

**CARTE INTERFACE
MULTIFONCTIONS RAPIDE
FASTLAB**

Pour PC-AT et COMPATIBLE

EUROSMART

MANUEL D'UTILISATION

Ce manuel et les routines contenues sur la disquette accompagnant la carte FASTLAB sont protégés par copyright 1994.

Tout ou partie de ces documents ne pourra être reproduit, transcrit, stocké dans n'importe quel système informatique ou autre, traduit dans n'importe quelle langue et n'importe quel langage informatique sans le consentement préalable et écrit d'EUROSMART.

EUROSMART se réserve le droit de réviser ultérieurement ces documents, de modifier les caractéristiques techniques de ses produits, sans préavis, ni aucune obligation de sa part.

- IBM PC, AT sont des marques déposées de International Business Machines Corporation
- INTEL est une marque déposée de Intel Corporation
- TURBO PASCAL et TURBO C sont des marques déposées de Borland International
- FASTLAB est une marque déposée d'Eurosmart

07-05-1997

**CARTE INTERFACE
MULTIFONCTIONS RAPIDE
FASTLAB**

Pour PC-AT et COMPATIBLE

EUROSMART

MANUEL D'UTILISATION

Ce manuel et les routines contenues sur la disquette accompagnant la carte FASTLAB sont protégés par copyright 1994.

Tout ou partie de ces documents ne pourra être reproduit, transcrit, stocké dans n'importe quel système informatique ou autre, traduit dans n'importe quelle langue et n'importe quel langage informatique sans le consentement préalable et écrit d'EUROSMART.

EUROSMART se réserve le droit de réviser ultérieurement ces documents, de modifier les caractéristiques techniques de ses produits, sans préavis, ni aucune obligation de sa part.

- IBM PC, AT sont des marques déposées de International Business Machines Corporation
- INTEL est une marque déposée de Intel Corporation
- TURBO PASCAL et TURBO C sont des marques déposées de Borland International
- FASTLAB est une marque déposée d'Eurosmart

07-05-1997

SOMMAIRE

PAGE

INTRODUCTION	3
CARACTERISTIQUES TECHNIQUES ET SPECIFICATIONS DE LA CARTE...	5
PREPARATION ET INSTALLATION DE LA CARTE.....	9
CONNEXION ET REGLAGE.....	11
FONCTIONNEMENT DE L'INTERFACE ANALOGIQUE/NUMERIQUE	19
FONCTIONNEMENT DE L'INTERFACE NUMERIQUE/ANALOGIQUE	25
L'INTERFACE LOGIQUE	29
FONCTIONNEMENT DES COUPLEURS D'ENTREE/SORTIE PIO 8255	31
FONCTIONNEMENT DES TIMERS PROGRAMMABLES 8254	35
UTILISATION DE LA CARTE SOUS INTERRUPTION.....	39
OUTILS LOGICIELS ET ROUTINES.....	41

SOMMAIRE

PAGE

INTRODUCTION	3
CARACTERISTIQUES TECHNIQUES ET SPECIFICATIONS DE LA CARTE...	5
PREPARATION ET INSTALLATION DE LA CARTE	9
CONNEXION ET REGLAGE	11
FONCTIONNEMENT DE L'INTERFACE ANALOGIQUE/NUMERIQUE.....	19
FONCTIONNEMENT DE L'INTERFACE NUMERIQUE/ANALOGIQUE.....	25
L'INTERFACE LOGIQUE	29
FONCTIONNEMENT DES COUPLEURS D'ENTREE/SORTIE PIO 8255.....	31
FONCTIONNEMENT DES TIMERS PROGRAMMABLES 8254.....	35
UTILISATION DE LA CARTE SOUS INTERRUPTION	39
OUTILS LOGICIELS ET ROUTINES	41

INTRODUCTION

La carte FASTLAB se compose d'un module de conversion analogique-numérique 12 bits avec un temps de conversion de $1,25\mu\text{s}$ (800kHz), d'un module de deux voies numérique-analogique 12 bits, associés à une interface d'entrée sortie double PIO 82C55 et de deux bases de temps triple décompteur timer 82C54.

Elle permet donc de :

- Mesurer des signaux en tension ou courant (possibilité de positionner des résistances de conversion). Ces signaux sont convertis en mots numériques de 12 bits et interfacés via le bus d'entrée/sortie 16 bits de votre ordinateur. On peut mesurer jusqu'à 8 entrées simples ou 4 entrées différentielles.
- Restituer des signaux en tension ou courant (option), résolution 12 bits.
- Connecter votre ordinateur sur un périphérique nécessitant une commande logique du type tout ou rien.
- Effectuer des comptages, générer des bases de temps, ou des interruptions.

La carte FASTLAB a été conçue pour répondre à vos besoins croissants. Après qu'elle ait été installée correctement, elle doit pouvoir assurer un fonctionnement sans problème pendant de nombreuses années.

La carte FASTLAB répond aux Directives Européennes en terme de Compatibilité Electromagnétique (CE).

Ce manuel d'utilisation, conjointement à la disquette contenant les routines, a été rédigé de manière à utiliser la carte FASTLAB dans les meilleures conditions.

Veuillez lire attentivement les instructions avant toute manipulation.

Note : La Société EUROSART décline toute responsabilité en cas d'utilisation de la FASTLAB qui ne soit pas conforme à la documentation.

INTRODUCTION

La carte FASTLAB se compose d'un module de conversion analogique-numérique 12 bits avec un temps de conversion de $1,25\mu\text{s}$ (800kHz), d'un module de deux voies numérique-analogique 12 bits, associés à une interface d'entrée sortie double PIO 82C55 et de deux bases de temps triple décompteur timer 82C54.

Elle permet donc de :

- Mesurer des signaux en tension ou courant (possibilité de positionner des résistances de conversion). Ces signaux sont convertis en mots numériques de 12 bits et interfacés via le bus d'entrée/sortie 16 bits de votre ordinateur. On peut mesurer jusqu'à 8 entrées simples ou 4 entrées différentielles.
- Restituer des signaux en tension ou courant (option), résolution 12 bits.
- Connecter votre ordinateur sur un périphérique nécessitant une commande logique du type tout ou rien.
- Effectuer des comptages, générer des bases de temps, ou des interruptions.

La carte FASTLAB a été conçue pour répondre à vos besoins croissants. Après qu'elle ait été installée correctement, elle doit pouvoir assurer un fonctionnement sans problème pendant de nombreuses années.

La carte FASTLAB répond aux Directives Européennes en terme de Compatibilité Electromagnétique (CE).

Ce manuel d'utilisation, conjointement à la disquette contenant les routines, a été rédigé de manière à utiliser la carte FASTLAB dans les meilleures conditions.

Veuillez lire attentivement les instructions avant toute manipulation.

Note : La Société EUROSART décline toute responsabilité en cas d'utilisation de la FASTLAB qui ne soit pas conforme à la documentation.

CARACTERISTIQUES TECHNIQUES

INTERFACE ANALOGIQUE/NUMERIQUE

- Résolution 12 bits (4096 pts)
- Voies d'entrée 8 voies simples ou 4 différentielles (mode programmable)
- Echelle d'entrée Bipolaire programmable $\pm 10\text{V}$ maximum - gain 1V/V
- Protection 100V/millisecondes - 60V continu
- Temps de conversion 1,25 μs (800kHz)
- Précision 12 bits ± 1 LSB
- Non-linéarité 12 bits ± 1 LSB sur la pleine échelle $\pm 10\text{V}$
- Codage numérique Binaire naturel
- Possibilité d'entrées en courant par composants passifs (en option)

Amplificateurs d'instrumentations

- 16 gains programmables 1,2,4,8,10,20,40,80,100,200,400,800,1000,2000,4000,8000V/V
- Impédance d'entrée 10 Gohms // 3pF
- Réjection de mode commun >80 dB gain = 1 10Kohms
- Bruit système

gain 1,2,4,8,10,20V/V	$\pm 1\text{LSB}$
gain 40,80V/V	$\pm 3\text{LSB}$
gain 100V/V	$\pm 3\text{LSB}$
gain 200V/V	$\pm 4\text{LSB}$
gain 400V/V	$\pm 10\text{LSB}$
gain 800V/V	$\pm 15\text{LSB}$
gain 1000V/V	$\pm 8\text{LSB}$
- Offset en fonction du gain (ajusté au gain de 100V/V)

gain 1,2,4,8V/V	$\pm 0,5\text{LSB}$
gain 10,20,40,80V/V	$\pm 2\text{LSB}$
gain 100,200,400,500V/V	$\pm 1\text{LSB}$
gain 1000V/V	30LSB typique ajustable à $\pm 1\text{LSB}$
- Vitesse de montée en tension 20V/ μs typique
- Temps d'établissement à 0,01% du signal

gain <100	5 μs
gain ≥ 100	20 μs
- Temps de recouvrement après une surcharge

gain <100	10 μs
gain ≥ 100	20 μs
- Déclenchement numérique par logiciel, timer ou signal externe (TTL)

CARACTERISTIQUES TECHNIQUES

INTERFACE ANALOGIQUE/NUMERIQUE

- Résolution 12 bits (4096 pts)
- Voies d'entrée 8 voies simples ou 4 différentielles (mode programmable)
- Echelle d'entrée Bipolaire programmable $\pm 10\text{V}$ maximum - gain 1V/V
- Protection 100V/millisecondes - 60V continu
- Temps de conversion 1,25 μs (800kHz)
- Précision 12 bits ± 1 LSB
- Non-linéarité 12 bits ± 1 LSB sur la pleine échelle $\pm 10\text{V}$
- Codage numérique Binaire naturel
- Possibilité d'entrées en courant par composants passifs (en option)

Amplificateurs d'instrumentations

- 16 gains programmables 1,2,4,8,10,20,40,80,100,200,400,800,1000,2000,4000,8000V/V
- Impédance d'entrée 10 Gohms // 3pF
- Réjection de mode commun >80 dB gain = 1 10Kohms
- Bruit système

gain 1,2,4,8,10,20V/V	$\pm 1\text{LSB}$
gain 40,80V/V	$\pm 3\text{LSB}$
gain 100V/V	$\pm 3\text{LSB}$
gain 200V/V	$\pm 4\text{LSB}$
gain 400V/V	$\pm 10\text{LSB}$
gain 800V/V	$\pm 15\text{LSB}$
gain 1000V/V	$\pm 8\text{LSB}$
- Offset en fonction du gain (ajusté au gain de 100V/V)

gain 1,2,4,8V/V	$\pm 0,5\text{LSB}$
gain 10,20,40,80V/V	$\pm 2\text{LSB}$
gain 100,200,400,500V/V	$\pm 1\text{LSB}$
gain 1000V/V	30LSB typique ajustable à $\pm 1\text{LSB}$
- Vitesse de montée en tension 20V/ μs typique
- Temps d'établissement à 0,01% du signal

gain <100	5 μs
gain ≥ 100	20 μs
- Temps de recouvrement après une surcharge

gain <100	10 μs
gain ≥ 100	20 μs
- Déclenchement numérique par logiciel, timer ou signal externe (TTL)

INTERFACE NUMERIQUE/ANALOGIQUE

- Voies de sortie Bipolaire 2 indépendantes
 - Résolution 12 bits (4096 pts) $\pm$ 1LSB
 - Echelle de sortie bipolaire $\pm$ 10V
 - Courant maxi 4mA
 - Temps de conversion 5 μ s max
 - Codage numérique Binaire naturel
- Possibilité de sorties en courant 4-20mA (en option)

INTERFACE D'ENTREE/SORTIE LOGIQUE : PIO 8255

- 32 lignes TTL sur le connecteur P2 type HE10 mâle 26 points (voir page 30) et le connecteur d'entrée P1 DB37 femelle (voir page 11).

TIMERS

- 6 décompteurs de 16 bits indépendants
- Fréquence de base 4 MHz disponible sur la carte
- Horloge externe 10 MHz MAX

ALIMENTATIONS

- Consommation alimentation +5V : 920 mA.
- Consommation alimentation +12V : 31 mA.
- Consommation alimentation -12V : 13 mA.

DIVERS

- Bus IBM PC-AT 16 bits
- Adressage commutable sur la carte par blocs de 32 adresses sélectables entre \$0000 à FF00
- 8 niveaux d'interruption (IRQ 3, 4, 5, 7, 10, 11, 12, 15), sélection programmable
- Connecteur d'entrée type DB37 femelle + connecteur sur carte type HE10 mâle 26 points
- Sorties des alimentations $\pm$ 5V et $\pm$ 12V du PC sur le connecteur P1 de la carte
- Dimensions : 230 x 108 x 20 mm
- Masse : 0,25kg

ETAT DE LA FASTLAB (lors de la mise sous tension de l'ordinateur)

- Entrées A/N : simples
 voie 0 sélectionnée
 gain 1
- Sorties N/A : 0V
- Timers : à l'arrêt
- Ports B et C des PIO 1 et 2 : en entrée

INTERFACE NUMERIQUE/ANALOGIQUE

- Voies de sortie Bipolaire 2 indépendantes
 - Résolution 12 bits (4096 pts) $\pm$ 1LSB
 - Echelle de sortie bipolaire $\pm$ 10V
 - Courant maxi 4mA
 - Temps de conversion 5 μ s max
 - Codage numérique Binaire naturel
- Possibilité de sorties en courant 4-20mA (en option)

INTERFACE D'ENTREE/SORTIE LOGIQUE : PIO 8255

- 32 lignes TTL sur le connecteur P2 type HE10 mâle 26 points (voir page 30) et le connecteur d'entrée P1 DB37 femelle (voir page 11).

TIMERS

- 6 décompteurs de 16 bits indépendants
- Fréquence de base 4 MHz disponible sur la carte
- Horloge externe 10 MHz MAX

ALIMENTATIONS

- Consommation alimentation +5V : 920 mA.
- Consommation alimentation +12V : 31 mA.
- Consommation alimentation -12V : 13 mA.

DIVERS

- Bus IBM PC-AT 16 bits
- Adressage commutable sur la carte par blocs de 32 adresses sélectables entre \$0000 à FF00
- 8 niveaux d'interruption (IRQ 3, 4, 5, 7, 10, 11, 12, 15), sélection programmable
- Connecteur d'entrée type DB37 femelle + connecteur sur carte type HE10 mâle 26 points
- Sorties des alimentations $\pm$ 5V et $\pm$ 12V du PC sur le connecteur P1 de la carte
- Dimensions : 230 x 108 x 20 mm
- Masse : 0,25kg

ETAT DE LA FASTLAB (lors de la mise sous tension de l'ordinateur)

- Entrées A/N : simples
 voie 0 sélectionnée
 gain 1
- Sorties N/A : 0V
- Timers : à l'arrêt
- Ports B et C des PIO 1 et 2 : en entrée

OPTIONS

- FASTEXT/B :** Boîtier de connexion externe blindé permettant l'accès direct à la majorité des fonctions de la carte sur 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.

SYNOPSIS DE LA FASTLAB**OPTIONS**

- FASTEXT/B :** Boîtier de connexion externe blindé permettant l'accès direct à la majorité des fonctions de la carte sur 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.

SYNOPSIS DE LA FASTLAB

IMPLANTATIONS DES COMPOSANTS DE LA FASTLAB

**EMPLACEMENT DES REGLAGES ET DES OPTIONS
DISPONIBLES SUR LA FASTLAB**

Figure 1

IMPLANTATIONS DES COMPOSANTS DE LA FASTLAB

**EMPLACEMENT DES REGLAGES ET DES OPTIONS
DISPONIBLES SUR LA FASTLAB**

Figure 1

PREPARATION ET INSTALLATION DE LA CARTE

Ce chapitre explique les différentes manipulations à effectuer avant la mise en fonctionnement de la carte. Il est vivement conseillé de **LIRE TOUTE LA DOCUMENTATION** avant manipulation.

PREPARATION

La carte possède un commutateur logique noté ADR pour définir l'adresse de base obligatoire sur le bus d'entrée/sortie de votre ordinateur.

ADRESSE DE BASE DE LA CARTE

La carte occupe un espace adressable de 24 adresses. L'adresse de base B est la première utilisée pour accéder aux 23 autres, soit B+X. La détermination de l'adresse de base s'effectue en additionnant les valeurs équivalentes aux commutateurs positionnés (OFF). Par convention l'adresse B représentera une adresse décimale, l'adresse B\$ une adresse hexadécimale.

Huit commutateurs (switch) ADR (12345678) permettent de sélectionner B.

SWITCH	ADRESSE BUS	DECIMAL	BINAIRE
1	A5	32	\$20
2	A6	64	\$40
3	A7	128	\$80
4	A8	256	\$100
5	A9	512	\$200
6	A10	1024	\$400
7	A11	2048	\$800
8	A12	4096	\$1000

Exemple : adresse \$320 = switchs 1,4,5 OFF.

La carte fonctionne en adressage d'entrées/sorties (I/O), espace mémoire de 65536 valeurs dans la famille 8088/6. IBM définit cet espace à 768 I/O qu'il utilise en partie pour les liaisons séries, les DMA, l'horloge, etc...

La carte FASTLAB ne doit pas avoir d'adresse commune avec une autre carte. Vous trouverez sur votre manuel IBM les adresses couramment occupées. Nous vous conseillons les adresses de base suivantes :

décimal	hexadécimal	
B	B\$	
576	\$240	la carte est livrée avec l'adresse \$240 = switchs 2,5 OFF
640	\$280	
768	\$300	
800	\$320	
832	\$340	

PREPARATION ET INSTALLATION DE LA CARTE

Ce chapitre explique les différentes manipulations à effectuer avant la mise en fonctionnement de la carte. Il est vivement conseillé de **LIRE TOUTE LA DOCUMENTATION** avant manipulation.

PREPARATION

La carte possède un commutateur logique noté ADR pour définir l'adresse de base obligatoire sur le bus d'entrée/sortie de votre ordinateur.

ADRESSE DE BASE DE LA CARTE

La carte occupe un espace adressable de 24 adresses. L'adresse de base B est la première utilisée pour accéder aux 23 autres, soit B+X. La détermination de l'adresse de base s'effectue en additionnant les valeurs équivalentes aux commutateurs positionnés (OFF). Par convention l'adresse B représentera une adresse décimale, l'adresse B\$ une adresse hexadécimale.

Huit commutateurs (switch) ADR (12345678) permettent de sélectionner B.

SWITCH	ADRESSE BUS	DECIMAL	BINAIRE
1	A5	32	\$20
2	A6	64	\$40
3	A7	128	\$80
4	A8	256	\$100
5	A9	512	\$200
6	A10	1024	\$400
7	A11	2048	\$800
8	A12	4096	\$1000

Exemple : adresse \$320 = switchs 1,4,5 OFF.

La carte fonctionne en adressage d'entrées/sorties (I/O), espace mémoire de 65536 valeurs dans la famille 8088/6. IBM définit cet espace à 768 I/O qu'il utilise en partie pour les liaisons séries, les DMA, l'horloge, etc...

La carte FASTLAB ne doit pas avoir d'adresse commune avec une autre carte. Vous trouverez sur votre manuel IBM les adresses couramment occupées. Nous vous conseillons les adresses de base suivantes :

décimal	hexadécimal	
B	B\$	
576	\$240	la carte est livrée avec l'adresse \$240 = switchs 2,5 OFF
640	\$280	
768	\$300	
800	\$320	
832	\$340	

Le tableau ci-dessous résume l'implantation des composants associés aux adresses de la carte :

Adresse Hexadécimale	Adresse Décimale	Registres	R/W	
B+0	B+0	PIO0 - PORTA	R/W	PIO0
B+1	B+1	PIO0 - PORTB	R/W	PIO0
B+2	B+2	PIO0 - PORTC	R/W	PIO0
B+3	B+3	PIO0 - CONTROL WORD	R/W	PIO0
B+4	B+4	PIO1 - PORTA	R/W	PIO1
B+5	B+5	PIO1 - PORTB	R/W	PIO1
B+6	B+6	PIO1 - PORTC	R/W	PIO1
B+7	B+7	PIO1 - CONTROL WORD	R/W	PIO1
B+8	B+8	TIMER0 - COMPTEUR0	R/W	TIMER0
B+9	B+9	TIMER0 - COMPTEUR1	R/W	TIMER0
B+A	B+10	TIMER0 - COMPTEUR2	R/W	TIMER0
B+B	B+11	TIMER0 - CONTROL WORD	R/W	TIMER0
B+C	B+12	TIMER1 - COMPTEUR0	R/W	TIMER1
B+D	B+13	TIMER1 - COMPTEUR1	R/W	TIMER1
B+E	B+14	TIMER1 - COMPTEUR2	R/W	TIMER1
B+F	B+15	TIMER1 - CONTROL WORD	R/W	TIMER1
B+10	B+16	N.U/DAC VOIE 1 (16 bits)	/W	DAC
B+11	B+17	N.U/LATCH	/W	DAC
B+12	B+18	RAZ DAC/DAC VOIE 2 (16 bits)	R/W	DAC
B+13	B+19	RAZ VOIE/MUX	R/W	ADC
B+14	B+20	N.U/GAIN	/W	ADC
B+15	B+21	ETAT/VALID	R/W	ADC
B+16	B+22	CONV (16 bits)/START	R/W	ADC
B+17	B+23	CLEAR IRQEXT/NR INT	R/W	ADC

INSTALLATION

Après avoir configuré la carte 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 doit être de 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.

Le tableau ci-dessous résume l'implantation des composants associés aux adresses de la carte :

Adresse Hexadécimale	Adresse Décimale	Registres	R/W	
B+0	B+0	PIO0 - PORTA	R/W	PIO0
B+1	B+1	PIO0 - PORTB	R/W	PIO0
B+2	B+2	PIO0 - PORTC	R/W	PIO0
B+3	B+3	PIO0 - CONTROL WORD	R/W	PIO0
B+4	B+4	PIO1 - PORTA	R/W	PIO1
B+5	B+5	PIO1 - PORTB	R/W	PIO1
B+6	B+6	PIO1 - PORTC	R/W	PIO1
B+7	B+7	PIO1 - CONTROL WORD	R/W	PIO1
B+8	B+8	TIMER0 - COMPTEUR0	R/W	TIMER0
B+9	B+9	TIMER0 - COMPTEUR1	R/W	TIMER0
B+A	B+10	TIMER0 - COMPTEUR2	R/W	TIMER0
B+B	B+11	TIMER0 - CONTROL WORD	R/W	TIMER0
B+C	B+12	TIMER1 - COMPTEUR0	R/W	TIMER1
B+D	B+13	TIMER1 - COMPTEUR1	R/W	TIMER1
B+E	B+14	TIMER1 - COMPTEUR2	R/W	TIMER1
B+F	B+15	TIMER1 - CONTROL WORD	R/W	TIMER1
B+10	B+16	N.U/DAC VOIE 1 (16 bits)	/W	DAC
B+11	B+17	N.U/LATCH	/W	DAC
B+12	B+18	RAZ DAC/DAC VOIE 2 (16 bits)	R/W	DAC
B+13	B+19	RAZ VOIE/MUX	R/W	ADC
B+14	B+20	N.U/GAIN	/W	ADC
B+15	B+21	ETAT/VALID	R/W	ADC
B+16	B+22	CONV (16 bits)/START	R/W	ADC
B+17	B+23	CLEAR IRQEXT/NR INT	R/W	ADC

INSTALLATION

Après avoir configuré la carte 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 doit être de 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 REGLAGE

CONNEXION D'ENTREE P1

La figure ci-dessous représente le brochage du connecteur d'entrée vue côté externe de votre ordinateur :

CONNECTEUR P1 DB37 FEMELLE
VUE FACE ARRIERE DE L'ORDINATEUR

ATTENTION : Veuillez à bien vérifier vos liaisons avant tout branchement. Si l'application le permet, la masse logique notée DGND sera utilisée pour l'interface d'entrée/sortie logique, GNDAD pour les voies analogiques d'entrées et GNDDA pour les voies analogiques de sorties permettant ainsi de réduire le bruit injecté par la masse logique sur l'analogique.

INFORMATION : Un boîtier de connexion externe blindé est proposé en option.
Réf. : FASTEXT/B. Il donne accès à la majorité des fonctions de la **FASTLAB** sur des douilles bananes Ø 4mm. Il permet ainsi une utilisation immédiate et aisée.

CABLAGE D'ENTREE SUR P1

Les schémas ci-après notent la liaison à effectuer entre la source de tension et l'entrée de la carte. L'utilisation de câbles blindés est préférable surtout en milieu industriel.

Les voies non utilisées doivent être reliées à la masse.

CONNEXION ET REGLAGE

CONNEXION D'ENTREE P1

La figure ci-dessous représente le brochage du connecteur d'entrée vue côté externe de votre ordinateur :

CONNECTEUR P1 DB37 FEMELLE
VUE FACE ARRIERE DE L'ORDINATEUR

ATTENTION : Veuillez à bien vérifier vos liaisons avant tout branchement. Si l'application le permet, la masse logique notée DGND sera utilisée pour l'interface d'entrée/sortie logique, GNDAD pour les voies analogiques d'entrées et GNDDA pour les voies analogiques de sorties permettant ainsi de réduire le bruit injecté par la masse logique sur l'analogique.

INFORMATION : Un boîtier de connexion externe blindé est proposé en option.
Réf. : FASTEXT/B. Il donne accès à la majorité des fonctions de la **FASTLAB** sur des douilles bananes Ø 4mm. Il permet ainsi une utilisation immédiate et aisée.

CABLAGE D'ENTREE SUR P1

Les schémas ci-après notent la liaison à effectuer entre la source de tension et l'entrée de la carte. L'utilisation de câbles blindés est préférable surtout en milieu industriel.

Les voies non utilisées doivent être reliées à la masse.

CONNEXION ENTREE 8 VOIES SIMPLES

CONNEXION ENTREE 4 VOIES DIFFERENTIELLES

REGLAGES GENERAUX

La carte est étalonnée en nos services et ne demande normalement aucun autre réétalonnage. Cependant, si l'utilisateur désire corriger l'offset introduit pour un gain égal à 1000V/V ou réajuster l'échelle d'entrée du convertisseur, la procédure suivante peut-être effectuée.

AJUSTEMENT DE L'OFFSET INTRODUIT PAR L'ETAGE D'AMPLIFICATION

L'étage d'amplification se compose de deux amplificateurs d'instrumentation permettant d'obtenir 16 pas de gain programmable. Chaque amplificateur possède un offset initial différent de zéro composé d'un offset d'entrée significatif au gain élevé et un offset de sortie significatif au gain unitaire. Le but de l'ajustement est d'annuler cet offset sur les différents gains utilisés.

A cet effet, chaque amplificateur possède deux potentiomètres notés I et S (figure 1).

A l'amplificateur 1 (gain décimal : 1, 10, 100, 1000) est associé le potentiomètre S1 pour le réglage de l'offset de sortie et I1 pour l'offset d'entrée. De même pour l'amplificateur 2 (gain binaire : 1, 2, 4, 8) sont associés respectivement les potentiomètres S2 et I2.

Pour effectuer la programmation des différents gains et de la voie positionnée à la masse, l'utilisateur peut se servir du programme FASTEST fourni sur la disquette accompagnant ce manuel.

Pour effectuer le réglage, l'utilisateur doit disposer d'un voltmètre et positionner la voie Ø à la masse de la carte EAØ→GNDAD.

CONNEXION ENTREE 8 VOIES SIMPLES

CONNEXION ENTREE 4 VOIES DIFFERENTIELLES

REGLAGES GENERAUX

La carte est étalonnée en nos services et ne demande normalement aucun autre réétalonnage. Cependant, si l'utilisateur désire corriger l'offset introduit pour un gain égal à 1000V/V ou réajuster l'échelle d'entrée du convertisseur, la procédure suivante peut-être effectuée.

AJUSTEMENT DE L'OFFSET INTRODUIT PAR L'ETAGE D'AMPLIFICATION

L'étage d'amplification se compose de deux amplificateurs d'instrumentation permettant d'obtenir 16 pas de gain programmable. Chaque amplificateur possède un offset initial différent de zéro composé d'un offset d'entrée significatif au gain élevé et un offset de sortie significatif au gain unitaire. Le but de l'ajustement est d'annuler cet offset sur les différents gains utilisés.

A cet effet, chaque amplificateur possède deux potentiomètres notés I et S (figure 1).

A l'amplificateur 1 (gain décimal : 1, 10, 100, 1000) est associé le potentiomètre S1 pour le réglage de l'offset de sortie et I1 pour l'offset d'entrée. De même pour l'amplificateur 2 (gain binaire : 1, 2, 4, 8) sont associés respectivement les potentiomètres S2 et I2.

Pour effectuer la programmation des différents gains et de la voie positionnée à la masse, l'utilisateur peut se servir du programme FASTEST fourni sur la disquette accompagnant ce manuel.

Pour effectuer le réglage, l'utilisateur doit disposer d'un voltmètre et positionner la voie Ø à la masse de la carte EAØ→GNDAD.

Ajustement de l'amplificateur 1 :

- 1/ Connecter le voltmètre entre les points test TAGND et T1.
- 2/ Programmer la voie Ø et le gain 1000V/V.
- 3/ Ajuster le potentiomètre I1 pour obtenir une tension voisine de $\pm 1\text{mV}$.
- 4/ Programmer la voie Ø et le gain 1V/V.
- 5/ Ajuster le potentiomètre S1 pour obtenir une tension voisine de $\pm 0,2\text{mV}$.
- 6/ Recommencer l'opération à partir du 2/ pour contrôler les variations et affiner le réglage.

Ajustement de l'amplificateur 2 :

- 1/ Connecter le voltmètre entre les points test TAGND et T2.
- 2/ Programmer la voie Ø et le gain 8V/V (calibre $\pm 1,25\text{V}$).
- 3/ Ajuster le potentiomètre I2 pour obtenir une tension voisine de $\pm 1\text{mV}$.
- 4/ Programmer la voie Ø et le gain 1V/V.
- 5/ Ajuster le potentiomètre S2 pour obtenir une tension voisine de $\pm 0,2\text{mV}$.
- 6/ Recommencer l'opération à partir du 2/ pour contrôler les variations et affiner le réglage.

L'étage d'amplification est maintenant ajusté. Pour utiliser la carte avec un gain de 1000V/V, il faudra tenir compte d'un offset introduit par l'amplificateur 1 ou réétalonner celui-ci pour un gain 1000 en programmant le gain 1000 à la place du gain 100 dans l'opération 2/. Dans ce cas l'offset se rapportera au gain 100.

AJUSTEMENT DU CONVERTISSEUR

L'ajustement du convertisseur doit suffire à l'étalonnage de la carte. Une grande attention est demandée à l'utilisateur. Pour effectuer ce réglage, il faudra utiliser le potentiomètre G pour ajuster la pleine échelle d'entrée et le potentiomètre S2 pour régler l'offset d'entrée (figure 1).

Résolution 12 bits pleine échelle d'entrée $\pm 10\text{V}$

- 1/ Eteindre et allumer l'ordinateur
- 2/ La carte est configurée à la mise sous tension en mode 8 voies simples et gain 1V/V
- 3/ Programmer la voie Ø et le gain 1V/V
- 4/ Etablir sur la voie Ø la valeur du milieu de la pleine échelle d'entrée soit : 0 V pleine échelle $\pm 10\text{V}$.
- 5/ Ajuster le potentiomètre S2 pour obtenir une valeur numérique décimale oscillant entre 2047 et 2048 (\$7FF et \$800).
- 6/ Alimenter avec la plus grande valeur de la pleine échelle d'entrée moins 1LSB soit : $+10\text{V} - 4,88\text{mV} = +9,995\text{V}$ (pleine échelle $\pm 10\text{V}$).
- 7/ Ajuster le potentiomètre G pour obtenir une valeur numérique décimale oscillant entre 4094 et 4095 (\$FFF).

L'utilisateur peut vérifier à l'aide des tableaux ci-après, les valeurs obtenues pour les tensions d'entrée pleine échelle considérées, avec les résolutions (LSB) correspondantes. Si des variations subsistent reprenez l'ajustement du convertisseur.

ATTENTION : Les valeurs converties sont en binaire naturel.

Ajustement de l'amplificateur 1 :

- 1/ Connecter le voltmètre entre les points test TAGND et T1.
- 2/ Programmer la voie Ø et le gain 1000V/V.
- 3/ Ajuster le potentiomètre I1 pour obtenir une tension voisine de $\pm 1\text{mV}$.
- 4/ Programmer la voie Ø et le gain 1V/V.
- 5/ Ajuster le potentiomètre S1 pour obtenir une tension voisine de $\pm 0,2\text{mV}$.
- 6/ Recommencer l'opération à partir du 2/ pour contrôler les variations et affiner le réglage.

Ajustement de l'amplificateur 2 :

- 1/ Connecter le voltmètre entre les points test TAGND et T2.
- 2/ Programmer la voie Ø et le gain 8V/V (calibre $\pm 1,25\text{V}$).
- 3/ Ajuster le potentiomètre I2 pour obtenir une tension voisine de $\pm 1\text{mV}$.
- 4/ Programmer la voie Ø et le gain 1V/V.
- 5/ Ajuster le potentiomètre S2 pour obtenir une tension voisine de $\pm 0,2\text{mV}$.
- 6/ Recommencer l'opération à partir du 2/ pour contrôler les variations et affiner le réglage.

L'étage d'amplification est maintenant ajusté. Pour utiliser la carte avec un gain de 1000V/V, il faudra tenir compte d'un offset introduit par l'amplificateur 1 ou réétalonner celui-ci pour un gain 1000 en programmant le gain 1000 à la place du gain 100 dans l'opération 2/. Dans ce cas l'offset se rapportera au gain 100.

AJUSTEMENT DU CONVERTISSEUR

L'ajustement du convertisseur doit suffire à l'étalonnage de la carte. Une grande attention est demandée à l'utilisateur. Pour effectuer ce réglage, il faudra utiliser le potentiomètre G pour ajuster la pleine échelle d'entrée et le potentiomètre S2 pour régler l'offset d'entrée (figure 1).

Résolution 12 bits pleine échelle d'entrée $\pm 10\text{V}$

- 1/ Eteindre et allumer l'ordinateur
- 2/ La carte est configurée à la mise sous tension en mode 8 voies simples et gain 1V/V
- 3/ Programmer la voie Ø et le gain 1V/V
- 4/ Etablir sur la voie Ø la valeur du milieu de la pleine échelle d'entrée soit : 0 V pleine échelle $\pm 10\text{V}$.
- 5/ Ajuster le potentiomètre S2 pour obtenir une valeur numérique décimale oscillant entre 2047 et 2048 (\$7FF et \$800).
- 6/ Alimenter avec la plus grande valeur de la pleine échelle d'entrée moins 1LSB soit : $+10\text{V} - 4,88\text{mV} = +9,995\text{V}$ (pleine échelle $\pm 10\text{V}$).
- 7/ Ajuster le potentiomètre G pour obtenir une valeur numérique décimale oscillant entre 4094 et 4095 (\$FFF).

L'utilisateur peut vérifier à l'aide des tableaux ci-après, les valeurs obtenues pour les tensions d'entrée pleine échelle considérées, avec les résolutions (LSB) correspondantes. Si des variations subsistent reprenez l'ajustement du convertisseur.

ATTENTION : Les valeurs converties sont en binaire naturel.

Exemple : tension d'entrée <-> valeur numérique associée

PLEINES ECHELLES				VALEURS	
± 1,25V	± 2,5V	± 5V	± 10V	HEX	DEC
+1,25V	+2,5V	+5,0V	+10V	FFF	4095
+0,625V	+1,25V	+2,5V	+5V	C00	3072
0V	0V	0V	0V	800	2048
-0,625V	-1,25V	-2,5V	-5V	400	1024
-1,25V	-2,5V	-5,0V	-10V	000	0000

Exemple : tension d'entrée <-> résolution

Pleine échelle d'entrée	Résolution LSB LSB=pleine échelle/2 ⁿ (n=12bits)
±10V	4,88mV
±5V	2,44mV
±2,5V	1,22mV
±1,25V	0,61mV

V=Volts

mV=millivolts

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.

REMARQUE : En mode différentiel : le filtrage ou le diviseur de tension peuvent être utilisés. Cependant, il est impératif de positionner les mêmes valeurs sur les voies associées.

Exemple : tension d'entrée <-> valeur numérique associée

PLEINES ECHELLES				VALEURS	
± 1,25V	± 2,5V	± 5V	± 10V	HEX	DEC
+1,25V	+2,5V	+5,0V	+10V	FFF	4095
+0,625V	+1,25V	+2,5V	+5V	C00	3072
0V	0V	0V	0V	800	2048
-0,625V	-1,25V	-2,5V	-5V	400	1024
-1,25V	-2,5V	-5,0V	-10V	000	0000

Exemple : tension d'entrée <-> résolution

Pleine échelle d'entrée	Résolution LSB LSB=pleine échelle/2 ⁿ (n=12bits)
±10V	4,88mV
±5V	2,44mV
±2,5V	1,22mV
±1,25V	0,61mV

V=Volts

mV=millivolts

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.

REMARQUE : En mode différentiel : le filtrage ou le diviseur de tension peuvent être utilisés. Cependant, il est impératif de positionner les mêmes valeurs sur les voies associées.

Figure 2

Figure 2

Figure 3

Figure 3

Figure 4

Figure 4

VITESSE D'ACQUISITION

La vitesse d'acquisition est fonction :

- Du module de conversion 1,25µs MAX. Ce temps est équivalent au temps de conversion $T1=1,1\mu s$ ajouté au temps de l'échantillonneur bloqueur $T2=150ns$ max.
- Du temps de test de fin de conversion et stockage du résultat.
- Du changement de voie et/ou de gain éventuel (écriture du gain - voie future).

Ces deux derniers temps sont fonction du logiciel de programmation utilisé. En général le temps de test et de stockage est suffisant pour respecter le temps nécessaire au S/H ($T2=150ns$ maximum) pour obtenir sa nouvelle tension. L'écriture d'une voie différente augmente ce temps (temps d'établissement sur changement de voie et/ou de gain). Pour des gains inférieurs à 1ØV/V, il est souhaitable, après chaque start de conversion, de commuter la future voie (ou gain). Ce qui permet d'utiliser les 1,25µs du temps de conversion pour établir le nouveau signal à l'entrée du S/H.

Attention : dans ce cas le résultat de la conversion correspond à la voie précédente.

Le diagramme ci-dessous détaille ce fonctionnement.

ATTENTION : En cas d'utilisation de plusieurs voies à des gains différents, il faut impérativement tenir compte des temps de recouvrement de chaque amplificateur (ces temps sont donnés au chapitre "CARACTERISTIQUES TECHNIQUES").

DIAGRAMME D'UNE ACQUISITION

VITESSE D'ACQUISITION

La vitesse d'acquisition est fonction :

- Du module de conversion 1,25µs MAX. Ce temps est équivalent au temps de conversion $T1=1,1\mu s$ ajouté au temps de l'échantillonneur bloqueur $T2=150ns$ max.
- Du temps de test de fin de conversion et stockage du résultat.
- Du changement de voie et/ou de gain éventuel (écriture du gain - voie future).

Ces deux derniers temps sont fonction du logiciel de programmation utilisé. En général le temps de test et de stockage est suffisant pour respecter le temps nécessaire au S/H ($T2=150ns$ maximum) pour obtenir sa nouvelle tension. L'écriture d'une voie différente augmente ce temps (temps d'établissement sur changement de voie et/ou de gain). Pour des gains inférieurs à 1ØV/V, il est souhaitable, après chaque start de conversion, de commuter la future voie (ou gain). Ce qui permet d'utiliser les 1,25µs du temps de conversion pour établir le nouveau signal à l'entrée du S/H.

Attention : dans ce cas le résultat de la conversion correspond à la voie précédente.

Le diagramme ci-dessous détaille ce fonctionnement.

ATTENTION : En cas d'utilisation de plusieurs voies à des gains différents, il faut impérativement tenir compte des temps de recouvrement de chaque amplificateur (ces temps sont donnés au chapitre "CARACTERISTIQUES TECHNIQUES").

DIAGRAMME D'UNE ACQUISITION

EXEMPLES DE PROGRAMMATION

La carte FASTLAB se programme très facilement. Des exemples sont donnés sur la disquette accompagnant ce manuel (voir le chapitre concernant les routines).

EXEMPLES DE PROGRAMMATION

La carte FASTLAB se programme très facilement. Des exemples sont donnés sur la disquette accompagnant ce manuel (voir le chapitre concernant les routines).

FONCTIONNEMENT DE L'INTERFACE ANALOGIQUE/NUMERIQUE

Ce chapitre décrit le fonctionnement général de l'interface Analogique/Numérique ainsi que toutes les informations concernant sa programmation.

DESCRIPTION FONCTIONNELLE

Le signal source est connecté sur l'une des entrées EA0 à EA7 via le connecteur P1. Il est ensuite dirigé à l'entrée d'un multiplexeur CMOS HI 509A. L'écriture de la voie désirée dans le registre MUX adresse B+19 permet de diriger le signal à l'entrée des amplificateurs d'instrumentation. Une fois "amplifié" et adapté, le signal est ensuite dirigé vers le module comprenant le convertisseur.

Une conversion peut alors débuter. Un START B+22 permet de démarrer la conversion. La valeur numérique en binaire naturel (1,25µs plus tard) est alors disponible. Une seule lecture du registre CONV permet de récupérer la valeur convertie sur 12 bits. Une autre conversion peut alors commencer si la voie est identique.

Le module d'acquisition possède plusieurs modes de déclenchement pour initialiser une conversion.

Le déclenchement d'une conversion peut être effectué par :

- Une écriture du registre START adresse B+22
- Un front montant provenant de la sortie OUT1 du timer 0 (déclenchement par timer).
- Une impulsion positive appliquée sur la broche SYNEXT du connecteur P1.

L'utilisateur peut aussi employer différentes méthodes pour tester la fin d'une conversion, soit:

- En scrutant un état bas sur la ligne FGRDY du registre ETAT.
- Sous interruption en validant le bit VIRQCONV du registre validation. Bien entendu la gestion devra être préalablement effectuée.

Toutes ces possibilités sont détaillées dans la description des registres de la carte et par des exemples de programmation (sur disquette accompagnant ce manuel).

DESCRIPTION DES DIFFERENTS REGISTRES

LES REGISTRES DE L'INTERFACE A/N

Le tableau ci-dessous résume l'utilisation des registres employés pour l'interface Analogique/numérique ADC.

FONCTIONNEMENT DE L'INTERFACE ANALOGIQUE/NUMERIQUE

Ce chapitre décrit le fonctionnement général de l'interface Analogique/Numérique ainsi que toutes les informations concernant sa programmation.

DESCRIPTION FONCTIONNELLE

Le signal source est connecté sur l'une des entrées EA0 à EA7 via le connecteur P1. Il est ensuite dirigé à l'entrée d'un multiplexeur CMOS HI 509A. L'écriture de la voie désirée dans le registre MUX adresse B+19 permet de diriger le signal à l'entrée des amplificateurs d'instrumentation. Une fois "amplifié" et adapté, le signal est ensuite dirigé vers le module comprenant le convertisseur.

Une conversion peut alors débuter. Un START B+22 permet de démarrer la conversion. La valeur numérique en binaire naturel (1,25µs plus tard) est alors disponible. Une seule lecture du registre CONV permet de récupérer la valeur convertie sur 12 bits. Une autre conversion peut alors commencer si la voie est identique.

Le module d'acquisition possède plusieurs modes de déclenchement pour initialiser une conversion.

Le déclenchement d'une conversion peut être effectué par :

- Une écriture du registre START adresse B+22
- Un front montant provenant de la sortie OUT1 du timer 0 (déclenchement par timer).
- Une impulsion positive appliquée sur la broche SYNEXT du connecteur P1.

L'utilisateur peut aussi employer différentes méthodes pour tester la fin d'une conversion, soit:

- En scrutant un état bas sur la ligne FGRDY du registre ETAT.
- Sous interruption en validant le bit VIRQCONV du registre validation. Bien entendu la gestion devra être préalablement effectuée.

Toutes ces possibilités sont détaillées dans la description des registres de la carte et par des exemples de programmation (sur disquette accompagnant ce manuel).

DESCRIPTION DES DIFFERENTS REGISTRES

LES REGISTRES DE L'INTERFACE A/N

Le tableau ci-dessous résume l'utilisation des registres employés pour l'interface Analogique/numérique ADC.

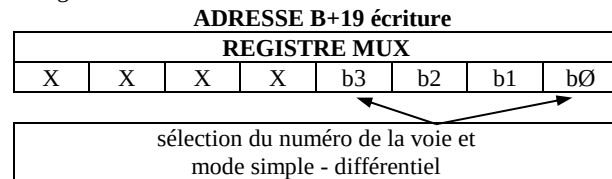
Adresse Hexadécimale	Adresse Décimale	Registres	R/W	
B+10	B+16	N.U / DAC VOIE 1 (16 bits)	/W	DAC
B+11	B+17	N.U / LATCH	/W	DAC
B+12	B+18	DAC VOIE 2 (16 bits)	R/W	DAC
B+13	B+19	RAZ VOIE/MUX	R/W	ADC
B+14	B+20	N.U / GAIN	/W	ADC
B+15	B+21	ETAT / VALID	R/W	ADC
B+16	B+22	CONV (16 bits) / START	R/W	ADC
B+17	B+23	CLEAR IRQEXT / NR INT	R/W	ADC

REGISTRE/MUX - ADRESSE B+19**-MUX** en écriture :

Le registre MUX permet de sélectionner une voie, soit 1 voie parmi 7, ou 1 parmi 4 et le mode de fonctionnement en voies simples ou différentielles.

Le tableau ci-après établit la correspondance entre la voie sélectionnée et la valeur écrite dans le registre MUX adresse B+19.

Configuration du registre :



	b3	b2	b1	b0	VOIE SELECTEE	
0	0	0	0	0	voie 0	Voies
1	0	0	0	1	voie 1	
2	0	0	1	0	voie 2	Simple
3	0	0	1	1	voie 3	
4	0	1	0	0	voie 4	8
5	0	1	0	1	voie 5	
6	0	1	1	0	voie 6	Voies
7	0	1	1	1	voie 7	
8	1	0	0	0	voie 0-4	Voies
9	1	0	0	1	voie 1-5	
10	1	0	1	0	voie 2-6	Différen-
11	1	0	1	1	voie 3-7	tielles
12	1	1	0	0	GNDAD	
13	1	1	0	1	GNDAD	4
14	1	1	1	0	GNDAD	
15	1	1	1	1	GNDAD	Voies

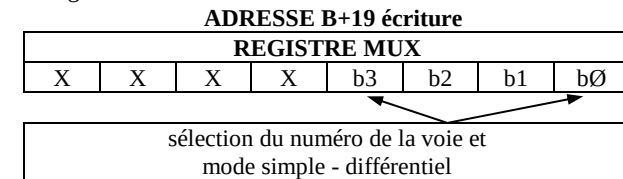
Adresse Hexadécimale	Adresse Décimale	Registres	R/W	
B+10	B+16	N.U / DAC VOIE 1 (16 bits)	/W	DAC
B+11	B+17	N.U / LATCH	/W	DAC
B+12	B+18	DAC VOIE 2 (16 bits)	R/W	DAC
B+13	B+19	RAZ VOIE/MUX	R/W	ADC
B+14	B+20	N.U / GAIN	/W	ADC
B+15	B+21	ETAT / VALID	R/W	ADC
B+16	B+22	CONV (16 bits) / START	R/W	ADC
B+17	B+23	CLEAR IRQEXT / NR INT	R/W	ADC

REGISTRE/MUX - ADRESSE B+19**-MUX** en écriture :

Le registre MUX permet de sélectionner une voie, soit 1 voie parmi 7, ou 1 parmi 4 et le mode de fonctionnement en voies simples ou différentielles.

Le tableau ci-après établit la correspondance entre la voie sélectionnée et la valeur écrite dans le registre MUX adresse B+19.

Configuration du registre :



	b3	b2	b1	b0	VOIE SELECTEE	
0	0	0	0	0	voie 0	Voies
1	0	0	0	1	voie 1	
2	0	0	1	0	voie 2	Simple
3	0	0	1	1	voie 3	
4	0	1	0	0	voie 4	8
5	0	1	0	1	voie 5	
6	0	1	1	0	voie 6	Voies
7	0	1	1	1	voie 7	
8	1	0	0	0	voie 0-4	Voies
9	1	0	0	1	voie 1-5	
10	1	0	1	0	voie 2-6	Différen-
11	1	0	1	1	voie 3-7	tielles
12	1	1	0	0	GNDAD	
13	1	1	0	1	GNDAD	4
14	1	1	1	0	GNDAD	
15	1	1	1	1	GNDAD	Voies

REGISTRE/GAIN - ADRESSE B+20**-GAIN** en écriture :

Le registre GAIN permet de sélectionner le gain de l'amplificateur d'instrumentation 1 parmi 16.

Configuration du registre :

ADRESSE B+20 écriture							
REGISTRE GAIN							
X	X	X	X	b3	b2	b1	b0
b3	b2	b1	b0	GAIN			
0	0	0	0	1 V/V			
0	0	0	1	2 V/V			
0	0	1	0	4 V/V			
0	0	1	1	8 V/V			
0	1	0	0	10 V/V			
0	1	0	1	20 V/V			
0	1	1	0	40 V/V			
0	1	1	1	80 V/V			
1	0	0	0	100 V/V			
1	0	0	1	200 V/V			
1	0	1	0	400 V/V			
1	0	1	1	800 V/V			
1	1	0	0	1000 V/V			
1	1	0	1	2000 V/V			
1	1	1	0	4000 V/V			
1	1	1	1	8000 V/V			

CHOIX DE LA CONFIGURATION MODE SIMPLE OU DIFFERENTIEL

- a/ Le mode 8 voies simples est utilisé pour des niveaux d'entrée élevés de $\pm 10V$ et $\pm 5V$ pleine échelle. Dans ce mode, le bruit et l'impédance du câble de liaison sont supposés négligeables.
- b/ Le mode 4 voies différentielles est utilisé pour des niveaux d'entrée faibles, inférieurs à 5V pleine échelle. Dans ce mode le bruit et l'impédance du câble sont diminués par la réjection du mode commun de l'amplificateur d'instrumentation.

Toutes les voies en service doivent être connectées à leur source. **Les voies non utilisées seront reliées à la masse de la carte. Aucune entrée ne doit être flottante.**

REGISTRE ETAT/VALID - ADRESSE B+21**-ETAT** en lecture :

Ce registre permet de lire l'état de 4 flags utilisés pour gérer le module de conversion A/N et les interruptions.

REGISTRE/GAIN - ADRESSE B+20**-GAIN** en écriture :

Le registre GAIN permet de sélectionner le gain de l'amplificateur d'instrumentation 1 parmi 16.

Configuration du registre :

ADRESSE B+20 écriture							
REGISTRE GAIN							
X	X	X	X	b3	b2	b1	b0
b3	b2	b1	b0	GAIN			
0	0	0	0	1 V/V			
0	0	0	1	2 V/V			
0	0	1	0	4 V/V			
0	0	1	1	8 V/V			
0	1	0	0	10 V/V			
0	1	0	1	20 V/V			
0	1	1	0	40 V/V			
0	1	1	1	80 V/V			
1	0	0	0	100 V/V			
1	0	0	1	200 V/V			
1	0	1	0	400 V/V			
1	0	1	1	800 V/V			
1	1	0	0	1000 V/V			
1	1	0	1	2000 V/V			
1	1	1	0	4000 V/V			
1	1	1	1	8000 V/V			

CHOIX DE LA CONFIGURATION MODE SIMPLE OU DIFFERENTIEL

- a/ Le mode 8 voies simples est utilisé pour des niveaux d'entrée élevés de $\pm 10V$ et $\pm 5V$ pleine échelle. Dans ce mode, le bruit et l'impédance du câble de liaison sont supposés négligeables.
- b/ Le mode 4 voies différentielles est utilisé pour des niveaux d'entrée faibles, inférieurs à 5V pleine échelle. Dans ce mode le bruit et l'impédance du câble sont diminués par la réjection du mode commun de l'amplificateur d'instrumentation.

Toutes les voies en service doivent être connectées à leur source. **Les voies non utilisées seront reliées à la masse de la carte. Aucune entrée ne doit être flottante.**

REGISTRE ETAT/VALID - ADRESSE B+21**-ETAT** en lecture :

Ce registre permet de lire l'état de 4 flags utilisés pour gérer le module de conversion A/N et les interruptions.

DESCRIPTIONS DU REGISTRE

ADRESSE B+21 lecture							
REGISTRE ETAT							
X	X	X	b4	b3	b2	X	b0

FGIRQPC	FGO1TIM1	FGIRQEXT	FGRDY
bx = 1 $\Rightarrow$ option validée bx = 0 $\Rightarrow$ option dévalidée			

Bit0 FGRDY : (actif à 0)

Ce bit est positionné à 0 à chaque fin de conversion. Il permet de déterminer aussi sous interruption l'origine de celle-ci (dans ce cas en fin de conversion). Il est remis à un par une lecture sur 16 bits de la donnée convertie. Cette remise à un est obligatoire dans la routine de gestion d'interruption pour acquitter celle-ci.

Bit2 FGIRQEXT :

Ce bit est positionné à 1 à chaque front montant de la ligne IRQEXT (P1). Cette ligne est dédiée à la génération d'interruption externe. Un front montant détecté sur cette ligne génère une interruption au PC. Ce bit permet de déterminer l'origine de celle-ci. Il est remis à zéro (acquiescement) par une lecture du registre en B+23 CLEAR IRQEXT.

Bit3 FGO1TIM1 :

Ce bit reflète l'état de la sortie OUT du timer 1. Ce timer est câblé pour générer des interruptions au PC. Il doit être programmé en mode 0, l'interruption se fait sur un front haut de cette sortie qui ne repassera à l'état bas que si l'on recharge le décompte. Ce bit à 1 montre donc que l'interruption au PC provient du timer 1.

Bit4 FGIRQPC :

Ce bit reflète la ligne d'interruption principale de la carte. Il permet à l'utilisateur de détecter quelle carte a généré une interruption dans l'ordinateur. Actif à 1 il indique donc une interruption en cours.

-VALID en écriture :

Ce registre en écriture permet de valider 5 options, principalement propres au module de conversion analogique/numérique. Il est remis à zéro à la mise sous tension, options inactives. La configuration et le détail de chaque bit sont décrits ci-après :

ADRESSE B+21 écriture							
REGISTRE DE VALIDATION							
X	X	X	b4	b3	b2	b1	b0

VIRQTIM1	VIRQEXT	VIRQCONV	VSYNEXT	VCONVT
bx = 1 $\Rightarrow$ état actif bx = 0 $\Rightarrow$ état inactif				

DESCRIPTIONS DU REGISTRE

ADRESSE B+21 lecture							
REGISTRE ETAT							
X	X	X	b4	b3	b2	X	b0

FGIRQPC	FGO1TIM1	FGIRQEXT	FGRDY
bx = 1 $\Rightarrow$ option validée bx = 0 $\Rightarrow$ option dévalidée			

Bit0 FGRDY : (actif à 0)

Ce bit est positionné à 0 à chaque fin de conversion. Il permet de déterminer aussi sous interruption l'origine de celle-ci (dans ce cas en fin de conversion). Il est remis à un par une lecture sur 16 bits de la donnée convertie. Cette remise à un est obligatoire dans la routine de gestion d'interruption pour acquitter celle-ci.

Bit2 FGIRQEXT :

Ce bit est positionné à 1 à chaque front montant de la ligne IRQEXT (P1). Cette ligne est dédiée à la génération d'interruption externe. Un front montant détecté sur cette ligne génère une interruption au PC. Ce bit permet de déterminer l'origine de celle-ci. Il est remis à zéro (acquiescement) par une lecture du registre en B+23 CLEAR IRQEXT.

Bit3 FGO1TIM1 :

Ce bit reflète l'état de la sortie OUT du timer 1. Ce timer est câblé pour générer des interruptions au PC. Il doit être programmé en mode 0, l'interruption se fait sur un front haut de cette sortie qui ne repassera à l'état bas que si l'on recharge le décompte. Ce bit à 1 montre donc que l'interruption au PC provient du timer 1.

Bit4 FGIRQPC :

Ce bit reflète la ligne d'interruption principale de la carte. Il permet à l'utilisateur de détecter quelle carte a généré une interruption dans l'ordinateur. Actif à 1 il indique donc une interruption en cours.

-VALID en écriture :

Ce registre en écriture permet de valider 5 options, principalement propres au module de conversion analogique/numérique. Il est remis à zéro à la mise sous tension, options inactives. La configuration et le détail de chaque bit sont décrits ci-après :

ADRESSE B+21 écriture							
REGISTRE DE VALIDATION							
X	X	X	b4	b3	b2	b1	b0

VIRQTIM1	VIRQEXT	VIRQCONV	VSYNEXT	VCONVT
bx = 1 $\Rightarrow$ état actif bx = 0 $\Rightarrow$ état inactif				

Bit0 VCONVT :

Lorsqu'il est positionné à 1, ce bit valide le déclenchement d'une conversion sur une impulsion positive provenant de la sortie OUT1 du timer 0.

Bit1 VSYNEXT :

Lorsqu'il est positionné à 1, ce bit valide le déclenchement d'une conversion sur un signal externe relié sur la broche SYNEXT du connecteur P1. Le déclenchement s'effectue sur une impulsion positive de niveau logique TTL. Pour être détectée, cette impulsion doit être obligatoirement supérieure à 1µs.

Bit2 VIRQCONV :

Lorsqu'il est positionné à 1, ce bit valide la génération d'une interruption après chaque fin de conversion. L'état du bit FGRDY du registre ETAT B+21 est utilisé pour en définir l'origine. Une lecture de la valeur convertie (registre CONV) remet à 1 FGRDY (acquittement).

Bit3 VIRQEXT :

Lorsqu'il est positionné à 1, ce bit valide la génération d'une interruption sur un signal externe relié sur la broche IRQEXT du connecteur P1. Le déclenchement s'effectue sur une impulsion positive de niveau logique TTL.

Bit4 VIRQTIM1 :

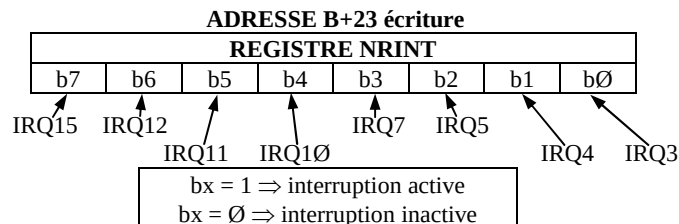
Lorsqu'il est positionné à 1, ce bit valide la génération d'une interruption calibrée à l'aide de la base de temps fournie par la sortie OUT1 du timer 1. Cette possibilité permet de réaliser un système multitâche.

REGISTRE CLEAR IRQEXT/NR INT - ADRESSE B+23**-CLEAR IRQEXT** en lecture :

Ce registre permet, par une simple lecture d'acquitter une interruption survenue sur la ligne IRQEXT du connecteur P1. Le flag FGIRQEXT du registre ETAT b+21 est remis à 0.

-NR INT en écriture :

Ce registre permet de diriger les différentes interruptions possibles sur la carte FASTLAB vers une des huit interruptions disponibles. Un seul des 8 bits doit être obligatoirement positionné à 1 (actif).

**Bit0 VCONVT :**

Lorsqu'il est positionné à 1, ce bit valide le déclenchement d'une conversion sur une impulsion positive provenant de la sortie OUT1 du timer 0.

Bit1 VSYNEXT :

Lorsqu'il est positionné à 1, ce bit valide le déclenchement d'une conversion sur un signal externe relié sur la broche SYNEXT du connecteur P1. Le déclenchement s'effectue sur une impulsion positive de niveau logique TTL. Pour être détectée, cette impulsion doit être obligatoirement supérieure à 1µs.

Bit2 VIRQCONV :

Lorsqu'il est positionné à 1, ce bit valide la génération d'une interruption après chaque fin de conversion. L'état du bit FGRDY du registre ETAT B+21 est utilisé pour en définir l'origine. Une lecture de la valeur convertie (registre CONV) remet à 1 FGRDY (acquittement).

Bit3 VIRQEXT :

Lorsqu'il est positionné à 1, ce bit valide la génération d'une interruption sur un signal externe relié sur la broche IRQEXT du connecteur P1. Le déclenchement s'effectue sur une impulsion positive de niveau logique TTL.

Bit4 VIRQTIM1 :

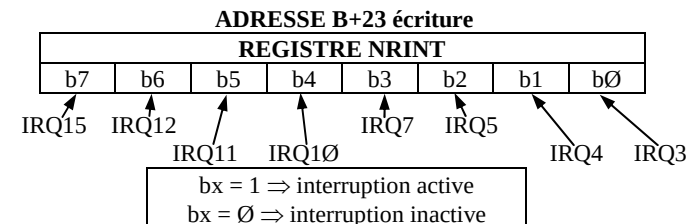
Lorsqu'il est positionné à 1, ce bit valide la génération d'une interruption calibrée à l'aide de la base de temps fournie par la sortie OUT1 du timer 1. Cette possibilité permet de réaliser un système multitâche.

REGISTRE CLEAR IRQEXT/NR INT - ADRESSE B+23**-CLEAR IRQEXT** en lecture :

Ce registre permet, par une simple lecture d'acquitter une interruption survenue sur la ligne IRQEXT du connecteur P1. Le flag FGIRQEXT du registre ETAT b+21 est remis à 0.

-NR INT en écriture :

Ce registre permet de diriger les différentes interruptions possibles sur la carte FASTLAB vers une des huit interruptions disponibles. Un seul des 8 bits doit être obligatoirement positionné à 1 (actif).



Un seul niveau d'interruption par carte est permis. L'utilisateur veillera à ne pas créer d'incompatibilité avec une autre carte. Plusieurs cartes peuvent avoir le même niveau d'interruption à condition que l'utilisateur crée un logiciel qui prenne en compte les conflits éventuels.

Les lignes utilisées sur la carte FASTLAB et leur utilisation courante sur un ordinateur PC-AT sont les suivantes :

IRQ3 : COM2 (communication asynchrone port2)
 IRQ4 : COM1 (communication asynchrone port1)
 IRQ5 : interface parallèle port 2
 IRQ7 : interface parallèle port 1
 IRQ10 : libre
 IRQ11 : libre
 IRQ12 : libre
 IRQ15 : libre

Les lignes notées libres par IBM peuvent être utilisées si aucune carte ne les utilise. (se référer au manuel des cartes utilisées).

Une demande d'interruption est activée lorsqu'un front montant et un état haut logique est détecté. Cette ligne sera remise à zéro dans la routine de gestion de l'interruption.

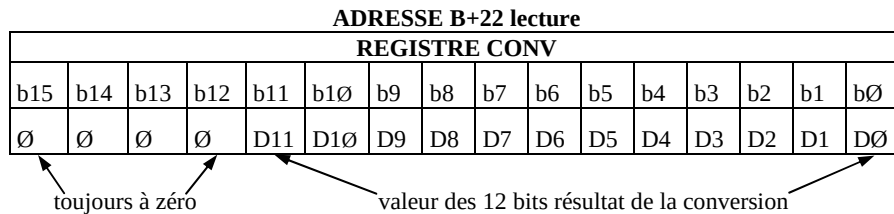
REGISTRE CONV/START - ADRESSE B+22

-START en écriture :

Ce registre permet par une simple écriture, de lancer une conversion start logiciel.

-CONV en lecture :

Une simple lecture de ce registre permet de lire le résultat d'une conversion sur un mot de 16 bits. Les figures ci-dessous représentent le format des données lues, noter que les 4 bits de poids forts sont toujours à zéro.



ATTENTION : Les valeurs converties sont en binaire naturel.

Un seul niveau d'interruption par carte est permis. L'utilisateur veillera à ne pas créer d'incompatibilité avec une autre carte. Plusieurs cartes peuvent avoir le même niveau d'interruption à condition que l'utilisateur crée un logiciel qui prenne en compte les conflits éventuels.

Les lignes utilisées sur la carte FASTLAB et leur utilisation courante sur un ordinateur PC-AT sont les suivantes :

IRQ3 : COM2 (communication asynchrone port2)
 IRQ4 : COM1 (communication asynchrone port1)
 IRQ5 : interface parallèle port 2
 IRQ7 : interface parallèle port 1
 IRQ10 : libre
 IRQ11 : libre
 IRQ12 : libre
 IRQ15 : libre

Les lignes notées libres par IBM peuvent être utilisées si aucune carte ne les utilise. (se référer au manuel des cartes utilisées).

Une demande d'interruption est activée lorsqu'un front montant et un état haut logique est détecté. Cette ligne sera remise à zéro dans la routine de gestion de l'interruption.

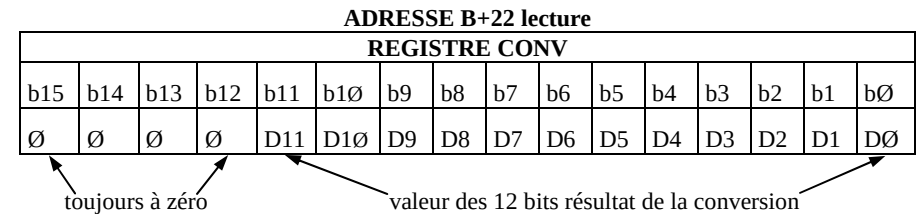
REGISTRE CONV/START - ADRESSE B+22

-START en écriture :

Ce registre permet par une simple écriture, de lancer une conversion start logiciel.

-CONV en lecture :

Une simple lecture de ce registre permet de lire le résultat d'une conversion sur un mot de 16 bits. Les figures ci-dessous représentent le format des données lues, noter que les 4 bits de poids forts sont toujours à zéro.



ATTENTION : Les valeurs converties sont en binaire naturel.

FONCTIONNEMENT DE L'INTERFACE NUMERIQUE/ANALOGIQUE

Ce chapitre décrit le fonctionnement général de l'interface Numérique/Analogique DAC ainsi que toutes les informations concernant sa programmation.

DESCRIPTION FONCTIONNELLE

Le composant utilisé est un double convertisseur N/A de 12 bits de résolution incluant une référence de tension et des amplificateurs de sortie. Le DAC1 est couplé avec le DAC2. La conversion N/A occupe 3 adresses logiques sur la carte. Au DAC X correspond respectivement la sortie analogique notée SAX. Chaque DAC possède 1 adresse pour sa programmation. La conversion Numérique-Analogique sur chaque voie est effectuée en 2 temps : un registre de 12 bits permet d'écrire la valeur désirée, la validation (mise à jour) est réalisée par une opération d'écriture sur une adresse (registre LATCH). Cette opération met à jour les deux voies couplées et les valide simultanément par une simple écriture. Si les deux voies sont à modifier, une seule écriture permet la mise à jour.

Le DAC possède en plus un registre de 8 bits en lecture (registre RAZ DAC) permettant d'effectuer un reset général des deux voies analogiques. Cette opération est réalisée automatiquement à la mise sous tension. Les DAC sont positionnées à la valeur bipolaire $\emptyset V$ (soit une valeur numérique \$800\$). Cette fonction est intéressante pour une initialisation ou une recalibration du système.

Une fois validé, le DAC délivre une tension proportionnelle à la donnée programmée. Deux amplificateurs opérationnels de précision permettent d'adapter et de protéger la sortie en tension. Deux potentiomètres par voie permettent le réglage de la tension de sortie GX et OX (figure 1).

ADRESSAGE DU CONVERTISSEUR N/A

Le tableau ci-dessous rappelle les différentes adresses et registres associés du module analogique.

Adresse Hexadécimale	Adresse Décimale	Registres	R/W	
B+10	B+16	N.U/DAC VOIE 1 (16 bits)	/W	DAC
B+11	B+17	N.U/LATCH	/W	DAC
B+12	B+18	RAZ DAC/DAC VOIE 2 (16 bits)	R/W	DAC
B+13	B+19	RAZ VOIE/MUX	R/W	ADC
B+14	B+20	N.U/GAIN	/W	ADC
B+15	B+21	ETAT/VALID	R/W	ADC
B+16	B+22	CONV (16 bits)/START	R/W	ADC
B+17	B+23	CLEAR IRQEXT/NR INT	R/W	ADC

FONCTIONNEMENT DE L'INTERFACE NUMERIQUE/ANALOGIQUE

Ce chapitre décrit le fonctionnement général de l'interface Numérique/Analogique DAC ainsi que toutes les informations concernant sa programmation.

DESCRIPTION FONCTIONNELLE

Le composant utilisé est un double convertisseur N/A de 12 bits de résolution incluant une référence de tension et des amplificateurs de sortie. Le DAC1 est couplé avec le DAC2. La conversion N/A occupe 3 adresses logiques sur la carte. Au DAC X correspond respectivement la sortie analogique notée SAX. Chaque DAC possède 1 adresse pour sa programmation. La conversion Numérique-Analogique sur chaque voie est effectuée en 2 temps : un registre de 12 bits permet d'écrire la valeur désirée, la validation (mise à jour) est réalisée par une opération d'écriture sur une adresse (registre LATCH). Cette opération met à jour les deux voies couplées et les valide simultanément par une simple écriture. Si les deux voies sont à modifier, une seule écriture permet la mise à jour.

Le DAC possède en plus un registre de 8 bits en lecture (registre RAZ DAC) permettant d'effectuer un reset général des deux voies analogiques. Cette opération est réalisée automatiquement à la mise sous tension. Les DAC sont positionnées à la valeur bipolaire $\emptyset V$ (soit une valeur numérique \$800\$). Cette fonction est intéressante pour une initialisation ou une recalibration du système.

Une fois validé, le DAC délivre une tension proportionnelle à la donnée programmée. Deux amplificateurs opérationnels de précision permettent d'adapter et de protéger la sortie en tension. Deux potentiomètres par voie permettent le réglage de la tension de sortie GX et OX (figure 1).

ADRESSAGE DU CONVERTISSEUR N/A

Le tableau ci-dessous rappelle les différentes adresses et registres associés du module analogique.

Adresse Hexadécimale	Adresse Décimale	Registres	R/W	
B+10	B+16	N.U/DAC VOIE 1 (16 bits)	/W	DAC
B+11	B+17	N.U/LATCH	/W	DAC
B+12	B+18	RAZ DAC/DAC VOIE 2 (16 bits)	R/W	DAC
B+13	B+19	RAZ VOIE/MUX	R/W	ADC
B+14	B+20	N.U/GAIN	/W	ADC
B+15	B+21	ETAT/VALID	R/W	ADC
B+16	B+22	CONV (16 bits)/START	R/W	ADC
B+17	B+23	CLEAR IRQEXT/NR INT	R/W	ADC

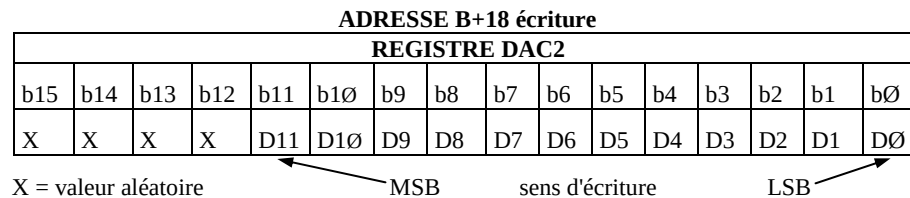
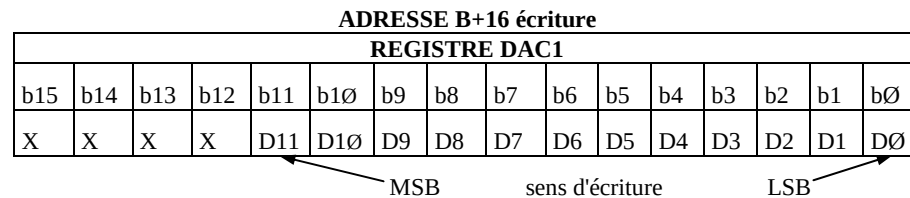
DESCRIPTION DES DIFFERENTS REGISTRES

REGISTRE DAC VOIE 1 ET 2 EN ECRITURE - ADRESSE B+16 ET B+18

Ces deux registres permettent d'écrire la valeur numérique à convertir dans la voie désirée. Cette opération est réalisée par une simple écriture 16 bits sur la carte FASTLAB.

Le format des données des registres DAC1 et 2 est décrit ci-après.

Les valeurs sont écrites en binaire naturel.



REGISTRE LATCH EN ECRITURE - ADRESSE B+17

Ce registre verrouille la valeur écrite dans les registres DAC 1 et 2 vers la sortie tension. Cette opération est effectuée par une simple écriture, elle est obligatoire.

REGISTRE RAZ DAC EN LECTURE - ADRESSE B+18

Ce registre permet par une simple lecture une remise à zéro asynchrone des deux voies de sortie analogique. Les registres DAC sont positionnés à la valeur bipolaire \$800. Cette valeur correspond à 0V en mode bipolaire.

La fonction réalisée est identique à la mise sous tension.

REGLAGES GENERAUX

Pour chaque sortie SAX correspondent deux potentiomètres notés respectivement :

GX = gain sortie X
OX = Offset sortie X

Se référer à l'implantation des composants et des options disponibles sur la carte (figure 1). Lors de l'ajustement, pour la programmation de chaque voie, utiliser le programme FASTEST fourni sur la disquette accompagnant ce manuel.

Réglage de l'offset de sortie

Le potentiomètre d'offset OX permet le réglage du zéro.

Pour ce faire : Ecrire dans le convertisseur la valeur -10V et régler le potentiomètre X de façon à obtenir la tension -10V en sortie.

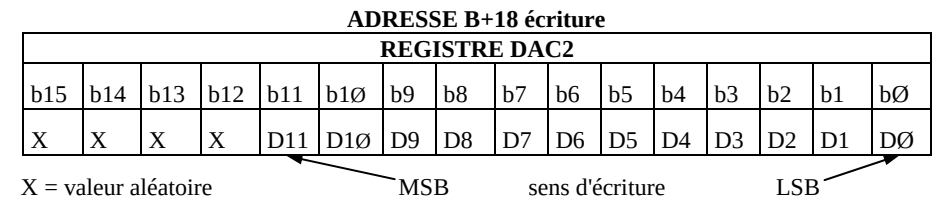
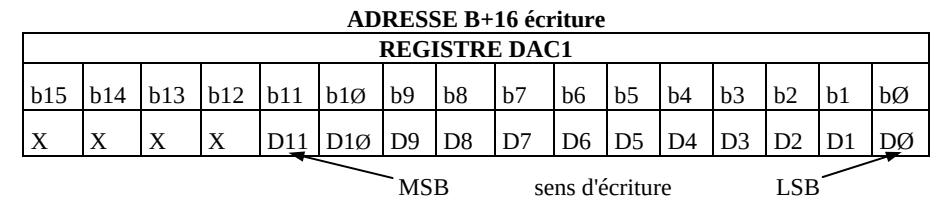
DESCRIPTION DES DIFFERENTS REGISTRES

REGISTRE DAC VOIE 1 ET 2 EN ECRITURE - ADRESSE B+16 ET B+18

Ces deux registres permettent d'écrire la valeur numérique à convertir dans la voie désirée. Cette opération est réalisée par une simple écriture 16 bits sur la carte FASTLAB.

Le format des données des registres DAC1 et 2 est décrit ci-après.

Les valeurs sont écrites en binaire naturel.



REGISTRE LATCH EN ECRITURE - ADRESSE B+17

Ce registre verrouille la valeur écrite dans les registres DAC 1 et 2 vers la sortie tension. Cette opération est effectuée par une simple écriture, elle est obligatoire.

REGISTRE RAZ DAC EN LECTURE - ADRESSE B+18

Ce registre permet par une simple lecture une remise à zéro asynchrone des deux voies de sortie analogique. Les registres DAC sont positionnés à la valeur bipolaire \$800. Cette valeur correspond à 0V en mode bipolaire.

La fonction réalisée est identique à la mise sous tension.

REGLAGES GENERAUX

Pour chaque sortie SAX correspondent deux potentiomètres notés respectivement :

GX = gain sortie X
OX = Offset sortie X

Se référer à l'implantation des composants et des options disponibles sur la carte (figure 1). Lors de l'ajustement, pour la programmation de chaque voie, utiliser le programme FASTEST fourni sur la disquette accompagnant ce manuel.

Réglage de l'offset de sortie

Le potentiomètre d'offset OX permet le réglage du zéro.

Pour ce faire : Ecrire dans le convertisseur la valeur -10V et régler le potentiomètre X de façon à obtenir la tension -10V en sortie.

Réglage de la pleine échelle de sortie

Le potentiomètre de gain GX permet le réglage de la tension pleine échelle de sortie.

Pour ce faire :

- 1/ Ecrire dans le convertisseur la valeur maximale 10V.
- 2/ A l'aide d'un voltmètre, mesurer la tension en sortie sur la broche du connecteur P1 correspondante à la voie choisie
- 3/ Régler le potentiomètre GX de façon à obtenir la tension $V_{max} = 10V$

En général, le réglage de l'offset s'effectue avant le réglage de la pleine échelle. L'utilisateur peut ajuster et vérifier son réglage à l'aide des deux tableaux ci-après. Si des variations subsistent, reprenez l'ajustement du convertisseur.

VALEUR BINAIRE DANS LE DAC X			VALEUR DECIMALE	TENSION DE SORTIE SAX
MSB		LSB		
1111	1111	1111	4095	$+V_{max} * (2047/2048)$
1000	0000	0001	2049	$+V_{max} * (1/2048)$
1000	0000	0000	2048	0V
0111	1111	1111	2047	$-V_{max} * (1/2048)$
0000	0000	0000	0000	$-V_{max} * (2048/2048) = -V_{max}$

OPTION SORTIE EN COURANT /SCF

cette option permet d'utiliser les deux voies de conversion numérique/analogique en sortie courant.

La tension de sortie 0 → 10V est convertie respectivement en courant 4 → 20mA.

Caractéristiques :

- courant de sortie : 4 → 20mA
- initial offset en courant : 0,2% max sur la pleine échelle
- erreur de gain initial : 0,6% max sur la pleine échelle
- temps d'établissement : 15µs à 0,01% de la pleine échelle

Si cela est nécessaire les réglages seront effectués à l'aide des potentiomètres GX et OX de la même manière que pour un réglage en tension mais en mesurant un courant.

ATTENTION : veuillez à positionner une charge (500 ohms) entre l'ampèremètre et la masse GNDDA lors des mesures.

La programmation des tensions négatives est interdite.

PROGRAMMATION DE LA CARTE

L'ordre des opérations est donc le suivant, nous prendrons comme exemple la voie 1.

- 1/ Ecriture des 12 bits dans le registre B+16
- 2/ Validation de la donnée par une écriture du registre LATCH en B+17.

Des routines et des exemples de programmation sont fournis sur la disquette accompagnant ce manuel.

Réglage de la pleine échelle de sortie

Le potentiomètre de gain GX permet le réglage de la tension pleine échelle de sortie.

Pour ce faire :

- 1/ Ecrire dans le convertisseur la valeur maximale 10V.
- 2/ A l'aide d'un voltmètre, mesurer la tension en sortie sur la broche du connecteur P1 correspondante à la voie choisie
- 3/ Régler le potentiomètre GX de façon à obtenir la tension $V_{max} = 10V$

En général, le réglage de l'offset s'effectue avant le réglage de la pleine échelle. L'utilisateur peut ajuster et vérifier son réglage à l'aide des deux tableaux ci-après. Si des variations subsistent, reprenez l'ajustement du convertisseur.

VALEUR BINAIRE DANS LE DAC X			VALEUR DECIMALE	TENSION DE SORTIE SAX
MSB		LSB		
1111	1111	1111	4095	$+V_{max} * (2047/2048)$
1000	0000	0001	2049	$+V_{max} * (1/2048)$
1000	0000	0000	2048	0V
0111	1111	1111	2047	$-V_{max} * (1/2048)$
0000	0000	0000	0000	$-V_{max} * (2048/2048) = -V_{max}$

OPTION SORTIE EN COURANT /SCF

cette option permet d'utiliser les deux voies de conversion numérique/analogique en sortie courant.

La tension de sortie 0 → 10V est convertie respectivement en courant 4 → 20mA.

Caractéristiques :

- courant de sortie : 4 → 20mA
- initial offset en courant : 0,2% max sur la pleine échelle
- erreur de gain initial : 0,6% max sur la pleine échelle
- temps d'établissement : 15µs à 0,01% de la pleine échelle

Si cela est nécessaire les réglages seront effectués à l'aide des potentiomètres GX et OX de la même manière que pour un réglage en tension mais en mesurant un courant.

ATTENTION : veuillez à positionner une charge (500 ohms) entre l'ampèremètre et la masse GNDDA lors des mesures.

La programmation des tensions négatives est interdite.

PROGRAMMATION DE LA CARTE

L'ordre des opérations est donc le suivant, nous prendrons comme exemple la voie 1.

- 1/ Ecriture des 12 bits dans le registre B+16
- 2/ Validation de la donnée par une écriture du registre LATCH en B+17.

Des routines et des exemples de programmation sont fournis sur la disquette accompagnant ce manuel.

L'INTERFACE LOGIQUE

L'interface logique se compose de 2 timers 82C54 et de 2 interfaces d'entrée/sortie PIO 82C55.

Le tableau ci-dessous résume l'utilisation des registres pour l'interface Logique.

Adresse Hexadécimale	Adresse Décimale	Registres	R/W	
B+0	B+0	PIO0 - PORTA	R/W	PIO0
B+1	B+1	PIO0 - PORTB	R/W	PIO0
B+2	B+2	PIO0 - PORTC	R/W	PIO0
B+3	B+3	PIO0 - CONTROL WORD	R/W	PIO0
B+4	B+4	PIO1 - PORTA	R/W	PIO1
B+5	B+5	PIO1 - PORTB	R/W	PIO1
B+6	B+6	PIO1 - PORTC	R/W	PIO1
B+7	B+7	PIO1 - CONTROL WORD	R/W	PIO1
B+8	B+8	TIMER0 - COMPTEUR0	R/W	TIMER0
B+9	B+9	TIMER0 - COMPTEUR1	R/W	TIMER0
B+A	B+10	TIMER0 - COMPTEUR2	R/W	TIMER0
B+B	B+11	TIMER0 - CONTROL WORD	R/W	TIMER0
B+C	B+12	TIMER1 - COMPTEUR0	R/W	TIMER1
B+D	B+13	TIMER1 - COMPTEUR1	R/W	TIMER1
B+E	B+14	TIMER1 - COMPTEUR2	R/W	TIMER1
B+F	B+15	TIMER1 - CONTROL WORD	R/W	TIMER1

CONNECTEUR LOGIQUE P1 ET P2

La figure ci-après représente le brochage du connecteur interne à la carte FASTLAB. Ce connecteur supporte les ports B et C du PIO1 et toutes les lignes de commandes du compteur 2 des timers 0 et 1 8254 et l'alimentation +5V directe du PC.

ATTENTION : Le port A du PIO1 est utilisé pour la validation des GATES des timers 0 et 1. Il n'est donc pas disponible.

L'INTERFACE LOGIQUE

L'interface logique se compose de 2 timers 82C54 et de 2 interfaces d'entrée/sortie PIO 82C55.

Le tableau ci-dessous résume l'utilisation des registres pour l'interface Logique.

Adresse Hexadécimale	Adresse Décimale	Registres	R/W	
B+0	B+0	PIO0 - PORTA	R/W	PIO0
B+1	B+1	PIO0 - PORTB	R/W	PIO0
B+2	B+2	PIO0 - PORTC	R/W	PIO0
B+3	B+3	PIO0 - CONTROL WORD	R/W	PIO0
B+4	B+4	PIO1 - PORTA	R/W	PIO1
B+5	B+5	PIO1 - PORTB	R/W	PIO1
B+6	B+6	PIO1 - PORTC	R/W	PIO1
B+7	B+7	PIO1 - CONTROL WORD	R/W	PIO1
B+8	B+8	TIMER0 - COMPTEUR0	R/W	TIMER0
B+9	B+9	TIMER0 - COMPTEUR1	R/W	TIMER0
B+A	B+10	TIMER0 - COMPTEUR2	R/W	TIMER0
B+B	B+11	TIMER0 - CONTROL WORD	R/W	TIMER0
B+C	B+12	TIMER1 - COMPTEUR0	R/W	TIMER1
B+D	B+13	TIMER1 - COMPTEUR1	R/W	TIMER1
B+E	B+14	TIMER1 - COMPTEUR2	R/W	TIMER1
B+F	B+15	TIMER1 - CONTROL WORD	R/W	TIMER1

CONNECTEUR LOGIQUE P1 ET P2

La figure ci-après représente le brochage du connecteur interne à la carte FASTLAB. Ce connecteur supporte les ports B et C du PIO1 et toutes les lignes de commandes du compteur 2 des timers 0 et 1 8254 et l'alimentation +5V directe du PC.

ATTENTION : Le port A du PIO1 est utilisé pour la validation des GATES des timers 0 et 1. Il n'est donc pas disponible.

FONCTIONNEMENT DES COUPLEURS D'ENTREE/SORTIE PIO 8255

Les PIO 8255 réalisent l'interface entre l'ordinateur et le monde extérieur. Il s'agit d'un système d'entrées sorties programmable. Ils se composent chacun de 24 entrées/sorties pouvant être programmées en 2 groupes de 12 bits et utilisées en 3 modes d'opérations.

ORGANISATION DU COMPOSANT

Le PIO est composé de 3 ports (A, B, C) d'entrées/sorties. Il possède un mot de contrôle programmable en trois modes :

- Dans le mode 0, chaque port peut être programmé en E/S.
- Dans le mode 1, les ports A et B sont programmables en E/S et possèdent comme lignes de contrôle et d'interruption le port C (bits 0 à 3 pour le port A et les bits 4 à 7 pour le port B).
- Dans le mode 2, le port A est bidirectionnel et possède comme lignes de contrôles et d'interruption les 3 à 7 du port C. (Les bits 0 à 2 étant en E/S).

DEFINITION DU MOT DE CONTROLE

Le mot de contrôle est un mot de huit bits dont chaque combinaison correspond à une utilisation des ports.

Les modes pour les ports A et B peuvent être séparément définis en ayant les bits du port C comme lignes de contrôle selon le mode.

Le tableau ci-dessous montre la correspondance des bits du mot de contrôle :

D7	D6	D5	D4	D3	D2	D1	D0	
								Port C (LOWER) 1 = INPUT 0 = OUTPUT
								Port B 1 = INPUT 0 = OUTPUT
								MODE SELECTION 0 = MODE 0 1 = MODE 1
								Port C (UPPER) 1 = INPUT 0 = OUTPUT
								Port A 1 = INPUT 0 = OUTPUT
								MODE SELECTION 00 = MODE 0 01 = MODE 1 1X = MODE 2
								MODE SET FLAG 1 = ACTIVE

FONCTIONNEMENT DES COUPLEURS D'ENTREE/SORTIE PIO 8255

Les PIO 8255 réalisent l'interface entre l'ordinateur et le monde extérieur. Il s'agit d'un système d'entrées sorties programmable. Ils se composent chacun de 24 entrées/sorties pouvant être programmées en 2 groupes de 12 bits et utilisées en 3 modes d'opérations.

ORGANISATION DU COMPOSANT

Le PIO est composé de 3 ports (A, B, C) d'entrées/sorties. Il possède un mot de contrôle programmable en trois modes :

- Dans le mode 0, chaque port peut être programmé en E/S.
- Dans le mode 1, les ports A et B sont programmables en E/S et possèdent comme lignes de contrôle et d'interruption le port C (bits 0 à 3 pour le port A et les bits 4 à 7 pour le port B).
- Dans le mode 2, le port A est bidirectionnel et possède comme lignes de contrôles et d'interruption les 3 à 7 du port C. (Les bits 0 à 2 étant en E/S).

DEFINITION DU MOT DE CONTROLE

Le mot de contrôle est un mot de huit bits dont chaque combinaison correspond à une utilisation des ports.

Les modes pour les ports A et B peuvent être séparément définis en ayant les bits du port C comme lignes de contrôle selon le mode.

Le tableau ci-dessous montre la correspondance des bits du mot de contrôle :

D7	D6	D5	D4	D3	D2	D1	D0	
								Port C (LOWER) 1 = INPUT 0 = OUTPUT
								Port B 1 = INPUT 0 = OUTPUT
								MODE SELECTION 0 = MODE 0 1 = MODE 1
								Port C (UPPER) 1 = INPUT 0 = OUTPUT
								Port A 1 = INPUT 0 = OUTPUT
								MODE SELECTION 00 = MODE 0 01 = MODE 1 1X = MODE 2
								MODE SET FLAG 1 = ACTIVE

MODE 0 :

Chaque port est programmé en entrée ou en sortie. Aucune ligne de contrôle n'est nécessaire, les données sont simplement écrites ou lues.

Définition fonctionnelle :

- 2 ports de 8 bits et 1 port de 2*4 bits soit 24 E/S
- Chaque port peut être en entrée ou en sortie
- Seules les entrées sont verrouillées
- 16 configurations différentes sont possibles dans ce mode

Le tableau ci-dessous représente les 16 configurations mode 0

A		B		GROUPE A			GROUPE B	
D4	D3	D1	D0	Port A	Port C (UPPER)		Port B	Port C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 1 :

Dans ce mode, les ports A et B utilisent les bits du port C comme lignes de contrôle.

Définition fonctionnelle :

- Deux groupes (A et B)
- Chaque groupe comprend un port de 8 bits et un port de 4 bits correspondant à la moitié du port C
- Chaque port peut être utilisé en entrée ou en sortie
- Les entrées et sorties sont verrouillées
- Le port de 4 bits est utilisé pour contrôler et transférer les états du port de 8 bits

MODE 0 :

Chaque port est programmé en entrée ou en sortie. Aucune ligne de contrôle n'est nécessaire, les données sont simplement écrites ou lues.

Définition fonctionnelle :

- 2 ports de 8 bits et 1 port de 2*4 bits soit 24 E/S
- Chaque port peut être en entrée ou en sortie
- Seules les entrées sont verrouillées
- 16 configurations différentes sont possibles dans ce mode

Le tableau ci-dessous représente les 16 configurations mode 0

A		B		GROUPE A			GROUPE B	
D4	D3	D1	D0	Port A	Port C (UPPER)		Port B	Port C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 1 :

Dans ce mode, les ports A et B utilisent les bits du port C comme lignes de contrôle.

Définition fonctionnelle :

- Deux groupes (A et B)
- Chaque groupe comprend un port de 8 bits et un port de 4 bits correspondant à la moitié du port C
- Chaque port peut être utilisé en entrée ou en sortie
- Les entrées et sorties sont verrouillées
- Le port de 4 bits est utilisé pour contrôler et transférer les états du port de 8 bits

MODE 2 :

Ce mode permet l'échange des données sur le même port en entrée et en sortie sans changer de mode. (Voir documentation du constructeur).

Définitions fonctionnelles :

- Utilisé pour le groupe A uniquement.
- Port de 8 bits bidirectionnel et 5 bits du port C utilisés en lignes de contrôle et d'état.
- Les entrées et sorties sont verrouillées.

MODE 2 :

Ce mode permet l'échange des données sur le même port en entrée et en sortie sans changer de mode. (Voir documentation du constructeur).

Définitions fonctionnelles :

- Utilisé pour le groupe A uniquement.
- Port de 8 bits bidirectionnel et 5 bits du port C utilisés en lignes de contrôle et d'état.
- Les entrées et sorties sont verrouillées.

FONCTIONNEMENT DES TIMERS PROGRAMMABLES 8254

La carte FASTLAB possède deux timers 0 et 1 correspondant à deux circuits type 8254. Le circuit 8254 est un timer programmable permettant la programmation de temporisations variables. Trois décompteurs de 16 bits indépendants associés à 6 modes de programmation permettant d'effectuer toutes sortes de fonctions : Horloge temps réel, compteur d'événements, générateur de signaux carrés, monostables, ...etc

DESCRIPTION DU COMPOSANT

Le timer 8254 se compose de trois compteurs de 16 bits indépendants. Chaque compteur est programmé à l'aide du registre de mot de contrôle unique CW (Control Word) associé à trois registres permettant l'écriture et la lecture de chaque compteur.

La figure ci-après représente le diagramme interne d'un compteur.

Toutes les opérations sont dirigées par le registre de mot de contrôle CW. Deux registres de comptage de 8 bits chacun, noté CR et CRL (Count Register) permettent de charger la valeur initiale du compteur (L signifiant poids faible LSB, M poids fort MSB). Le compteur réel noté CE pour élément de comptage est le décompteur synchrone, l'utilisateur ne peut y accéder. Deux autres registres au fonctionnement identique au CR noté OLL et OLM (output LATCH) gèrent la lecture du compteur sans modifier le comptage. CRM-CRL et OLM-OLL ne représente qu'une adresse dans l'espace adressable de l'ordinateur. La programmation et la récupération des données s'effectuent par adressage interne au composant suivant la valeur du registre CW. Chaque compteur possède une logique de contrôle pour gérer son entrée d'horloge CLK, son entrée de déclenchement GATen et sa sortie OUTn.

Pour fonctionner, chaque compteur doit posséder sa base de temps (CLKn). Les opérations de chargement et de comptage sont synchronisées sur le front descendant de cette horloge. A cet effet une horloge de fréquence 4 MHz est disponible sur la carte.

FONCTIONNEMENT DES TIMERS PROGRAMMABLES 8254

La carte FASTLAB possède deux timers 0 et 1 correspondant à deux circuits type 8254. Le circuit 8254 est un timer programmable permettant la programmation de temporisations variables. Trois décompteurs de 16 bits indépendants associés à 6 modes de programmation permettant d'effectuer toutes sortes de fonctions : Horloge temps réel, compteur d'événements, générateur de signaux carrés, monostables, ...etc

DESCRIPTION DU COMPOSANT

Le timer 8254 se compose de trois compteurs de 16 bits indépendants. Chaque compteur est programmé à l'aide du registre de mot de contrôle unique CW (Control Word) associé à trois registres permettant l'écriture et la lecture de chaque compteur.

La figure ci-après représente le diagramme interne d'un compteur.

Toutes les opérations sont dirigées par le registre de mot de contrôle CW. Deux registres de comptage de 8 bits chacun, noté CR et CRL (Count Register) permettent de charger la valeur initiale du compteur (L signifiant poids faible LSB, M poids fort MSB). Le compteur réel noté CE pour élément de comptage est le décompteur synchrone, l'utilisateur ne peut y accéder. Deux autres registres au fonctionnement identique au CR noté OLL et OLM (output LATCH) gèrent la lecture du compteur sans modifier le comptage. CRM-CRL et OLM-OLL ne représente qu'une adresse dans l'espace adressable de l'ordinateur. La programmation et la récupération des données s'effectuent par adressage interne au composant suivant la valeur du registre CW. Chaque compteur possède une logique de contrôle pour gérer son entrée d'horloge CLK, son entrée de déclenchement GATen et sa sortie OUTn.

Pour fonctionner, chaque compteur doit posséder sa base de temps (CLKn). Les opérations de chargement et de comptage sont synchronisées sur le front descendant de cette horloge. A cet effet une horloge de fréquence 4 MHz est disponible sur la carte.

UTILISATION SUR LA CARTE

La carte FASTLAB possède deux timers sur la carte soit au total 6 décompteurs de 16 bits indépendants. 4 de ces décompteurs sont configurés pour des utilisations bien précises sur la carte. Les compteurs 0 et 1 du timer 0 sont cascades pour obtenir un compteur 32 bits décrémente par l'horloge de base 4 MHz disponible sur la carte, la sortie OUT1 permettant ainsi de déclencher des conversions analogiques numériques.

utilisation pour le déclenchement des conversions par timer
compteur 0 et 1 cascades

De même les compteurs 0 et 1 du timer 1 sont cascades pour effectuer des comptages externes sur 32 bits en permettant la génération d'une interruption en fin de comptage sur la sortie OUT1. L'entrée horloge de comptage peut ainsi utiliser la ligne CLK0TIM1 sur le connecteur P1 ou l'horloge interne CLKB= 4MHz pour effectuer le comptage.

UTILISATION SUR LA CARTE

La carte FASTLAB possède deux timers sur la carte soit au total 6 décompteurs de 16 bits indépendants. 4 de ces décompteurs sont configurés pour des utilisations bien précises sur la carte. Les compteurs 0 et 1 du timer 0 sont cascades pour obtenir un compteur 32 bits décrémente par l'horloge de base 4 MHz disponible sur la carte, la sortie OUT1 permettant ainsi de déclencher des conversions analogiques numériques.

utilisation pour le déclenchement des conversions par timer
compteur 0 et 1 cascades

De même les compteurs 0 et 1 du timer 1 sont cascades pour effectuer des comptages externes sur 32 bits en permettant la génération d'une interruption en fin de comptage sur la sortie OUT1. L'entrée horloge de comptage peut ainsi utiliser la ligne CLK0TIM1 sur le connecteur P1 ou l'horloge interne CLKB= 4MHz pour effectuer le comptage.

Seuls les deux décompteurs 2 des timers Ø et 1 sont exploitables par l'utilisateur. Les trois lignes propres à chaque décompteur sont disponibles sur le connecteur interne P2.

Les jumpers soudables A et B situés côté soudure sous chaque composant permettent :

- JUMPER SOUDE : de valider la programmation des gates de chaque compteur 2 à l'aide des lignes PA2 et PA5 du PIO1 port A.
- JUMPER DESSOUDE : d'utiliser les lignes GATE2TIMØ et 1 disponibles sur le connecteur P2 pour une validation externe.

ATTENTION : Les gates de chaque compteur sont programmables en standard. Les bits Ø à 5 port A du PIO 1 sont utilisés pour la validation. Ces lignes sont à zéro à la mise sous tension (compteurs dévalidés). Un 1 sur une ligne, valide le compteur correspondant. La correspondance entre ces lignes est définie ci-dessous :

PIO1 PORTA	GATE TIMER	
PAØ	→	GATEØ TIMERØ
PA1	→	GATE1 TIMERØ
PA2	(si jumper B soudé)→	GATE2 TIMERØ
PA3	→	GATEØ TIMER1
PA4	→	GATE1 TIMER1
PA5	(si jumper A soudé)→	GATE2 TIMER1
PA6-PA7	non utilisables	

PROGRAMMATION DU COMPOSANT

Les registres compteurs (n) utilisent implicitement CR pour l'écriture et OL pour la lecture des valeurs de comptage.

ECRITURE

La programmation de chaque compteur doit débuter obligatoirement par l'écriture du registre de contrôle CW. Ce registre définit le numéro du compteur n (Ø-1-2) bits SCØ-SC1, le transfert de la valeur du compteur (bits RWØ-RW1) en lecture ou écriture : une seule opération poids faible ou poids fort, deux opérations poids faible et poids fort ensuite, le mode de fonctionnement (MØ-M1-M2) et sa base numérique binaire ou BCD.

Une fois le CW programmé, la valeur initiale du compteur (n) spécifié peut être écrite. Aucune séquence d'écriture n'est à respecter (le chargement de la valeur initiale du compteur Ø peut être écrite après celle du compteur 2 et/ou 1, il suffit de respecter le sens du chargement).

LECTURE

La lecture de chaque compteur peut s'effectuer suivant trois méthodes :

- Par une simple opération de lecture.
- Par le CLC (Counter Latch Commande)
- Par le RBC (Read Back Commande)

Toutes ces opérations ne modifient en rien le sens de transfert et le mode de chaque compteur.

Seuls les deux décompteurs 2 des timers Ø et 1 sont exploitables par l'utilisateur. Les trois lignes propres à chaque décompteur sont disponibles sur le connecteur interne P2.

Les jumpers soudables A et B situés côté soudure sous chaque composant permettent :

- JUMPER SOUDE : de valider la programmation des gates de chaque compteur 2 à l'aide des lignes PA2 et PA5 du PIO1 port A.
- JUMPER DESSOUDE : d'utiliser les lignes GATE2TIMØ et 1 disponibles sur le connecteur P2 pour une validation externe.

ATTENTION : Les gates de chaque compteur sont programmables en standard. Les bits Ø à 5 port A du PIO 1 sont utilisés pour la validation. Ces lignes sont à zéro à la mise sous tension (compteurs dévalidés). Un 1 sur une ligne, valide le compteur correspondant. La correspondance entre ces lignes est définie ci-dessous :

PIO1 PORTA	GATE TIMER	
PAØ	→	GATEØ TIMERØ
PA1	→	GATE1 TIMERØ
PA2	(si jumper B soudé)→	GATE2 TIMERØ
PA3	→	GATEØ TIMER1
PA4	→	GATE1 TIMER1
PA5	(si jumper A soudé)→	GATE2 TIMER1
PA6-PA7	non utilisables	

PROGRAMMATION DU COMPOSANT

Les registres compteurs (n) utilisent implicitement CR pour l'écriture et OL pour la lecture des valeurs de comptage.

ECRITURE

La programmation de chaque compteur doit débuter obligatoirement par l'écriture du registre de contrôle CW. Ce registre définit le numéro du compteur n (Ø-1-2) bits SCØ-SC1, le transfert de la valeur du compteur (bits RWØ-RW1) en lecture ou écriture : une seule opération poids faible ou poids fort, deux opérations poids faible et poids fort ensuite, le mode de fonctionnement (MØ-M1-M2) et sa base numérique binaire ou BCD.

Une fois le CW programmé, la valeur initiale du compteur (n) spécifié peut être écrite. Aucune séquence d'écriture n'est à respecter (le chargement de la valeur initiale du compteur Ø peut être écrite après celle du compteur 2 et/ou 1, il suffit de respecter le sens du chargement).

LECTURE

La lecture de chaque compteur peut s'effectuer suivant trois méthodes :

- Par une simple opération de lecture.
- Par le CLC (Counter Latch Commande)
- Par le RBC (Read Back Commande)

Toutes ces opérations ne modifient en rien le sens de transfert et le mode de chaque compteur.

- La première méthode effectue une simple opération de lecture à l'adresse du compteur n correspondant (LSB et/ou MSB). La CLK_n sera inhibée obligatoirement, cependant le compteur peut être modifié durant l'opération de lecture et le résultat faussé.
- La seconde méthode utilise les bits D5 et D4 du CW. L'opération d'écriture du CW avec D5 et D4 à 0 verrouille la valeur actuelle du compteur n sélectionné par SC1-SC2 dans le registre de sortie OL. Cette valeur est alors stockée jusqu'à sa lecture. ATTENTION si une autre opération CLC est effectuée sans avoir lu au préalable le comptage, celle-ci est ignorée. Les bits restant D3 et D0 seront positionnés à 0.
- La troisième permet de lire les valeurs d'un ou des compteurs, leurs modes de fonctionnement, les états des sorties OUT et les drapeaux (null count). Cette opération est effectuée par une écriture du CW. Le RBC équivaut à de multiples CLC. Elle permet de verrouiller, à un instant donné, toutes les informations utiles d'un ou de plusieurs compteurs à la fois. Les bits D7 et D6 à 1 du CW définissent le RBC, le bit D5 l'opération de verrouillage du ou des compteurs sélectionnés par D3-D2-D1 pour une lecture du comptage et D4 verrouille l'état de fonctionnement soit :
Les valeurs des sorties OUT_n - l'indication du dernier chargement du compteur pour une lecture éventuelle (null count) et les références du CW programmé (chargement, mode, BCD).

Lorsque les deux opérations sont effectuées simultanément (D5-D4 à 0) la première lecture du registre compteur représente l'octet d'état, la suivante la valeur du comptage.

La sélection du mode est définie en fonction de l'application désirée. Le mode est programmé à la première écriture du CW (D3-D2-D1). La signification de chaque mode est spécifiée ci-dessous:

- MODE 0 : Interruption de fin de comptage - compteur d'évènement
- MODE 1 : Bascule réamorçable
- MODE 2 : Générateur de signaux
- MODE 3 : Signaux carrés, identique au mode 2 mais au rapport cyclique différent
- MODE 4 : Trace repère amorçable par logiciel
- MODE 5 : Trace repère "ré"amorçable par l'extérieur

Pour plus d'information sur le fonctionnement du composant et l'utilisation des différents modes, se référer à la notice technique du constructeur.

- La première méthode effectue une simple opération de lecture à l'adresse du compteur n correspondant (LSB et/ou MSB). La CLK_n sera inhibée obligatoirement, cependant le compteur peut être modifié durant l'opération de lecture et le résultat faussé.
- La seconde méthode utilise les bits D5 et D4 du CW. L'opération d'écriture du CW avec D5 et D4 à 0 verrouille la valeur actuelle du compteur n sélectionné par SC1-SC2 dans le registre de sortie OL. Cette valeur est alors stockée jusqu'à sa lecture. ATTENTION si une autre opération CLC est effectuée sans avoir lu au préalable le comptage, celle-ci est ignorée. Les bits restant D3 et D0 seront positionnés à 0.
- La troisième permet de lire les valeurs d'un ou des compteurs, leurs modes de fonctionnement, les états des sorties OUT et les drapeaux (null count). Cette opération est effectuée par une écriture du CW. Le RBC équivaut à de multiples CLC. Elle permet de verrouiller, à un instant donné, toutes les informations utiles d'un ou de plusieurs compteurs à la fois. Les bits D7 et D6 à 1 du CW définissent le RBC, le bit D5 l'opération de verrouillage du ou des compteurs sélectionnés par D3-D2-D1 pour une lecture du comptage et D4 verrouille l'état de fonctionnement soit :
Les valeurs des sorties OUT_n - l'indication du dernier chargement du compteur pour une lecture éventuelle (null count) et les références du CW programmé (chargement, mode, BCD).

Lorsque les deux opérations sont effectuées simultanément (D5-D4 à 0) la première lecture du registre compteur représente l'octet d'état, la suivante la valeur du comptage.

La sélection du mode est définie en fonction de l'application désirée. Le mode est programmé à la première écriture du CW (D3-D2-D1). La signification de chaque mode est spécifiée ci-dessous:

- MODE 0 : Interruption de fin de comptage - compteur d'évènement
- MODE 1 : Bascule réamorçable
- MODE 2 : Générateur de signaux
- MODE 3 : Signaux carrés, identique au mode 2 mais au rapport cyclique différent
- MODE 4 : Trace repère amorçable par logiciel
- MODE 5 : Trace repère "ré"amorçable par l'extérieur

Pour plus d'information sur le fonctionnement du composant et l'utilisation des différents modes, se référer à la notice technique du constructeur.

UTILISATION DE LA CARTE SOUS INTERRUPTION

La carte FASTLAB permet l'utilisation des interruptions. Pour cela l'utilisateur devra effectuer un programme de gestion en langage machine ou évolué.

ATTENTION : l'utilisateur désirant travailler sous interruption doit avoir une grande connaissance des langages de programmation, ainsi que du système d'exploitation de l'appareil.

Sur l'IBM PC-AT la gestion d'interruption matérielle est gérée par deux contrôleurs INTEL PIC 8259. Ces composants permettent de gérer chacun 8 niveaux physiques d'interruption. Un état haut sur l'une de ces lignes déclenche une interruption système.

Le contrôleur 1 gère les lignes notées IRQ0 à IRQ7 à l'aide de 2 registres OCW1 adresse \$20 et OCW2 adresse \$21.

Le contrôleur 2 gère les lignes notées IRQ8 à IRQ15 à l'aide de 2 registres OCW1 adresse \$A0 et OCW2 adresse \$A1.

8 de ces lignes sont utilisables sur la carte. Une de ces 8 lignes peut être programmée dans le registre NRINT B+23. Le numéro validé définira le vecteur (adresse de la gestion) et le contrôleur associé.

PROGRAMMATION

La programmation d'une telle fonction comporte différentes étapes nécessaires au bon fonctionnement du système. Les étapes énumérées ci-dessous devront être gérées impérativement pour éviter tout problème.

- initialisation du vecteur d'interruption dans l'ordinateur.
- initialisation du contrôleur PIC 8259 correspondant.
- initialisation du module générant l'interruption (INTEXT-TIMER1-convertisseur).
- initialisation du registre VALID B+21
- gestion de la routine d'interruption - acquittement.

Se référer aux routines d'interruptions fournies sur la disquette accompagnant ce manuel.

UTILISATION DE LA CARTE SOUS INTERRUPTION

La carte FASTLAB permet l'utilisation des interruptions. Pour cela l'utilisateur devra effectuer un programme de gestion en langage machine ou évolué.

ATTENTION : l'utilisateur désirant travailler sous interruption doit avoir une grande connaissance des langages de programmation, ainsi que du système d'exploitation de l'appareil.

Sur l'IBM PC-AT la gestion d'interruption matérielle est gérée par deux contrôleurs INTEL PIC 8259. Ces composants permettent de gérer chacun 8 niveaux physiques d'interruption. Un état haut sur l'une de ces lignes déclenche une interruption système.

Le contrôleur 1 gère les lignes notées IRQ0 à IRQ7 à l'aide de 2 registres OCW1 adresse \$20 et OCW2 adresse \$21.

Le contrôleur 2 gère les lignes notées IRQ8 à IRQ15 à l'aide de 2 registres OCW1 adresse \$A0 et OCW2 adresse \$A1.

8 de ces lignes sont utilisables sur la carte. Une de ces 8 lignes peut être programmée dans le registre NRINT B+23. Le numéro validé définira le vecteur (adresse de la gestion) et le contrôleur associé.

PROGRAMMATION

La programmation d'une telle fonction comporte différentes étapes nécessaires au bon fonctionnement du système. Les étapes énumérées ci-dessous devront être gérées impérativement pour éviter tout problème.

- initialisation du vecteur d'interruption dans l'ordinateur.
- initialisation du contrôleur PIC 8259 correspondant.
- initialisation du module générant l'interruption (INTEXT-TIMER1-convertisseur).
- initialisation du registre VALID B+21
- gestion de la routine d'interruption - acquittement.

Se référer aux routines d'interruptions fournies sur la disquette accompagnant ce manuel.

GENERATION DES INTERRUPTIONS SUR LA CARTE

Trois sources IRQEXT - sortie OUT1 du TIMER1 - CONVERTISSEUR permettent de générer des interruptions. Avant toute validation, l'utilisateur s'assurera qu'aucune interruption n'est active en effectuant soit une lecture du convertisseur FGRDY=1 - un test sur l'état logique de la sortie OUT1 du timer 1 - une lecture du registre CLEAR IRQEXT.

Ligne IRQEXT

Cette ligne est connectée au module d'interruption par une bascule. Un front montant survenant sur cette ligne déclenche une interruption.

LE TIMER 1

Un passage à l'état haut sur la sortie OUT1 du compteur 1 génère une interruption. Le mode Ø (interruption en fin de comptage) est conseillé.

LE CONVERTISSEUR

A chaque fin de conversion FGRDY=Ø, une interruption est générée.

Ces trois lignes peuvent être validées indépendamment pour générer des interruptions par le registre VALID B+21 et leur état lu par le registre ETAT situé à la même adresse. l'acquittement de ces lignes sur la carte est effectué :

- Par une lecture du registre CONV résultat de la conversion.
- Par une écriture de la valeur des compteurs (cette opération est fonction du mode utilisé)
- Par une lecture du registre CLEAR IRQEXT

GENERATION DES INTERRUPTIONS SUR LA CARTE

Trois sources IRQEXT - sortie OUT1 du TIMER1 - CONVERTISSEUR permettent de générer des interruptions. Avant toute validation, l'utilisateur s'assurera qu'aucune interruption n'est active en effectuant soit une lecture du convertisseur FGRDY=1 - un test sur l'état logique de la sortie OUT1 du timer 1 - une lecture du registre CLEAR IRQEXT.

Ligne IRQEXT

Cette ligne est connectée au module d'interruption par une bascule. Un front montant survenant sur cette ligne déclenche une interruption.

LE TIMER 1

Un passage à l'état haut sur la sortie OUT1 du compteur 1 génère une interruption. Le mode Ø (interruption en fin de comptage) est conseillé.

LE CONVERTISSEUR

A chaque fin de conversion FGRDY=Ø, une interruption est générée.

Ces trois lignes peuvent être validées indépendamment pour générer des interruptions par le registre VALID B+21 et leur état lu par le registre ETAT situé à la même adresse. l'acquittement de ces lignes sur la carte est effectué :

- Par une lecture du registre CONV résultat de la conversion.
- Par une écriture de la valeur des compteurs (cette opération est fonction du mode utilisé)
- Par une lecture du registre CLEAR IRQEXT

OUTILS LOGICIELS ET ROUTINES

PROGRAMMES FOURNIS (*Disquette routines pour FASTLAB*)

FASTEST.EXE	Programme de test et de gestion de la carte (Voir descriptif plus loin)
FASTEST.C	Programme source en TURBO C de FASTEST.EXE
FASTEST.SCR	Masque écran pour FASTEST
FASTEST.PRJ	Editions des liens entre FASTEST.C et les drivers de la carte
FAST_C.C	Routines écrites en C. Correspond aux interfaçages LOGIQUES de la carte
FAST_C.OBJ	Routines C compilées
FAST_ASM.ASM	Routines écrites en Assembleur pour TURBO C. Correspond aux interfaçages ANALOGIQUES de la carte
FAST_ASM.OBJ	Routines Assembleur compilées pour TURBO C
FASTLAB.H	Prototypage des fonctions C et Assembleur
FASTLAB.PAS	Routines PASCAL
FASTLAB.TPU	Routines PASCAL compilées (UNITE)
FLAB_UNI.EXE	Acquisition simple sur une seule voie
FLAB_UNI.PAS	Programme source en TURBO PASCAL de FLAB_UNI.EXE
FLAB_MUL.EXE	Acquisitions multiples sur plusieurs voies
FLAB_MUL.PAS	Programme source en TURBO PASCAL de FLAB_MUL.EXE
FLAB_TIM.EXE	Acquisitions multiples sous contrôle du timer
FLAB_TIM.PAS	Programme source en TURBO PASCAL de FLAB_TIM.EXE
FLAB_CNA.EXE	Emission sur les sorties analogiques
FLAB_CNA.PAS	Programme source en TURBO PASCAL de FLAB_CNA.EXE
FLAB_LOG.EXE	Utilisation des ports logiques
FLAB_LOG.PAS	Programme source en TURBO PASCAL de FLAB_LOG.EXE
PBASE.TPU	Unité de gestion d'écran pour programmes FLAB_*.PAS.
PBASE.PAS	Programme source de l'unité PBASE.TPU

DESCRIPTIONS DES FONCTIONS DE FASTEST.EXE

FASTEST est un logiciel de test général et de gestion de la carte FASTLAB. Sa fonction essentielle est de contrôler l'ensemble des fonctionnalités de la carte et de faire les réglages si ceux-ci s'avèrent nécessaires.

Il permet donc de :

- mesurer une tension sur les entrées analogiques
- programmer une tension sur chacune des sorties analogiques
- programmer l'état des ports logiques
- vérifier l'état du timer utilisateur (T1). Ce test est réalisé par le déroutage d'une IRQ en fin de comptage

Pour lancer ce programme de test, taper **FASTEST** et suivre les indications portées à l'écran. Pour modifier chaque paramètre, il suffit de taper la lettre clé correspondante. Voir tableau ci-après.

OUTILS LOGICIELS ET ROUTINES

PROGRAMMES FOURNIS (*Disquette routines pour FASTLAB*)

FASTEST.EXE	Programme de test et de gestion de la carte (Voir descriptif plus loin)
FASTEST.C	Programme source en TURBO C de FASTEST.EXE
FASTEST.SCR	Masque écran pour FASTEST
FASTEST.PRJ	Editions des liens entre FASTEST.C et les drivers de la carte
FAST_C.C	Routines écrites en C. Correspond aux interfaçages LOGIQUES de la carte
FAST_C.OBJ	Routines C compilées
FAST_ASM.ASM	Routines écrites en Assembleur pour TURBO C. Correspond aux interfaçages ANALOGIQUES de la carte
FAST_ASM.OBJ	Routines Assembleur compilées pour TURBO C
FASTLAB.H	Prototypage des fonctions C et Assembleur
FASTLAB.PAS	Routines PASCAL
FASTLAB.TPU	Routines PASCAL compilées (UNITE)
FLAB_UNI.EXE	Acquisition simple sur une seule voie
FLAB_UNI.PAS	Programme source en TURBO PASCAL de FLAB_UNI.EXE
FLAB_MUL.EXE	Acquisitions multiples sur plusieurs voies
FLAB_MUL.PAS	Programme source en TURBO PASCAL de FLAB_MUL.EXE
FLAB_TIM.EXE	Acquisitions multiples sous contrôle du timer
FLAB_TIM.PAS	Programme source en TURBO PASCAL de FLAB_TIM.EXE
FLAB_CNA.EXE	Emission sur les sorties analogiques
FLAB_CNA.PAS	Programme source en TURBO PASCAL de FLAB_CNA.EXE
FLAB_LOG.EXE	Utilisation des ports logiques
FLAB_LOG.PAS	Programme source en TURBO PASCAL de FLAB_LOG.EXE
PBASE.TPU	Unité de gestion d'écran pour programmes FLAB_*.PAS.
PBASE.PAS	Programme source de l'unité PBASE.TPU

DESCRIPTIONS DES FONCTIONS DE FASTEST.EXE

FASTEST est un logiciel de test général et de gestion de la carte FASTLAB. Sa fonction essentielle est de contrôler l'ensemble des fonctionnalités de la carte et de faire les réglages si ceux-ci s'avèrent nécessaires.

Il permet donc de :

- mesurer une tension sur les entrées analogiques
- programmer une tension sur chacune des sorties analogiques
- programmer l'état des ports logiques
- vérifier l'état du timer utilisateur (T1). Ce test est réalisé par le déroutage d'une IRQ en fin de comptage

Pour lancer ce programme de test, taper **FASTEST** et suivre les indications portées à l'écran. Pour modifier chaque paramètre, il suffit de taper la lettre clé correspondante. Voir tableau ci-après.

CONVERTISSEUR A/D VOIES : Simples								CONVERTISSEUR D/A				ADRESSE			
GAIN : 1				CALIBRE : ±10V				Saisie de la Valeur				(Hexadécimale)			
Num. Volt								Sortie 1 (V) : 5.00				Saisie Adresse			
Entrée 0 : 2049 0.0049								Sortie 2 (V) : 10.00				X= 240			
Entrée 1 : 1891 -0.7666								E : Convertisseur A/D (Configuration)							
Entrée 2 : 1712 -1.6406								S : Convertisseur D/A (Programmation)							
Entrée 3 : 1584 -2.2656								T : Tests TIMER 1 (TIMER Utilisateur)							
Entrée 4 : 1231 -3.9892								L : E/S Logiques (Configuration)							
Entrée 5 : 1243 -3.9307								P : E/S Logiques (Programmation PORTS)							
Entrée 6 : 1254 -3.8769								X : Programmation adresse FASTLAB							
Entrée 7 : 1291 -3.6963								Q : Quitter FASTEST.EXE							
Touche + pour modifier Paramètres															
ENTREES/SORTIES LOGIQUES												TIMER			
Port B:		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Test sur déclenchement d'IRQ (Sélect° = IRQ10) Temps entre 2 interrupt° → 1 seconde ESCAPE = Arrêt du test					
		0	0	0	0	0	0	0	0						
Port C:		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0						
		0	0	0	0	0	0	0	0						
Etat Ports : B: Sortie C: Sortie N°PIO: O												© EUROSMART - FASTEST			
Touche + pour modifier Paramètres															

DESCRIPTION DES FONCTIONNALITES DES ROUTINES:

Des routines d'utilisation de la carte **FASTLAB** ont été développées en langages C et PASCAL, intégrant des appels à des routines écrites en ASSEMBLEUR (Pour version C).

Le nom des différentes routines est le même d'un langage à l'autre. Le chapitre ci-dessous décrit les fonctionnalités de celle-ci dans leur généralité.

Les codes sources de ces routines (.PAS, .H, .C et .ASM) vous sont également donnés, à titre d'exemple de programmation de la carte, ou pour toutes modifications personnelles.

En cas de modifications, les routines devront être recompilées afin de prendre en compte les ajouts ou suppressions effectués.

Certains paramètres sont à passer dans un langage, mais pas obligatoirement dans l'autre. Les spécifications de tels cas vous sont indiqués en italique.

A noter que pour une programmation en langage C, l'adresse de base de la carte devra être passée à toutes les procédures, excepté INSTALL_IRQ et DESINSTAL_IRQ.

La description de chaque routine contient :

Nom de la routine
Paramètre(s) transmis
Valeur retournée
Fonctionnalité

L'absence d'une ou plusieurs rubriques décrites ci-dessus, lors de la description d'une routine, témoigne de leur inutilité.

CONVERTISSEUR A/D								VOIES : Simples								CONVERTISSEUR D/A								ADRESSE											
GAIN : 1								CALIBRE : $\pm 10V$								Saisie de la Valeur								(Hexadécimale)											
Num. Volt																Sortie 1 (V) : 5.00								Saisie Adresse											
Entrée 0 : 2049								0.0049								E : Convertisseur A/D (Configuration) S : Convertisseur D/A (Programmation) T : Tests TIMER 1 (TIMER Utilisateur) L : E/S Logiques (Configuration) P : E/S Logiques (Programmation PORTS) X : Programmation adresse FASTLAB Q : Quitter FASTEST.EXE								X= 240											
Entrée 1 : 1891								-0.7666																											
Entrée 2 : 1712								-1.6406																											
Entrée 3 : 1584								-2.2656																											
Entrée 4 : 1231								-3.9892																											
Entrée 5 : 1243								-3.9307																											
Entrée 6 : 1254								-3.8769																											
Entrée 7 : 1291								-3.6963																											
Touche + pour modifier Paramètres																																			
ENTREES/SORTIES LOGIQUES																TIMER																			
Port B:								Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Test sur déclenchement d'IRQ (Sélect° = IRQ10)																			
								0	0	0	0	0	0	0	0	Temps entre 2 interrupt°																			
Port C:								Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	→ 1 seconde																			
								0	0	0	0	0	0	0	0	ESCAPE = Arrêt du test																			
Etat Ports :								B: Sortie				C: Sortie				N°PIO: O				© EUROSMART - FASTEST															
Touche + pour modifier Paramètres																																			

DESCRIPTION DES FONCTIONNALITES DES ROUTINES:

Des routines d'utilisation de la carte **FASTLAB** ont été développées en langages C et PASCAL, intégrant des appels à des routines écrites en ASSEMBLEUR (Pour version C).

Le nom des différentes routines est le même d'un langage à l'autre. Le chapitre ci-dessous décrit les fonctionnalités de celle-ci dans leur généralité.

Les codes sources de ces routines (.PAS, .H, .C et .ASM) vous sont également donnés, à titre d'exemple de programmation de la carte, ou pour toutes modifications personnelles.

En cas de modifications, les routines devront être recompilées afin de prendre en compte les ajouts ou suppressions effectués.

Certains paramètres sont à passer dans un langage, mais pas obligatoirement dans l'autre. Les spécifications de tels cas vous sont indiqués en italique.

A noter que pour une programmation en langage C, l'adresse de base de la carte devra être passée à toutes les procédures, excepté INSTALL_IRQ et DESINSTAL_IRQ.

La description de chaque routine contient :

Nom de la routine
Paramètre(s) transmis
Valeur retournée
Fonctionnalité

L'absence d'une ou plusieurs rubriques décrites ci-dessus, lors de la description d'une routine, témoigne de leur inutilité.

ATTENTION : Au sein de votre IDE, le paramètre "DEFAULT CHAR TYPE" du menu "OPTIONS", sous menu "COMPILER" option "CODE GENERATION" doit être validé sur UNSIGNED.

INITFASTLAB() *EXCLUSIVEMENT DISPONIBLE EN PASCAL*

Paramètre transmis : Adresse de base de la carte.
 Fonction : Initialise la carte FASTLAB dans sa totalité.
 Remise à zéro de l'IRQ programmée sur la carte
 Initialisation des 2 TIMERS
 Remise à zéro flag IRQEXT
 Remise à zéro des options d'acquisition et d'interruptions
 Initialisation des 2 PIO

SETCHANNEL()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Mode d'entrée : 0 = 8 Voies simples
 1 = 4 Voies différentielles
 Numéro de la voie à programmer
 Fonction : Programme le type d'entrée analogique (Simple ou Différentiel)
 Programme la voie choisie en entrée

SETGAIN()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur du gain à programmer (*Langage C*)
 Valeurs du Gain et du Calibre (*Langage PASCAL*)
 Fonction : Programme un gain analogique sur une entrée (*Pour langage C*)
 (0:x1,1:x2,2:x4,3:x8,4:x10,5:x20,6:x40,7:x80,8:x100,9:x200,10:x400,
 11:x800,12:x1000,13:x2000,14:x4000,15:x8000)
 Programme un gain et un calibre sur une entrée (*Pour langage Pascal*)
 (Gain: 0:x1,1:x10,2: x100,3:x1000 / Calibre: 0:10V,1:5V,2:2.5V,3:1.25V)

START_READAD()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur retournée : Résultat de la numérisation
 Fonction : Déclenche une acquisition A/D
 Test fin de conversion
 Retourne la valeur convertie

MULTI_AD()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Segment adresse de stockage du tampon
 Offset adresse de stockage du tampon
 Nombre de voies à scruter
 Nombre de valeurs à acquérir
 Tempo. entre commutation des voies (établissement)
 Fonction : Stocke dans un tampon le résultat de P points sur N voies A/D

EXT_READAD()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur retournée : Résultat de la numérisation
 Fonction : Teste la fin de conversion
 Retourne la valeur convertie

ATTENTION : Au sein de votre IDE, le paramètre "DEFAULT CHAR TYPE" du menu "OPTIONS", sous menu "COMPILER" option "CODE GENERATION" doit être validé sur UNSIGNED.

INITFASTLAB() *EXCLUSIVEMENT DISPONIBLE EN PASCAL*

Paramètre transmis : Adresse de base de la carte.
 Fonction : Initialise la carte FASTLAB dans sa totalité.
 Remise à zéro de l'IRQ programmée sur la carte
 Initialisation des 2 TIMERS
 Remise à zéro flag IRQEXT
 Remise à zéro des options d'acquisition et d'interruptions
 Initialisation des 2 PIO

SETCHANNEL()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Mode d'entrée : 0 = 8 Voies simples
 1 = 4 Voies différentielles
 Numéro de la voie à programmer
 Fonction : Programme le type d'entrée analogique (Simple ou Différentiel)
 Programme la voie choisie en entrée

SETGAIN()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur du gain à programmer (*Langage C*)
 Valeurs du Gain et du Calibre (*Langage PASCAL*)
 Fonction : Programme un gain analogique sur une entrée (*Pour langage C*)
 (0:x1,1:x2,2:x4,3:x8,4:x10,5:x20,6:x40,7:x80,8:x100,9:x200,10:x400,
 11:x800,12:x1000,13:x2000,14:x4000,15:x8000)
 Programme un gain et un calibre sur une entrée (*Pour langage Pascal*)
 (Gain: 0:x1,1:x10,2: x100,3:x1000 / Calibre: 0:10V,1:5V,2:2.5V,3:1.25V)

START_READAD()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur retournée : Résultat de la numérisation
 Fonction : Déclenche une acquisition A/D
 Test fin de conversion
 Retourne la valeur convertie

MULTI_AD()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Segment adresse de stockage du tampon
 Offset adresse de stockage du tampon
 Nombre de voies à scruter
 Nombre de valeurs à acquérir
 Tempo. entre commutation des voies (établissement)
 Fonction : Stocke dans un tampon le résultat de P points sur N voies A/D

EXT_READAD()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur retournée : Résultat de la numérisation
 Fonction : Teste la fin de conversion
 Retourne la valeur convertie

RAZ_FLAG_CONV()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Remise à zéro du flag de fin de conversion

RAZ_VALID()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Remise à zéro du registre de validation

READ_BIT_ETAT()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Masque du bit à tester
 Fonction : Test la validité d'un bit dans le registre d'état

WRITE_VALID_BIT()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Masque du bit à valider
 Ancien masque de validation
 Fonction : Valide un bit du registre d'état

WRITE_DEVALID_BIT()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Masque du bit à dévalider
 Ancien masque de validation
 Fonction : Dévalide un bit du registre d'état

INSTALL_IRQ()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro de l'interruption à installer
 Pointeur sur la fonction d'interruption
 Valeur retournée : Pointeur sur ancienne routine d'interruption (*Langage C*)
 Fonction : Configure le vecteur d'interruption sur la routine correspondante.

DESINSTAL_IRQ()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro de l'interruption à désinstaller
 Pointeur sur ancienne routine d'interruption (*Langage C*)
 Fonction : Restaure le vecteur d'interruption sur la routine précédente à l'appel d'**INSTALL_IRQ()**

VALID_IRQ()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro de l'interruption
 Fonction : Valide le contrôleur d'interruption (8259) du PC
 Valide la ligne de la carte correspondante au N° d'interruption

DEVALID_IRQ()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro d'interruption
 Fonction : Dévalide le contrôleur d'interruption du PC
 Dévalide la ligne de la carte correspondante au N° d'interruption

RAZ_FLAG_CONV()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Remise à zéro du flag de fin de conversion

RAZ_VALID()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Remise à zéro du registre de validation

READ_BIT_ETAT()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Masque du bit à tester
 Fonction : Test la validité d'un bit dans le registre d'état

WRITE_VALID_BIT()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Masque du bit à valider
 Ancien masque de validation
 Fonction : Valide un bit du registre d'état

WRITE_DEVALID_BIT()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Masque du bit à dévalider
 Ancien masque de validation
 Fonction : Dévalide un bit du registre d'état

INSTALL_IRQ()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro de l'interruption à installer
 Pointeur sur la fonction d'interruption
 Valeur retournée : Pointeur sur ancienne routine d'interruption (*Langage C*)
 Fonction : Configure le vecteur d'interruption sur la routine correspondante.

DESINSTAL_IRQ()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro de l'interruption à désinstaller
 Pointeur sur ancienne routine d'interruption (*Langage C*)
 Fonction : Restaure le vecteur d'interruption sur la routine précédente à l'appel d'**INSTALL_IRQ()**

VALID_IRQ()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro de l'interruption
 Fonction : Valide le contrôleur d'interruption (8259) du PC
 Valide la ligne de la carte correspondante au N° d'interruption

DEVALID_IRQ()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro d'interruption
 Fonction : Dévalide le contrôleur d'interruption du PC
 Dévalide la ligne de la carte correspondante au N° d'interruption

CLEAR_FLAG_IRQEXT()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Remise à zéro du flag de déclenchement d'une interruption externe

RAZ_VECTEUR()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Dévalide toutes les lignes de la carte connectées aux IRQ du PC

TEST_VALID_EOI()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro d'interruption
 Fonction : Permet d'acquitter le contrôleur d'interruption du PC en fonction du numéro d'IRQ transmis

SET_TAMPON() **EXCLUSIVEMENT DISPONIBLE EN PASCAL**

Paramètres transmis : Pointeur sur tampon à affecter
 Nombre de valeurs à enregistrer
 Valeur retournée : Valeur du pointeur.
 Fonction : Allocation d'un tampon de stockage pour acquisition par interruption

FREE_TAMPON() **EXCLUSIVEMENT DISPONIBLE EN PASCAL**

Paramètre transmis : Pointeur sur tampon à désallouer
 Fonction : Libère l'emplacement mémoire alloué par **SET_TAMPON()**

SETDA1()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur (16 bits) désirée en sortie.
 Fonction : Permet d'écrire une valeur dans le convertisseur DA en voie 1
 Valide la valeur programmée en sortie

SETDA2()

Fonction identique à **SETDA1**, mais pour le convertisseur DA voie 2

MULTI_SETDA()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur (16 bits) à écrire sur convertisseur DA voie 1
 Valeur (16 bits) à écrire sur convertisseur DA voie 2
 Fonction : Permet d'écrire sur le convertisseur DA en voie 1 et en voie 2
 Valide les valeurs sur les voies correspondantes

RAZ_DAC()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Remise à zéro des 2 sorties analogiques SA1 et SA2

CLEAR_FLAG_IRQEXT()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Remise à zéro du flag de déclenchement d'une interruption externe

RAZ_VECTEUR()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Dévalide toutes les lignes de la carte connectées aux IRQ du PC

TEST_VALID_EOI()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro d'interruption
 Fonction : Permet d'acquitter le contrôleur d'interruption du PC en fonction du numéro d'IRQ transmis

SET_TAMPON() **EXCLUSIVEMENT DISPONIBLE EN PASCAL**

Paramètres transmis : Pointeur sur tampon à affecter
 Nombre de valeurs à enregistrer
 Valeur retournée : Valeur du pointeur.
 Fonction : Allocation d'un tampon de stockage pour acquisition par interruption

FREE_TAMPON() **EXCLUSIVEMENT DISPONIBLE EN PASCAL**

Paramètre transmis : Pointeur sur tampon à désallouer
 Fonction : Libère l'emplacement mémoire alloué par **SET_TAMPON()**

SETDA1()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur (16 bits) désirée en sortie.
 Fonction : Permet d'écrire une valeur dans le convertisseur DA en voie 1
 Valide la valeur programmée en sortie

SETDA2()

Fonction identique à **SETDA1**, mais pour le convertisseur DA voie 2

MULTI_SETDA()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Valeur (16 bits) à écrire sur convertisseur DA voie 1
 Valeur (16 bits) à écrire sur convertisseur DA voie 2
 Fonction : Permet d'écrire sur le convertisseur DA en voie 1 et en voie 2
 Valide les valeurs sur les voies correspondantes

RAZ_DAC()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)
 Fonction : Remise à zéro des 2 sorties analogiques SA1 et SA2

SETTIMER()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du TIMER (0 ou 1)
 Numéro du compteur (0,1 ou 2)
 Mode de fonctionnement du compteur choisi
 Comptage en Binaire ou en BCD (0 ou 1)
 Valeur de comptage (0 à 65535)

Fonction : Routine d'initialisation d'un compteur

READTIMER()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du Timer (0 ou 1)
 Numéro du Compteur (0, 1 ou 2)

Valeur retournée : Valeur contenue par le Compteur

Fonction : Lecture du compteur choisi (16 bits)

RAZ_TIMER()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)

Fonction : Dévalide les comptages sur les 2 Timers

START_TIMER()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du Timer (0 ou 1)
 Numéro du Compteur (0, 1 ou 2)
 Ancien masque de validation

Valeur retournée : Nouveau masque de validation

Fonction : Valide le démarrage d'un compteur en validant sa gate

STOP_TIMER()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du Timer (0 ou 1)
 Numéro du Compteur (0, 1 ou 2)
 Ancien masque de validation

Valeur retournée : Nouveau masque de validation

Fonction : Stoppe un compteur en dévalidant sa gate

INPUTIO()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du PIO 8255 (0 ou 1)
 Numéro du Port (1=PORT B 2=PORT C)

Valeur retournée : Valeur lue sur Port (octet)

OUTPUTIO()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du PIO 8255 (0 ou 1)
 Numéro de Port (1=PORT B 2=PORT C)
 Valeur à écrire (octet)

Fonction : Permet d'écrire sur Port de sorties logiques

SETTIMER()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du TIMER (0 ou 1)
 Numéro du compteur (0,1 ou 2)
 Mode de fonctionnement du compteur choisi
 Comptage en Binaire ou en BCD (0 ou 1)
 Valeur de comptage (0 à 65535)

Fonction : Routine d'initialisation d'un compteur

READTIMER()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du Timer (0 ou 1)
 Numéro du Compteur (0, 1 ou 2)

Valeur retournée : Valeur contenue par le Compteur

Fonction : Lecture du compteur choisi (16 bits)

RAZ_TIMER()

Paramètre transmis : Adresse de base de la carte (*Langage C uniquement*)

Fonction : Dévalide les comptages sur les 2 Timers

START_TIMER()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du Timer (0 ou 1)
 Numéro du Compteur (0, 1 ou 2)
 Ancien masque de validation

Valeur retournée : Nouveau masque de validation

Fonction : Valide le démarrage d'un compteur en validant sa gate

STOP_TIMER()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du Timer (0 ou 1)
 Numéro du Compteur (0, 1 ou 2)
 Ancien masque de validation

Valeur retournée : Nouveau masque de validation

Fonction : Stoppe un compteur en dévalidant sa gate

INPUTIO()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du PIO 8255 (0 ou 1)
 Numéro du Port (1=PORT B 2=PORT C)

Valeur retournée : Valeur lue sur Port (octet)

OUTPUTIO()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du PIO 8255 (0 ou 1)
 Numéro de Port (1=PORT B 2=PORT C)
 Valeur à écrire (octet)

Fonction : Permet d'écrire sur Port de sorties logiques

SETPIO()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du PIO 8255 (0 ou 1)
 Numéro du Port (Ø= PORT A 1=PORT B 2=PORT C)
 Ancien masque de configuration
 Mode de programmation du Port (1=Entrée 0=Sortie)
 Valeur retournée : Nouveau masque de configuration

En langage C, le Port A du PIO1 doit être initialisé comme suit :
masque=Setpio(adr_carte,1,0,0x9B,0);

**TABLEAU RECAPITULATIF DES PROCEDURES ET DES
PASSAGES DES PARAMETRES**

NOMS PROCEDURES	PARAMETRES C	PARAMETRES PASCAL
Initfastlab()	<i>Non disponible</i>	adr_carte : word
Setchannel()	unsigned int adr_carte unsigned char mode unsigned char voie	mode : byte voie : byte
Setgain()	unsigned int adr_carte unsigned char gain	codecalibre : byte codegain : byte
Start_readad()	unsigned int adr_carte Retour : int	Retour : integer
Multi_ad()	unsigned in adr_carte int segment int offset int nb_voie int nb_acq int tempo	segment : word offset : word nb_voie : word nb_acq : word tempo : word
Ext_readad()	unsigned int adr_carte Retour : int	Retour : integer
Raz_flag_conv()	unsigned int adr_carte	
Raz_valid()	unsigned int adr_carte Retour : char	Retour : byte
Read_bit_etat()	unsigned int adr_carte char nr_bit Retour : char	nr_bit : byte Retour : boolean

SETPIO()

Paramètres transmis : Adresse de base de la carte (*Langage C uniquement*)
 Numéro du PIO 8255 (0 ou 1)
 Numéro du Port (Ø= PORT A 1=PORT B 2=PORT C)
 Ancien masque de configuration
 Mode de programmation du Port (1=Entrée 0=Sortie)
 Valeur retournée : Nouveau masque de configuration

En langage C, le Port A du PIO1 doit être initialisé comme suit :
masque=Setpio(adr_carte,1,0,0x9B,0);

**TABLEAU RECAPITULATIF DES PROCEDURES ET DES
PASSAGES DES PARAMETRES**

NOMS PROCEDURES	PARAMETRES C	PARAMETRES PASCAL
Initfastlab()	<i>Non disponible</i>	adr_carte : word
Setchannel()	unsigned int adr_carte unsigned char mode unsigned char voie	mode : byte voie : byte
Setgain()	unsigned int adr_carte unsigned char gain	codecalibre : byte codegain : byte
Start_readad()	unsigned int adr_carte Retour : int	Retour : integer
Multi_ad()	unsigned in adr_carte int segment int offset int nb_voie int nb_acq int tempo	segment : word offset : word nb_voie : word nb_acq : word tempo : word
Ext_readad()	unsigned int adr_carte Retour : int	Retour : integer
Raz_flag_conv()	unsigned int adr_carte	
Raz_valid()	unsigned int adr_carte Retour : char	Retour : byte
Read_bit_etat()	unsigned int adr_carte char nr_bit Retour : char	nr_bit : byte Retour : boolean

suite du tableau

NOMS PROCEDURES	PARAMETRES C	PARAMETRES PASCAL
Write_valid_bit()	unsigned int adr_carte char nr_bit char masque_valid Retour : char	nr_bit : byte masque_valid : byte Retour : byte
Write_devalid_bit()	unsigned int adr_carte char nr_bit char masque_valid Retour : char	nr_bit : byte masque_valid : byte Retour : byte
Install_irq()	int IRQ void interrupt (*ptr)()	IRQ : byte ptr : pointer
Desinstal_irq()	unsigned int IRQ void interrupt (*ptr)()	IRQ : byte
Valid_irq()	unsigned int adr_carte int IRQ	IRQ : byte
Devalid_irq()	unsigned int adr_carte int IRQ	IRQ : byte
Clear_flag_irqext()	unsigned int adr_carte	
Raz_vecteur()	unsigned int adr_carte	
Test_valid_eoi()	unsigned int adr_carte int IRQ	IRQ : byte
Set_tampon	Non disponible	ptr : pointer nr_max : word
Free_tampon()	Non disponible	ptr : pointer nr_max : word
Setda1() & Setda2()	unsigned int adr_carte unsigned int valeur	valeur : word
Multi_setda()	unsigned int adr_carte int valeur1 int valeur2	valeur1 : word valeur2 : word
Raz_dac()	unsigned int adr_carte	

suite du tableau

NOMS PROCEDURES	PARAMETRES C	PARAMETRES PASCAL
Write_valid_bit()	unsigned int adr_carte char nr_bit char masque_valid Retour : char	nr_bit : byte masque_valid : byte Retour : byte
Write_devalid_bit()	unsigned int adr_carte char nr_bit char masque_valid Retour : char	nr_bit : byte masque_valid : byte Retour : byte
Install_irq()	int IRQ void interrupt (*ptr)()	IRQ : byte ptr : pointer
Desinstal_irq()	unsigned int IRQ void interrupt (*ptr)()	IRQ : byte
Valid_irq()	unsigned int adr_carte int IRQ	IRQ : byte
Devalid_irq()	unsigned int adr_carte int IRQ	IRQ : byte
Clear_flag_irqext()	unsigned int adr_carte	
Raz_vecteur()	unsigned int adr_carte	
Test_valid_eoi()	unsigned int adr_carte int IRQ	IRQ : byte
Set_tampon	Non disponible	ptr : pointer nr_max : word
Free_tampon()	Non disponible	ptr : pointer nr_max : word
Setda1() & Setda2()	unsigned int adr_carte unsigned int valeur	valeur : word
Multi_setda()	unsigned int adr_carte int valeur1 int valeur2	valeur1 : word valeur2 : word
Raz_dac()	unsigned int adr_carte	

suite du tableau

NOMS PROCEDURES	PARAMETRES C	PARAMETRES PASCAL
Settimer()	unsigned int adr_carte char nr_timer char nr_compteur char mode char bcd unsigned int valeur	nr_timer : byte nr_compteur : byte mode : byte bcd : byte valeur : word
Readtimer()	unsigned int adr_carte char nr_timer char nr_compteur Retour : unsigned int	nr_timer : byte nr_compteur : byte Retour : word
Raz_timer()	unsigned int adr_carte Retour : char	Retour : byte
Start_timer()	unsigned int adr_carte char nr_timer char nr_compteur char msq_gate Retour : char	nr_timer : byte nr_compteur : byte msq_gate : byte Retour : byte
Stop_timer()	unsigned int adr_carte char nr_timer char nr_compteur char msq_gate Retour : char	nr_timer : byte nr_compteur : byte msq_gate : byte Retour : byte
Inputio()	unsigned int adr_carte char nr_pio char nr_port Retour : char	nr_pio : byte nr_port : byte Retour : byte
Outputio()	unsigned int adr_carte char nr_pio char nr_port char valeur	nr_pio : byte nr_port : byte valeur : byte
Setpio()	unsigned int adr_carte char nr_pio char nr_port char msq_io char mode	nr_pio : byte nr_port : byte msq_io : byte mode : byte

suite du tableau

NOMS PROCEDURES	PARAMETRES C	PARAMETRES PASCAL
Settimer()	unsigned int adr_carte char nr_timer char nr_compteur char mode char bcd unsigned int valeur	nr_timer : byte nr_compteur : byte mode : byte bcd : byte valeur : word
Readtimer()	unsigned int adr_carte char nr_timer char nr_compteur Retour : unsigned int	nr_timer : byte nr_compteur : byte Retour : word
Raz_timer()	unsigned int adr_carte Retour : char	Retour : byte
Start_timer()	unsigned int adr_carte char nr_timer char nr_compteur char msq_gate Retour : char	nr_timer : byte nr_compteur : byte msq_gate : byte Retour : byte
Stop_timer()	unsigned int adr_carte char nr_timer char nr_compteur char msq_gate Retour : char	nr_timer : byte nr_compteur : byte msq_gate : byte Retour : byte
Inputio()	unsigned int adr_carte char nr_pio char nr_port Retour : char	nr_pio : byte nr_port : byte Retour : byte
Outputio()	unsigned int adr_carte char nr_pio char nr_port char valeur	nr_pio : byte nr_port : byte valeur : byte
Setpio()	unsigned int adr_carte char nr_pio char nr_port char msq_io char mode	nr_pio : byte nr_port : byte msq_io : byte mode : byte