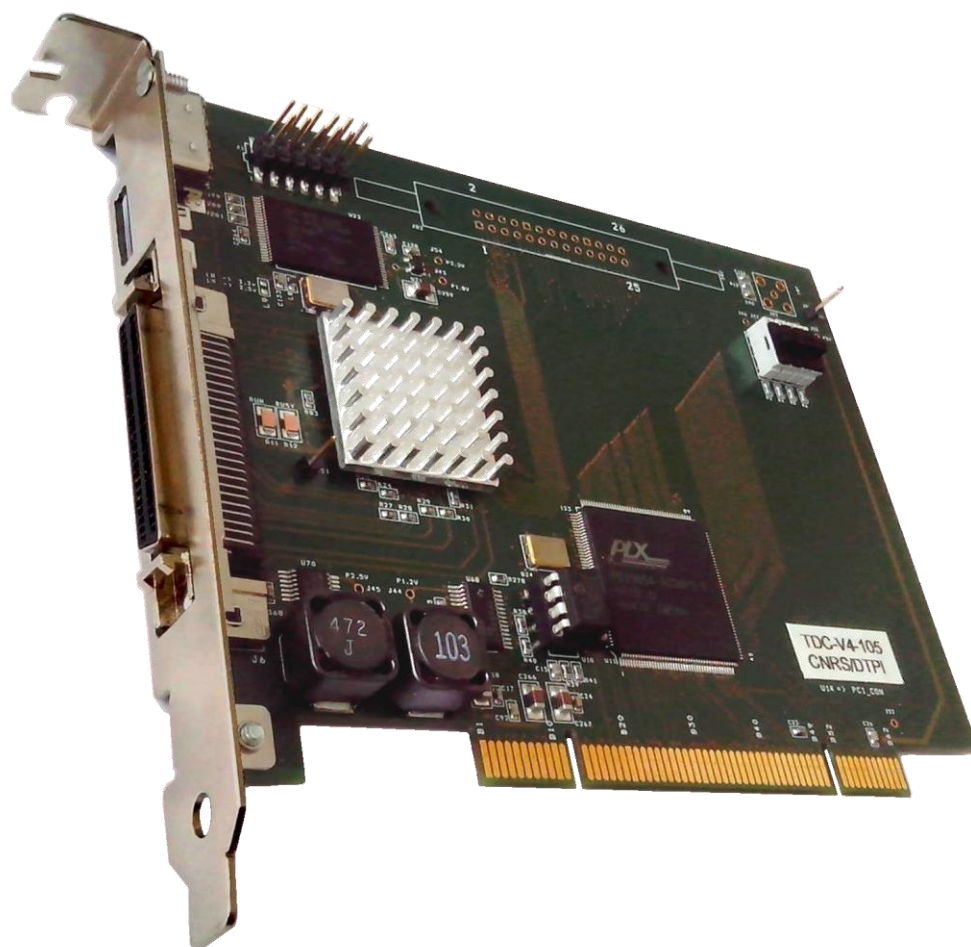


# TDC-V4

Time to Digital Converter



**Notice Utilisateur**  
*v1.2 - Février 2020*

## Historique des modifications

- Améliorations de mise en forme
- Reformulations dans le but d'améliorer la clarté des explications
- Correction de l'inversion TRIGGER\_GATE et EXTERNAL\_GATE de registre de configuration FORWARD\_MODE
- Installation du pilote v12.21 signé (Windows 7/8/10) et prise en charge du TDC-V4-E

# Table des matières

Historique des modifications.....	2
1. Remarques préliminaires .....	5
2. L'architecture de la carte TDC-V4.....	6
2.a. la localisation des composants électroniques principaux.....	6
2.b. l'architecture logique .....	7
2.c. les Entrées-Sorties (connecteur SCSI-2 50b) .....	8
3. Les Modes de Fonctionnement .....	9
3.a. la définition de l'Evènement (EVENT) .....	9
3.b. les Modes d'Acquisition.....	10
3.c. les Modes de Déclenchement .....	12
3.d. les Durées d'Analyse.....	13
3.e. les Modes de Codage .....	16
3.f. les Modes d'Acquittement .....	20
3.g. récapitulatif des Modes de Fonctionnement.....	21
4. Les commandes .....	22
4.a. les Commandes d'Initialisation.....	22
4.b. la Commande RUN .....	23
4.c. la Commande HOST_ACK .....	23
4.d. la Commande HOST_START.....	24
5. Sélection de la polarité des signaux de Service.....	25
6. Le Format des Données .....	26
7. L'encapsulation des Données .....	28
8. La collecte des Données par le bus PCI .....	30
9. Caractéristiques techniques .....	33

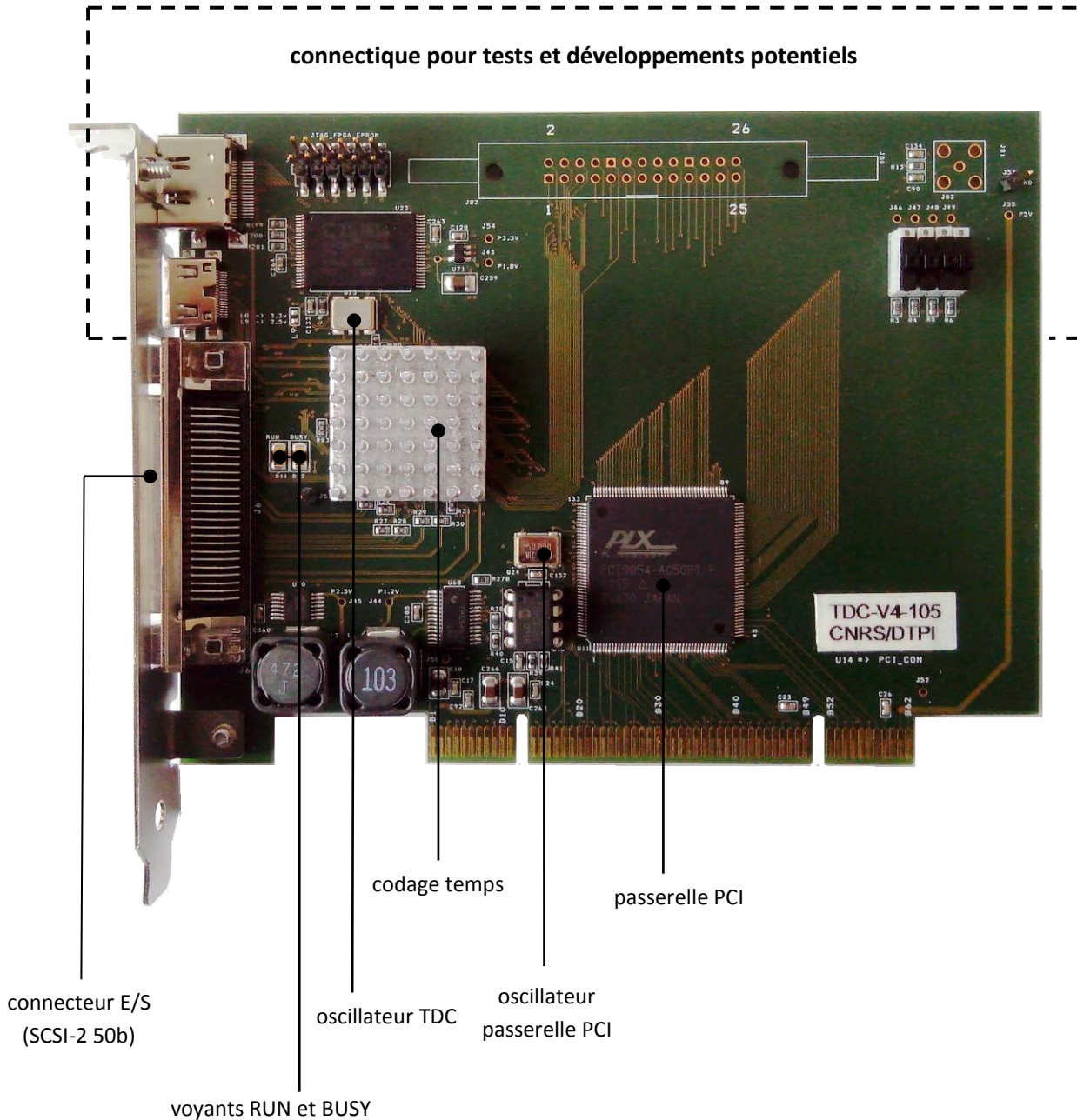
10.	Débit de collecte des données .....	34
10.a.	les 3 modes de collecte des Données.....	34
10.b.	les performances en débit maximal sans pertes.....	36
11.	Installation de TDC-V4.....	39
11.a.	la configuration requise.....	39
11.b.	l'installation .....	39
12.	Fonctions de la DLL.....	41
12.a.	les fichiers requis.....	41
12.b.	les fonctions .....	41
13.	Exemple de programme .....	43
14.	Récapitulatif des registres de TDC-V4 .....	45

# 1. Remarques préliminaires

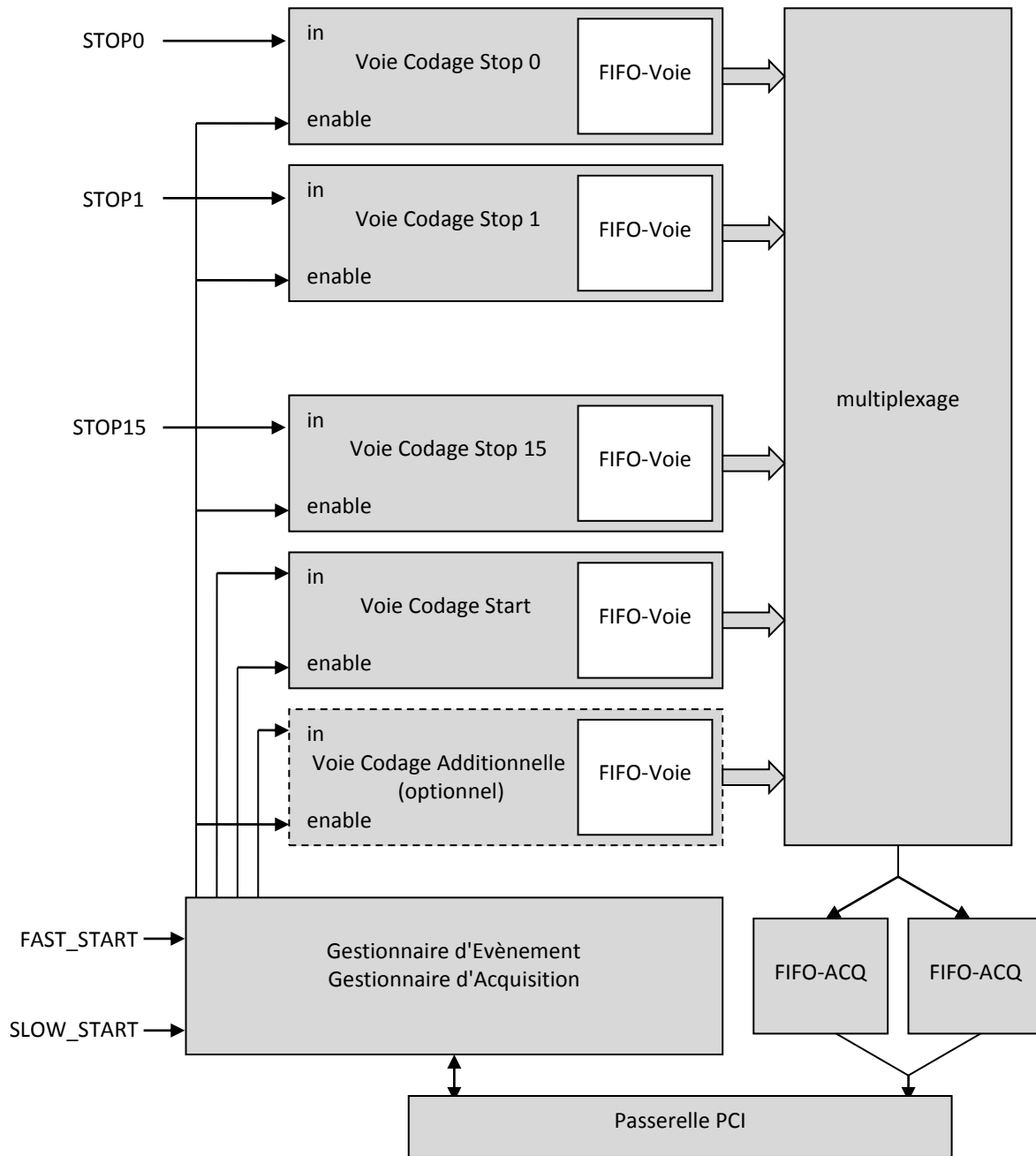
- La carte TDC-V4 doit être insérée dans un emplacement libre d'un bus PCI 32 ou 64 bits.
- Le fonctionnement de la carte TDC-V4 est garanti pour une température de l'air environnant la carte inférieure à 60°C. Si les conditions expérimentales empêchent de respecter cette température, un ventilateur peut être installé sur le composant dissipatif (puce de codage temps doté d'un dissipateur, voir la figure du chapitre 2.a); une alimentation de 5,0V est prévue à cet effet. Une autre approche consiste à limiter à 10 le nombre de voies de codage (option TDC-V4-10 : 1 voie Start + 8 voies Stop + 1 voie Additionnelle).
- **Attention :**  
L'affectation des broches du connecteur SCSI-2 est spécifique à la carte TDC-V4 ; celle-ci ne peut être connectée à un bus SCSI. Le connecteur SCSI-2 comporte notamment deux broches qui délivrent une tension d'alimentation +12v / 800mA permettant l'auto alimentation du rack d'interface ISIBox (cf. chapitre 2.c et la Notice Utilisateur ISIBox).
- Lors de l'insertion de la carte TDC-V4 dans l'emplacement PCI, il est impératif de mettre hors tension l'ordinateur.
- Du point de vue du logiciel d'extraction des données, la carte TDC-V4 est compatible (compatibilité ascendante) avec la famille de TDC ISITime01.
- La Notice Utilisateur adopte la convention graphique suivante :
  - en bleu, les signaux d'Entrée-Sortie, ex: **FAST\_START**; **BUSY**
  - en rouge, les modes de fonctionnement, ex: **STATIC\_STOP\_ENABLE**; **EBE\_HOST\_ACK**
  - en vert, les commandes, ex: **RUN**; **HOST\_START**
  - en noir, les états internes, ex : Busy
- Les caractéristiques chiffrées contenues dans cette notice sont relatives à la version TDC-V4 au pas de 120 ps.
- Lorsque le rack d'Interface ISIBox est utilisé en interface de TDC-V4, le standard des Entrées-Sorties et les fonctionnalités sont décrits dans la Notice Utilisateur ISIBox.

## 2. L'architecture de la carte TDC-V4

### 2.a. la localisation des composants électroniques principaux



## 2.b. l'architecture logique



**FIFO-Voie** : - tampon simple ; profondeur : 512 mots de Codages Temps

- "tamponne" les Codages Temps associés à un même Evènement

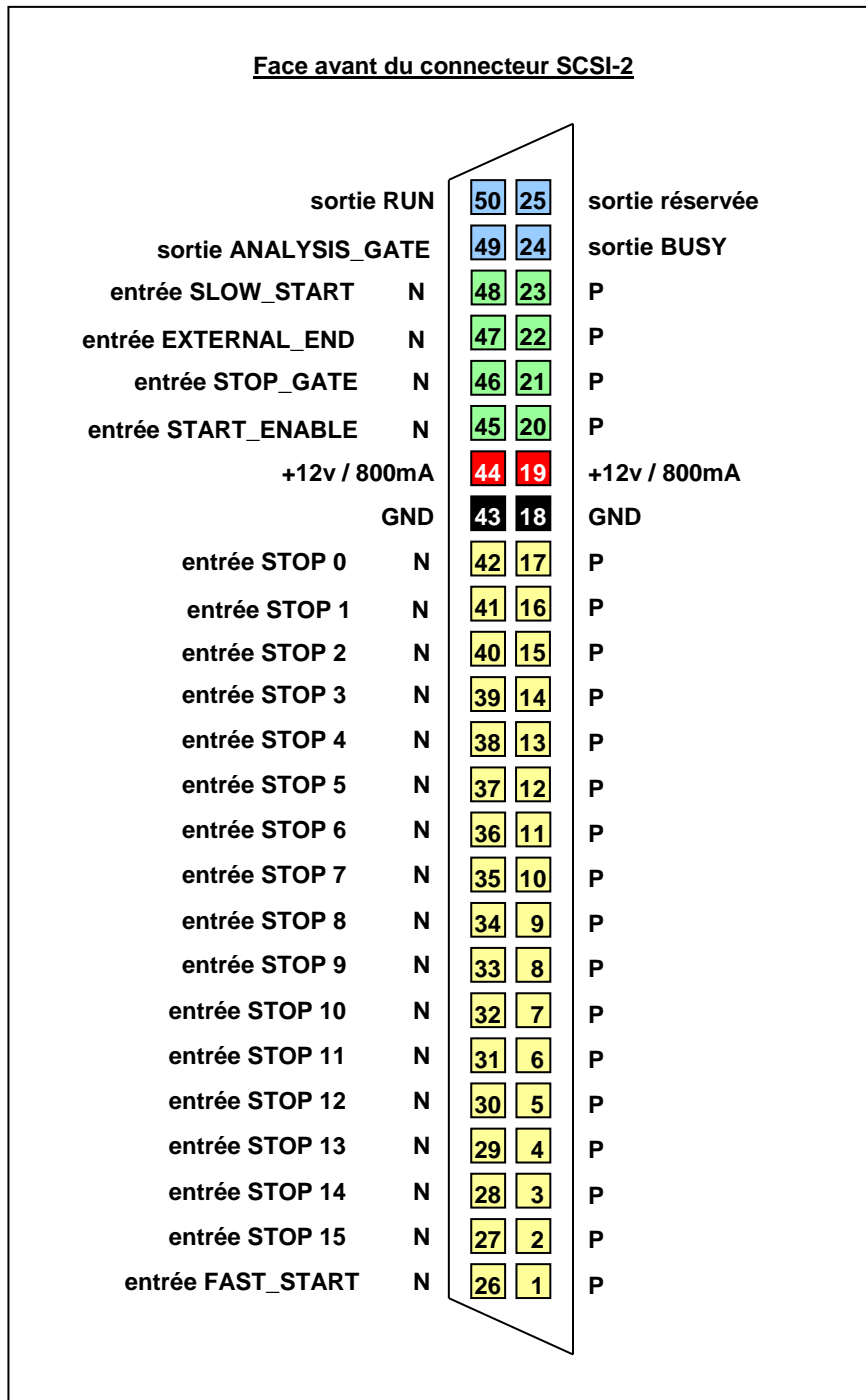
- un nouvel Evènement ne peut être accepté tant que la FIFO-Voie n'est pas vidée des codages relatifs à l'Evènement précédent ; plusieurs Evènements ne peuvent donc coexister dans une FIFO-Voie.

**FIFO-ACQ** : 2 tampons en bascule ; profondeur 509 mots (32 bits) pour chaque tampon

➤ en mode Event By Event (**EBE\_HOST\_ACK**, **EBE\_TDC\_AUTO** et **EBE\_EXT\_END**), ne tamponnent que les mots de Codage ou de Service d'un même Evènement

➤ en mode Accumulation (**ACC\_TDC\_AUTO**), plusieurs Evènements peuvent cohabiter dans un tampon FIFO-ACQ.

## 2.c. les Entrées-Sorties (connecteur SCSI-2 50b)



■ TTL

■ LVDS (signaux lents)

■ LVDS (signaux rapides)



## 3. Les Modes de Fonctionnement

### 3.a. la définition de l'Evènement (EVENT)

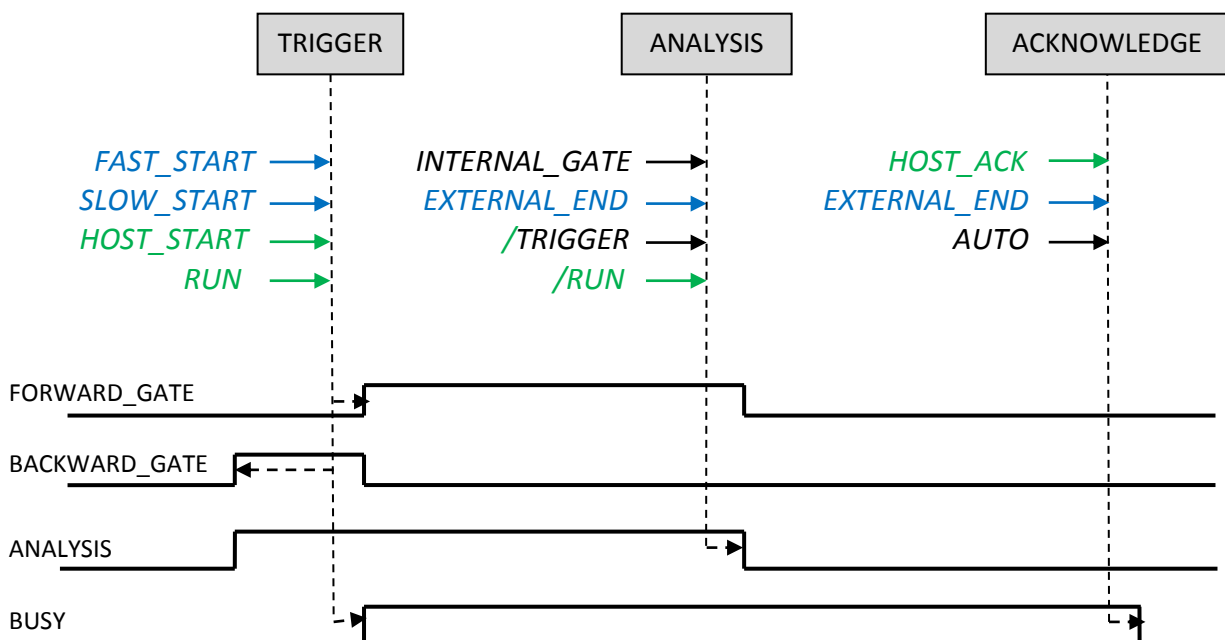
Un Evènement regroupe toutes les informations (mots de Codage et mots de Service) relatifs à une action physique. Un Evènement peut donc être constitué d'un nombre faible de mots (ex : localisation de quelques particules corrélées après interaction), d'un nombre de mots relativement grand (ex : spectroscopie de masse par temps de vol) ou d'un très grand nombre de mots (ex : expériences « atomes froids »).

D'un point de vue logique, un Evènement est caractérisé par :

- un signal de Déclenchement (TRIGGER)
- une durée d'Analyse (ANALYSIS)
- un signal d'Acquittement (ACKNOWLEDGE)
- un mode d'Acquisition.

Entre les signaux TRIGGER et ACKNOWLEDGE, TDC-V4 est dans l'état Busy : il ne peut traiter un nouvel Evènement. L'état Busy = 1 est visualisé par un voyant sur le circuit imprimé de TDC-V4 ainsi qu'en face avant du rack d'interface ISIBox.

Le traitement d'un Evènement est représenté par le chronogramme ci-après :



### **3.b. les Modes d'Acquisition**

Chaque Voie de Codage dispose d'un tampon FIFO-Voie de 512 mots et tous ces tampons sont vides au moment du déclenchement d'un nouvel Evènement. Ainsi, sauf pour des Evènements présentant un débit important de signaux Stop, une Voie ne peut subir de pertes dues au débit présent sur une autre.

Les données issues des tampons FIFO-Voies de la Voie de Codage Start et des Voies de codage Stop sont multiplexées et stockées dans l'un ou l'autre des 2 tampons de sortie FIFO-ACQ montés en bascule, au fur et à mesure que les signaux sont codés et dès le début de la durée d'Analyse (cf. chapitre 2.b : l'architecture logique).

Le mécanisme de multiplexage assure une probabilité d'accès égale pour toutes les Voies ; seul le premier codage de la Voie de Codage Start dispose d'un accès prioritaire. Ce mécanisme assure qu'une Voie à fort débit ne peut imposer de pertes à des Voies à faible débit. Dans les tampons de sortie, l'ordre chronologique est respecté pour les codages d'une même Voie de Codage ; il peut ne pas être respecté entre les codages de Voies différentes.

Quel que soit le mode de collecte des données sélectionné (voir ci-dessous), le nombre de données constituant un Evènement est illimité : le logiciel d'acquisition est alerté chaque fois qu'un tampon de sortie FIFO-ACQ est plein et disponible pour lecture et informé de la taille des données pertinentes ( $\leq 509$  mots de 32 bits).

Une fois la dernière donnée relative à un Evènement (mot de Service EOE) stockée dans l'un des tampons de sortie FIFO-ACQ, TDC-V4 propose 2 modes de collecte des données :

- Collecte en Mode Evènement par Evènement
- Collecte en Mode Accumulation

#### **➤ collecte en Mode Evènement par Evènement**

Le logiciel d'acquisition est alerté qu'un tableau de données est disponible pour la lecture et informé de la taille des données pertinentes ( $\leq 509$  mots de 32 bits). TDC-V4 attend l'acquiescement Evènement (cf. chapitre 3.3.f : les Modes d'Acquiescement) avant d'accepter un nouveau Déclenchement.

Du fait de la latence du logiciel d'acquisition, ce mode de régulation est peu efficace du point de vue du débit moyen d'acquisition et d'autant moins efficace que la taille de l'Evènement est faible (cf. chapitre 10 : débit moyen de collecte des données). Ce mode de régulation est néanmoins obligatoire si TDC-V4 est corrélé avec d'autres tiroirs de codage car on doit attendre que toutes les sources de données constitutives d'un Evènement aient été collectées par le logiciel d'acquisition avant d'accepter le codage d'un nouvel Evènement.

### ➤ **collecte en Mode Accumulation**

TDC-V4 auto-acquitte l'Evènement en cours et est prêt à accepter un nouveau Déclenchement. TDC-V4 empile donc, à la suite, dans les tampons de sortie FIFO-ACQ les données relatives à des Evènements successifs.

Ce mode de régulation est plus efficace du point de vue du débit moyen d'acquisition et d'autant plus efficace que la taille de l'Evènement est faible (cf. chapitre 10 : débit moyen de collecte des données). Il est adapté aux expériences ne comportant que TDC-V4 comme source de données.

### **Cas particulier : collecte en Mode Analyse Continue (Continuing Analysis)**

Dans cette configuration, TDC-V4 ne filtre plus les signaux au moyen de Déclenchement, Durée d'Analyse et Acquiescement : tous les signaux arrivant sur les entrées validées des Voies Start et Stop sont codés à partir d'un Déclenchement unique et pendant tout le temps où le TDC reste en Run.

Par exemple, si on a choisi le Déclenchement par la **TRIGGER\_SOURCE = RUN + HOST\_START** (cf. chapitre 3.3.c : les Modes de Déclenchement), l'ensemble du Run est logiquement considéré comme un seul Evènement de durée d'analyse infinie, tous les signaux sont codés durant la durée de Run et tous les codages sont empilés en continu dans les tampons FIFO-ACQ.

C'est le mode de travail le plus efficace du point de vue du débit (cf. chapitre 10 : débit moyen de collecte des données) car la tâche de filtrage des données et de construction d'Evènements est reportée sur le logiciel d'acquisition.

Ce mode de travail est obtenu en sélectionnant **FORWARD\_MODE = EXTERNAL\_GATE** et en ne connectant rien sur l'entrée **EXTERNAL\_END**.

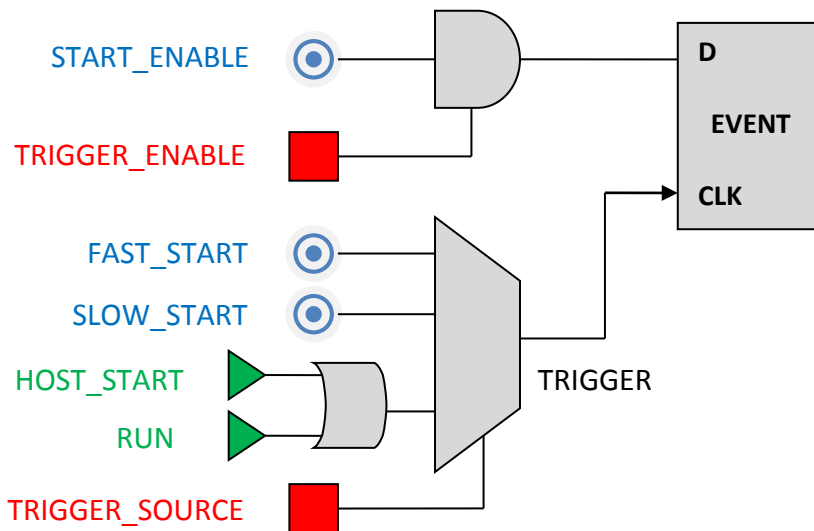
### 3.c. les Modes de Déclenchement

Un Evènement peut être déclenché à partir de 4 sources :

- l'un des 2 signaux d'entrée **FAST\_START** ou **SLOW\_START** ; lorsque le rack d'interface ISIBox est utilisé, **FAST\_START** est un signal NIM et **SLOW\_START** est un signal TTL
- la commande **HOST\_START** ; ce type de déclenchement peut être utilisé en test ou pour mesurer un bruit de fond
- la commande **RUN** ; cette commande permet de déclencher l'Analyse de durée infinie mentionnée ci-dessus sans obligation de générer un signal de déclenchement matériel ou logiciel

Par ailleurs, le déclenchement peut être validé ou inhibé par le signal d'entrée **START\_ENABLE**.

Les sources de déclenchement sont résumées dans la figure suivante :



Les Modes de Déclenchement sont contrôlés par les registres suivants :

nom	adresse PCI	data		
<b>TRIGGER_SOURCE</b>	240 0000h	$2^1$	$2^0$	<i>source</i>
		0	0	reserved
		0	1	FAST_START
		1	0	SLOW_START
		1	1	RUN + HOST_START
<b>TRIGGER_ENABLE</b>	220 0000h	$2^5$	$2^4$	<i>validation</i>
		0	1	START_ENABLE as Enable On
		0	0	START_ENABLE as Enable Off

### 3.d. les Durées d'Analyse

#### ➤ FORWARD ANALYSIS

Une fois l'Evènement déclenché, les signaux entrant dans les Voies de Codage Temps sont codés durant une durée suivant immédiatement l'instant de déclenchement et se terminant selon l'un des 4 modes Forward suivants, par :

- un compteur interne à TDC-V4 (**INTERNAL\_GATE**)
- le signal d'entrée **EXTERNAL\_END** (**EXTERNAL\_GATE**)
- le plus rapide des 2 signaux d'entrée : **EXTERNAL\_END** ou du compteur interne à TDC-V4 (**EXTERNAL\_GATE / WATCHDOG**)
- la durée du signal TRIGGER lui-même

Le dernier mode est le plus simple à mettre en œuvre lorsqu'on souhaite ajuster précisément la durée d'analyse : par exemple, sélectionner **FAST\_START** comme source de déclenchement et régler la durée du signal d'entrée **FAST\_START** en conséquence.

Une durée Forward infinie est obtenue en sélectionnant le mode **EXTERNAL\_GATE** et en supprimant tout signal sur l'entrée **EXTERNAL\_END**. De plus, le déclenchement d'une analyse de durée infinie peut être obtenu sans signaux de déclenchement en sélectionnant la source de trigger **RUN + HOST\_START** (cf. chapitre 3.3.c).

Le **FORWARD\_MODE**, et la Durée d'Analyse **FORWARD\_DURATION** si **INTERNAL\_GATE** est choisi, sont contrôlés par les registres suivants :

nom	adresse PCI	data		
<b>FORWARD_MODE</b>	1C0 0000h	$2^6$	$2^5$	<i>source de la fin d'analyse</i>
		0	0	TRIGGER_GATE
		0	1	INTERNAL_GATE
		1	0	EXTERNAL_GATE
		1	1	EXTERNAL_GATE / WATCHDOG
<b>FORWARD_DURATION</b>	1C0 0000h	$2^4 2^3 2^2 2^1 2^0$		<i>Durée d'Analyse</i>
		0		180 ns
		1		260 ns
		2		340 ns
		3		500 ns
		4		660 ns
		5		980 ns
		6		1,3 µs
		7		1,9 µs
		8		2,5 µs
		9		3,8 µs
		10		5,1 µs
		11		7,6 µs
		12		10 µs
		13		15 µs
		14		20 µs
		15		30 µs
		16		41 µs
		17		61 µs
		18		82 µs
		19		120 µs
		20		160 µs
		21		240 µs
		22		320 µs
		23		490 µs
		24		650 µs
		25		980 µs
		26		1,3 ms
		27		1,9 ms
		28		2,6 ms
		29		3,9 ms
		30		5,2 ms
31		7,8 ms		

➤ **BACKWARD ANALYSIS**

Par ailleurs, TDC-V4 peut coder les signaux entrant dans les Voies de Codage Temps durant une durée **BACKWARD\_DURATION** précédant immédiatement l'instant de déclenchement. Ce mode évite d'avoir à retarder les signaux d'entrée STOP qui arriveraient avant le signal utilisé pour le déclenchement de l'Evènement.

Le **BACKWARD\_MODE** et sa durée d'analyse sont contrôlés par les registres suivants :

nom	adresse PCI	data	
<b>BACKWARD_MODE</b>	180 0000h	$2^5$	<i>validation</i>
		0	OFF
		1	ON
<b>BACKWARD_DURATION</b>	180 0000h	$2^3 2^2 2^1 2^0$	<i>Durée d'Analyse</i>
		0	0 ns
		1	0 ns
		2	120 ns
		3	240 ns
		4	360 ns
		5	610 ns
		6	860 ns
		7	1,3 $\mu$ s
		8	1,8 $\mu$ s
		9	2,8 $\mu$ s
		10	3,8 $\mu$ s
		11	5,7 $\mu$ s
		12	7,7 $\mu$ s
13	11,6 $\mu$ s		

Remarque :

Lorsque **BACKWARD\_MODE** n'est pas sélectionné, TDC-V4 présente un temps mort entre l'instant de déclenchement de l'Evènement et l'instant où les Voies de Codage Temps sont utilisables (cf. Start Latency Time au chapitre 9 : Caractéristiques techniques).

Sélectionner **BACKWARD\_MODE** avec **BACKWARD\_DURATION** égale à 0 ns permet d'annuler le Start Latency Time.

### 3.e. les Modes de Codage

#### ➤ Voie de Codage Start : mode **NEXT\_START**

Cette Voie de Codage Temps est dédiée au codage de l'instant de Déclenchement de l'Evènement. Lors de l'encapsulation des données (cf. chapitre 7 : L'encapsulation des données) le Mot de Codage délivré par cette Voie est obligatoirement placé en tête des données relatives à l'Evènement.

L'encapsulation des données prévoyant nécessairement de clore les données relatives à un Evènement par le Mot de Service EOE, la Voie de codage Start peut, sans risque de confusion au dépouillement, coder des signaux arrivant après l'instant de Déclenchement.

En mode **NEXT\_START**, TDC-V4 permet de sélectionner comme Start suivant l'un des 3 signaux suivants (cf. le schéma de principe de la page suivante) :

- **FAST\_START**
- **SLOW\_START**
- **FORWARD\_GATE**

L'utilisateur peut notamment mettre en œuvre avec le rack d'Interface ISIBox les 4 montages du tableau ci-dessous :

signal de Déclenchement	signaux suivants	commentaire
FAST_START NIM	FAST_START NIM	détection d'empilement d'Evènement
FAST_START NIM	SLOW_START TTL	Voie supplémentaire de codage Stop (entrée TTL)
SLOW_START TTL	FAST_START NIM	Voie supplémentaire de codage Stop (entrée NIM)
SLOW_START TTL	SLOW_START TTL	détection d'empilement d'Evènement

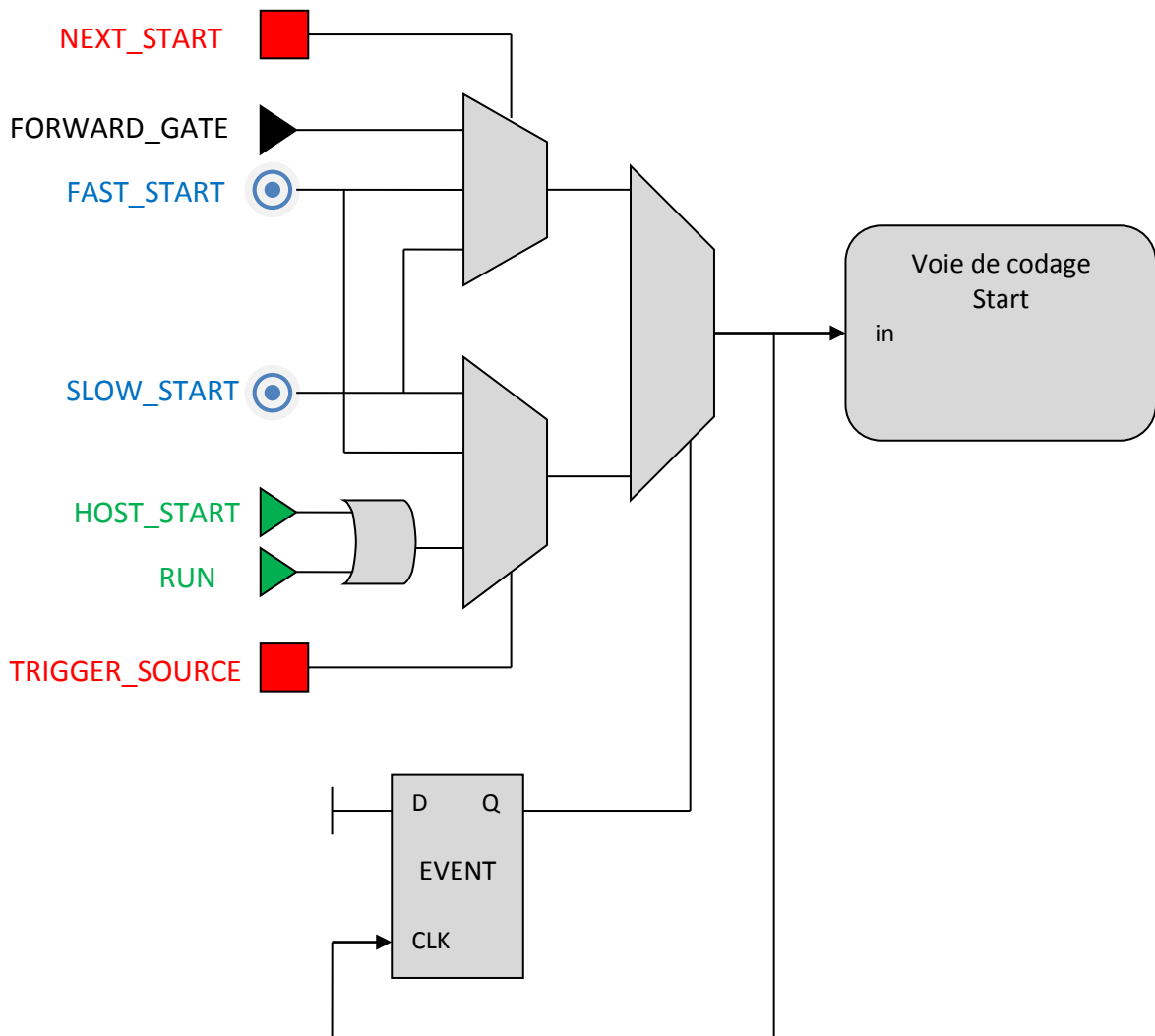
La sélection du signal **FORWARD\_GATE** comme Start suivant permet de contrôler la valeur effective de la durée d'analyse lorsque celle-ci est variable (Mode **EXTERNAL\_GATE**).

Le mode **NEXT\_START** est contrôlé par le registre suivant :

nom	adresse PCI	data		
<b>NEXT_START</b>	220 0000h	2 <sup>1</sup>	2 <sup>0</sup>	<i>source des signaux suivants</i>
		0	0	OFF
		0	1	FAST_START
		1	0	SLOW_START
		1	1	FORWARD_GATE



principe de fonctionnement du mode **NEXT START** avec la Voie de codage Start



➤ **Voie de Codage Start : EVENT LABELLING**

L'état du signal d'entrée **START\_ENABLE** à l'instant du Déclenchement peut être mémorisé et restitué dans le Mot de Codage de la Voie Start (bit EL ; cf. chapitre 6 : Le Format des Données).

Cette fonction est contrôlée par le registre suivant :

nom	adresse PCI	data		
<b>EVENT LABELLING</b>	220 0000h	$2^5$	$2^4$	<i>validation EVENT LABELLING</i>
		1	1	START_ENABLE as Label On
		1	0	START_ENABLE as Label Off

➤ **Voie de Codage Additionnelle (optionnel)**

La Voie de Codage Additionnelle peut coder l'un des 2 signaux d'entrée suivants :

- SLOW\_START
- EXTERNAL\_END

Si l'un ou l'autre de ces 2 signaux d'entrée n'est pas utilisé (Déclenchement pour SLOW\_START ; Durée d'Analyse pour EXTERNAL\_END), la Voie de Codage Additionnelle permet d'étendre le nombre de Voies Stop de 16 à 17 et, si le rack d'Interface ISIBox est utilisé, le signal d'entrée de la Voie de Codage Additionnelle est en niveau TTL.

La Voie de Codage Additionnelle ne bénéficie pas des possibilités de validation des Voies de Codage Stop (voir paragraphe suivant).

La sélection du signal d'entrée est contrôlée par le registre suivant :

nom	adresse PCI	data	
EXTRA_CHANNEL	à définir	$2^0$	sélection signal d'entrée
		0	SLOW_START
		1	EXTERNAL_END

➤ **Voies de Codage Stop : validation**

Les Voies de Codage Stop peuvent être validées ou inhibées :

- statiquement pour chacune des Voies
- dynamiquement pour l'ensemble des 16 Voies par le signal d'entrée STOP\_GATE

La sélection de la validation des Voies de Codage Stop est contrôlée par les registres suivants :

nom	adresse PCI	data	
STATIC_STOP_ENABLE	120 0000h	$2^i$	validation statique de la voie <i>i</i>
		0	OFF
		1	ON
DYNAM_STOP_ENABLE	220 0000h	$2^3$	validation par STOP_GATE
		0	OFF
		1	ON

➤ **Voies de Codage Stop : indication de Débordement**

TDC-V4 vide les tampons FIFO-Voie de chacune des Voies de Codage dès le début de l'Analyse. Néanmoins, si le débit de l'expérience est supérieur au débit d'acquisition de données, le tampon FIFO-Voie d'une Voie de Codage peut déborder.

Le tampon FIFO-Voie passe alors à l'état "Full" et le premier Mot de Codage qui sera délivré par la Voie de Codage sera marqué dans son champ de Label par le bit OF (cf. chapitre 6 : Le Format des Données).

➤ **Voie de Service Range Extension (REXT ; optionnel)**

TDC-V4 délivre des Mots de Codage et des Mots de Service au format 32 bits. Le mot de 32 bits est décomposé en 2 champs (cf. chapitre 6 : Le Format des Données) :

- le champ DATA de 26 bits
- le champ LABEL de 6 bits.

Le champ DATA limite donc la plage de mesure en temps des Voies de Codage Temps à  $2^{26} \times 120 \text{ ps} = 7,8 \text{ ms}$ .

La mise en œuvre de la Voie de Service Range Extension permet l'extension de la plage de mesure en temps au-delà de 7,8 ms pour toutes les voies de codage en restant avec un format de 32 bits. L'extension du codage au-delà de 26 bits est alors à la charge du logiciel d'acquisition.

La Voie de Service Range Extension n'est pas équipée de tampon FIFO-Voie.

➤ **Voies de Codage Temps: optimisation RESOLUTION/DNL**

TDC-V4 dispose d'un mécanisme permettant de privilégier soit la caractéristique de résolution (au dépend de la caractéristique de DNL<sup>1</sup>) soit la caractéristique de DNL (au dépend de la caractéristique de résolution). Cette optimisation s'applique pour l'ensemble des Voies de Codage Temps (Voies Stop, Voie Additionnelle, Voie Range Extension).

Cette optimisation est contrôlée par le registre suivant :

nom	adresse PCI	data	
<b>OPTIMIZATION</b>	220 0000h	2 <sup>2</sup>	<i>caractéristique optimisée</i>
		0	RESOLUTION
		1	DNL

Note 1 : Differential Non Linearity (quantifie l'inégalité de la largeur réelle des bins)

### 3.f. les Modes d'Acquittement

Les possibilités d'Acquittement de l'Evènement dépendent du Mode d'Acquisition sélectionné selon le tableau suivant :

Mode d'Acquisition	Mode d'Acquittement	
Evènement par Evènement	EBE_HOST_ACK	TDC-V4 ne présente au logiciel d'acquisition que les données relatives à un seul Evènement et n'accepte un nouveau Déclenchement d'Evènement qu'après avoir été acquitté par le logiciel d'acquisition par la commande HOST_ACK.
	EBE_EXT_END	TDC-V4 ne présente au logiciel d'acquisition que les données relatives à un seul Evènement et n'accepte un nouveau Déclenchement d'Evènement qu'après avoir été acquitté par le signal EXTERNAL_END.
	EBE_TDC_AUTO	TDC-V4 ne présente au logiciel d'acquisition que les données relatives à un seul Evènement et accepte un nouveau Déclenchement d'Evènement à partir du moment où la totalité des données relatives à cet Evènement ont été stockées dans les tampons de sortie FIFO-ACQ. <i>Ce Mode d'Acquittement suppose que l'utilisateur maîtrise le signal de Déclenchement afin d'assurer la corrélation des diverses sources de données.</i>
Accumulation	ACC_TDC_AUTO	TDC-V4 empile à la suite dans les tampons de sortie FIFO-ACQ les données relatives à des Evènements successifs. Chaque Evènement est automatiquement acquitté du moment que la totalité des données relatives à cet Evènement ont été stockées.

La sélection du Mode d'Acquittement est contrôlée par le registre suivant :

nom	adresse PCI	data		
		2 <sup>1</sup>	2 <sup>0</sup>	modes d'acquittement
ACK_MODE	260 0000h	0	0	ACC_TDC_AUTO
		0	1	EBE_TDC_AUTO
		1	0	EBE_EXT_END (optionnel)
		1	1	EBE_HOST_ACK

### 3.g. récapitulatif des Modes de Fonctionnement

valeurs par défaut ↘

<p><b>Les Modes de Déclenchement</b></p>	<ul style="list-style-type: none"> <li>• TRIGGER SOURCE FAST_START SLOW_START RUN + HOST_START</li> <li>• TRIGGER ENABLE START_ENABLE as Enable</li> </ul>	<p>3-c</p>	<p><b>FAST_START</b></p> <p><b>OFF</b></p>
<p><b>Les Durées d'Analyse</b></p>	<ul style="list-style-type: none"> <li>• FORWARD ANALYSIS INTERNAL_GATE / duration table EXTERNAL_GATE EXTERNAL_GATE / WATCHDOG TRIGGER_GATE</li> <li>• BACKWARD ANALYSIS / duration table</li> </ul>	<p>3-d</p>	<p><b>INTERNAL (180ns)</b></p> <p><b>OFF (0ns)</b></p>
<p><b>Les Modes de Codage</b></p>	<p><u>Voie de Codage START</u></p> <ul style="list-style-type: none"> <li>• NEXT_START FAST_START SLOW_START FORWARD_GATE</li> <li>• EVENT LABELLING START_ENABLE as Label</li> </ul> <p><u>Voie de Codage ADDITIONNELLE (optionnel)</u></p> <ul style="list-style-type: none"> <li>• INPUT SLOW_START EXTERNAL_END</li> </ul> <p><u>Voies de Codage STOP</u></p> <ul style="list-style-type: none"> <li>• STATIC_STOP_ENABLE</li> <li>• DYNAM_STOP_ENABLE</li> </ul> <p><u>Voie de Service RANGE EXTENSION (optionnel)</u></p> <p><u>Débordement Voies de Codage Stop</u></p> <p><u>RESOLUTION/DNL Optimisation</u></p>	<p>3-e</p>	<p><b>OFF</b></p> <p><b>OFF</b></p> <p><b>toutes actives OFF</b></p> <p><b>RESOLUTION</b></p>
<p><b>Les Modes d'Acquittement</b></p>	<p><u>Mode de collecte des données</u></p> <p>EBE_HOST_ACK EBE_EXT_END (optionnel) EBE_TDC_AUTO ACC_TDC_AUTO</p>	<p>3-f</p>	<p><b>ACC_TDC_AUTO</b></p>

Remarque :

Les registres de modes de configuration ne sont accessibles qu'en écriture.

## 4. Les commandes

### 4.a. les Commandes d'Initialisation

#### ➤ Initialisation par défaut des Modes de Fonctionnement

La Commande est contrôlée par le registre suivant :

nom	adresse PCI	data	
INIT_CONF	140 0000h	2 <sup>0</sup>	<i>commande</i>
		0	OFF
		1	ON

Une Commande **INIT\_CONF** = ON nécessite d'être suivie de la Commande **INIT\_CONF** = OFF.

Cette commande doit être utilisée avec prudence : après avoir été lancée, les Modes de Fonctionnement de TDC-V4 prennent leurs valeurs par défaut comme définies dans le tableau récapitulatif chapitre 3.g et ne correspondent plus à celles éventuellement affichées par l'utilitaire de Configuration.

#### ➤ Initialisation de l'état logique du TDC

La Commande est contrôlée par le registre suivant :

nom	adresse PCI	data	
INIT_TDC	160 0000h	2 <sup>0</sup>	<i>commande</i>
		0	ON
		1	OFF

Une Commande **INIT\_TDC** = ON nécessite d'être suivie d'une Commande **INIT\_TDC** = OFF.  
L'état logique du TDC est automatiquement initialisé après la mise sous tension de TDC-V4.

## 4.b. la Commande RUN

La Commande RUN a plusieurs fonctions :

- dans tous les cas, RUN = ON conditionne l'acceptation de Déclenchement d'Evènement
- dans le cas où TRIGGER\_SOURCE = RUN + HOST\_START (cf. chapitre 3.c : les Modes de Déclenchement), RUN = ON déclenche le traitement d'un Evènement
- dans le cas d'une Durée Forward infinie (cf. chapitre 3.d : les Durées d'Analyse), RUN = OFF clôt la Durée d'Analyse
- dans tous les cas RUN = OFF :
  - invalide l'acceptation de Déclenchement d'Evènement
  - transfert au logiciel d'acquisition, si celui-ci continue la collecte, de toutes les données encore présentes dans TDC-V4
  - clôt le dernier tableau de données avec le Mot de Service EOR (cf. chapitre 6 : Le Format des Données).

La Commande est contrôlée par le registre suivant :

nom	adresse PCI	data	
RUN	1A0 0000h	2 <sup>0</sup>	commande
		0	OFF
		1	ON

L'état RUN = ON est visualisé par un voyant sur le circuit imprimé de TDC-V4 ainsi qu'en face avant du rack d'Interface ISIBox.

## 4.c. la Commande HOST\_ACK

La Commande HOST\_ACK est utilisée dans le Mode d'Acquittement EBE\_HOST\_ACK (cf. chapitre 3.f : les Modes d'Acquittement) c'est-à-dire lors de la régulation du flux Event By Event.

La Commande est contrôlée par le registre suivant :

nom	adresse PCI	data	
HOST_ACK	2C0 0000h	2 <sup>0</sup>	commande
		0	OFF
		1	ON

Une Commande HOST\_ACK = ON nécessite d'être suivie d'une Commande HOST\_ACK = OFF.

#### 4.d. la Commande HOST\_START

La Commande **HOST\_START** est utilisée dans le Mode Déclenchement **TRIGGER\_SOURCE = RUN + HOST\_START** (cf. chapitre 3.c : les Modes de Déclenchement).

La Commande est contrôlée par le registre suivant :

nom	adresse PCI	data	
HOST_START	2E0 0000h	2 <sup>0</sup>	<i>commande</i>
		0	OFF
		1	ON

Une Commande **HOST\_START = ON** nécessite d'être suivie d'une Commande **HOST\_START = OFF**.

#### Remarque :

Les registres de commandes ne sont accessibles qu'en écriture.



## 5. Sélection de la polarité des signaux de Service

La sélection de la polarité de chacun des 4 signaux d'entrée **START\_ENABLE**, **STOP\_GATE**, **EXTERNAL\_END** et **SLOW\_START** ainsi que de chacun des 2 signaux de sortie **ANALYSIS\_GATE** et **BUSY** est contrôlée par le registre suivant :

nom	adresse	data	
<b>POLARITY</b>	280 0000h	$2^5$ $2^4$ $2^3$ $2^2$ $2^1$ $2^0$	<i>polarité</i>
		x x x x x 0	entrée START_ENABLE : <i>actif bas</i>
		x x x x x 1	entrée START_ENABLE : <i>actif haut</i>
		x x x x 0 x	entrée STOP_GATE : <i>actif bas</i>
		x x x x 1 x	entrée STOP_GATE : <i>actif haut</i>
		x x x 0 x x	entrée EXTERNAL_END : <i>actif bas</i>
		x x x 1 x x	entrée EXTERNAL_END : <i>actif haut</i>
		x x 0 x x x	entrée SLOW_START : <i>actif bas</i>
		x x 1 x x x	entrée SLOW_START : <i>actif haut</i>
		x 0 x x x x	sortie BUSY: <i>actif bas</i>
		x 1 x x x x	sortie BUSY: <i>actif haut</i>
		0 x x x x x	sortie ANALYSIS_GATE : <i>actif bas</i>
		1 x x x x x	sortie ANALYSIS_GATE : <i>actif haut</i>

Remarque : lorsque le rack d'Interface ISIBox est utilisé, l'état du voyant Busy (en panneau avant) dépend de la polarité choisie pour le signal de sortie **BUSY** (Lemo00 en panneau avant) :

- si le signal **BUSY** est en polarité positive, le voyant est allumé lorsque TDC-V4 est dans l'état Busy
- si le signal **BUSY** est en polarité négative, le voyant est éteint lorsque TDC-V4 est dans l'état Busy

Remarque :

Les registres de sélection de polarité ne sont accessibles qu'en écriture.  
Les polarités sont configurées sur Actif haut par défaut.

## 6. Le Format des Données

TDC-V4 délivre des Mots de Codage et des Mots de Service au format 32 bits. Le mot de 32 bits est décomposé en 2 champs :

- le champ DATA de 26 bits
- le champ LABEL de 6 bits.

Le champ LABEL de 6 bits permet 64 Labels différents et donc de distinguer 64 sources de données de types différents. Parmi eux, 50 Labels sont attribués et sont répertoriés dans le tableau de la page suivante :

- les Labels en caractères gras correspondent aux types de source de données développés dans la version de TDC-V4 décrite dans la présente Notice Utilisateur
- les Labels en caractères grisés correspondent aux types de source de donnée qu'il est envisagé de développer en fonction de besoins évoqués par les utilisateurs.

Les 14 Labels restants sont disponibles pour des sources de donnée de type exotique telles que Echelle de comptage, Débitmètre, Alarme, ... .

LABEL	FONCTION	DATA
<b>2<sup>31</sup> 2<sup>30</sup> 2<sup>29</sup> 2<sup>28</sup> 2<sup>27</sup> 2<sup>26</sup></b>		<b>2<sup>25</sup> 2<sup>24</sup> ..... 2<sup>1</sup> 2<sup>0</sup></b>
<b>0 n n n n m</b>	<ul style="list-style-type: none"> <li>- <u>TDC 16 Voies Stop / codage en 1 mot avec marquage</u> nnnn = numéro de voie m = OF (OverFlow FIFO-Voie)</li> <li>- <u>TDC 16 Voies Stop / codage en 2 mots / durée</u> nnnn = numéro de voie m = 0 front montant ; m = 1 front descendant</li> <li>- <u>TDC 16 Voies Stop / codage en 2 mots / h<sup>te</sup> résolution</u> nnnn = numéro de voie m = 1 poids forts ; m = 0 poids faibles</li> </ul>	<p>codage temps absolu</p> <p>codage temps absolu</p> <p>codage temps absolu</p>
<b>0 n n n n n</b>	- <u>TDC 32 Voies Stop / codage en 1 mot</u> nnnnn = numéro de voie	codage temps absolu
<b>1 0 0 0 0 EL</b>	- <u>Voie Start / codage en 1 mot avec marquage</u> EL = entrée START_ENABLE as Label (sinon EL = 0)	codage temps absolu
<b>1 0 0 1 0 1</b>	- <u>Voie Temps Additionnelle / codage en 1 mot</u>	codage temps absolu
<b>1 0 0 0 1 EL</b>	- <u>Voie Start / codage en 2 mots avec marquage / msb</u> EL = entrée START_ENABLE as label (sinon EL = 0)	codage temps absolu
<b>1 0 0 1 0 0</b>	- <u>Voie Start / codage en 2 mots / lsb</u>	codage temps absolu
<b>1 0 0 1 1 m</b>	- <u>Voie Temps Additionnelle / codage en 2 mots</u> m = 1 poids forts ; m = 0 poids faibles	codage temps absolu
<b>1 1 1 0 0 m</b>	- <u>REXT (Range Extension)</u> m = 0 1 <sup>ère</sup> demi-période 7,8 ms m = 1 2 <sup>ème</sup> demi-période 7,8 ms	(comptage)*
<b>1 1 0 0 0 0</b>	- <u>EOE (End Of Event) / 1 mot</u>	libre
<b>1 1 0 0 1 m</b>	- <u>EOE-N (End Of Event / Event Number) / 2 mots</u> m = 1 poids forts m = 0 poids faibles	numéro d'Evènement
<b>1 1 0 1 0 m</b>	- <u>EOE-T (End Of Event / Event Time) / 2 mots</u> m = 1 poids forts m = 0 poids faibles	time flag
<b>1 1 0 1 1 0</b>	- <u>SOR (Start Of Run)</u>	(numéro de run)
<b>1 1 0 0 0 1</b>	- <u>EOR (End Of Run)</u>	(numéro de run)*
<b>1 1 0 1 1 0</b>	- <u>EOS (End Of Source)</u>	numéro de source

\* reste à préciser

Voie de Codage

Voie de Service

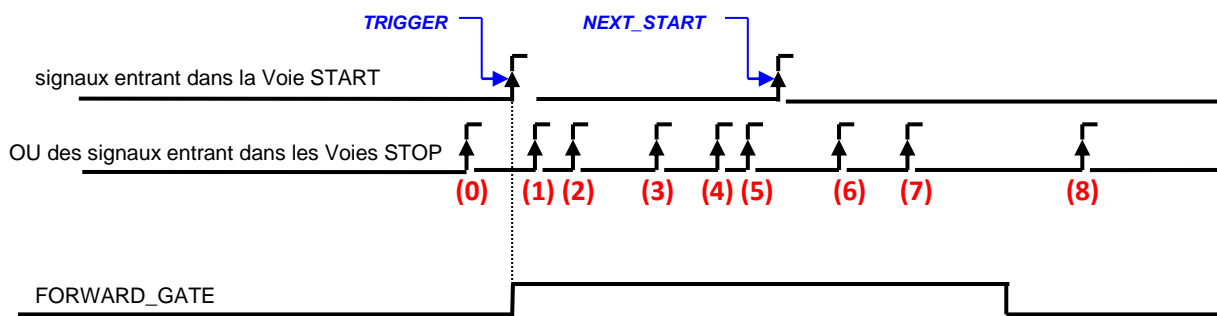
## 7. L'encapsulation des Données

L'encapsulation des données est indépendante de tout Mode de Fonctionnement.

Les Données relatives à un Evènement et délivrées par les Voies de Codage Stop, la Voie de Codage Start (si le mode **NEXT\_START** est sélectionné), la Voie de codage Additionnelle (si l'option est implémentée) et la Voie de Service Range Extension (si l'option est implémentée) sont groupées en un tableau unique (qui peut être réparti sur plusieurs tampons consécutifs selon la taille de l'Evènement de le mode d'acquisition) et encadrées par les 2 mots suivants :

- en tête de tableau : le Mot de Codage de la Voie Start correspondant à l'instant de déclenchement
- en queue de tableau : le Mot de Service EOE (End Of Event)

La figure ci-dessous représente un exemple d'encapsulation de données :

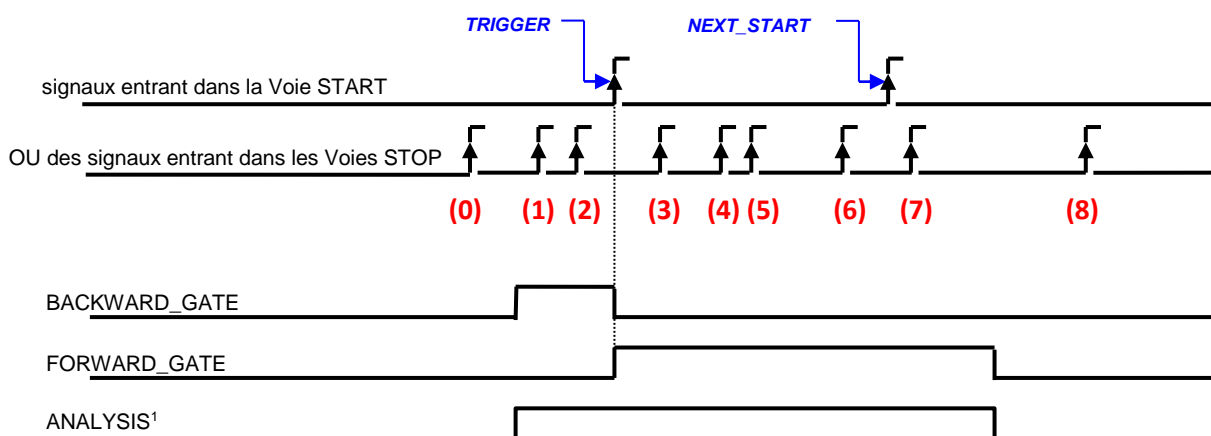


<b>START_WORD</b>
STOP_WORD (1)
STOP_WORD (2)
STOP_WORD (3)
STOP_WORD (4)
STOP_WORD (5)
<b>START_WORD (*)</b>
STOP_WORD (6)
STOP_WORD (7)
<b>EOE_WORD</b>

(\*) si le mode **NEXT\_START** est sélectionné

## Remarques

- L'ordre chronologique des signaux entrant dans une même Voie de Codage Stop est respecté après encapsulation.
- L'ordre chronologique des signaux entrant dans des Voies différentes de Codage Stop n'est pas être respecté après encapsulation.
- L'ordre chronologique des signaux entrant dans la Voie de Codage Start en mode **NEXT\_START** est respecté après encapsulation. Le premier Mot de Codage délivré par la Voie de Codage Start se distingue des suivants par le fait qu'il se trouve au début du Run ou qu'il est précédé du Mot de Service EOE.
- Comme indiqué dans la figure ci-dessous, l'encapsulation des données reste inchangée si le mode **BACKWARD\_MODE** est sélectionné :



<b>START_WORD</b>
STOP_WORD (1)
STOP_WORD (2)
STOP_WORD (3)
STOP_WORD (4)
STOP_WORD (5)
STOP_WORD (6)
<b>START_WORD<sup>2</sup></b>
STOP_WORD (7)
<b>EOE_WORD</b>

Note 1 : la sortie **ANALYSIS\_GATE** est active durant la durée **FORWARD\_GATE**

Note 2 : si le mode **NEXT\_START** est sélectionné

- Suite à la Commande **RUN = OFF**, le Mot de Service EOR est délivré au logiciel d'acquisition après le transfert du tableau de données correspondant au dernier Evènement traité par TDC-V4.

## 8. La collecte des Données par le bus PCI

La collecte des Données délivrées par TDC-V4 se fait au travers des 3 registres suivants (les valeurs des adresses PCI sont différentes pour la version de TDC-V4 compatible avec le logiciel d'acquisition Narval) :

- *Le registre SEMAPHORE (en Lecture et Ecriture)*

nom	adresse PCI	data						
SEMAPHORE	2AFC04h	$2^{31} \dots 2^{16}$	$2^{15} \dots 2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
		X ... X	0 ... 0	0	1	0	0	<i>tampon disponible: non</i>
		n ... n	0 ... 0	0	1	1	0	<i>tampon disponible: oui</i> n ... n : Taille

Le reset du registre SEMAPHORE est effectué par Ecriture de la valeur 4h.

- *Le registre SIZE (en Lecture)*

nom	adresse PCI	data				
SIZE	2AFC08h	$2^{31} \dots 2^{16}$	$2^{15} \dots 2^0$			
		0 ... 0	n ... n			n ... n : Taille

- *Le registre DATA (en Lecture)*

nom	adresse PCI	data				
DATA	2AFC18h	$2^{31} \dots 2^0$				
		d ... d				d ... d : Donnée

Dans TDC-V4, les deux tampons de sortie FIFO-ACQ sont montés en bascule. Quand le tampon est-il dit disponible ?

- *Si le mode Evènement par Evènement est sélectionné (EBE\_HOST\_ACK, EBE\_EXT\_END ou EBE\_TDC\_AUTO), un tampon est disponible si :*
  - la FIFO en cours de remplissage est pleine (509 mots)
  - ou le Mot de Service EOE (End of Event) est enregistré
  - ou le Mot de Service EOR (End of Run) est enregistré.
- *Si le mode Accumulation est sélectionné (ACC\_TDC\_AUTO), un tampon est disponible si :*
  - la FIFO en cours de remplissage est pleine (509 mots)
  - ou le Mot de Service EOR (End of Run) est enregistré.

L'accès au SEMAPHORE par le logiciel d'acquisition, c'est-à-dire à l'information "tampon disponible", peut se faire par 2 moyens :

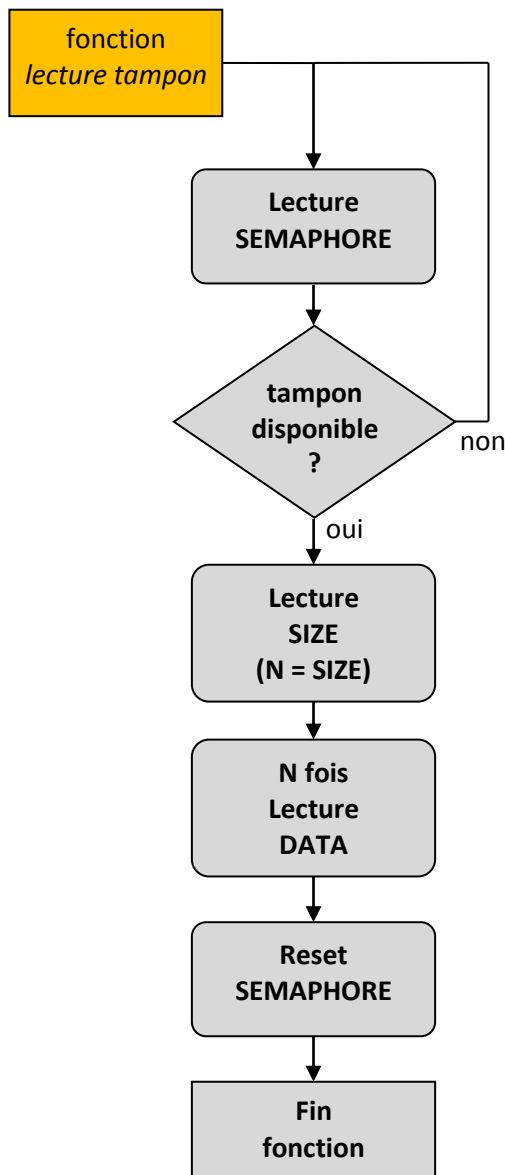
- interrogation du contenu du registre SEMAPHORE (polling)
- traitement d'interruption matérielle (ligne d'interruption INT<sub>A</sub> du bus PCI)

Pour chacun de ces 2 modes d'accès (Polling ou Interruption), TDC-V4 est compatible avec 2 protocoles possibles de collecte des Données :

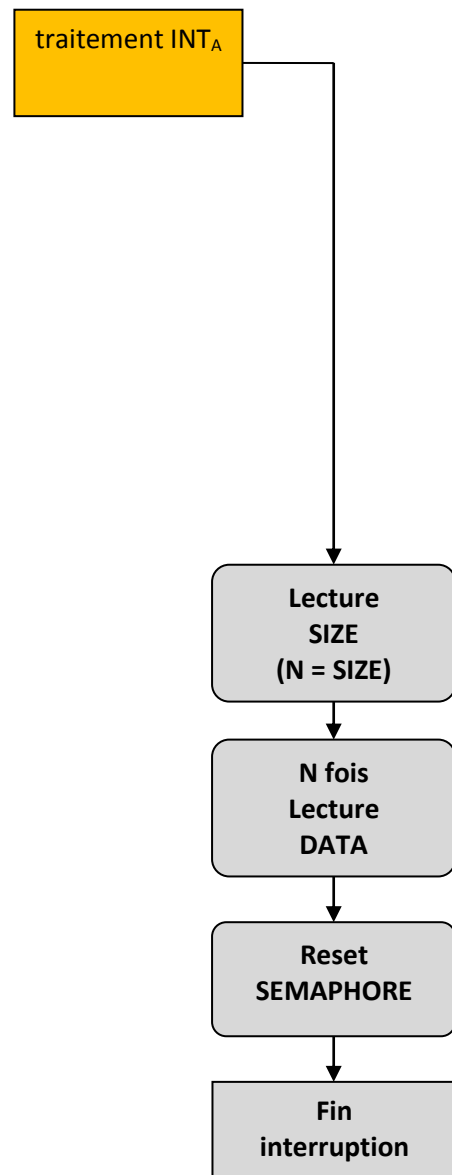
- le Protocole Standard, le plus souvent utilisé par les versions antérieures de TDC
- le Protocole Simplifié, permettant d'accélérer les débits en collecte Evènement par Evènement

### Protocole Standard

Polling



Interruption



## Protocole Simplifié

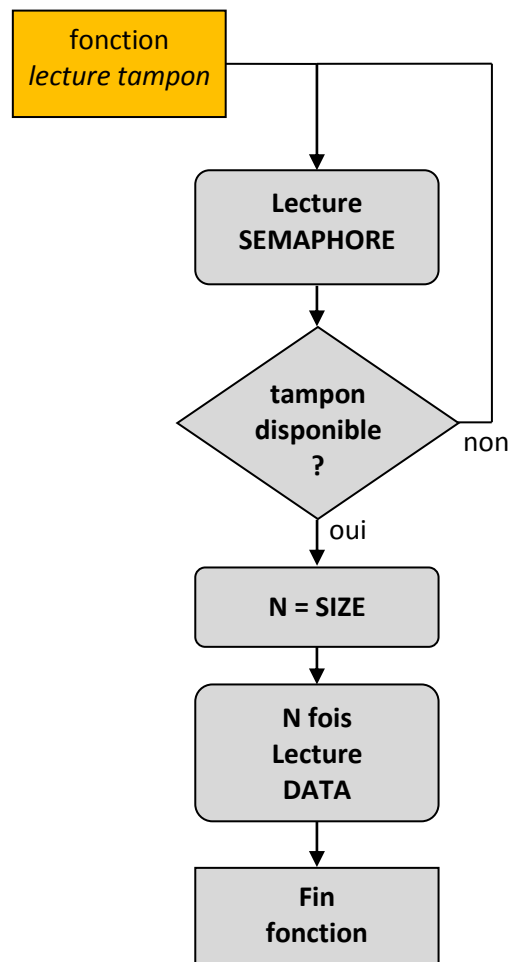
En accès par Polling, ce protocole supprime :

- la Lecture du registre SIZE (le protocole utilise le champ SIZE du registre SEMAPHORE)
- la réinitialisation par Ecriture dans le registre SEMAPHORE

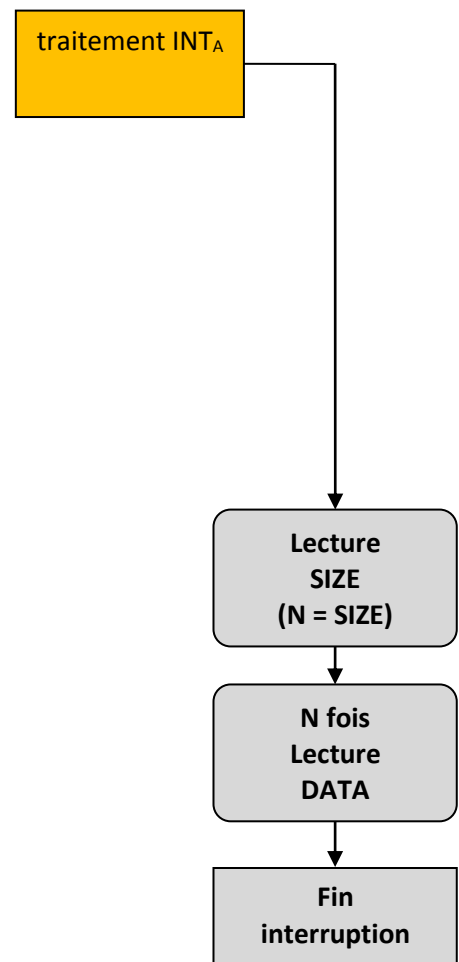
En accès par Interruption, ce protocole supprime :

- la réinitialisation par Ecriture dans le registre SEMAPHORE

Polling



Interruption





## 9. Caractéristiques techniques

avec rack ISIBox

<b>pas de codage</b>	120 ps	
<b>résolution (<math>\sigma</math>)</b>	60 ps	
<b>résolution double hit (max)</b> <ul style="list-style-type: none"> <li>• isolé</li> <li>• rafale (max)</li> <li>• rafale (moyenne)</li> </ul>	2,5 ns 5 ns 4 ns	
<b>latence après déclenchement</b> <ul style="list-style-type: none"> <li>• sans mode backward</li> <li>• avec mode backward</li> </ul>	4 ns 0 ns	
<b>INL</b>	0 %	
<b>DNL</b>	4 %	
<b>plage</b> <ul style="list-style-type: none"> <li>• sans REXT</li> <li>• avec REXT (optionnel)</li> </ul>	7,8 ms sans limite	
<b>tampon d'entrée par voie</b>	512 mots	
<b>tampon de sortie</b>	2 x 509 mots	
<b>voies de codage temps</b> <ul style="list-style-type: none"> <li>• Start</li> <li>• Stop</li> <li>• Additionnelle (optionnel)</li> </ul>	1 16 1	NIM / ECL
<b>niveaux entrées/sorties SCSI</b>	LVDS	
<b>sorties de contrôle</b>	RUN BUSY FORWARD_GATE	TTL et voyant TTL et voyant TTL
<b>entrées de contrôle</b>	TRIGGER_EN. / EVENT_LABEL. STOP_GATE EXTERNAL_END	TTL TTL TTL
<b>sources de déclenchement</b>	FAST_START SLOW_START HOST	NIM TTL
<b>fenêtre d'analyse</b>	INTERNAL EXTERNAL BACKWARD sans limite	
<b>modes d'acquisition</b>	EVENT BY EVENT ACCUMULATION	
<b>communication PCI</b>	POLLING INTERRUPTION	

## 10. Débit de collecte des données

Le débit moyen de collecte des Données dépend de nombreux paramètres : puissance, architecture et système d'exploitation de l'ordinateur d'acquisition, programme de traitement des données, séquençage des Evènements (périodique, poissonnier), constitution des Evènement (taille fixe, taille variable ; durée de la fenêtre d'Analyse), taux de pertes, ...

De plus, la performance en débit dépend fortement du Mode d'Acquisition et d'Acquittement d'Evènement adopté (cf. chapitres 3.b et 3.f). Ce chapitre précise en 10.10.a les modalités d'utilisation des 3 modes :

- Continuing Analysis
- Accumulation
- Event By Event

et en 10.10.b les performances en débit relevées.

### ***10.a. les 3 modes de collecte des Données***

#### ➤ ***Continuing Analysis***

Dans ce mode de collecte des Données, tous les signaux validés arrivant sur les entrées Start et Stop sont codés de l'instant où le TDC est déclenché (un seul signal de déclenchement est utile) (cf. chapitre 3.c) jusqu'à l'instant de remise à zéro du Run. Les codages résultants sont empilés dans les tampons de sortie.

Ce mode de collecte est obtenu en sélectionnant **EXTERNAL\_GATE** comme Durée d'Analyse et en ne branchant pas de signal sur l'entrée **EXTERNAL\_END** ce qui entraîne une analyse de durée illimitée (cf. chapitre 3.d).

Ainsi, l'ensemble du Run étant considéré comme un seul Evènement, les Modes d'Acquittement Event By Event ou Accumulation peuvent être indifféremment sélectionnés (cf. chapitre 3.f). La sélection du mode **BACKWARD\_MODE** n'a aucun effet.

Si l'on souhaite coder tous les signaux aiguillés sur l'entrée de cette voie, sélectionner le Mode de Codage **NEXT\_START** (cf. chapitre 3.e).

Le flot des données est encadré par le Mot de Codage Start au début du Run et les Mots de Service EOE + EOR à la fin. A l'exception du dernier tableau du Run, tous les tableaux acquis comportent 509 mots de 32 bits.

C'est le mode de collecte le plus efficace que peut permettre TDC-V4 du point de vue du débit ; TDC-V4 n'effectue aucun filtrage sur les signaux d'entrée et ne construit pas d'évènements. Il revient au logiciel d'acquisition d'effectuer ces opérations.

### ➤ **Accumulation**

Dans ce mode, chaque Start déclenche une Fenêtre d'Analyse de durée finie (180 ns minimum, 7,8 ms maximum) et TDC-V4 empile les codages relatifs à plusieurs Evènements successifs dans le tampon de sortie.

Pour chaque Evènement, TDC-V4 regroupe les codages durant la fenêtre d'analyse Backward (si **BACKWARD\_MODE** est sélectionné, cf. chapitre 3.d) et la fenêtre d'analyse Forward. TDC-V4 encadre les codages des Voies Stop (et de la Voie Start si le Mode de Codage **NEXT\_START** est sélectionné, cf. chapitre 3.e) par le Mot de Codage Start au début du tableau d'un Evènement et le Mot de Service EOE à la fin.

Dans le mode Accumulation, le logiciel d'acquisition de données est donc plus simple que dans le mode Continuing Analysis, mais le débit est inférieur.

Excepté le dernier tampon du Run, tous les tampons contiennent 509 mots de 32 bits.

### ➤ **Event By Event**

Dans ce mode, chaque Start déclenche une Fenêtre d'Analyse de durée finie (180 ns minimum, 7,8 ms maximum), TDC-V4 regroupe les codages durant la fenêtre d'analyse Backward (si **BACKWARD\_MODE** est sélectionné, cf. chapitre 3.d) et la fenêtre d'analyse Forward, et attend d'être acquitté avant d'accepter un nouvel Evènement. TDC-V4 est acquitté automatiquement seulement en mode **EBE\_TDC\_AUTO**.

TDC-V4 encadre les codages des Voies Stop (et de la Voie Start si le Mode de Codage **NEXT\_START** est sélectionné, cf. chapitre 3.e) par le Mot de Codage Start au début du tableau d'un Evènement et le Mot de Service EOE à la fin.

Les tampons lus sont de taille variable, comprise entre 2 et 509 mots de 32 bits.

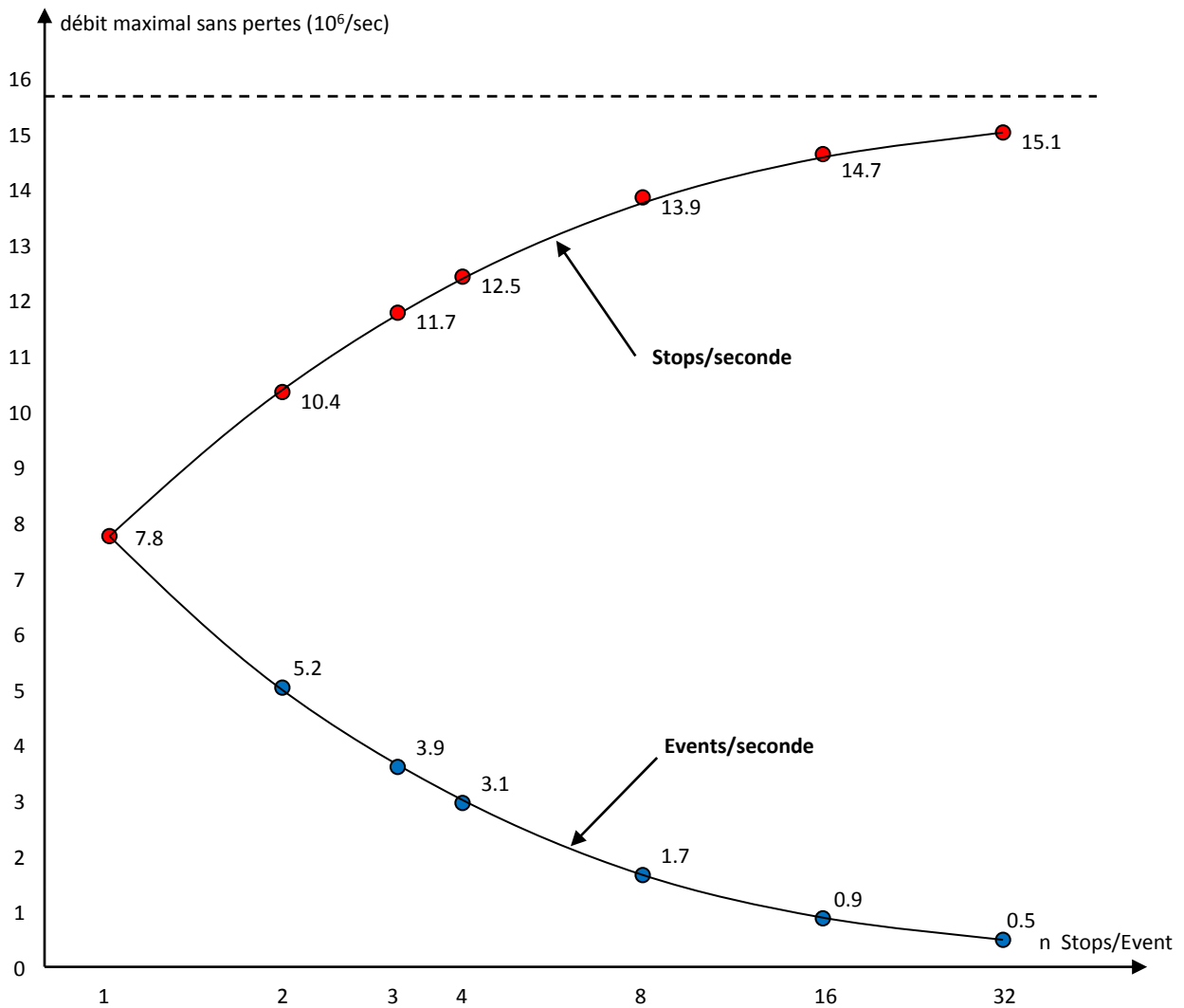
Ce mode de collecte est obligatoire lorsque TDC-V4 doit être corrélé avec d'autres sources de données. C'est le mode le moins efficace du point de vue des débits et sa performance est d'autant plus médiocre que le nombre moyen de Stops par Evènement est faible.

## 10.b. les performances en débit maximal sans pertes

Les conditions dans lesquelles ont été obtenues les performances ci-après sont les suivantes :

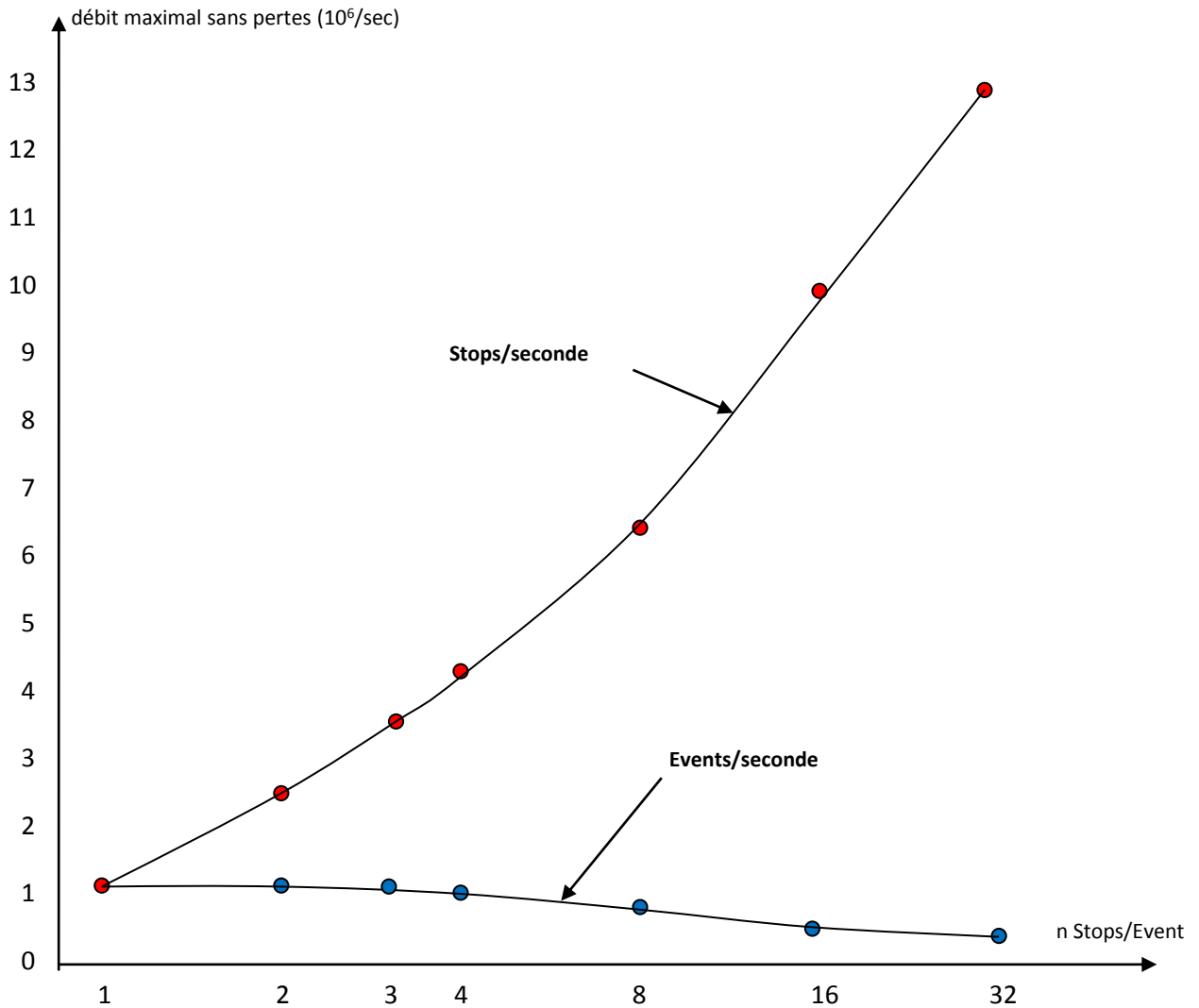
- unité centrale : PC multi-cœurs
- système d'exploitation : Windows XP
- traitement des données : aucun
- séquençage des Evènements : périodique
- taille de l'Evènement : fixe, constitué de 1 Start + n Stops
- taux de pertes : 0%

### ➤ collecte des données en Continuing Analysis



*Débit sans pertes et à fréquence périodique, en fonction de la taille de l'Evènement*  
**Mode : Continuing Analysis**

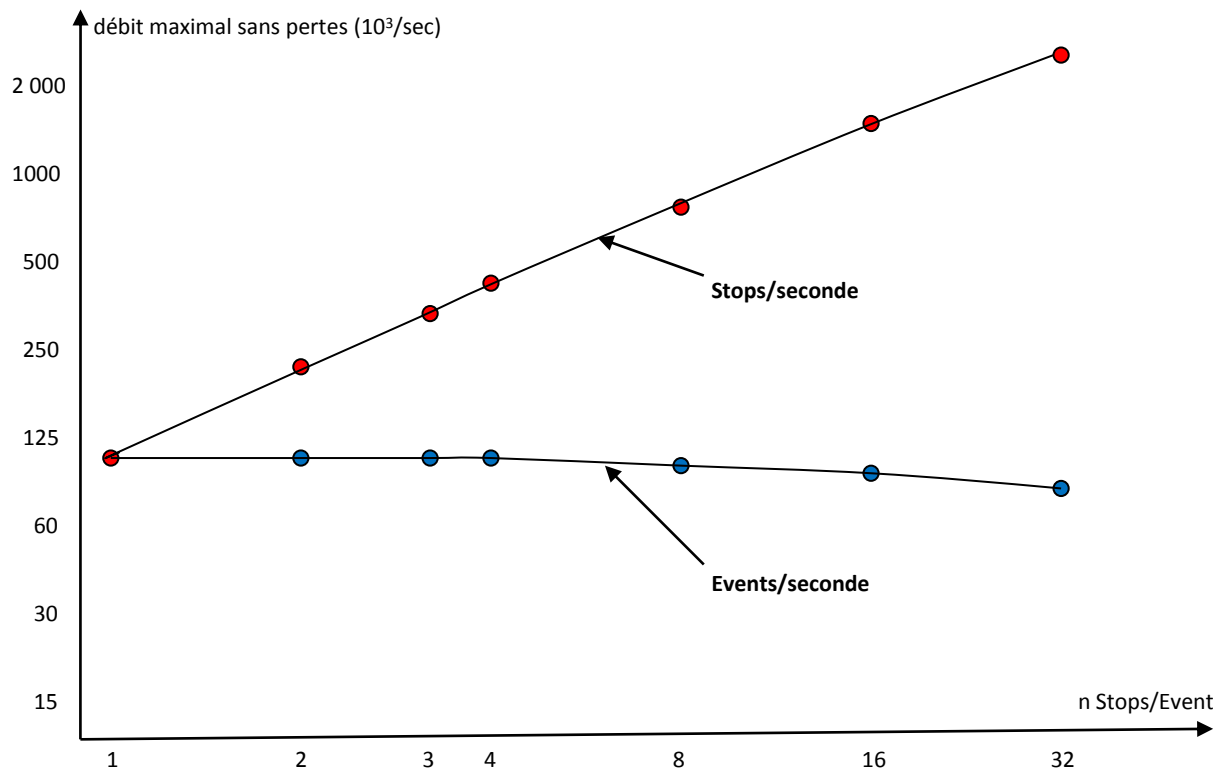
➤ **collecte des données en ACCUMULATION**



*Débit sans pertes et à fréquence périodique, en fonction de la taille de l'Evènement*  
**Mode d'Acquittement : ACC\_TDC\_AUTO INTERNAL\_GATE : 180 ns**

nombre moyen de Stops par Evènement	débit sans pertes MegaEvents/sec	débit sans pertes MegaStops/sec
1	1.25	1.25
2	1.25	2.5
3	1.25	3.75
4	1.1	4.4
8	0.8	6.4
16	0.6	10.0
32	0.4	12.8

➤ **collecte des données en Event By Event**



Débit, sans pertes et à fréquence périodique, en fonction de la taille de l'Evènement  
**Modes d'Acquittement: EBE\_HOST\_ACK ou EBE\_TDC\_AUTO**  
**INTERNAL\_GATE : 180 ns**

nombre moyen de Stops par Evènement	débit sans pertes KiloEvents/sec	débit sans pertes KiloStops/sec
1	91.3	91.3
2	90.5	181
3	89.7	269
4	88.9	356
8	85.8	687
16	80.3	1285
32	71.2	2280

# 11. Installation de TDC-V4

## 11.a. la configuration requise

L'installation de la carte TDC-V4 ou TDC-V4-E, ainsi que du pilote et de la DLL livrés avec la carte, nécessite un ordinateur de type PC ayant les caractéristiques minimales suivantes :

- Un emplacement PCI (32 bits ou 64 bits) libre pour la version TDC-V4 en PCI
- Un emplacement PCI Express (x1, x4, x8 ou x16) libre pour la version TDC-V4-E en PCI Express
- Système d'exploitation : Windows 7/8/10 en 64 bits (pilote v12.21 signé)

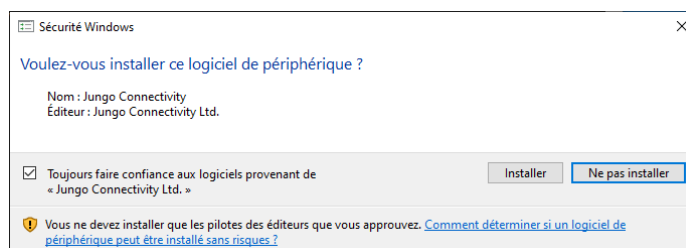
## 11.b. l'installation (pilote v12.21 signé)

Installation de la carte TDC-V4 ou TDC-V4-E dans le PC :

- Eteindre l'ordinateur, débrancher le câble d'alimentation, puis retirer le capot du boîtier du PC
- Insérer la carte dans un emplacement libre (PCI pour le TDC-V4, PCI Express pour le TDC-V4-E) en la tenant fermement par les bords (si un cache obstrue le châssis arrière du PC, le retirer), et l'enfoncer à fond. Fixer sa languette sur le châssis arrière à l'aide d'une vis si nécessaire
- Replacer le capot et rebrancher le câble d'alimentation

Installation du pilote (Windows 7/8/10) :

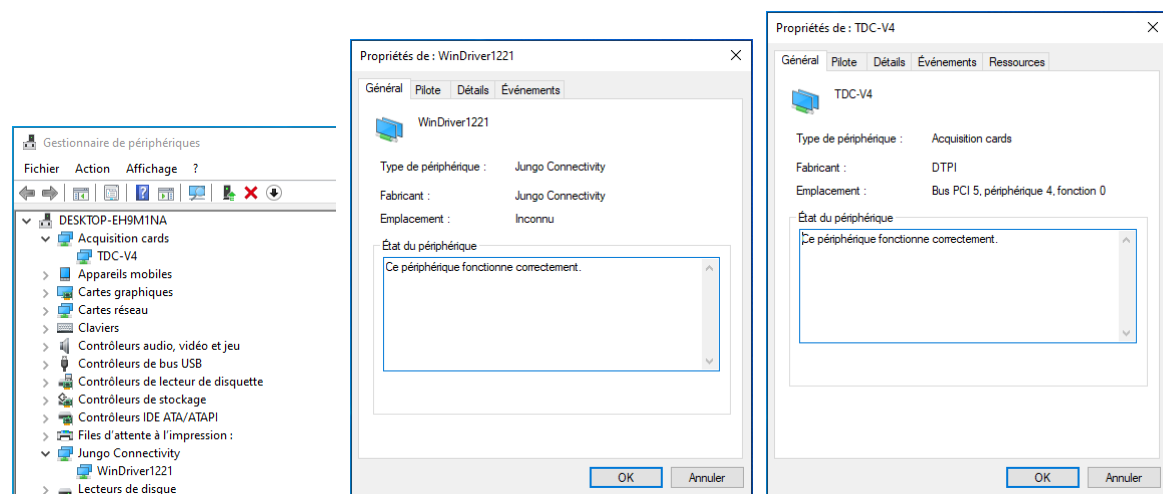
- Dans le dossier *Driver* du package logiciel fourni, faire un clic droit sur *install.bat*, puis choisir *Exécuter en tant qu'administrateur*. Attendre (moins d'une minute)
- Pour Windows 7 uniquement: si le système n'est pas à jour par internet avec Windows Update, une mise à jour Windows (KB4474419) est susceptible de s'installer. Attendre l'installation (qui peut durer plusieurs minutes), et choisir *Oui* lorsqu'une fenêtre demande de redémarrer le système. Relancer *install.bat* en tant qu'administrateur puis terminer l'installation du pilote
- Dans la fenêtre qui apparaît *Voulez-vous installer ce logiciel de périphérique*, cliquer sur *Installer*



- Pour Windows 7 uniquement: une deuxième fenêtre *Voulez-vous installer ce logiciel de périphérique* apparaît, cliquer sur *Installer*

### Vérification que la carte est bien reconnue :

- Cliquer sur *Démarrer*, puis taper *Gestionnaire de Périphériques*
- Faire un double clic sur *Jungo Connectivity* (fournisseur du pilote), puis sur *Windriver1221* (désignation générique du pilote). Dans la fenêtre qui apparaît *Propriétés*, vérifier que le message *Ce périphérique fonctionne correctement* est affiché
- Faire un double clic sur *Acquisition cards*, puis sur *TDC-V4* (désigne la carte, TDC-V4 est affiché pour le TDC-V4-E). Dans la fenêtre qui apparaît *Propriétés*, vérifier que le message *Ce périphérique fonctionne correctement* est affiché



### Remarque:

Si l'icône ⚠ apparaît à gauche de *TDC-V4* ou *Windriver1221*, le pilote n'est pas correctement installé

### Désinstallation du pilote :

- Dans le dossier *Driver* du package logiciel fourni, faire un clic droit sur *uninstall.bat*, puis choisir *Exécuter en tant qu'administrateur*. Le pilote est désinstallé, *TDC-V4* et *Windriver1221* n'apparaissent plus dans le *Gestionnaire de Périphériques*



## 12. Fonctions de la DLL

Les indications suivantes permettent d'utiliser les fonctions de la DLL livrée avec la carte TDC-V4 dans les applications développées en langage C/C++. Celle-ci a été développée par la société ISITech pour la carte ISITime01 et TDC-V4 est compatible avec cette DLL. Une DLL optimisée pour TDC-V4 sera développée ultérieurement.

### 12.a. les fichiers requis

Pour écrire une application en langage C/C++, les fichiers suivants, fournis dans le dossier *DLL* du package TDC-V4, doivent être inclus dans le programme :

- *isitime\_dll.dll* : fichier DLL
- *import\_dll.h* : fichier header
- *isitime\_dll.lib* : fichier ressource

### 12.b. les fonctions

➤ Ouverture du pilote :

**Void OuvertureDriver(char \*flag, unsigned long vID, unsigned long dID, int mc)**

Cette fonction doit être utilisée à chaque lancement du programme. Elle ouvre le pilote en passant les paramètres suivants:

- **flag** retourne la valeur 20 lorsque la fonction s'est déroulée avec succès
- **vID** = 0x10B5
- **dID** = 0x5406
- **mc** = -1

➤ Fermeture du pilote :

**Void FermetureDriver()**

Cette fonction doit être utilisée à chaque fermeture du programme pour fermer le pilote.

➤ Ecriture dans un registre :

**Bool Data\_Ecriture(int \*register\_val, int size, unsigned long address, int mode)**

Cette fonction permet d'écrire dans les registres modes et commandes :

- **register\_val** : mot de donnée (32 bits)
- **size** = 1
- **address** : voir le récapitulatif chapitre 14
- **mode** = 6

- Lecture d'un registre :  
**Bool Data\_Lecture(int \*register\_val, int size, unsigned long address, int mode)**

Cette fonction permet de lire dans les registres DATA, SEMAPHORE et SIZE :

- **register\_val** : mot de donnée (32 bits)
- **size** : nombre de mots à lire
- **address** : voir le récapitulatif chapitre 14
- **mode** = 3 pour registre DATA; 0 pour les autres registres

- Lecture d'un tampon d'acquisition FIFO-ACQ :  
**int Read\_dsp\_bis(int \*data\_buffer, int \*size)**

Cette fonction permet de lire le tampon de données disponible dans le tampon de sortie du TDC-V4 dans le protocole standard. Elle gère :

- ✓ la lecture du registre SEMAPHORE
- ✓ le test de la disponibilité des données :
  - si aucune donnée n'est disponible : la fonction retourne 0
  - si des données sont disponibles : la fonction lit la taille à partir du registre TAILLE, lit le nombre de mots correspondant dans les tampons de sortie de TDC-V4, réinitialise le sémaphore, retourne 6 et remplit les valeurs suivantes :
    - **data\_buffer** : tampon lu (tableau de mots de 32 bits)
    - **size** : nombre de mots lus ( $\leq 509$ )

## 13. Exemple de programme

L'exemple suivant montre la structure chronologique à respecter pour tout programme d'acquisition (protocole standard).

### 1. Ouverture du pilote

- Utiliser la fonction `OuvertureDriver`

Exemple de syntaxe:

```
char flag = 0;
OuvertureDriver(&flag, 0x10B5, 0x5406, -1);
```

### 2. Initialisation (non indispensable avant le lancement d'une acquisition)

- Utiliser la fonction `Data_Ecriture`

Exemple de syntaxe:

```
#define INIT_TDC 0x1600000
int register_val = 0;
Data_Ecriture(&register_val, 1, INIT_TDC, 6); //Activer Reset TDC
register_val = 1;
Data_Ecriture(&register_val, 1, INIT_TDC, 6); //Désactiver Reset TDC
```

Attention: Reset TDC doit-être une impulsion, une instruction *désactiver* doit suivre une instruction *activer*

### 3. Configuration (peut être réalisé à tout moment)

- Utiliser la fonction `Data_Ecriture`

Exemple de syntaxe : activer les voies 1 et 3, désactiver les autres voies (les autres registres de configuration restent à leurs valeurs par défaut):

```
#define STATIC_STOP_ENABLE 0x1200000
register_val = 0xA; // binaire: 1010
Data_Ecriture(&register_val, 1, STATIC_STOP_ENABLE, 6);
```

### 4. Lancement d'un Run

- Utiliser la fonction `Data_Ecriture`

Exemple de syntaxe:


```
#define RUN 0x1A00000
register_val = 1;
Data_Ecriture(&register_val, 1, RUN, 6);
```

## 5. Collecte des données (protocole standard)

- Utiliser la fonction `Read_dsp_bis`

Exemple de syntaxe:

```
int size, data_buffer[509], status = 0;

while(status == 0 ) // Polling "test tampon disponible" pendant la durée
du Run
{
 status = Read_dsp_bis(data_buffer, &size)
}
```

## 6. Arrêt du Run

- Utiliser la fonction `Data_Ecriture`

Exemple de syntaxe:

```
register_val = 0;
Data_Ecriture(&register_val, 1, RUN, 6);
```

## 7. Mise en arrêt de la collecte des données

Pour s'assurer que tous les mots codage restant dans les tampons de TDC-V4 soient lus après la mise en arrêt, il est recommandé de continuer la collecte des données jusqu'à la lecture du mot EOR

## 8. Fermeture du Driver

- Utiliser la fonction `FermetureDriver`

Exemple de syntaxe:

```
FermetureDriver();
```

---

Un exemple de programme fonctionnel en C++ est fourni dans le dossier *Example program* du package TDC-V4.

## 14. Récapitulatif des registres de TDC-V4

nom	adresse	page
<b>REGISTRES DE CONFIGURATION</b>		
TRIGGER_SOURCE	240 0000h	12
TRIGGER_ENABLE	220 0000h	12
FORWARD_MODE	1C0 0000h	14
FORWARD_DURATION	1C0 0000h	14
BACKWARD_MODE	180 0000h	15
BACKWARD_DURATION	180 0000h	15
NEXT_START	220 0000h	16
EVENT LABELLING	220 0000h	17
EXTRA_CHANNEL	NA	18
STATIC_STOP_ENABLE	120 0000h	18
DYNAM_STOP_ENABLE	220 0000h	18
OPTIMIZATION	220 0000h	19
ACK_MODE	260 0000h	20
INIT_CONF	140 0000h	22
INIT_TDC	160 0000h	22
RUN	1A0 0000h	23
HOST_ACK	2C0 0000h	23
HOST_START	2E0 0000h	24
POLARITY	280 0000h	25
<b>REGISTRES DE COMMANDE</b>		
SEMAPHORE	2AFC04h	30
SIZE	2AFC08h	30
DATA	2AFC18h	30