

SOMMAIRE

	Page
Caractéristiques techniques.....	3
Synoptique de la PC-MES3	5
Emplacements des réglages et des options disponibles	6
Préparation et installation de la PC-MES3	7
Connexion et réglages	10
Zone de Conditionnement d'entrée RX-CX	13
Entrées/Sorties logiques	15
Utilisation du timer en fréquencemètre	16
Programmation de la PC-MES3	17
Outils logiciels et routines	22
Routines disponibles	25
Routines pour le C	25
Routines pour le Pascal	29
Routines pour le Basic	31

SOMMAIRE

	Page
Caractéristiques techniques	3
Synoptique de la PC-MES3	5
Emplacements des réglages et des options disponibles	6
Préparation et installation de la PC-MES3	7
Connexion et réglages	10
Zone de Conditionnement d'entrée RX-CX.....	13
Entrées/Sorties logiques	15
Utilisation du timer en fréquencemètre	16
Programmation de la PC-MES3	17
Outils logiciels et routines.....	22
Routines disponibles	25
Routines pour le C.....	25
Routines pour le Pascal	29
Routines pour le Basic	31

CARACTERISTIQUES TECHNIQUES

CONVERSION ANALOGIQUE/DIGITALE

- 8 voies analogiques simples
- Impédance d'entrée : 1 Mohm
- 2 calibres programmables : 0/+10V et -5/+5V
- Conversion 12 bits $\pm 1\text{LSB}$
- Temps de conversion : 8 micro-secondes (120KHz)
- Protection jusqu'à 70V CC
- Interruption possible en fin de conversion
- Possibilité d'entrées en courant par composants passifs (en option)

CONVERSION DIGITALE/ANALOGIQUE

- 2 voies indépendantes
- Calibre programmable : 0/+10V
- Une 3ème sortie couplée à un D/A générant -10/+10V
- Courant max. : 10mA C.C.
- Conversion 12 bits $\pm 1\text{LSB}$
- Temps de conversion : 1,5 micro-seconde

ENTREES/SORTIES LOGIQUES

- 8 entrées TTL
- 8 sorties à collecteur ouvert tension maxi : 50V, courant maxi : 200mA

TIMER

- 1 timer programmable de type 8254 Intel, comportant 3 compteurs indépendants
- Registres décompteurs 16 bits
- Compteur 0 disponible seul, Compteur 2 chaîné avec le compteur 1 (pour comptages longs)
- Signaux : entrées, horloge et portes, tous TTL
- Horloge interne 2,4576MHz (divisible par 2^1 à 2^{15})
- Horloge externe
- Le compteur 1 peut être source d'interruptions

CARACTERISTIQUES TECHNIQUES

CONVERSION ANALOGIQUE/DIGITALE

- 8 voies analogiques simples
- Impédance d'entrée : 1 Mohm
- 2 calibres programmables : 0/+10V et -5/+5V
- Conversion 12 bits $\pm 1\text{LSB}$
- Temps de conversion : 8 micro-secondes (120KHz)
- Protection jusqu'à 70V CC
- Interruption possible en fin de conversion
- Possibilité d'entrées en courant par composants passifs (en option)

CONVERSION DIGITALE/ANALOGIQUE

- 2 voies indépendantes
- Calibre programmable : 0/+10V
- Une 3ème sortie couplée à un D/A générant -10/+10V
- Courant max. : 10mA C.C.
- Conversion 12 bits $\pm 1\text{LSB}$
- Temps de conversion : 1,5 micro-seconde

ENTREES/SORTIES LOGIQUES

- 8 entrées TTL
- 8 sorties à collecteur ouvert tension maxi : 50V, courant maxi : 200mA

TIMER

- 1 timer programmable de type 8254 Intel, comportant 3 compteurs indépendants
- Registres décompteurs 16 bits
- Compteur 0 disponible seul, Compteur 2 chaîné avec le compteur 1 (pour comptages longs)
- Signaux : entrées, horloge et portes, tous TTL
- Horloge interne 2,4576MHz (divisible par 2^1 à 2^{15})
- Horloge externe
- Le compteur 1 peut être source d'interruptions

ADRESSAGE ET BUS

- Possibilité d'installer plusieurs cartes dans l'ordinateur
- 2 groupes de 8 adresses sélectables entre 000h et 3FFh
- Cartes au format 16 bits (BUS AT) permettant d'utiliser les interruptions supplémentaires disponibles sur un AT. La **PC-MES3** fonctionne aussi sur un bus 8 bits (XT).

ALIMENTATION

- A partir du +5V du PC, une alimentation statique (DC/DC) sur la **PC-MES3**, alimente les circuits analogiques
- Consommation totale : 600mA environ sous +5V

CONNECTIQUE

- Signaux analogiques : connecteur SUB-D 15 points femelles
- Signaux logiques : connecteur SUB-D 25 points femelles

DIVERS

- Dimensions : 195 x 109 x 19 mm
- Masse : 0,170 kg

DOCUMENTATION ET LOGICIEL

La **PC-MES3** est livrée avec :

- Une documentation technique détaillée
- Une disquette contenant des routines d'utilisation en C, PASCAL, BASIC et en Assembleur pour TURBO C et TURBO PASCAL. Ainsi qu'un programme complet de contrôle "MESTEST" et un "fréquencemètre".

OPTIONS

- MESEXT/B** : Boîtier de connexion externe blindé permettant l'accès à la majorité des fonctions de la **PC-MES3** sur des douilles bananes Ø4mm.
- SYNCHRONIE** : Logiciel Universel d'acquisition et de traitements de données sous Windows.
- PHYSCOPE** : Logiciel Universel d'acquisition et de traitements de données sous DOS.

ADRESSAGE ET BUS

- Possibilité d'installer plusieurs cartes dans l'ordinateur
- 2 groupes de 8 adresses sélectables entre 000h et 3FFh
- Cartes au format 16 bits (BUS AT) permettant d'utiliser les interruptions supplémentaires disponibles sur un AT. La **PC-MES3** fonctionne aussi sur un bus 8 bits (XT).

ALIMENTATION

- A partir du +5V du PC, une alimentation statique (DC/DC) sur la **PC-MES3**, alimente les circuits analogiques
- Consommation totale : 600mA environ sous +5V

CONNECTIQUE

- Signaux analogiques : connecteur SUB-D 15 points femelles
- Signaux logiques : connecteur SUB-D 25 points femelles

DIVERS

- Dimensions : 195 x 109 x 19 mm
- Masse : 0,170 kg

DOCUMENTATION ET LOGICIEL

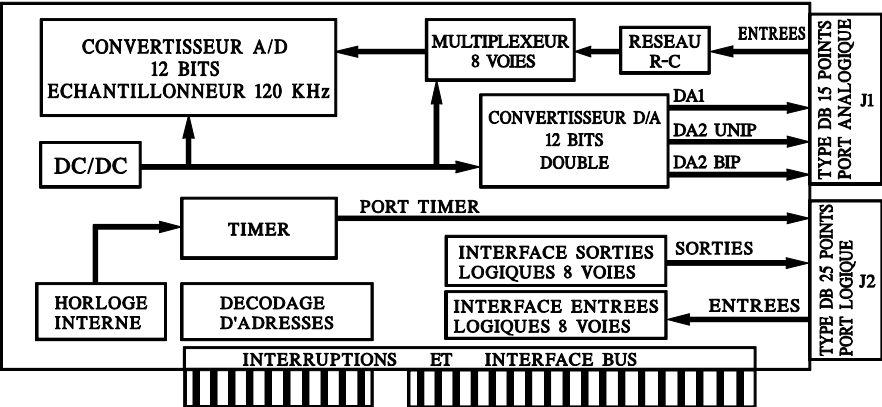
La **PC-MES3** est livrée avec :

- Une documentation technique détaillée
- Une disquette contenant des routines d'utilisation en C, PASCAL, BASIC et en Assembleur pour TURBO C et TURBO PASCAL. Ainsi qu'un programme complet de contrôle "MESTEST" et un "fréquencemètre".

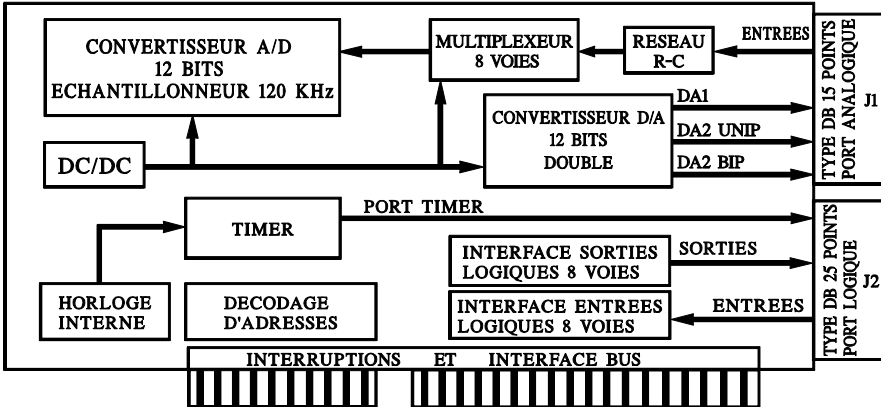
OPTIONS

- MESEXT/B** : Boîtier de connexion externe blindé permettant l'accès à la majorité des fonctions de la **PC-MES3** sur des douilles bananes Ø4mm.
- SYNCHRONIE** : Logiciel Universel d'acquisition et de traitements de données sous Windows.
- PHYSCOPE** : Logiciel Universel d'acquisition et de traitements de données sous DOS.

SYNOPTIQUE DE LA PC-MES3



SYNOPTIQUE DE LA PC-MES3



EMPLACEMENT DES REGLAGES ET DES OPTIONS DISPONIBLES SUR LA PC-MES3

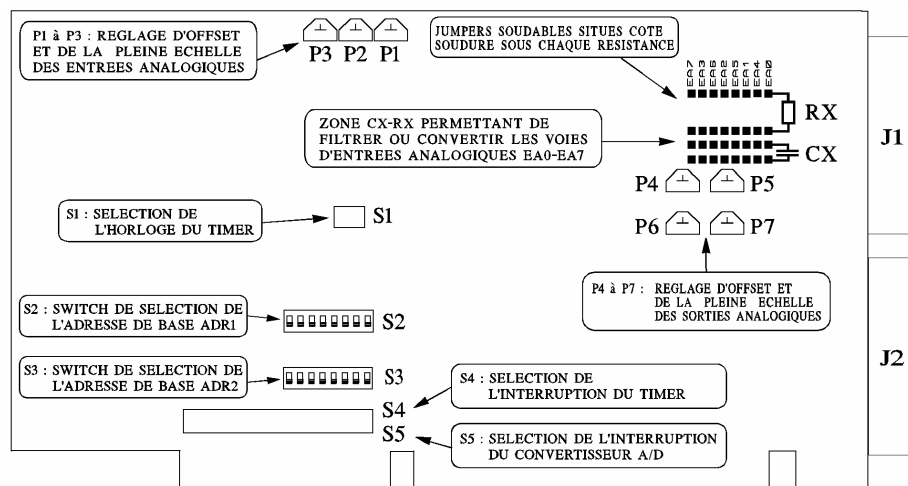


Figure 1

POTENTIOMETRES DE REGLAGE

CONVERTISSEUR A/D

P1 : réglage pleine échelle de l'A/D
P2 : réglage de l'offset bipolaire
P3 : réglage de l'offset unipolaire

CONVERTISSEURS D/A

P4 : réglage offset D/A N°1
P5 : réglage pleine échelle des D/A 1&2
P6 : réglage offset D/A N°2 unipolaire
P7 : réglage offset D/A N°2 bipolaire

EMPLACEMENT DES REGLAGES ET DES OPTIONS DISPONIBLES SUR LA PC-MES3

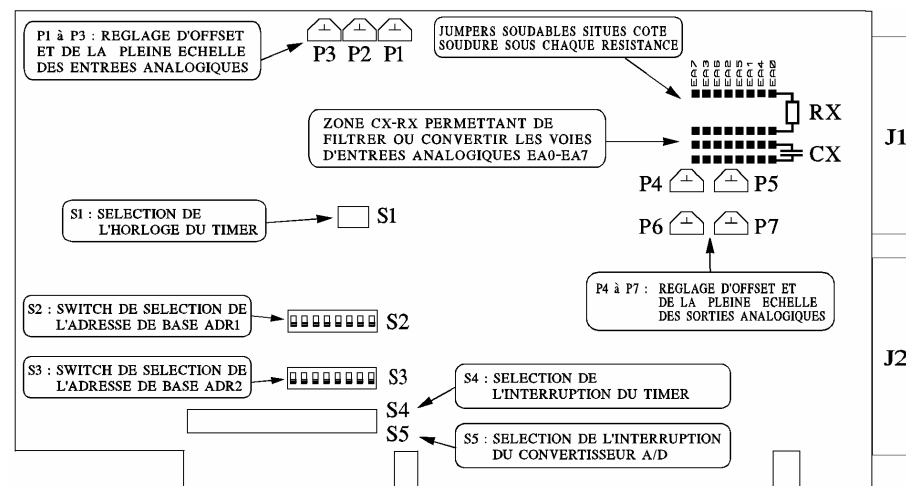


Figure 1

POTENTIOMETRES DE REGLAGE

CONVERTISSEUR A/D

P1 : réglage pleine échelle de l'A/D
P2 : réglage de l'offset bipolaire
P3 : réglage de l'offset unipolaire

CONVERTISSEURS D/A

P4 : réglage offset D/A N°1
P5 : réglage pleine échelle des D/A 1&2
P6 : réglage offset D/A N°2 unipolaire
P7 : réglage offset D/A N°2 bipolaire

PREPARATION ET INSTALLATION DE LA PC-MES3

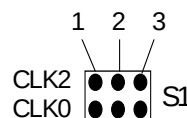
Ce chapitre explique les différentes manipulations à effectuer avant la mise en fonctionnement de la **PC-MES3**. Il est vivement conseillé de **lire toute la documentation avant manipulation**.

PREPARATION

La carte **PC-MES3** possède plusieurs cavaliers permettant de la configurer (figure 1).

S1 : Sélection de l'horloge du timer

- **CLK0** : sélection de l'horloge du compteur 0
 - **CLK2** : sélection de l'horloge du compteur 2
- Position 1-2 : horloge interne (2,4576MHz)
Position 2-3 : horloge externe
configuration usine, CLK2 interne et CLK0 externe



S2 : Sélection de l'adresse de base ADR1

S3 : Sélection de l'adresse de base ADR2

S4 : Sélection de l'interruption du timer

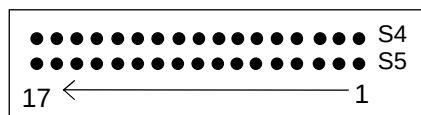
- Si un cavalier est placé, une interruption sera générée lorsque la sortie du compteur 1 (**OUT1**) passera à 1.

S5 : Sélection de l'interruption du convertisseur A/D

- Si un cavalier est placé, une interruption sera générée à chaque fin de conversion (pendant environ 8 micro-secondes).

Dans les deux cas, le type d'interruption généré est décrit ci-dessous :

Position du cavalier	Type d'interruption
1-2	IRQ4
3-4	IRQ3
4-5	IRQ5
6-7	IRQ6
7-8	IRQ7
9-10	IRQ2 sur PC-XT seul
10-11	IRQ10 sur PC-AT seul
12-13	IRQ11 sur PC-AT seul
13-14	IRQ12 sur PC-AT seul
15-16	IRQ15 sur PC-AT seul
16-17	IRQ14 sur PC-AT seul



PREPARATION ET INSTALLATION DE LA PC-MES3

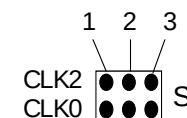
Ce chapitre explique les différentes manipulations à effectuer avant la mise en fonctionnement de la **PC-MES3**. Il est vivement conseillé de **lire toute la documentation avant manipulation**.

PREPARATION

La carte **PC-MES3** possède plusieurs cavaliers permettant de la configurer (figure 1).

S1 : Sélection de l'horloge du timer

- **CLK0** : sélection de l'horloge du compteur 0
 - **CLK2** : sélection de l'horloge du compteur 2
- Position 1-2 : horloge interne (2,4576MHz)
Position 2-3 : horloge externe
configuration usine, CLK2 interne et CLK0 externe



S2 : Sélection de l'adresse de base ADR1

S3 : Sélection de l'adresse de base ADR2

S4 : Sélection de l'interruption du timer

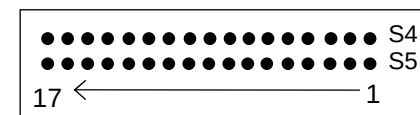
- Si un cavalier est placé, une interruption sera générée lorsque la sortie du compteur 1 (**OUT1**) passera à 1.

S5 : Sélection de l'interruption du convertisseur A/D

- Si un cavalier est placé, une interruption sera générée à chaque fin de conversion (pendant environ 8 micro-secondes).

Dans les deux cas, le type d'interruption généré est décrit ci-dessous :

Position du cavalier	Type d'interruption
1-2	IRQ4
3-4	IRQ3
4-5	IRQ5
6-7	IRQ6
7-8	IRQ7
9-10	IRQ2 sur PC-XT seul
10-11	IRQ10 sur PC-AT seul
12-13	IRQ11 sur PC-AT seul
13-14	IRQ12 sur PC-AT seul
15-16	IRQ15 sur PC-AT seul
16-17	IRQ14 sur PC-AT seul



ADRESSES DE BASE DE LA PC-MES3

La carte **PC-MES3** occupe deux groupes d'adresses modifiables par l'utilisateur, grâce à des micro-interrupteurs (dip-switchs). Le premier groupe (débutant à l'adresse **ADR1**) occupe 8 adresses et correspond aux convertisseurs A/D et D/A ainsi qu'au timer. Le deuxième groupe (débutant à l'adresse **ADR2**) occupe 2 adresses et correspond aux entrées sorties logiques, multiplexage des entrées et polarité de la carte.

La carte PC-MES3 est livrée configurée aux adresses :

- \$330 (ADR1) et \$340 (ADR2).

Ces adresses sont généralement libres et il n'y a donc probablement pas à les modifier.

Dans le cas contraire, les micro-interrupteurs S2 et S3 permettront la modification de ces adresses.

MODIFICATION D'ADRESSES

La validation d'un bit se fait en positionnant le micro-interrupteur correspondant à OFF.

ADR1 est du type \$ABC

Exemple de configuration de **S2** :

1	2	3	4	5	6	7	8
N.U.	1	1	0	0	1	1	0
non utilisé	A		B				C

Dans cet exemple ADR1 a la valeur hexadécimale 330.

ADR2 est du type \$XYZ

Exemple de configuration de **S3** :

1	2	3	4	5	6	7	8
1	1	0	1	0	0	0	0
X		Y				Z	

Dans cet exemple ADR2 a la valeur hexadécimale 340.

ADRESSES DE BASE DE LA PC-MES3

La carte **PC-MES3** occupe deux groupes d'adresses modifiables par l'utilisateur, grâce à des micro-interrupteurs (dip-switchs). Le premier groupe (débutant à l'adresse **ADR1**) occupe 8 adresses et correspond aux convertisseurs A/D et D/A ainsi qu'au timer. Le deuxième groupe (débutant à l'adresse **ADR2**) occupe 2 adresses et correspond aux entrées sorties logiques, multiplexage des entrées et polarité de la carte.

La carte PC-MES3 est livrée configurée aux adresses :

- \$330 (ADR1) et \$340 (ADR2).

Ces adresses sont généralement libres et il n'y a donc probablement pas à les modifier.

Dans le cas contraire, les micro-interrupteurs S2 et S3 permettront la modification de ces adresses.

MODIFICATION D'ADRESSES

La validation d'un bit se fait en positionnant le micro-interrupteur correspondant à OFF.

ADR1 est du type \$ABC

Exemple de configuration de **S2** :

1	2	3	4	5	6	7	8
N.U.	1	1	0	0	1	1	0
non utilisé	A		B				C

Dans cet exemple ADR1 a la valeur hexadécimale 330.

ADR2 est du type \$XYZ

Exemple de configuration de **S3** :

1	2	3	4	5	6	7	8
1	1	0	1	0	0	0	0
X		Y				Z	

Dans cet exemple ADR2 a la valeur hexadécimale 340.

POSITION DES MICRO-INTERRUPTEURS				VALEUR HEXADECIMALE
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A

IMPORTANT : Lire les micro-interrupteurs S2 et S3 sur la **PC-MES3**, de droite à gauche.
ON : 0
OFF : 1

ATTENTION : La **PC-MES3** ne doit pas avoir d'adresse commune avec une autre carte. Vous trouverez sur votre manuel IBM les adresses couramment occupées.

INSTALLATION

Après avoir configuré la **PC-MES3** selon votre application, vous pouvez l'installer dans l'un des slots disponibles de votre ordinateur.

MANIPULATION

Pour éviter d'endommager le matériel :

AVANT D'INSERER OU D'ENLEVER UNE CARTE, VOTRE ORDINATEUR DOIT ETRE HORS TENSION

- 1/ Ouvrez le capot de votre ordinateur.
- 2/ Retirez le cache correspondant à l'emplacement choisi.
- 3/ Insérez la carte dans la fente en question
(le slot peut être de 8 ou 16 bits).
- 4/ Fixez la carte à l'aide de la vis de sécurité récupérée en 2/.
- 5/ Se reporter aux pages suivantes pour la connectique externe.

POSITION DES MICRO-INTERRUPTEURS				VALEUR HEXADECIMALE
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A

IMPORTANT : Lire les micro-interrupteurs S2 et S3 sur la **PC-MES3**, de droite à gauche.
ON : 0
OFF : 1

ATTENTION : La **PC-MES3** ne doit pas avoir d'adresse commune avec une autre carte. Vous trouverez sur votre manuel IBM les adresses couramment occupées.

INSTALLATION

Après avoir configuré la **PC-MES3** selon votre application, vous pouvez l'installer dans l'un des slots disponibles de votre ordinateur.

MANIPULATION

Pour éviter d'endommager le matériel :

AVANT D'INSERER OU D'ENLEVER UNE CARTE, VOTRE ORDINATEUR DOIT ETRE HORS TENSION

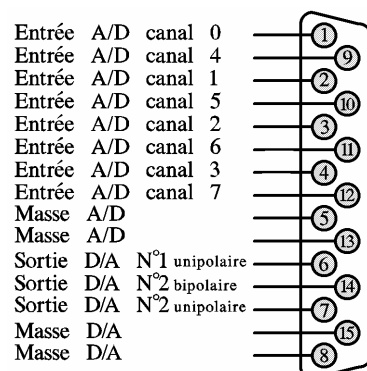
- 1/ Ouvrez le capot de votre ordinateur.
- 2/ Retirez le cache correspondant à l'emplacement choisi.
- 3/ Insérez la carte dans la fente en question
(le slot peut être de 8 ou 16 bits).
- 4/ Fixez la carte à l'aide de la vis de sécurité récupérée en 2/.
- 5/ Se reporter aux pages suivantes pour la connectique externe.

CONNEXION ET REGLAGES

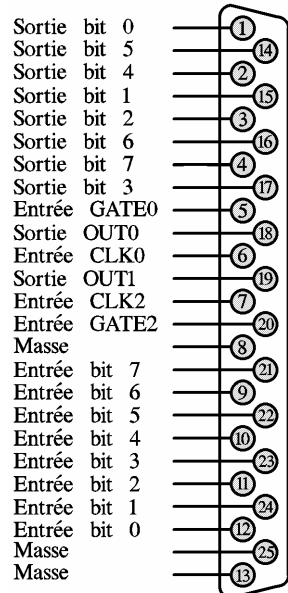
La partie analogique de la **PC-MES3** est accessible sur un connecteur SUB-D 15 points femelles. La partie logique est sur un connecteur SUB-D 25 points femelles. (les numéros indiqués sont imprimés sur les connecteurs).

CONNEXION D'ENTREES J1 ET J2

La figure ci-dessous représente le brochage des connecteurs d'entrées vus côté externe de votre ordinateur :



CONNECTEUR J1
DB15 FEMELLE
VUE FACE ARRIERE DE L'ORDINATEUR



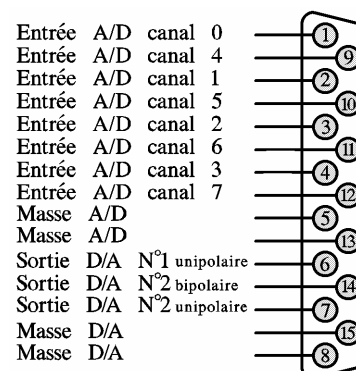
CONNECTEUR J2
DB25 FEMELLE
VUE FACE ARRIERE DE L'ORDINATEUR

CONNEXION ET REGLAGES

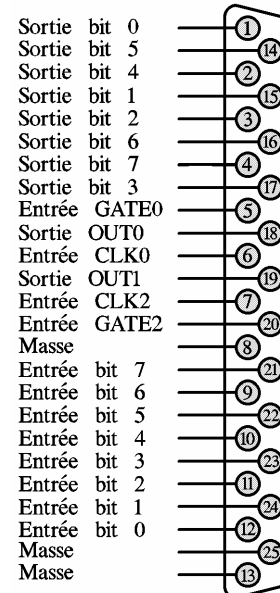
La partie analogique de la **PC-MES3** est accessible sur un connecteur SUB-D 15 points femelles. La partie logique est sur un connecteur SUB-D 25 points femelles. (les numéros indiqués sont imprimés sur les connecteurs).

CONNEXION D'ENTREES J1 ET J2

La figure ci-dessous représente le brochage des connecteurs d'entrées vus côté externe de votre ordinateur :



CONNECTEUR J1
DB15 FEMELLE
VUE FACE ARRIERE DE L'ORDINATEUR



CONNECTEUR J2
DB25 FEMELLE
VUE FACE ARRIERE DE L'ORDINATEUR

ATTENTION : Veuillez à bien vérifier vos liaisons avant tout branchement. En milieu industriel ou dans un environnement parasité, il est préférable d'utiliser des câbles blindés.

INFORMATION : Un boîtier de connexion externe blindé est proposé en option. Réf : MESEXT/B. Il donne accès à la majorité des fonctions de la **PC-MES3** sur des douilles bananes Ø4mm. Il permet ainsi une utilisation immédiate et aisée.

REGLAGES GENERAUX

La **PC-MES3** est étalonnée en nos services et ne demande normalement aucun autre réétalonnage. Cependant, si l'utilisateur désire réajuster les réglages, la procédure suivante peut être effectuée. En cas de doute, ne pas toucher à ces réglages.

ENTREES ANALOGIQUES

Les 8 entrées du convertisseur A/D possèdent 2 calibres programmables, un pour une mesure unipolaire (0 à +10V) et l'autre pour une mesure bipolaire (-5V à +5V). Pour des raisons d'arrondi, la pleine échelle n'est pas réglée à 10V mais à 10,24V ce qui donne une résolution de **2,5mV** par point de mesure. La correspondance entre la tension mesurée et la valeur lue est :

TENSION D'ENTREE		VALEUR DECIMALE
mode unipolaire	mode bipolaire	
0V	-5,12V	0
+5,12V	0V	2048
+10,24V	+5,12V	4095

La résolution de la **PC-MES3** est calculée comme suit :

$$\text{Résolution LSB} = \frac{\text{Pleine échelle}}{2^n} = \frac{10,24V}{2^{12}} = \frac{10,24V}{4096} = 2,5mV \quad (n = 12bits)$$

V : Volts mV : millivolts

Réglage des offsets

- 1/ Connecter l'entrée analogique 0 sur la masse analogique.
- 2/ Configurer les entrées analogiques en mode bipolaire.
- 3/ Agir sur le potentiomètre P2 de façon à obtenir une tension voisine de 0V ± 2.5mV (valeur décimale oscillant entre 2047 et 2048).
- 4/ Configurer les entrées analogiques en mode unipolaire.
- 5/ Agir sur le potentiomètre P3 de façon à obtenir une tension voisine de 0V ± 2.5mV (valeur décimale oscillant entre 0 et 1).

ATTENTION : Veuillez à bien vérifier vos liaisons avant tout branchement. En milieu industriel ou dans un environnement parasité, il est préférable d'utiliser des câbles blindés.

INFORMATION : Un boîtier de connexion externe blindé est proposé en option. Réf : MESEXT/B. Il donne accès à la majorité des fonctions de la **PC-MES3** sur des douilles bananes Ø4mm. Il permet ainsi une utilisation immédiate et aisée.

REGLAGES GENERAUX

La **PC-MES3** est étalonnée en nos services et ne demande normalement aucun autre réétalonnage. Cependant, si l'utilisateur désire réajuster les réglages, la procédure suivante peut être effectuée. En cas de doute, ne pas toucher à ces réglages.

ENTREES ANALOGIQUES

Les 8 entrées du convertisseur A/D possèdent 2 calibres programmables, un pour une mesure unipolaire (0 à +10V) et l'autre pour une mesure bipolaire (-5V à +5V). Pour des raisons d'arrondi, la pleine échelle n'est pas réglée à 10V mais à 10,24V ce qui donne une résolution de **2,5mV** par point de mesure. La correspondance entre la tension mesurée et la valeur lue est :

TENSION D'ENTREE		VALEUR DECIMALE
mode unipolaire	mode bipolaire	
0V	-5,12V	0
+5,12V	0V	2048
+10,24V	+5,12V	4095

La résolution de la **PC-MES3** est calculée comme suit :

$$\text{Résolution LSB} = \frac{\text{Pleine échelle}}{2^n} = \frac{10,24V}{2^{12}} = \frac{10,24V}{4096} = 2,5mV \quad (n = 12bits)$$

V : Volts mV : millivolts

Réglage des offsets

- 1/ Connecter l'entrée analogique 0 sur la masse analogique.
- 2/ Configurer les entrées analogiques en mode bipolaire.
- 3/ Agir sur le potentiomètre P2 de façon à obtenir une tension voisine de 0V ± 2.5mV (valeur décimale oscillant entre 2047 et 2048).
- 4/ Configurer les entrées analogiques en mode unipolaire.
- 5/ Agir sur le potentiomètre P3 de façon à obtenir une tension voisine de 0V ± 2.5mV (valeur décimale oscillant entre 0 et 1).

Réglage de la pleine échelle

- 1/ Configurer les entrées analogiques en mode unipolaire.
- 2/ Connecter une tension continue de 10V sur l'entrée analogique 0.
- 3/ Agir sur P1 de façon à obtenir une tension voisine de $10V \pm 2.5mV$ (valeur décimale oscillant entre 3999 et 4000).

Recommencer les opérations depuis le début afin d'affiner les réglages.

SORTIES ANALOGIQUES

Deux convertisseurs D/A sont disponibles sur la carte **PC-MES3**, DA1 et DA2 programmés respectivement par les fonctions **SETDA1** et **SETDA2**. Ces convertisseurs génèrent une tension comprise entre 0 et +10V. Cependant, le convertisseur DA2 génère en plus une tension bipolaire comprise entre -10V et +10V. La correspondance entre les valeurs programmées et la tension générée est :

VALEUR DECIMALE PROGRAMMEE	TENSION DE SORTIE	
	Mode unipolaire	Mode bipolaire
0	0V	+10V
2048	+5V	0V
4095	+10V	-10V

V : Volts

Réglage des offsets

- 1/ Connecter un voltmètre entre la sortie analogique 1 et la masse.
- 2/ Programmer une tension de 0V sur la sortie 1.
- 3/ Agir sur le potentiomètre P4 de façon à obtenir sur celle-ci une tension de sortie voisine de 0V.
- 4/ Connecter le voltmètre entre la sortie analogique 2 unipolaire et la masse.
- 5/ Programmer une tension de 0V sur la sortie analogique 2 unipolaire.
- 6/ Agir sur le potentiomètre P6 de façon à obtenir sur celle-ci une tension de sortie voisine de 0V.
- 7/ Connecter le voltmètre entre la sortie analogique 2 bipolaire et la masse.
- 8/ Programmer une tension de 0V sur la sortie analogique 2 bipolaire.
- 9/ Agir sur le potentiomètre P7 de façon à obtenir sur celle-ci une tension de sortie voisine de 0V.

Réglage de la pleine échelle des DA1 et DA2

- 1/ Connecter le voltmètre entre la sortie analogique 1 et la masse.
- 2/ Programmer une tension de 10V sur la sortie analogique 1.
- 3/ Agir sur le potentiomètre P5 de façon à obtenir sur celle-ci une tension de sortie voisine de 10V.

Recommencer les opérations depuis le début afin d'affiner les réglages.

Réglage de la pleine échelle

- 1/ Configurer les entrées analogiques en mode unipolaire.
- 2/ Connecter une tension continue de 10V sur l'entrée analogique 0.
- 3/ Agir sur P1 de façon à obtenir une tension voisine de $10V \pm 2.5mV$ (valeur décimale oscillant entre 3999 et 4000).

Recommencer les opérations depuis le début afin d'affiner les réglages.

SORTIES ANALOGIQUES

Deux convertisseurs D/A sont disponibles sur la carte **PC-MES3**, DA1 et DA2 programmés respectivement par les fonctions **SETDA1** et **SETDA2**. Ces convertisseurs génèrent une tension comprise entre 0 et +10V. Cependant, le convertisseur DA2 génère en plus une tension bipolaire comprise entre -10V et +10V. La correspondance entre les valeurs programmées et la tension générée est :

VALEUR DECIMALE PROGRAMMEE	TENSION DE SORTIE	
	Mode unipolaire	Mode bipolaire
0	0V	+10V
2048	+5V	0V
4095	+10V	-10V

V : Volts

Réglage des offsets

- 1/ Connecter un voltmètre entre la sortie analogique 1 et la masse.
- 2/ Programmer une tension de 0V sur la sortie 1.
- 3/ Agir sur le potentiomètre P4 de façon à obtenir sur celle-ci une tension de sortie voisine de 0V.
- 4/ Connecter le voltmètre entre la sortie analogique 2 unipolaire et la masse.
- 5/ Programmer une tension de 0V sur la sortie analogique 2 unipolaire.
- 6/ Agir sur le potentiomètre P6 de façon à obtenir sur celle-ci une tension de sortie voisine de 0V.
- 7/ Connecter le voltmètre entre la sortie analogique 2 bipolaire et la masse.
- 8/ Programmer une tension de 0V sur la sortie analogique 2 bipolaire.
- 9/ Agir sur le potentiomètre P7 de façon à obtenir sur celle-ci une tension de sortie voisine de 0V.

Réglage de la pleine échelle des DA1 et DA2

- 1/ Connecter le voltmètre entre la sortie analogique 1 et la masse.
- 2/ Programmer une tension de 10V sur la sortie analogique 1.
- 3/ Agir sur le potentiomètre P5 de façon à obtenir sur celle-ci une tension de sortie voisine de 10V.

Recommencer les opérations depuis le début afin d'affiner les réglages.

ZONE DE CONDITIONNEMENT D'ENTREE RX-CX

Cette zone permet :

- 1/ De positionner un filtre passe bas sur une des voies d'entrée analogique (figure 2). L'utilisateur fixe les différentes valeurs R et C en fonction de sa fréquence de coupure et les positionne dans l'alignement respectif à la voie désirée RX et CX. $F_c = 0,159/R.C$
- 2/ De positionner un pont diviseur en entrée pour utiliser un signal d'entrée supérieur à 10V (figure 3). L'utilisateur positionnera la résistance R1 à l'emplacement RX et R2 en CX sur la voie désirée. La tension d'entrée correspondra alors à $V_{entrée} = V_{source} \times R2/(R1+R2)$.
- 3/ De convertir un courant provenant d'un capteur 0 ou 4-20 mA en tension (figure 4). L'utilisateur disposera alors une résistance de 500 Ohms à l'emplacement CX de la voie désirée. Une résistance de 1% couche métallique est le plus souvent utilisée.

ATTENTION : Les opérations 1/ et 2/ nécessitent de dessouder le jumper situé côté soudure sous chaque résistance RX utilisée.

Figure 2

ZONE DE CONDITIONNEMENT D'ENTREE RX-CX

Cette zone permet :

- 1/ De positionner un filtre passe bas sur une des voies d'entrée analogique (figure 2). L'utilisateur fixe les différentes valeurs R et C en fonction de sa fréquence de coupure et les positionne dans l'alignement respectif à la voie désirée RX et CX. $F_c = 0,159/R.C$
- 2/ De positionner un pont diviseur en entrée pour utiliser un signal d'entrée supérieur à 10V (figure 3). L'utilisateur positionnera la résistance R1 à l'emplacement RX et R2 en CX sur la voie désirée. La tension d'entrée correspondra alors à $V_{entrée} = V_{source} \times R2/(R1+R2)$.
- 3/ De convertir un courant provenant d'un capteur 0 ou 4-20 mA en tension (figure 4). L'utilisateur disposera alors une résistance de 500 Ohms à l'emplacement CX de la voie désirée. Une résistance de 1% couche métallique est le plus souvent utilisée.

ATTENTION : Les opérations 1/ et 2/ nécessitent de dessouder le jumper situé côté soudure sous chaque résistance RX utilisée.

Figure 2

Figure 3

Figure 4

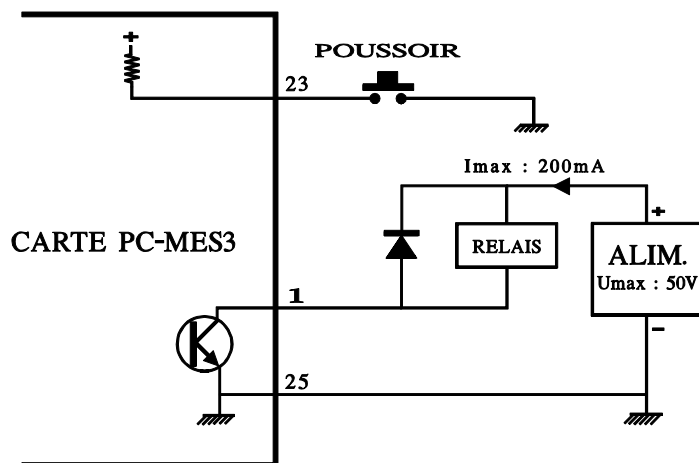
Figure 3

Figure 4

ENTREES/SORTIES LOGIQUES

Comme les entrées logiques possèdent une résistance de rappel au +, pour les rendre actives, il suffit de les relier à la masse (logique inversée). Le schéma ci-dessous indique un exemple d'utilisation permettant de lire l'état d'un bouton poussoir sur le bit d'entrée 3.

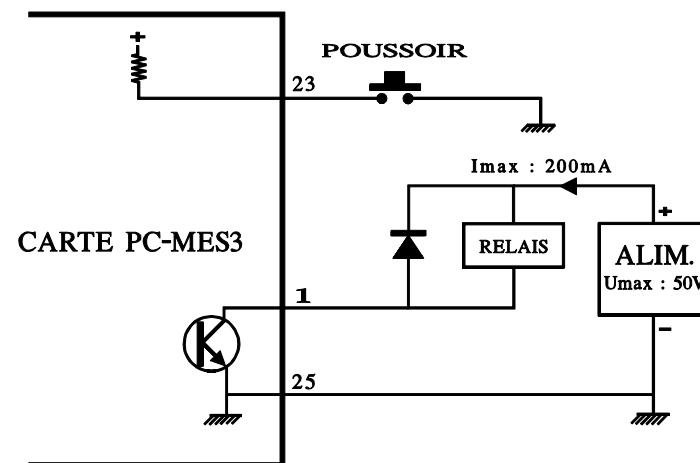
Les sorties logiques étant à collecteur ouvert, elles se comportent, lorsqu'elles sont actives, comme un interrupteur fermé à la masse. Le schéma ci-dessous donne un exemple d'utilisation de la sortie logique 0 pour commander un relais. La diode est nécessaire pour supprimer la tension inverse aux bornes de la bobine du relais (risque de destruction du transistor interne).



ENTREES/SORTIES LOGIQUES

Comme les entrées logiques possèdent une résistance de rappel au +, pour les rendre actives, il suffit de les relier à la masse (logique inversée). Le schéma ci-dessous indique un exemple d'utilisation permettant de lire l'état d'un bouton poussoir sur le bit d'entrée 3.

Les sorties logiques étant à collecteur ouvert, elles se comportent, lorsqu'elles sont actives, comme un interrupteur fermé à la masse. Le schéma ci-dessous donne un exemple d'utilisation de la sortie logique 0 pour commander un relais. La diode est nécessaire pour supprimer la tension inverse aux bornes de la bobine du relais (risque de destruction du transistor interne).



UTILISATION DU TIMER EN FREQUENCEMETRE

L'ensemble des compteurs peut être utilisé en tant que fréquencemètre. Pour cela, effectuer les liaisons suivantes :

- Relier **OUT1** à **GATE0**.
Les compteurs 1 et 2 constituent alors la base de temps du fréquencemètre qui autorisera le comptage du compteur 0 lorsque **OUT1** sera à 1.
- Relier aussi **OUT1** à l'entrée logique **BIT0**.
Cela permet au programme de savoir si le compteur 0 est en cours de comptage (niveau 1) ou arrêté (0).
- Relier l'horloge interne à **CLK2** (cavalier **S1**).

Envoyer votre signal à mesurer sur **CLK0**.

Les compteurs 1 et 2 sont programmés en mode 3 (oscillateur symétrique) afin de générer un signal dont la fréquence est par exemple de 0,5Hz (comptage pendant 1 seconde).

Dès que **OUT1** est à 0 (testée avec l'entrée logique **BIT0**), le compteur 0 est programmé en mode 0 avec une valeur de 65535.
Lorsque **OUT1** passera à 1, le compteur 0 commencera à décompter à la fréquence du signal à mesurer et ce, jusqu'à ce que **OUT1** repasse à 0. Le programme attend cela puis va lire le contenu du compteur 0.
En faisant la différence de cette valeur avec 65535, on obtient le nombre d'impulsions du signal à mesurer en 1 seconde (si la base de temps est à 1 seconde) d'où sa fréquence.

ATTENTION : Ne jamais envoyer un signal dépassant les normes TTL (0/+5V) sur l'entrée **CLK0**.

UTILISATION DU TIMER EN FREQUENCEMETRE

L'ensemble des compteurs peut être utilisé en tant que fréquencemètre. Pour cela, effectuer les liaisons suivantes :

- Relier **OUT1** à **GATE0**.
Les compteurs 1 et 2 constituent alors la base de temps du fréquencemètre qui autorisera le comptage du compteur 0 lorsque **OUT1** sera à 1.
- Relier aussi **OUT1** à l'entrée logique **BIT0**.
Cela permet au programme de savoir si le compteur 0 est en cours de comptage (niveau 1) ou arrêté (0).
- Relier l'horloge interne à **CLK2** (cavalier **S1**).

Envoyer votre signal à mesurer sur **CLK0**.

Les compteurs 1 et 2 sont programmés en mode 3 (oscillateur symétrique) afin de générer un signal dont la fréquence est par exemple de 0,5Hz (comptage pendant 1 seconde).

Dès que **OUT1** est à 0 (testée avec l'entrée logique **BIT0**), le compteur 0 est programmé en mode 0 avec une valeur de 65535.
Lorsque **OUT1** passera à 1, le compteur 0 commencera à décompter à la fréquence du signal à mesurer et ce, jusqu'à ce que **OUT1** repasse à 0. Le programme attend cela puis va lire le contenu du compteur 0.
En faisant la différence de cette valeur avec 65535, on obtient le nombre d'impulsions du signal à mesurer en 1 seconde (si la base de temps est à 1 seconde) d'où sa fréquence.

ATTENTION : Ne jamais envoyer un signal dépassant les normes TTL (0/+5V) sur l'entrée **CLK0**.

PROGRAMMATION DE LA PC-MES3

Ce chapitre n'est utile qu'aux personnes qui n'utilisent pas les routines fournies avec la carte **PC-MES3** et qui souhaitent développer leurs propres interfaces avec cette carte. Les personnes désirant utiliser le timer programmable doivent également lire ce qui suit. Les routines fournies avec la **PC-MES3** (Assembleur, C, Pascal et Basic) conviennent dans la plupart des cas et vous simplifient donc le travail. Ces routines sont décrites plus loin et sont données sur la disquette accompagnant ce manuel.

Les adresses des ports indiquées ci-après sont relatives aux adresses de base **ADR1** et **ADR2**. Si vous ne modifiez pas ces adresses, vous pouvez utiliser celles qui sont indiquées plus loin entre parenthèses.

Toutes les adresses sont spécifiées en hexadécimal.

Type indique un registre à lecture seule (R), écriture (W) ou lecture/écriture (R/W).

Le nom imprimé en gras correspond au nom donné au registre décrit.

DESCRIPTION DES REGISTRES

Timer programmable :

Le timer est un 8254 de chez Intel. Pour une documentation plus complète que celle donnée ci-dessous, consulter le databook Intel

Nom : COUNT0	Adresse : ADR1 (330)	Type : R/W
Nom : COUNT1	Adresse : ADR1+1 (331)	Type : R/W
Nom : COUNT2	Adresse : ADR1+2 (332)	Type : R/W

7	6	5	4	3	2	1	0	n° du bit
CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8	
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0	

Le timer comporte 3 compteurs indépendants mais identiques 16 bits, numérotés de 0 à 2. Chaque compteur peut être utilisé en binaire ou BCD. Ces trois compteurs sont **COUNT0**, **COUNT1**, **COUNT2**. Telle qu'est câblée la **PC-MES3**, les compteurs 2 et 1 sont cascades afin d'obtenir éventuellement des durées plus importantes. Le compteur 0 est seul (figure 5). Les 16 bits de ces compteurs sont accessibles en 2 fois 8 bits.

PROGRAMMATION DE LA PC-MES3

Ce chapitre n'est utile qu'aux personnes qui n'utilisent pas les routines fournies avec la carte **PC-MES3** et qui souhaitent développer leurs propres interfaces avec cette carte. Les personnes désirant utiliser le timer programmable doivent également lire ce qui suit. Les routines fournies avec la **PC-MES3** (Assembleur, C, Pascal et Basic) conviennent dans la plupart des cas et vous simplifient donc le travail. Ces routines sont décrites plus loin et sont données sur la disquette accompagnant ce manuel.

Les adresses des ports indiquées ci-après sont relatives aux adresses de base **ADR1** et **ADR2**. Si vous ne modifiez pas ces adresses, vous pouvez utiliser celles qui sont indiquées plus loin entre parenthèses.

Toutes les adresses sont spécifiées en hexadécimal.

Type indique un registre à lecture seule (R), écriture (W) ou lecture/écriture (R/W).

Le nom imprimé en gras correspond au nom donné au registre décrit.

DESCRIPTION DES REGISTRES

Timer programmable :

Le timer est un 8254 de chez Intel. Pour une documentation plus complète que celle donnée ci-dessous, consulter le databook Intel

Nom : COUNT0	Adresse : ADR1 (330)	Type : R/W
Nom : COUNT1	Adresse : ADR1+1 (331)	Type : R/W
Nom : COUNT2	Adresse : ADR1+2 (332)	Type : R/W

7	6	5	4	3	2	1	0	n° du bit
CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8	
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0	

Le timer comporte 3 compteurs indépendants mais identiques 16 bits, numérotés de 0 à 2. Chaque compteur peut être utilisé en binaire ou BCD. Ces trois compteurs sont **COUNT0**, **COUNT1**, **COUNT2**. Telle qu'est câblée la **PC-MES3**, les compteurs 2 et 1 sont cascades afin d'obtenir éventuellement des durées plus importantes. Le compteur 0 est seul (figure 5). Les 16 bits de ces compteurs sont accessibles en 2 fois 8 bits.

La figure 5 décrit cette partie. Seules les connexions indiquées sont sorties sur le connecteur de la **PC-MES3**. Les entrées d'horloge (**CLKO** et **CLK2**) peuvent être reliées à une horloge externe ou interne à la carte (2,4576 MHz) suivant la position des cavaliers **S1**.

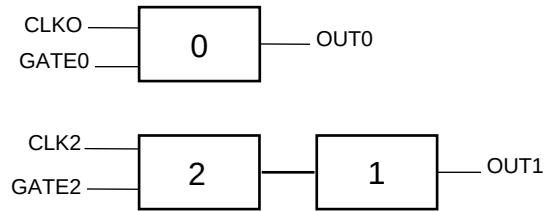


Figure 5

L'ensemble des compteurs 1 et 2 peut générer une interruption si un cavalier est placé sur **S4** (ce qui relie **OUT1** à la ligne d'interruption sélectionnée). Si tel est le cas, une interruption sera générée lorsque **OUT1** sera au niveau 1.

Nom : TIMER		Adresse : ADR1+3 (333)						Type : R/W
7	6	5	4	3	2	1	0	n° du bit
SC1	SC0	RL1	RL0	M2	M1	M0	BCD	

Registre de contrôle du timer. Ce registre permet de programmer chaque compteur en sélectionnant son numéro dans les bits SC1-SC0

SC1	SC0	Compteur sélectionné
0	0	Compteur 0
0	1	Compteur 1
1	0	Compteur 2

RL1	RL0	Mode de lecture/écriture dans les compteurs
0	0	Echantillonne la valeur en cours de comptage
0	1	Lit/Ecrit l'octet de poids faible seul
1	0	Lit/Ecrit l'octet de poids fort seul
1	1	Lit/Ecrit l'octet de poids faible puis fort

Remarque : Si RL0=RL1=0, le contenu du compteur spécifié est échantillonné pour ensuite pouvoir être lu (afin de ne pas lire une valeur en cours de transition).

M2	M1	M0	Mode de fonctionnement du compteur
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

La figure 5 décrit cette partie. Seules les connexions indiquées sont sorties sur le connecteur de la **PC-MES3**. Les entrées d'horloge (**CLKO** et **CLK2**) peuvent être reliées à une horloge externe ou interne à la carte (2,4576 MHz) suivant la position des cavaliers **S1**.

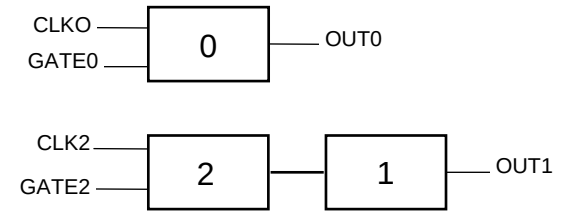


Figure 5

L'ensemble des compteurs 1 et 2 peut générer une interruption si un cavalier est placé sur **S4** (ce qui relie **OUT1** à la ligne d'interruption sélectionnée). Si tel est le cas, une interruption sera générée lorsque **OUT1** sera au niveau 1.

Nom : TIMER		Adresse : ADR1+3 (333)						Type : R/W
7	6	5	4	3	2	1	0	n° du bit
SC1	SC0	RL1	RL0	M2	M1	M0	BCD	

Registre de contrôle du timer. Ce registre permet de programmer chaque compteur en sélectionnant son numéro dans les bits SC1-SC0

SC1	SC0	Compteur sélectionné
0	0	Compteur 0
0	1	Compteur 1
1	0	Compteur 2

RL1	RL0	Mode de lecture/écriture dans les compteurs
0	0	Echantillonne la valeur en cours de comptage
0	1	Lit/Ecrit l'octet de poids faible seul
1	0	Lit/Ecrit l'octet de poids fort seul
1	1	Lit/Ecrit l'octet de poids faible puis fort

Remarque : Si RL0=RL1=0, le contenu du compteur spécifié est échantillonné pour ensuite pouvoir être lu (afin de ne pas lire une valeur en cours de transition).

M2	M1	M0	Mode de fonctionnement du compteur
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

MODE0 : La sortie passe à 0 dès la programmation de ce mode. Après l'écriture de la valeur dans le compteur, la sortie reste à 0 jusqu'à ce que le compteur arrive à 0.

MODE1 : Monostable. La sortie passe à 0 dès que la GATE passe à 1. Elle repasse à 1 à la fin du comptage (compteur arrivé à 0).

MODE2 : Oscillateur. Génère un signal dont la période est programmée dans le compteur. Le signal n'est pas symétrique, il est à 0 pendant 1 période de l'horloge CLK.

MODE3 : Oscillateur. Identique au mode 2 mais génère des signaux symétriques de la période spécifiée.

MODE4 : La sortie passe à 1 après la sélection de ce mode. Elle reste à 1 durant la période programmée puis passe à 0 pendant seulement 1 période de CLK.

MODE5 : La sortie passe à 1 dès la programmation de ce mode. Le compteur démarre dès que la GATE passe de 0 à 1. La sortie passe alors à 0 après le temps programmé et ce pour une durée égale à la période de CLK.

Bit 0 du timer : (BCD)

0 : Binaire

1 : Binaire codé décimal

Nom : ADHIGH	Adresse : ADR1+4 (334)				Type : R		
7 6 5 4 3 2 1 0							N° du bit
BUSY				AD11	AD10	AD9	AD8

Contient les 4 bits de poids fort du convertisseur A/D ainsi qu'un bit indiquant si une conversion est en cours (BUSY = 1) ou pas.

Nom : ADLOW	Adresse : ADR1+5 (335)				Type : R		
7 6 5 4 3 2 1 0							N° du bit
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Contient les 8 bits de poids faible du convertisseur A/D (résultat d'une conversion).

Nom : DA1LOW	Adresse : ADR1+4 (334)				Type : W		
7 6 5 4 3 2 1 0							N° du bit
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

Registre où l'on doit écrire les 8 bits de poids faible du convertisseur D/A numéro 1.

MODE0 : La sortie passe à 0 dès la programmation de ce mode. Après l'écriture de la valeur dans le compteur, la sortie reste à 0 jusqu'à ce que le compteur arrive à 0.

MODE1 : Monostable. La sortie passe à 0 dès que la GATE passe à 1. Elle repasse à 1 à la fin du comptage (compteur arrivé à 0).

MODE2 : Oscillateur. Génère un signal dont la période est programmée dans le compteur. Le signal n'est pas symétrique, il est à 0 pendant 1 période de l'horloge CLK.

MODE3 : Oscillateur. Identique au mode 2 mais génère des signaux symétriques de la période spécifiée.

MODE4 : La sortie passe à 1 après la sélection de ce mode. Elle reste à 1 durant la période programmée puis passe à 0 pendant seulement 1 période de CLK.

MODE5 : La sortie passe à 1 dès la programmation de ce mode. Le compteur démarre dès que la GATE passe de 0 à 1. La sortie passe alors à 0 après le temps programmé et ce pour une durée égale à la période de CLK.

Bit 0 du timer : (BCD)

0 : Binaire

1 : Binaire codé décimal

Nom : ADHIGH	Adresse : ADR1+4 (334)				Type : R		
7 6 5 4 3 2 1 0							N° du bit
BUSY				AD11	AD10	AD9	AD8

Contient les 4 bits de poids fort du convertisseur A/D ainsi qu'un bit indiquant si une conversion est en cours (BUSY = 1) ou pas.

Nom : ADLOW	Adresse : ADR1+5 (335)				Type : R		
7 6 5 4 3 2 1 0							N° du bit
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Contient les 8 bits de poids faible du convertisseur A/D (résultat d'une conversion).

Nom : DA1LOW	Adresse : ADR1+4 (334)				Type : W		
7 6 5 4 3 2 1 0							N° du bit
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

Registre où l'on doit écrire les 8 bits de poids faible du convertisseur D/A numéro 1.

Nom : **DA1HIGH** Adresse : ADR1+5 (335) Type : W
 7 6 5 4 3 2 1 0 N° du bit
 DA11 DA10 DA9 DA8

Registre où l'on doit écrire les 4 bits de poids fort du convertisseur D/A numéro 1.

Nom : **DA2LOW** Adresse : ADR1+6 (336) Type : W
 Nom : **DA2HIGH** Adresse : ADR1+7 (337) Type : W

Ces deux registres sont indentiques aux registres DA1LOW et DA1HIGH, mais pour le convertisseur D/A numéro 2.

Remarque : Les valeurs écrites dans les 4 registres des convertisseurs D/A ne sont effectivement validées que lorsque l'on va lire le registre **UPDDA** (voir plus loin).

Nom : **STARTAD** Adresse : ADR1+6 (336) Type : R

Déclenche une mesure sur le convertisseur A/D dès que l'on a effectué une lecture de ce registre. On peut ensuite savoir si la conversion est terminée en testant le bit 7 de **ADHIGH**.

Nom : **UPDDA** Adresse : ADR1+7 (337) Type : R

Une lecture dans ce registre valide les valeurs programmées dans les 4 registres de données des D/A.

Nom : **IOOUT** Adresse : ADR2 (340) Type : W
 7 6 5 4 3 2 1 0 N° du bit
 OUT7 OUT6 OUT5 OUT4 OUT3 OUT2 OUT1 OUT0

La valeur écrite dans ce registre est envoyée sur le port de sortie. Un bit à 1 met la sortie à 0 (collecteur ouvert).

Nom : **DA1HIGH** Adresse : ADR1+5 (335) Type : W
 7 6 5 4 3 2 1 0 N° du bit
 DA11 DA10 DA9 DA8

Registre où l'on doit écrire les 4 bits de poids fort du convertisseur D/A numéro 1.

Nom : **DA2LOW** Adresse : ADR1+6 (336) Type : W
 Nom : **DA2HIGH** Adresse : ADR1+7 (337) Type : W

Ces deux registres sont indentiques aux registres DA1LOW et DA1HIGH, mais pour le convertisseur D/A numéro 2.

Remarque : Les valeurs écrites dans les 4 registres des convertisseurs D/A ne sont effectivement validées que lorsque l'on va lire le registre **UPDDA** (voir plus loin).

Nom : **STARTAD** Adresse : ADR1+6 (336) Type : R

Déclenche une mesure sur le convertisseur A/D dès que l'on a effectué une lecture de ce registre. On peut ensuite savoir si la conversion est terminée en testant le bit 7 de **ADHIGH**.

Nom : **UPDDA** Adresse : ADR1+7 (337) Type : R

Une lecture dans ce registre valide les valeurs programmées dans les 4 registres de données des D/A.

Nom : **IOOUT** Adresse : ADR2 (340) Type : W
 7 6 5 4 3 2 1 0 N° du bit
 OUT7 OUT6 OUT5 OUT4 OUT3 OUT2 OUT1 OUT0

La valeur écrite dans ce registre est envoyée sur le port de sortie. Un bit à 1 met la sortie à 0 (collecteur ouvert).

Nom : IOIN		Adresse : ADR2		(340)		Type : R		
7	6	5	4	3	2	1	0	N° du bit
IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0	

La valeur lue est celle de l'état du port d'entrée. Un bit à 1 signifie que l'entrée correspondante est en l'air, à 0 qu'elle est reliée à la masse.

Nom : MULPLEX		Adresse : ADR2+1		(341)		Type : W		
7	6	5	4	3	2	1	0	N° du bit
			MUL2	MUL1	MUL0	UNIP		

Ecriture du mode du calibre du convertisseur A/D et du convertisseur DA2 :

- UNIP = 1 : mesure unipolaire (0/+10V)
- UNIP = 0 : mesure bipolaire (-5/+5V) et calibre $\pm 10V$ disponible sur DA2 bipolaire
- MULx : bits sélectionnant la voie à mesurer (0 à 7)

Nom : IOIN		Adresse : ADR2		(340)		Type : R		
7	6	5	4	3	2	1	0	N° du bit
IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0	

La valeur lue est celle de l'état du port d'entrée. Un bit à 1 signifie que l'entrée correspondante est en l'air, à 0 qu'elle est reliée à la masse.

Nom : MULPLEX		Adresse : ADR2+1		(341)		Type : W		
7	6	5	4	3	2	1	0	N° du bit
			MUL2	MUL1	MUL0	UNIP		

Ecriture du mode du calibre du convertisseur A/D et du convertisseur DA2 :

- UNIP = 1 : mesure unipolaire (0/+10V)
- UNIP = 0 : mesure bipolaire (-5/+5V) et calibre $\pm 10V$ disponible sur DA2 bipolaire
- MULx : bits sélectionnant la voie à mesurer (0 à 7)

OUTILS LOGICIELS ET ROUTINES

PROGRAMMES FOURNIS (disquette routines pour **PC-MES3**)

La carte **PC-MES3** se programme très facilement. Pour ce faire, vous pouvez utiliser les routines décrites dans les chapitres suivants et données sur la disquette "Routines pour la **PC-MES3**".

MESTEST.EXE : Programme de test et de gestion de la carte (voir descriptif plus loin)
MESTEST.C : Source de MESTEST.EXE en C
MESTEST.SCR : Masque d'écran pour MESTEST
MESTEST.PRJ : Fichier projet de MESTEST (pour TURBO C) permettant de linker les routines
FREQ.EXE : Programme utilisant la **PC-MES3** en mode fréquencemètre
FREQ.PAS : Source de FREQ.EXE en PASCAL
C_AD.ASM : Source Assembleur des routines pour TURBOC
C_AD.OBJ : Résultat de l'assemblage de C_AD.ASM
AD.H : Définition des fonctions C et Assembleur. Définition des adresses de la carte
AD.C : Sources des routines C de la **PC-MES3**
IRQ.C : Sources des routines de gestion des interruptions matérielles en langage C
P_AD.ASM : Source Assembleur des routines pour TURBO PASCAL
P_AD.OBJ : Résultat de l'assemblage de P_AD.ASM
AD1.PAS : Source de l'unité PASCAL d'utilisation de la **PC-MES3** utilisant les routines en Assembleur
AD2.PAS : Source des routines d'utilisation en PASCAL standard
AD1.TPU et **AD2.TPU** : Unités AD1.PAS et AD2.PAS compilées
IRQ.PAS : Identique à IRQ.C mais pour le PASCAL
AD.BAS : Routines pour le Basic

MESTEST

MESTEST est un logiciel de test général et de gestion de la carte **PC-MES3**. Sa fonction essentielle est de contrôler l'ensemble des fonctionnalités de la carte **PC-MES3** et de faire les réglages qui s'avèrent nécessaires.

OUTILS LOGICIELS ET ROUTINES

PROGRAMMES FOURNIS (disquette routines pour **PC-MES3**)

La carte **PC-MES3** se programme très facilement. Pour ce faire, vous pouvez utiliser les routines décrites dans les chapitres suivants et données sur la disquette "Routines pour la **PC-MES3**".

MESTEST.EXE : Programme de test et de gestion de la carte (voir descriptif plus loin)
MESTEST.C : Source de MESTEST.EXE en C
MESTEST.SCR : Masque d'écran pour MESTEST
MESTEST.PRJ : Fichier projet de MESTEST (pour TURBO C) permettant de linker les routines
FREQ.EXE : Programme utilisant la **PC-MES3** en mode fréquencemètre
FREQ.PAS : Source de FREQ.EXE en PASCAL
C_AD.ASM : Source Assembleur des routines pour TURBOC
C_AD.OBJ : Résultat de l'assemblage de C_AD.ASM
AD.H : Définition des fonctions C et Assembleur. Définition des adresses de la carte
AD.C : Sources des routines C de la **PC-MES3**
IRQ.C : Sources des routines de gestion des interruptions matérielles en langage C
P_AD.ASM : Source Assembleur des routines pour TURBO PASCAL
P_AD.OBJ : Résultat de l'assemblage de P_AD.ASM
AD1.PAS : Source de l'unité PASCAL d'utilisation de la **PC-MES3** utilisant les routines en Assembleur
AD2.PAS : Source des routines d'utilisation en PASCAL standard
AD1.TPU et **AD2.TPU** : Unités AD1.PAS et AD2.PAS compilées
IRQ.PAS : Identique à IRQ.C mais pour le PASCAL
AD.BAS : Routines pour le Basic

MESTEST

MESTEST est un logiciel de test général et de gestion de la carte **PC-MES3**. Sa fonction essentielle est de contrôler l'ensemble des fonctionnalités de la carte **PC-MES3** et de faire les réglages qui s'avèrent nécessaires.

Il permet donc de :

- Mesurer une tension sur les 8 entrées analogiques.
- Programmer une tension sur chacune des sorties analogiques.
- Lire l'état des entrées logiques.
- Programmer les sorties logiques.
- Programmer la fréquence d'un signal carré généré par le timer (compteur 0).
- Vérifier l'état du timer. Ce test est réalisé par la génération d'une IRQ en fin de comptage.

Pour lancer ce programme de test, taper **MESTEST** et suivre les instructions données à l'écran.

Pour modifier chaque paramètre, il suffit de taper la lettre clé correspondante. Voir tableau ci-dessous.

ENTREES ANALOGIQUES				CONVERTISSEUR D/A				ADRESSE			
Calibre : 0V/10.24V				Sortie 1 (V) : 0				(Hexadécimale)			
Num. Volt				Sortie 2 (V) : 0				Adr1 : \$ 330			
Entrée 0 : +0 +0.000				Sortie 2B (V) : N.U.				Adr2 : \$ 340			
Entrée 1 : +0 +0.000				P : Changement de polarité A/D et D/A							
Entrée 2 : +0 +0.000				S : Programmation des sorties D/A							
Entrée 3 : +0 +0.000				L : Gestion des sorties logiques							
Entrée 4 : +0 +0.000				I : Génération d'IRQ par Timer							
Entrée 5 : +0 +0.000				T : Test du compteur 0 du timer							
Entrée 6 : +0 +0.000				Q : Quitter MESTEST							
Entrée 7 : +0 +0.000											
TIMER - Test du compteur 0								IRQ TIMER			
Valeur du compteur (0 à 65535) : 0								Test compteurs 1 et 2 sur déclenchement IRQ (Sélect°= IRQ5) Temps entre 2 interrupt° --> 1 seconde			
ENTREES/SORTIES LOGIQUES								© EUROS MESTEST			
Entrées : Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7											
0 0 0 0 0 0 0 0											
Sorties : Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7											
0 0 0 0 0 0 0 0											

Il permet donc de :

- Mesurer une tension sur les 8 entrées analogiques.
- Programmer une tension sur chacune des sorties analogiques.
- Lire l'état des entrées logiques.
- Programmer les sorties logiques.
- Programmer la fréquence d'un signal carré généré par le timer (compteur 0).
- Vérifier l'état du timer. Ce test est réalisé par la génération d'une IRQ en fin de comptage.

Pour lancer ce programme de test, taper **MESTEST** et suivre les instructions données à l'écran.

Pour modifier chaque paramètre, il suffit de taper la lettre clé correspondante. Voir tableau ci-dessous.

ENTREES ANALOGIQUES				CONVERTISSEUR D/A				ADRESSE			
Calibre : 0V/10.24V				Sortie 1 (V) : 0				(Hexadécimale)			
Num. Volt				Sortie 2 (V) : 0				Adr1 : \$ 330			
Entrée 0 : +0 +0.000				Sortie 2B (V) : N.U.				Adr2 : \$ 340			
Entrée 1 : +0 +0.000				P : Changement de polarité A/D et D/A S : Programmation des sorties D/A L : Gestion des sorties logiques I : Génération d'IRQ par Timer T : Test du compteur 0 du timer Q : Quitter MESTEST							
Entrée 2 : +0 +0.000											
Entrée 3 : +0 +0.000											
Entrée 4 : +0 +0.000											
Entrée 5 : +0 +0.000											
Entrée 6 : +0 +0.000											
Entrée 7 : +0 +0.000											
TIMER - Test du compteur 0								IRQ TIMER			
Valeur du compteur (0 à 65535) : 0								Test compteurs 1 et 2			
ENTREES/SORTIES LOGIQUES								sur déclenchement			
Entrées : Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7								IRQ (Sélect° = IRQ5°)			
0 0 0 0 0 0 0 0								Temps entre 2 interrupt°			
Sorties : Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7								--> 1 seconde			
0 0 0 0 0 0 0 0								© EUROSMA			
								MESTEST			

FREQUENCEMETRE

Ce programme permet à la carte **PC-MES3** de mesurer des fréquences comprises entre le Hertz et environ 60 kHz. Pour cela, il est nécessaire d'effectuer le branchement indiqué dans le chapitre "Utilisation du timer en Fréquence-mètre". Lancer le programme en tapant **FREQ**. L'écran de votre ordinateur affiche alors la fréquence du signal envoyée sur **CLK0**.

ATTENTION : Le signal envoyé sur **CLK0** doit être compatible TTL, c'est à dire être un signal carré d'amplitude 0-5V. Le non respect de ces valeurs détruira le timer 8254.

Remarque : Pour que ce programme puisse fonctionner, la **PC-MES3** doit être adressée aux adresses standard, c'est-à-dire \$330 et \$340. Si les adresses ont été changées, vous devez alors recompiler ce programme en tenant compte des nouvelles adresses.

FREQUENCEMETRE

Ce programme permet à la carte **PC-MES3** de mesurer des fréquences comprises entre le Hertz et environ 60 kHz. Pour cela, il est nécessaire d'effectuer le branchement indiqué dans le chapitre "Utilisation du timer en Fréquence-mètre". Lancer le programme en tapant **FREQ**. L'écran de votre ordinateur affiche alors la fréquence du signal envoyée sur **CLK0**.

ATTENTION : Le signal envoyé sur **CLK0** doit être compatible TTL, c'est à dire être un signal carré d'amplitude 0-5V. Le non respect de ces valeurs détruira le timer 8254.

Remarque : Pour que ce programme puisse fonctionner, la **PC-MES3** doit être adressée aux adresses standard, c'est-à-dire \$330 et \$340. Si les adresses ont été changées, vous devez alors recompiler ce programme en tenant compte des nouvelles adresses.

ROUTINES DISPONIBLES

Plusieurs routines d'utilisation de la carte **PC-MES3** ont été développées dans trois langages différents : C, PASCAL et BASIC. Pour C et PASCAL, ces routines ont été aussi écrites en Assembleur pour une plus grande rapidité. Vous trouverez plus loin un chapitre décrivant ces routines pour chaque langage. Celles-ci étant très ressemblantes d'un langage à l'autre, tout ce qui est dit dans le chapitre concernant le C est aussi valable pour BASIC et PASCAL. Il est donc conseillé de lire ce qui suit.

ROUTINES POUR LE C

Deux types de routines ont été écrits, un type en Assembleur (pour le TURBO C seul) et l'autre entièrement écrit en C standard. Dans tous les cas vous devez inclure à votre programme le fichier de définitions des fonctions AD.H. Les définitions sont les suivantes :

CONSTANTES

ADR1 et **ADR2** sont les deux adresses qui définissent les adresses de base de la **PC-MES3**. Ce sont ces adresses qu'il faut modifier si vous changez les cavaliers **S2** ou **S3**. Les valeurs par défaut sont **\$330** et **\$340**.

IOIN : adresse du port d'entrées logiques
IOOUT : adresse du port de sorties logiques
MULPLEX : adresse du registre de sélection du multiplexeur et de la polarité de mesure
TIMER : registre de contrôle du timer (8254)
COUNT0 : compteur 0 du timer
COUNT1 : compteur 1 du timer
COUNT2 : compteur 2 du timer
ADHIGH : registre des 4 bits de poids fort du convertisseur A/D
ADLOW : registre des 8 bits de poids faible du convertisseur A/D
STARTAD : déclenchement d'une mesure sur l'A/D
UPDDA : validation des valeurs précédemment écrites dans les convertisseurs D/A
DA1LOW : registre des 8 bits de poids faible du convertisseur DA1
DA1HIGH : registre des 4 bits de poids fort du convertisseur DA1
DA2LOW et **DA2HIGH** : identiques à **DA1LOW** et **DA1HIGH** pour le convertisseur DA2

ROUTINES DISPONIBLES

Plusieurs routines d'utilisation de la carte **PC-MES3** ont été développées dans trois langages différents : C, PASCAL et BASIC. Pour C et PASCAL, ces routines ont été aussi écrites en Assembleur pour une plus grande rapidité. Vous trouverez plus loin un chapitre décrivant ces routines pour chaque langage. Celles-ci étant très ressemblantes d'un langage à l'autre, tout ce qui est dit dans le chapitre concernant le C est aussi valable pour BASIC et PASCAL. Il est donc conseillé de lire ce qui suit.

ROUTINES POUR LE C

Deux types de routines ont été écrits, un type en Assembleur (pour le TURBO C seul) et l'autre entièrement écrit en C standard. Dans tous les cas vous devez inclure à votre programme le fichier de définitions des fonctions AD.H. Les définitions sont les suivantes :

CONSTANTES

ADR1 et **ADR2** sont les deux adresses qui définissent les adresses de base de la **PC-MES3**. Ce sont ces adresses qu'il faut modifier si vous changez les cavaliers **S2** ou **S3**. Les valeurs par défaut sont **\$330** et **\$340**.

IOIN : adresse du port d'entrées logiques
IOOUT : adresse du port de sorties logiques
MULPLEX : adresse du registre de sélection du multiplexeur et de la polarité de mesure
TIMER : registre de contrôle du timer (8254)
COUNT0 : compteur 0 du timer
COUNT1 : compteur 1 du timer
COUNT2 : compteur 2 du timer
ADHIGH : registre des 4 bits de poids fort du convertisseur A/D
ADLOW : registre des 8 bits de poids faible du convertisseur A/D
STARTAD : déclenchement d'une mesure sur l'A/D
UPDDA : validation des valeurs précédemment écrites dans les convertisseurs D/A
DA1LOW : registre des 8 bits de poids faible du convertisseur DA1
DA1HIGH : registre des 4 bits de poids fort du convertisseur DA1
DA2LOW et **DA2HIGH** : identiques à **DA1LOW** et **DA1HIGH** pour le convertisseur DA2

Remarques :

Pour plus de précisions sur ces registres, voir le chapitre "PROGRAMMATION DE LA **PC-MES3**".

Il est indispensable d'appeler la fonction **INITDRVAD** avant toute utilisation d'une autre fonction.

Activer une sortie du port de sortie (c'est-à-dire mettre le bit correspondant à 1) met la sortie au niveau 0V (GND). Ceci est dû au fait que les sorties sont à collecteur ouvert.

De même, comme lorsqu'une entrée n'est pas active (en l'air), des résistances de rappel mettent cette entrée au niveau +5V, le fait de la rendre active (en la reliant à la masse) rend un résultat 1 par la fonction **TESTBIT**.

FONCTIONS

INITDRVAD : initialisation des drivers de la **PC-MES3**

Syntaxe : **void initdrvad(int adr1, int adr2)**

Doit être impérativement appelée avant toute autre fonction

SETADR : initialisation des adresses de la **PC-MES3**

Syntaxe : **Void setadr(int adr1, int adr2)**

Permet d'initialiser toutes les routines avec des nouvelles adresses définies sur la **PC-MES3**

SETPOLAR : sélection de la polarité courante

Syntaxe : **void setpolar(char pol)**

Si pol = "U", mesures unipolaires (0/+10,24V)

Si pol = "B", mesures bipolaires (-5,12/+5,12V)

SETCHANNEL : sélection du canal courant

Syntaxe : **void setchannel(int ch)**

Permet de sélectionner le numéro du canal (0 à 7) qui sera mesuré

READAD : lancement d'une mesure du convertisseur A/D

Syntaxe : **int readad(void)**

Lance une mesure sur le canal et la polarité courants et retourne le résultat. Si la mesure est bipolaire, retourne une valeur comprise entre -2048 et +2047, sinon entre 0 et 4095

Remarques :

Pour plus de précisions sur ces registres, voir le chapitre "PROGRAMMATION DE LA **PC-MES3**".

Il est indispensable d'appeler la fonction **INITDRVAD** avant toute utilisation d'une autre fonction.

Activer une sortie du port de sortie (c'est-à-dire mettre le bit correspondant à 1) met la sortie au niveau 0V (GND). Ceci est dû au fait que les sorties sont à collecteur ouvert.

De même, comme lorsqu'une entrée n'est pas active (en l'air), des résistances de rappel mettent cette entrée au niveau +5V, le fait de la rendre active (en la reliant à la masse) rend un résultat 1 par la fonction **TESTBIT**.

FONCTIONS

INITDRVAD : initialisation des drivers de la **PC-MES3**

Syntaxe : **void initdrvad(int adr1, int adr2)**

Doit être impérativement appelée avant toute autre fonction

SETADR : initialisation des adresses de la **PC-MES3**

Syntaxe : **Void setadr(int adr1, int adr2)**

Permet d'initialiser toutes les routines avec des nouvelles adresses définies sur la **PC-MES3**

SETPOLAR : sélection de la polarité courante

Syntaxe : **void setpolar(char pol)**

Si pol = "U", mesures unipolaires (0/+10,24V)

Si pol = "B", mesures bipolaires (-5,12/+5,12V)

SETCHANNEL : sélection du canal courant

Syntaxe : **void setchannel(int ch)**

Permet de sélectionner le numéro du canal (0 à 7) qui sera mesuré

READAD : lancement d'une mesure du convertisseur A/D

Syntaxe : **int readad(void)**

Lance une mesure sur le canal et la polarité courants et retourne le résultat. Si la mesure est bipolaire, retourne une valeur comprise entre -2048 et +2047, sinon entre 0 et 4095

SETDA1 : programme le convertisseur D/A numéro 1

Syntaxe : **void setda1(unsigned int value)**

Programme une tension sur le D/A numéro 1, **value** devant être compris entre 0 (0V) et 4095 (10V)

SETDA2 : programme le convertisseur D/A numéro 2

Identique à SETDA1 mais pour le D/A numéro 2

OUTPUTIO : écriture d'une valeur dans le port de sortie

Syntaxe : **void outputio(unsigned char value)**

Ecrit l'octet **value** dans le port de sortie

SETBIT : positionne un bit sur le port de sortie

Syntaxe : **void setbit(int bit, int value)**

Positionne le bit de numéro **bit** (0-7) à la valeur **value** (0-1)

INPUTIO : lecture du port d'entrée

Syntaxe : **int inputio(void)**

Retourne l'état du port 8 bits d'entrée

TESTBIT : test d'un bit du port d'entrée

Syntaxe : **int testbit(int bit)**

Retourne 0 ou 1 selon l'état du bit de numéro **bit** du port d'entrée

SETTIMER : programmation des compteurs du timer

Syntaxe : **void settimer(int numtimer, int mode, int bcd, unsigned int val)**

Programme le compteur de numéro **numtimer** (0-2) dans le mode **mode** (0-5) pour un comptage en binaire (**bcd** à 0) ou en BCD (**bcd** à 1) avec une valeur **val** (0-65535)

READTIMER : lecture des compteurs du timer

Syntaxe : **unsigned int readtimer(int numtimer)**

Retourne la valeur actuelle du compteur correspondant au timer de numéro **numtimer**

STARTAD : lectures analogiques sur plusieurs voies

Syntaxe : **void startad(Void far *tab, int tps1, int tps2, int nbmes, int voie1, int voie2, int voie3, int voie4)**

Lance une conversion de plusieurs points sur différentes voies.

***tab** correspond à un pointeur sur le tableau des résultats.

TPS1 et TPS2 sont les valeurs respectives des compteurs 1 et 2, et correspondant à la base de temps des acquisitions.

SETDA1 : programme le convertisseur D/A numéro 1

Syntaxe : **void setda1(unsigned int value)**

Programme une tension sur le D/A numéro 1, **value** devant être compris entre 0 (0V) et 4095 (10V)

SETDA2 : programme le convertisseur D/A numéro 2

Identique à SETDA1 mais pour le D/A numéro 2

OUTPUTIO : écriture d'une valeur dans le port de sortie

Syntaxe : **void outputio(unsigned char value)**

Ecrit l'octet **value** dans le port de sortie

SETBIT : positionne un bit sur le port de sortie

Syntaxe : **void setbit(int bit, int value)**

Positionne le bit de numéro **bit** (0-7) à la valeur **value** (0-1)

INPUTIO : lecture du port d'entrée

Syntaxe : **int inputio(void)**

Retourne l'état du port 8 bits d'entrée

TESTBIT : test d'un bit du port d'entrée

Syntaxe : **int testbit(int bit)**

Retourne 0 ou 1 selon l'état du bit de numéro **bit** du port d'entrée

SETTIMER : programmation des compteurs du timer

Syntaxe : **void settimer(int numtimer, int mode, int bcd, unsigned int val)**

Programme le compteur de numéro **numtimer** (0-2) dans le mode **mode** (0-5) pour un comptage en binaire (**bcd** à 0) ou en BCD (**bcd** à 1) avec une valeur **val** (0-65535)

READTIMER : lecture des compteurs du timer

Syntaxe : **unsigned int readtimer(int numtimer)**

Retourne la valeur actuelle du compteur correspondant au timer de numéro **numtimer**

STARTAD : lectures analogiques sur plusieurs voies

Syntaxe : **void startad(Void far *tab, int tps1, int tps2, int nbmes, int voie1, int voie2, int voie3, int voie4)**

Lance une conversion de plusieurs points sur différentes voies.

***tab** correspond à un pointeur sur le tableau des résultats.

TPS1 et TPS2 sont les valeurs respectives des compteurs 1 et 2, et correspondant à la base de temps des acquisitions.

Nbmes correspond au nombre de mesures par voie que l'on souhaite faire, voie1 à voie4 sont les numéros des voies que l'on veut acquérir. Combiner l'état de la polarité de la carte avec le numéro de la voie ("ou" logique)

CETTE FONCTION N'EST DISPONIBLE QUE DANS LE MODULE C_AD.ASM, ET N'EST DONC UTILISABLE QU'EN LANGAGE C AVEC APPEL ASSEMBLEUR

Nbmes correspond au nombre de mesures par voie que l'on souhaite faire, voie1 à voie4 sont les numéros des voies que l'on veut acquérir. Combiner l'état de la polarité de la carte avec le numéro de la voie ("ou" logique)

CETTE FONCTION N'EST DISPONIBLE QUE DANS LE MODULE C_AD.ASM, ET N'EST DONC UTILISABLE QU'EN LANGAGE C AVEC APPEL ASSEMBLEUR

ROUTINES POUR LE PASCAL

Les routines pour le PASCAL sont très ressemblantes à celles du C.
Les différences sont détaillées ci-dessous.

De même que pour le C, deux types de routines sont disponibles, **AD1.PAS** pour l'utilisation des routines PASCAL avec appel Assembleur et **AD2.PAS** pour l'utilisation sous PASCAL standard (routines écrites en Pascal).

CONSTANTES

Dans les deux cas, les définitions de constantes sont les mêmes que pour le C, consulter le chapitre précédent.

FONCTIONS

Les procédures et fonctions sont identiques à celles du C, seule la syntaxe est différente. Voir le paragraphe sur le C pour plus de détails.

INITDRVAD : initialisation des drivers de la **PC-MES3**

Syntaxe : **procedure InitDrvAd(Adr1, Adr2 : integer)**

Doit impérativement être appelée en premier

SETADR : initialisation des adresses de la PC-MES3

Syntaxe : **Void setadr (adr1, adr2 : integer)**

N'est disponible que dans le module Pascal avec Appel Assembleur

SETPOLAR : sélection de la polarité courante

Syntaxe : **procedure SetPolar(Pol : Char);**

SETCHANNEL : sélection du canal courant

Syntaxe : **procedure SetChanel(Ch : integer);**

READAD : lancement d'une mesure sur le convertisseur A/D

Syntaxe : **function ReadAD : integer;**

SETDA1 : programme le convertisseur D/A numéro 1

Syntaxe : **procedure SetDA1 (Value : integer);**

SETDA2 : programme le convertisseur D/A numéro 2

Identique à SETDA1 mais pour le D/A numéro 2

ROUTINES POUR LE PASCAL

Les routines pour le PASCAL sont très ressemblantes à celles du C.
Les différences sont détaillées ci-dessous.

De même que pour le C, deux types de routines sont disponibles, **AD1.PAS** pour l'utilisation des routines PASCAL avec appel Assembleur et **AD2.PAS** pour l'utilisation sous PASCAL standard (routines écrites en Pascal).

CONSTANTES

Dans les deux cas, les définitions de constantes sont les mêmes que pour le C, consulter le chapitre précédent.

FONCTIONS

Les procédures et fonctions sont identiques à celles du C, seule la syntaxe est différente. Voir le paragraphe sur le C pour plus de détails.

INITDRVAD : initialisation des drivers de la **PC-MES3**

Syntaxe : **procedure InitDrvAd(Adr1, Adr2 : integer)**

Doit impérativement être appelée en premier

SETADR : initialisation des adresses de la PC-MES3

Syntaxe : **Void setadr (adr1, adr2 : integer)**

N'est disponible que dans le module Pascal avec Appel Assembleur

SETPOLAR : sélection de la polarité courante

Syntaxe : **procedure SetPolar(Pol : Char);**

SETCHANNEL : sélection du canal courant

Syntaxe : **procedure SetChanel(Ch : integer);**

READAD : lancement d'une mesure sur le convertisseur A/D

Syntaxe : **function ReadAD : integer;**

SETDA1 : programme le convertisseur D/A numéro 1

Syntaxe : **procedure SetDA1 (Value : integer);**

SETDA2 : programme le convertisseur D/A numéro 2

Identique à SETDA1 mais pour le D/A numéro 2

OUTPUTIO : écriture d'une valeur dans le port de sortie

Syntaxe : **procedure OutputIO(Value : integer);**

SETBIT : positionne un bit sur le port de sortie

Syntaxe : **procedure SetBit(Bit, Value : integer);**

INPUTIO : lecture du port d'entrée

Syntaxe : **function InputIO : integer;**

TESTBIT : test d'un bit du port d'entrée

Syntaxe : **function TestBit(Bit : integer) : integer;**

SETTIMER : programmation d'un des compteurs du timer

Syntaxe : **procedure SetTimer(Numtimer, Mode, BCD, Val:integer);**

READTIMER : lit un compteur du timer

Syntaxe : **function ReadTimer(NumTimer : integer) : integer;**

Les mêmes remarques que pour le C sont valables ici.

Remarques : - La fonction STARTAD n'est définie ni pour le PASCAL, ni pour le PASCAL avec appel Assembleur.
- La fonction SETADR n'est définie que pour le Pascal avec appel Assembleur.

OUTPUTIO : écriture d'une valeur dans le port de sortie

Syntaxe : **procedure OutputIO(Value : integer);**

SETBIT : positionne un bit sur le port de sortie

Syntaxe : **procedure SetBit(Bit, Value : integer);**

INPUTIO : lecture du port d'entrée

Syntaxe : **function InputIO : integer;**

TESTBIT : test d'un bit du port d'entrée

Syntaxe : **function TestBit(Bit : integer) : integer;**

SETTIMER : programmation d'un des compteurs du timer

Syntaxe : **procedure SetTimer(Numtimer, Mode, BCD, Val:integer);**

READTIMER : lit un compteur du timer

Syntaxe : **function ReadTimer(NumTimer : integer) : integer;**

Les mêmes remarques que pour le C sont valables ici.

Remarques : - La fonction STARTAD n'est définie ni pour le PASCAL, ni pour le PASCAL avec appel Assembleur.
- La fonction SETADR n'est définie que pour le Pascal avec appel Assembleur.

ROUTINES POUR LE BASIC

Les routines en BASIC sont, elles aussi, identiques à celles en C, seuls le passage de paramètres et l'appel diffèrent.

NOTE : Ces routines ont été écrites en GWBASIC (ou BASICA) et peuvent demander certaines modifications sur un autre BASIC.

CONSTANTES

Des constantes identiques à celles en C sont définies, du fait de leur type entier, le nom est suivi d'un %.

FONCTIONS

Les sous-programmes ont la même fonction que ceux en C et PASCAL, seul l'appel et le passage des paramètres changent. L'appel doit être fait par un GOSUB N°Ligne, les paramètres doivent passer dans les variables indiquées. Toutes les variables sont de type entier, d'où le signe % à la fin de leurs noms. Pour plus de détails, voir la description en C.

INITDRVAD : initialisation des drivers de la **PC-MES3**

N°de ligne : 10000

Paramètres : Aucun

Doit être impérativement appelée en premier

SETPOLAR : sélection de la polarité courante

N°de ligne : 10100

Paramètres : POLAR\$ contient "U" ou "B"

SETCHANNEL : sélection du canal courant

N°de ligne : 10200

Paramètres : CHANNEL% contient le numéro du canal

READAD : lancement d'une mesure sur le convertisseur A/D

N°de ligne : 10800

Paramètres : VALUE% retourne le résultat de la mesure

SETDA1 : programme le convertisseur D/A numéro 1

N°de ligne : 10300

Paramètres : VALUE% contient la valeur à écrire dans DA1

ROUTINES POUR LE BASIC

Les routines en BASIC sont, elles aussi, identiques à celles en C, seuls le passage de paramètres et l'appel diffèrent.

NOTE : Ces routines ont été écrites en GWBASIC (ou BASICA) et peuvent demander certaines modifications sur un autre BASIC.

CONSTANTES

Des constantes identiques à celles en C sont définies, du fait de leur type entier, le nom est suivi d'un %.

FONCTIONS

Les sous-programmes ont la même fonction que ceux en C et PASCAL, seul l'appel et le passage des paramètres changent. L'appel doit être fait par un GOSUB N°Ligne, les paramètres doivent passer dans les variables indiquées. Toutes les variables sont de type entier, d'où le signe % à la fin de leurs noms. Pour plus de détails, voir la description en C.

INITDRVAD : initialisation des drivers de la **PC-MES3**

N°de ligne : 10000

Paramètres : Aucun

Doit être impérativement appelée en premier

SETPOLAR : sélection de la polarité courante

N°de ligne : 10100

Paramètres : POLAR\$ contient "U" ou "B"

SETCHANNEL : sélection du canal courant

N°de ligne : 10200

Paramètres : CHANNEL% contient le numéro du canal

READAD : lancement d'une mesure sur le convertisseur A/D

N°de ligne : 10800

Paramètres : VALUE% retourne le résultat de la mesure

SETDA1 : programme le convertisseur D/A numéro 1

N°de ligne : 10300

Paramètres : VALUE% contient la valeur à écrire dans DA1

SETDA2 : programme le convertisseur D/A numéro 2

N° de ligne : 10400

Paramètres : VALUE% contient la valeur à écrire dans DA2

OUTPUTIO : écriture dans le port de sortie

N° de ligne : 10500

Paramètres : VALUE% contient la valeur à écrire dans le port de sortie

SETBIT : positionne un bit sur le port de sortie

N° de ligne : 10600

Paramètres : BIT% contient le numéro du bit à modifier VALUE% contient 0 ou 1 selon la valeur à écrire

INPUTIO : lecture du port d'entrée

N° de ligne : 10900

Paramètres : VALUE% retourne la valeur du port d'entrée

TESTBIT : test d'un bit du port d'entrée

N° de ligne : 11000

Paramètres : BIT% contient le numéro du bit à tester VALUE% retourne 0 ou 1 selon l'état de l'entrée

SETTIMER : programmation d'un des compteurs du timer

N° de ligne : 10700

Paramètres : NUMTIMER% contient le N° du timer
MODE% contient le mode du timer (0-5)
BCD% contient le mode binaire ou BCD
VALUE% contient la valeur à programmer

READTIMER : lit un compteur du timer

N° de ligne : 11200

Paramètres : NUMTIMER% contient le N° du timer VALUE% retourne la valeur lue dans le compteur

Les remarques faites pour le C sont valables ici.

Les fonctions STARTAD et SETADR ne sont pas définies pour le BASIC.

NOTE : Les valeurs programmées dans les compteurs, pouvant varier entre 0 et 65535, la variable VALUE% peut poser des problèmes avec certains BASIC qui considèrent les variables entières comme des nombres signés entre -32768 et +32767.

SETDA2 : programme le convertisseur D/A numéro 2

N° de ligne : 10400

Paramètres : VALUE% contient la valeur à écrire dans DA2

OUTPUTIO : écriture dans le port de sortie

N° de ligne : 10500

Paramètres : VALUE% contient la valeur à écrire dans le port de sortie

SETBIT : positionne un bit sur le port de sortie

N° de ligne : 10600

Paramètres : BIT% contient le numéro du bit à modifier VALUE% contient 0 ou 1 selon la valeur à écrire

INPUTIO : lecture du port d'entrée

N° de ligne : 10900

Paramètres : VALUE% retourne la valeur du port d'entrée

TESTBIT : test d'un bit du port d'entrée

N° de ligne : 11000

Paramètres : BIT% contient le numéro du bit à tester VALUE% retourne 0 ou 1 selon l'état de l'entrée

SETTIMER : programmation d'un des compteurs du timer

N° de ligne : 10700

Paramètres : NUMTIMER% contient le N° du timer
MODE% contient le mode du timer (0-5)
BCD% contient le mode binaire ou BCD
VALUE% contient la valeur à programmer

READTIMER : lit un compteur du timer

N° de ligne : 11200

Paramètres : NUMTIMER% contient le N° du timer VALUE% retourne la valeur lue dans le compteur

Les remarques faites pour le C sont valables ici.

Les fonctions STARTAD et SETADR ne sont pas définies pour le BASIC.

NOTE : Les valeurs programmées dans les compteurs, pouvant varier entre 0 et 65535, la variable VALUE% peut poser des problèmes avec certains BASIC qui considèrent les variables entières comme des nombres signés entre -32768 et +32767.