de la carte CAN TRIO

4.1 Les variables de configuration

val_servo_period = 2000	Valeur Servo Period nécessaire
nb_var = 1	Nombre de variateurs présents sur le réseau CAN
slt_0 = 0	Emplacement carte CAN P290 ou P293
slt_1 = 1	Emplacement carte CAN P290 ou P293
slt_2 = 2	Emplacement carte CAN P290 ou P293
slt_3 = 3	Emplacement carte CAN P290 ou P293
can_speed_0 = 0	Vitesse Can Open slot 0 "0=1MHz, 1=500KHz, 2=250KHz"
can_speed_1 = 0	Vitesse Can Open slot 0 "0=1MHz, 1=500KHz, 2=250KHz"
can_speed_2 = 0	Vitesse Can Open slot 0 "0=1MHz, 1=500KHz, 2=250KHz"
can_speed_3 = 0	Vitesse Can Open slot 0 "0=1MHz, 1=500KHz, 2=250KHz"

val_servo_period	C'est la valeur de la SERVO PERIOD de la TRIO
nb_var	Nombre de variateurs reliés à piloter en CAN Open
slt_	Emplacement carte CAN Open
can_speed_	Vitesse du CAN Open de la carte « n »

4.2 Contrôle du nombre de cartes CAN

```

PRINT#pv5,"commande Emplacements CN "
FOR i=0 TO 3'Init tableau de configuration
  VR(vr_base_can+i)=0
NEXT i
IF commande=224 THEN nb_emplacement=4 ELSE nb_emplacement=1
FOR i=0 TO nb_emplacement-1
  PRINT#pv5,"SLOT",i[0],"=",COMMSTYPE SLOT(i)[0];
  IF COMMSTYPE SLOT(i)=20 OR COMMSTYPE SLOT(i)=29 THEN
    nb_carte_can=nb_carte_can+1
    IF COMMSTYPE SLOT(i)=20 THEN VR(vr_base_can+i)=290
    IF COMMSTYPE SLOT(i)=29 THEN VR(vr_base_can+i)=293
    PRINT#pv5,"  P",VR(vr_base_can+i)[0],"CARTE CAN OPEN"
  ELSE
    VR(vr_base_can+i)=0 : PRINT#pv5,"  Autre CARTE ou pas de CARTE"
  ENDIF
NEXT i

```

Ces lignes de programme permettent de vérifier le nombre de cartes connectées à la commande numérique TRIO MC224. Pour une MC206X, l'emplacement est obligatoirement le slot « 0 ».

4.3 Paramétrage des axes

'CID =adr CAN du var"n" // AX=Num axe TRIO // DR_MODE: 1=Vit 2=Inter position

```

IF commande=207 THEN
  cid1=1 : ax1=5 : dr_mode1=2'           Configuration variateur 1
  cid2=2 : ax2=6 : dr_mode2=2'           Configuration variateur 2
  cid3=3 : ax3=7 : dr_mode3=2'           Configuration variateur 3
  cid4=4 : ax4=8 : dr_mode4=2'           Configuration variateur 4
  IF slt>0 THEN PRINT#pv5,"Erreur sur paramètre SLT 206" : STOP
  IF nb_var>4 THEN PRINT#pv5,"NB VAR ERROR 206" : STOP
IF can_speed_0<0 OR can_speed_0>2 THEN PRINT#pv5,"DEFAULT VITESSE SLT 0" : STOP

ELSEIF commande=224 THEN
  cid1=1 : ax1=1 : dr_mode1=2'           Configuration variateur 1
  cid2=2 : ax2=2 : dr_mode2=2'           Configuration variateur 2
  cid3=3 : ax3=3 : dr_mode3=2'           Configuration variateur 3
  cid4=4 : ax4=4 : dr_mode4=2'           Configuration variateur 4
  cid5=5 : ax5=5 : dr_mode5=2'           Configuration variateur 5
  cid6=6 : ax6=6 : dr_mode6=2'           Configuration variateur 6
  cid7=7 : ax7=7 : dr_mode7=2'           Configuration variateur 7
  cid8=8 : ax8=8 : dr_mode8=2'           Configuration variateur 8
  cid9=9 : ax9=9 : dr_mode9=2'           Configuration variateur 9
  cid10=10 : ax10=10 : dr_mode10=2'       Configuration variateur 10
  cid11=11 : ax11=11 : dr_mode11=2'       Configuration variateur 11
  cid12=12 : ax12=12 : dr_mode12=2'       Configuration variateur 12
  cid13=13 : ax13=13 : dr_mode13=2'       Configuration variateur 13
  cid14=14 : ax14=14 : dr_mode14=2'       Configuration variateur 14
  cid15=15 : ax15=15 : dr_mode15=2'       Configuration variateur 15
  cid16=16 : ax16=16 : dr_mode16=2'       Configuration variateur 16
  IF slt<0 OR slt>3 THEN PRINT#pv5,"Erreur sur paramètre SLT 224" : STOP
  IF nb_var>nb_carte_can*4 THEN PRINT#pv5,"NB VAR ERROR 224" : STOP
IF can_speed_0<0 OR can_speed_0>2 THEN PRINT#pv5,"DEFAULT VITESSE SLT 0" : STOP
IF can_speed_1<0 OR can_speed_1>2 THEN PRINT#pv5,"DEFAULT VITESSE SLT 1" : STOP
IF can_speed_2<0 OR can_speed_2>2 THEN PRINT#pv5,"DEFAULT VITESSE SLT 2" : STOP
IF can_speed_3<0 OR can_speed_3>2 THEN PRINT#pv5,"DEFAULT VITESSE SLT 3" : STOP

ELSE
  PRINT#pv5,"Type CN incompatible avec ce programme" : STOP
ENDIF

```

Ces lignes de programme permettent de configurer les adresses CAN des variateurs, le numéro d'axe TRIO utilisé et le mode de pilotage souhaité.

Pour chaque axe:

cid	Adresse CAN du variateur
ax	Numéro d'axe TRIO
dr_mode	Mode de pilotage 1= Mode of operation « 3 »(Profile velocity mode) 2= Mode of operation « 7 » (Interpolated position mode)

4.4 Configuration des buffers CAN de la carte TRIO

CAN(slt_0,2,can_speed_0)' Paramétrage vitesse réseau CAN OPEN

PRINT# pv5,"Interrupt on Message : 1"

CAN(slt_0,1,6,2)

CAN(slt_0,1,\$A,1)

PRINT# pv5,"Adressage des Buffers"

CAN(slt_0,5,15,\$80,0,1)' Buffer 15 == Synchronisation message

CAN(slt_0,5,14,\$0,2,1)' Buffer 14 == NMT message

CAN(slt_0,5,8,\$200+cid1,6,1)' Buffer 08 == CW + Consigne axis 1

CAN(slt_0,5,9,\$200+cid2,6,1)' Buffer 09 == CW + Consigne axis 2

CAN(slt_0,5,10,\$200+cid3,6,1)' Buffer 10 == CW + Consigne axis 3

CAN(slt_0,5,11,\$200+cid3,6,1)' Buffer 11 == CW + Consigne axis 4

CAN(slt_0,5,4,\$180+cid1,6,0)' Buffer 04 == SW + Feedback axis 1

CAN(slt_0,5,5,\$180+cid2,6,0)' Buffer 05 == SW + Feedback axis 2

CAN(slt_0,5,6,\$180+cid3,6,0)' Buffer 06 == SW + Feedback axis 3

CAN(slt_0,5,7,\$180+cid4,6,0)' Buffer 07 == SW + Feedback axis 4

CAN(slt_0,5,13,\$600+cid1,8,1)' Buffer 13 == Tx SDO

CAN(slt_0,5,1,\$580+cid1,8,0)' Buffer 01 == Rx SDO

WA(200)

CAN(slt_0,7,14,1,0) : PRINT #pv5,"Active tous les axes grâce au NMT"

WA(50)

Ici on attribue les 16 buffers de la carte CAN à des fonctions.

CAN(slt_0,2,can_speed_0)	Paramétrage de la vitesse côté TRIO
CAN(slt_0,1,6,2) CAN(slt_0,1,\$A,1)	Paramétrage TRIO
CAN(slt_0,5,15,\$80,0,1)	Affectation du Buffer 15 au message de synchronisation
CAN(slt_0,5,14,\$0,2,1)	Affectation du Buffer 14 au message NMT
CAN(slt_0,5,8,\$200+cid1,6,1)	Affectation du Buffer 8 pour le PDO 1 Rx Mot de Commande(16bits) + Consigne de vitesse ou position (32bits) soit au total 48 bits (6 octets)
CAN(slt_0,5,4,\$180+cid1,6,0)	Affectation du Buffer 8 pour le PDO 1 Rx Mot d'État(16bits) + Position actuelle (32bits) « en unité du variateur » soit au total 48 bits (6 octets)
CAN(slt_0,7,14,1,0)	Activation NMT

4.5 Configuration axe par axe Mapping des données à échanger.

4.5.1 Configuration axe

```

PRINT #pv5,"-----Set up Axis-----"
vr_nbvar_can=0 : status_can = 0
IF nb_var>=1 THEN
  slt=slt_0 : axe=ax1 : id=cid1 : dr_mode=dr_mode1 : GOSUB drive_setup
  IF nb_var=1 THEN status_can=1 : GOTO m001
ENDIF
IF nb_var>=2 THEN
  slt=slt_0 : axe=ax2 : id=cid2 : dr_mode=dr_mode2 : GOSUB drive_setup
  IF nb_var=2 THEN vr_stat_can=1 : GOTO m001
ENDIF
.....

```

slt	Slot Can utilisé pour piloter l'axe
axe	Axe Trio
id	Affectation du Buffer 15 au message de synchronisation
dr_mode	Mode de pilotage

drive_setup:

```

PRINT #pv5,"-----"
PRINT #pv5,"Setup Drive // num ",id[0]
OUTLIMIT AXIS(axe)=327670
'-->commande Mode de fonctionnement CAN
IF dr_mode<1 OR dr_mode>2 THEN PRINT#pv5,"==>MODE CAN ERROR":STOP

CAN(slt,5,1,$580+id,8,0) : CAN(slt,5,13,$600+id,8,1)

```

CAN(slt,5,1,\$580+id,8,0)	Paramétrage pour l'utilisation du SDO Rx
CAN(slt,5,13,\$600+id,8,1)	Paramétrage pour l'utilisation du SDO Tx

Si on charge la valeur 1 dans la variable « dr_mode », on active le mode « profile velocity » et si on charge la valeur 2, on active le mode « Interpolated position ».

```

IF dr_mode=1 THEN'   Mode= 3=Velocity mode // 7=Interpolated position mode
  PRINT #pv5,"interpolated speed"
  IF CAN(slt,9,13,1,8,$6060,0,$0003,$0000)=0 THEN PRINT #pv5,"$6060" : STOP
ELSEIF dr_mode=2 THEN
  PRINT #pv5,"interpolated position"
  IF CAN(slt,9,13,1,8,$6060,0,$0007,$0000)=0 THEN PRINT #pv5,"$6060" : STOP
ENDIF

```

4.5.2 Paramètre ATYPE.

Chaque axe de la TRIO peut être paramétré individuellement.

```
ATYPE AXIS(axe)=28-dr_mode
PRINT#pv5,"ATYPE AXIS(",axe[0],")=",ATYPE AXIS(axe)[0]
AXIS_ADDRESS AXIS(axe)=id+(256*slt)
PRINT#pv5,"AXIS_ADDRESS AXIS(",axe[0],")=",AXIS_ADDRESS AXIS(axe)[0]
```

ATYPE=26	CANopen position axis DS402 = Mot de commande + Consigne de position
ATYPE=27	CANopen speed axis DS402 = Mot de commande + Consigne de vitesse

4.5.3 Mapping des PDO.

Ces lignes de programmes permettent d'effacer le nombre d'objets à échanger dans le PDO1:

```
$1600 pour le PDO1 Rx
$1a00 pour le PDO1 Tx
```

```
PRINT#pv5,"Clear the PDO1 variable counts"
IF CAN(slt,9,13,1,32,$1600,0,$0000,$0000)=FALSE THEN STOP
IF CAN(slt,9,13,1,32,$1a00,0,$0000,$0000)=FALSE THEN STOP
```

Mapping PDO1 RX:

Les lignes de programmes ci-dessous permettent de « mapper » le mot de commande + consigne de vitesse ou consigne de position suivant la valeur de la variable « dr_mode ».

```
PRINT#pv5,"-----RX PDO1 mapping-----"
PRINT#pv5,"Set the PDO1 Rx Mapping for commande Word($6040, $0010)"
IF CAN(slt,9,13,1,32,$1600,1,$0010,$6040)=FALSE THEN STOP
IF dr_mode=1 THEN
  PRINT#pv5,"Set the PDO1 Rx Mapping for Velocity mode ($60ff, $0020)"
  IF CAN(slt,9,13,1,32,$1600,2,$0020,$60ff)=FALSE THEN STOP
ELSEIF dr_mode=2 THEN
  PRINT#pv5,"Set the PDO1 Rx Mapping for Position mode ($60C1, $0020)"
  IF CAN(slt,9,13,1,32,$1600,2,$0120,$60C1)=FALSE THEN STOP
ENDIF
PRINT #pv5,"2 variable(s) in PDO1 Rx"
IF CAN(slt,9,13,1,32,$1600,0,$0002,$0000)=FALSE THEN STOP
```

CAN(slt,9 13,1,32,\$1600,1,\$0010,\$6040)	On place le mot de commande (objet \$6040) dans le sous-index 1 du PDO1 Rx (\$1600) \$0010 = longueur 16 bits (2 octets)
CAN(slt,9 ,13,1,32,\$1600,2,\$0020,\$60ff)	Mode « consigne de vitesse » Dans ce cas, on place la consigne de vitesse (objet \$60FF dans sous-index 2 du PDO1Rx(\$1600) \$0120= longueur 32 bits (4 octets)
CAN(slt,9,13,1,32,\$1600,2,\$0020,\$60C1)	Mode « consigne de position » Dans ce cas on place la consigne de position (objet \$60C1 sous index 1 dans sous-index 2 du PDO1 Rx (\$1600) \$0120=sous index 1 & longueur 32 bits (4 octets)
CAN(slt,9 ,13,1,32,\$1600,0,\$0002,\$0000)	Le PD01 est constitué de 2 éléments: Mot de commande + consigne de vitesse ou de position

Paramétrage « Transmission type » & « InhibitTime »:

IF CAN(slt,9 ,13,1,32,\$1400,2,\$0001,\$0000)=FALSE THEN STOP
IF CAN(slt,9 ,13,1,32,\$1400,3,\$0000,\$0000)=FALSE THEN STOP

CAN(slt,9 ,13,1,32,\$1400,2,\$0001,\$0000)	\$1400 sous index 2 = Transmission type si = 0 --> Acyclic synchronous (voir documentation CAN du variateur CDE)
CAN(slt,9 ,13,1,32,\$1400,3,\$0000,\$0000)	\$1400 sous index 3 = InhibitTime

Mapping PDO1 TX:

Les lignes de programmes ci-dessus permettent de « mapper » le mot d'état+ la position actuelle en unité du variateur.

```
PRINT#pv5,"-----TX PDO1 mapping-----"
PRINT#pv5,"Set the PDO1 index 1 Tx Mapping for Status Word ($6041, $0010)"
IF CAN(slt,9 ,13,1,32,$1a00,1,$0010,$6041)=FALSE THEN STOP
PRINT#pv5,"Set the PDO1 index 2 Tx Mapping for Actual Position ($6064, $0020)"
IF CAN(slt,9 ,13,1,32,$1a00,2,$0020,$6064)=FALSE THEN STOP
PRINT#pv5,"2 variables in PDO1 Tx"
IF CAN(slt,9 ,13,1,32,$1a00,0,$0002,$0000)=FALSE THEN STOP
PRINT#pv5,"-----"
```

CAN(slt,9 ,13,1,32,\$1a00,1,\$0010,\$6041)	On place le mot de status (objet \$6041) dans le sous-index 1 du PDO1 Tx (\$1a00) \$0010 = longueur 16 bits (2 octets)
CAN(slt,9 ,13,1,32,\$1600,2,\$0020,\$60ff)	On place la position actuelle (objet \$60FF dans sous-index 2 du PDO1 Tx (\$1a00) \$0020= longueur 32 bits (4 octets)
CAN(slt,9 ,13,1,32,\$1600,0,\$0002,\$0000)	Le PDO1 est constitué de 2 éléments: Mot de status + position actuelle

Activation mode « cyclic » du PDO1:

```
PRINT#pv5,"Set PDO1 to cyclic"
IF CAN(slt,9 ,13,1,32,$1800,2,$0001,$0000)=FALSE THEN STOP
```

Temps de cycle CAN:

Le temps de cycle CAN doit être réglé à la même valeur que la SERVO PERIOD de la TRIO.

```
IF SERVO_PERIOD=1000 THEN
  PRINT#pv5,"Set CAN update period 3e8 = 1 msec"
  IF CAN(slt,9 ,13,1,32,$1006,0,$03e8,$0000)=FALSE THEN STOP
  PRINT#pv5,"Set up interpolation time in sp_mode=1"
  IF CAN(slt,9 ,13,1,8,$60C2,1,1)=FALSE THEN STOP
  IF CAN(slt,9 ,13,1,8,$60C2,2,-3)=FALSE THEN STOP
ELSEIF SERVO_PERIOD=2000 THEN
  PRINT#pv5,"Set CAN update period 7d0 = 2 msec"
  IF CAN(slt,9 ,13,1,32,$1006,0,$07d0,$0000)=FALSE THEN STOP
  PRINT#pv5,"Set up interpolation time in sp_mode=1"
  IF CAN(slt,9 ,13,1,8,$60C2,1,2)=FALSE THEN STOP
  IF CAN(slt,9 ,13,1,8,$60C2,2,-3)=FALSE THEN STOP
ENDIF
```

Figure 1 shows a diagram of the CANbus cycle.

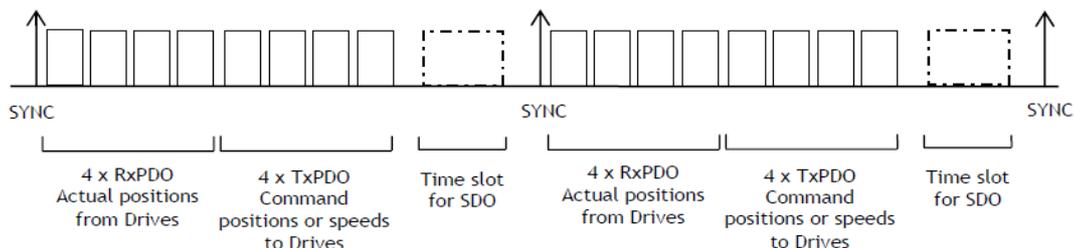


Fig. 1 CANbus cycle - 1Mbit/sec, SYNC telegram sent once per millisecond

5 Commande et état des variateurs par CANopen

Après avoir réalisé correctement la configuration de la communication CANopen, on peut réaliser le pilotage des variateurs par le bus de terrain.

5.1 Paramètre DRIVE ENABLE:

Ce paramètre permet d'activer ou d'arrêter le message de synchronisation CANopen (voir page précédente).

DRIVE_ENABLE = ON ou DRIVE_ENABLE = OFF

5.2 Mot de Commande variateur:

Description des différents bits du mot de commande

The bits of the control word are defined as follows:

15	11	10	9	8	7	6	4	3	2	1	0
manufacturer-specific	reserved	halt	Fault reset	Operation mode specific	Enable operation	Quick Stop	Enable voltage	Switch on			
0	0	0	M	0	M	M	M	M			

MSB

LSB

Optional

M - Mandatory

Table 6.1 Control word DS402

5.2 Mot d'Etat variateur:

Status word bits

Bit	Description	M / O
0	Ready to switch on	M
1	Switched on	M
2	Operation enabled	M
3	Fault	M
4	Voltage enabled	M
5	Quick stop	M
6	Switch on disabled	M
7	Warning	O
8	Manufacturer specific	O
9	Remote	M
10	Target reached	M
11	Internal limit active	M
12 - 13	Operation mode specific	O
14 - 15	Manufacturer specific	O

Table 6.4 Bits in the status word

5.4 Dévalidation variateur:

Dévalidation du variateur en mettant à 0 tous les bits du mot de commande. Mettre 16#0000 dans le mot de commande.

```
PRINT#pv5,"Disable drive"
DRIVE_CONTROL AXIS(axe)=$0 : WA(20)
```

5.5 Acquiescement défaut variateur:

L'acquiescement d'un défaut variateur se fait en activant le bit 7 du mot de commande soit 16#0080.

```
' set RAZ Error
DRIVE_CONTROL AXIS(axe)=$80 : WA(20)
```

