

## Manuel ADC-200 / 212 / 216 v1.0

1. Introduction
2. Connexion au PC
3. Caractéristiques
4. Principes d'Opération



---

### Ecrire ses propres logiciels

#### vue d'ensemble



C



Turbo  
Pascal



C/C++  
Win



Delphi



Visual  
Basic



Excel



HP Vee



LabView



LabWindows



Linux

---

### Informations supplémentaires

Contact



Pico

Technical



Support

Signal



Conditioners



Safety



Legal



Print Thi  
Manual

## Caractéristiques

Produit	200/20	200/50	200/100	212/50	212/100	212/3	216
Résolution	8 bits	8 bits	8 bits	12 bits	12 bits	12 bits	16 bits
Voies d'entrée	2 Connecteurs BNC à impédance d'1M ohm						
Couplage	AC/DC sous contrôle des logiciels						
Lancement externe	Le connecteur EXT BNC sert soit d'entrée de lancement, soit de sortie de générateur de signal carré.						
Gammes de tension	+/-50mV à 20V par intervalles de 1,2,5 (9 gammes)					+/-10mV à 20V par intervalles de 1,2,5 (11 gammes)	
Précision (%)	+/-3	+/-3	+/-3	+/-1	+/-1	+/-1	+/-1
Protection surcharge (V)	+/-100	+/-100	+/-100	+/-100	+/-100	+/-100	+/-100
Taux d'échant. (1 voie)	20M	50M	100M	50M	100M	3M	333k
(éch/s) (2 voies)	20M	50M	50M	50M	50M	1.5M	166.6k
Bande passante (Hz)	10M	25M	50M	25M	50M	1.5M	166k
Tampon (éch)	8K	16K	32K	128k	128k	32K	32K
Alimentation	12V DC nominal à 500mA maxi. Broche DC1.3mm (centre positif)						
Dimensions	140 x 190 x 45mm						

## Mises à jour

Nos logiciels sont mises à jour régulièrement par l'ajout de nouvelles fonctions. Pour vérifier quelle version du logiciel on utilise, lancer PicoScope ou PicoLog et sélectionner l'option **Infos Sur...** du menu **Aide**. On peut télécharger gratuitement la dernière version depuis le site web

**<http://www.picotech.com>**.

On peut également acheter une mise à jour sur CD auprès de votre distributeur ou de Pico Technology.

Pour obtenir gratuitement notre bulletin d'informations électronique, s'inscrire par le biais de notre site web **<http://www.picotech.com/maillist.html>**

## Consignes de sécurité

Pour tous les ADC Pico, la connexion terre de l' ADC est raccordée directement à la fiche terre de l' ordinateur, afin de minimiser les interférences électriques.

Ne brancher la connexion terre de l' ADC à aucun appareil d'une tension autre que terre, car cela risquerait d'endommager l' ADC et l' ordinateur. Par ailleurs, si l' on connecte l' ADC terre à quelque chose qui est sous tension, le châssis de l'ordinateur peut passer sous tension.

Dans le doute, vérifier en branchant un voie d'introduction à la sortie terre douteuse et s' assurer que la tension est presque zéro.

## Législation

### Licence

A l'achat d'un convertisseur analogique-numérique, Pico Technology accorde une licence autorisant une personne à utiliser les logiciels à tout moment.

L'utilisateur actuel est autorisé à installer les logiciels sur plusieurs ordinateurs et à utiliser plusieurs exemplaires de chaque logiciel, mais ne doit pas autoriser plusieurs personnes à utiliser les logiciels en même temps.

L'utilisateur actuel est autorisé à télécharger du site Web de Pico Technology les logiciels gratuits de mise à niveau pour une période d'un an à compter de la date d'achat.

### Garantie

Pico Technology ne peut garantir ni l'exactitude des logiciels, et du produit avec lequel ils ont été fournis, ni leur bon fonctionnement dans tout environnement d'exploitation.

**Il incombe à l'utilisateur de vérifier que les logiciels et le produit, ainsi que les données qu'ils fournissent, répondent à leurs besoins, avant d'en faire un emploi définitif.**

Si vous rencontrez un problème, veuillez nous le signaler afin que nous puissions, dans la mesure du possible, le rectifier. Si malgré cela le produit ne vous apporte pas entière satisfaction, retournez-le; il vous sera remboursé intégralement.

### Marques Déposées

Borland, Delphi et Turbo Pascal sont des marques ou marques déposées de Borland International, Inc.

Microsoft, MS-DOS, Windows, Windows NT, Visual C++, Visual Basic sont des marques ou marques déposées de Microsoft Corporation aux Etats-Unis et dans d'autres pays

National Instruments et LabVIEW sont des marques de National Instruments Corporation, déposées aux Etats-Unis et dans d'autres pays

Pico Technology Limited, PicoScope et PicoLog sont des marques de Pico Technology Limited, déposées au Royaume-Uni et dans d'autres pays

## **Exemples de programmation**

C (DOS)

Pascal

C (WINDOWS)

Delphi

Visual Basic

## **Logiciels d'Application**

Excel

LabVIEW

LabWindows

HP-Vee

## Principes de fonctionnement

Le gamme de convertisseurs ADC-2xx comprend des produits à grande vitesse (ADC-200/100) ainsi que d'autres versions à haute précision (ADC-216). Un dispositif ADC-2xx prend des mesures et les transmet à un ordinateur.

### Echantillonnage en bloc

A grande vitesse, l'ADC-2xx peut recueillir des données beaucoup plus vite que le PC ne peut les lire. L'ADC-2xx recueille alors un bloc de données dans sa mémoire interne, et les transmet ensuite au PC, une fois que le bloc est complet. A basses vitesses, il n'est pas possible d'analyser les premières lectures avant le recueil complet du bloc. L'ADC-2xx offre alors deux modes de fonctionnement pour des vitesses différentes.

### Mode rapide

En mode rapide, l'ordinateur donne l'ordre à l'ADC-2xx de recueillir un bloc de données dans sa mémoire. Lorsque le recueil est complet, l'ordinateur arrête l'ADC et transfère tout le bloc de l'ADC à sa propre mémoire.

Le nombre maximum d'échantillons dépend de la taille de la mémoire de l'ADC2xx. Le convertisseur peut prendre des échantillons à plusieurs taux: le plus rapide est la vitesse d'horloge du dispositif: les autres taux correspondent à la vitesse de l'horloge divisé par les puissances de deux (demi, quart, huitième, etc). Il existe entre 16 et 20 taux d'échantillonnage suivant le modèle d'ADC.

Pour les ADC200, il existe un tampon individuel par voie: sur la version ADC200-100, une voie peut être dirigée vers les deux tampons, doublant ainsi le taux d'échantillonnage efficace. Pour l'ADC212 et l'ADC216, les deux voies se partagent le même tampon.

Le pilote ADC-2xx effectue normalement plusieurs opérations de réglage avant de recueillir chaque bloc de données, ce que peut prendre jusqu'à 50 millisecondes. Pour recueillir des blocs de données à un intervalle de temps minimum entre les blocs, utiliser l'option `adc200_set_rapid`.

### Mode lent

En mode lent, l'ADC-2xx utilise sa mémoire interne en 'premier entré, premier sorti': l'ordinateur peut lire les valeurs de la mémoire dès que les lectures sont recueillies. Dans ce mode, l'ADC-2xx fonctionne en continu, et le nombre de lectures est illimité.

A noter: le mode lent n'est pas disponible actuellement: en attendant, voir `adc200_get_single`.

### Lancement

L'ADC-2xx peut commencer à recueillir des données, soit immédiatement, soit au moment d'un événement de lancement. L'événement peut être, soit le dépassement d'une tension de seuil par une voie (A ou B), soit un changement d'état de l'entrée de lancement externe (numérique). Le dépassement peut être montant ou descendant.

L'événement de lancement peut être réglé pour le début du bloc de données, pour la fin, ou entre les deux.

L'entrée de lancement étant la même que la sortie du générateur de signaux, ces deux fonctions ne peuvent être utilisées simultanément.

### Gammes de tension

On peut régler les tensions d'entrée des voies sur  $\pm 50\text{mV}$  et jusqu'à  $\pm 20\text{V}$ . Les gammes  $\pm 10\text{mV}$  et  $\pm 20\text{mV}$  existent également pour les versions ADC-212/3 et ADC-216.

### Opération AC/DC

On peut régler les voies sur mode AC ou mode DC. En mode AC, tout composant DC du signal est éliminé.

## Suréchantillonnage

Lorsque le dispositif fonctionne à des vitesses inférieures au maximum, il peut arriver de **suréchantillonner** - c'est à dire de prendre plus d'une mesure pendant chaque intervalle de temps, ce qui réduit les effets de repliement et augmente la résolution apparente du convertisseur.

## Echelonnage

L'ADC-200 est un convertisseur à 8 bits. Il restitue une valeur entre 0 et 255, représentant la gamme de tension entière. Afin de faciliter le développement des logiciels, le pilote convertit les valeurs de façon qu'une valeur de 0 correspond à 0 volts. Les valeurs sont toujours échelonnées comme si le suréchantillonnage était réglé sur x16. La tension maximum positive de la gamme actuelle est donc représentée par 2047, et la tension maximum négative par -2047.

L'ADC-212 est un convertisseur à 12 bits. Le pilote restitue une valeur entre 2047 et -2047, sans tenir compte du réglage du suréchantillonnage.

L'ADC-216 est un convertisseur à 16 bits. Le pilote restitue une valeur entre 32767 et -32767.

## Générateur de signaux

L'ADC-200 est doté d'un **générateur de signaux** intégré produisant une série de fréquences précises de 1kHz à 250kHz. Ces dernières sont sélectionnées sous le contrôle des logiciels. La forme d'onde est presque carrée à basses fréquences, mais s'arrondit au dessus de 100kHz.

La sortie du générateur de signaux étant la même que l'entrée de lancement, ces deux fonctions ne peuvent être utilisées en même temps.

## Exploitation multidispositifs

It is possible to collect data using up to three ADC200 units at the same time. Each AC-200 must be connected to a separate printer port. The routine `adc200_set_unit` select which unit the driver should access next.

Il est possible de recueillir des données à l'aide de trois convertisseurs ADC200 en même temps.

Chaque convertisseur doit être connecté à un port imprimante différent. La procédure `adc200_set_unit` sélectionne le dispositif à utiliser à partir de ce moment.

Par exemple, pour recueillir des données des convertisseurs sur les ports LPT1 et LPT3 en même temps:

```
adc200_open (1)
adc200_open (3)

adc200_set_unit (1)
..... régler dispositif sur LPT1
adc200_run

adc200_set_unit (3)
.... régler dispositif sur LPT3
adc200_run

ready = FALSE
while not ready
  adc200_set_unit (1)
  ready = adc200_ready
  adc200_set_unit (3)
  ready = ready & adc200_ready ()

adc200_set_unit (1)
adc200_get_values
adc200_set_unit (3)
adc200_get_values
```

## Mode lent

Le mode lent sert à recueillir des échantillons à intervalles réguliers sur de longues périodes.

Il n'est pas disponible actuellement. `Adc200_get_single` sert de remplacement en attendant sa mise en service. Ses fonctions sont les mêmes, mais l'utilisateur contrôle lui-même le moment de la prise de chaque échantillon.

## Procédures du pilote

### Formats

The drivers are available in five formats:

- as a **DOS standard mode object file**
- as a **DOS protected mode DLL**
- as a **Windows 3.1/Windows 95 16-bit DLL**
- as a **Windows 95 32-bit DLL**
- as a **Windows NT DLL**

Les pilotes son disponibles en cinq formats:

- un fichier OBJ pour DOS en mode standard
- un fichier DLL pour DOS en mode protégé
- un fichier DLL à 16 bits pour Windows 3.1 et Windows 95/98
- un fichier DLL à 32 bits pour Windows 95/98
- un fichier DLL à 32 bits pour Windows NT/2000
- un daemon pour Linux

### Procédures

Le pilote comprend les procédures suivantes:

procédure	Description
<code>adc200_get_driver_version</code>	Détermine la version du pilote
<code>adc200_open_unit</code>	Ouvre un dispositif <code>adc2xx</code>
<code>adc200_set_unit</code>	Sélectionne un dispositif Exploitation multidispositifs seulement
<code>adc200_close_unit</code>	Ferme un dispositif
<code>adc200_set_dc</code>	Règle l'interrupteur AC/DC
<code>adc200_set_range</code>	Règle la gamme de tension
<code>adc200_set_channels</code>	Sélectionne les voies à utiliser (A, B, les deux)
<code>adc200_set_oversample</code>	Règle le suréchantillonnage
<code>adc200_set_timebase</code>	Règle le taux d'échantillonnage
<code>adc200_set_trigger</code>	Règle les paramètres de lancement
<code>adc200_set_rapid</code>	Active mode renouvellement rapide
<code>adc200_max_samples</code>	Détermine le nombre maximum d'échantillons pouvant être recueillis à l'aide des réglages actuels.
<code>adc200_run</code>	Lance le recueil de données
<code>adc200_ready</code>	Détermine si le recueil est terminé
<code>adc200_stop</code>	Arrête l' ADC-2xx
<code>adc200_get_values</code>	Transfère un bloc de données de l'ADC-2xx vers l'ordinateur
<code>adc200_get_overflow</code>	Détermine si un débordement s'est produit au cours du dernier recueil
<code>adc200_get_single</code>	Prend une lecture unique de chaque voie
<code>adc200_get_unit_info</code>	En cas d'échec d'ouverture: donne des infos sur l'erreur En cas de succès: donne des infos sur le dispositif
<code>adc200_has_relays</code>	Détermine si le dispositif est doté d' interrupteurs AC/DC manuels ou de relais
<code>adc200_get_status</code>	Fournit le code erreur de l'exploitation la plus

	récente de la procédure <code>adc200_open_unit</code>
<code>adc200_get_product</code>	Détermine le type de dispositif (200/212/216)
<code>adc200_set_frequency</code>	Controls the signal generator

## Ordre d'exploitation des procédures

Opérations de base:

- ouvrir l'ADC-2xx
- sélectionner les gammes de tension et les interrupteurs AC/DC
- régler les voies, le lancement et le suréchantillonnage
- sélectionner la base de temps
- déterminer le nombre maximum d'échantillons
- régler la fréquence du générateur de signaux (le cas échéant)

Opérations renouvelées:

- démarrer l'ADC2xx
- attendre la fin du recueil de données (`adc200_ready`)
- arrêter l'ADC2xx
- transférer un bloc de données de l'ADC2xx
- afficher les données

### `adc200_get_driver_version`

```
unsigned short adc200_get_driver_version (void)
```

Cette procédure fournit un code de 2 octets, identifiant la version du pilote. Si vos logiciels risquent d'être utilisés avec d'autres versions du pilote, cette routine sert à vérifier que le pilote est suffisamment récent.

L'octet supérieur renferme la version principale, et l'octet inférieur la version secondaire.

### `adc200_open_unit`

```
unsigned short adc200_open_unit (unsigned short port)
```

Cette procédure ouvre un dispositif ADC2xx sur le port imprimante précisé. Si l'opération est réussie, la valeur de retour est différente de zéro. Le processus d'initialisation prend environ 2 secondes.

port                    Le numéro du port parallèle sur lequel l'ADC-2xx est branché (1 for LPT1, 2 for LPT2 etc).

Le pilote peut exploiter jusqu'à trois dispositifs ADC-2xx en même temps: si l'on souhaite utiliser plusieurs dispositifs, appeler `adc200_open_unit` une fois pour chaque dispositif, puis appeler `adc200_set_unit` pour sélectionner les dispositifs.

Remarques: Le pilote Windows NT ne peut pas déterminer les adresses des ports imprimante. Il présume qu'elles sont les suivantes:

```
LPT1  0x278
LPT2  0x378
LPT3  0x3BC
```

Si votre ordinateur n'est pas conforme à cette norme, utiliser le numéro du port correspondant à l'adresse figurant sur cette liste.

### `adc200_set_unit`

```
unsigned short adc200_set_unit (unsigned short port)
```

Le pilote peut exploiter jusqu'à trois dispositifs ADC-2xx en même temps: si l'on souhaite utiliser plusieurs dispositifs, appeler `adc200_open_unit` une fois pour chaque dispositif, puis appeler `adc200_set_unit` pour sélectionner les dispositifs.

## **adc200\_close\_unit**

```
void adc200_close_unit (unsigned short port)
```

Cette procédure arrête l'ADC-2xx et ferme le pilote.

## **adc200\_get\_unit\_info**

```
short adc200_get_unit_info (char * str, short str_lth, short line,  
short port);
```

Si l'appel à l'`adc200_open_unit` a échoué, cette procédure fournit en format texte la raison de l'échec. Si `adc200_open_unit` a réussi, cette procédure fournit plusieurs lignes de texte d'informations sur le pilote, le dispositif et le taux d'échantillonnage maximum.

Str - tampon pour le texte fourni  
str\_lth - longueur du tampon  
line - 0 à 4: précise la ligne du texte à fournir  
port - numéro du port imprimante (1..3)

## **adc200\_set\_dc**

```
unsigned short adc200_set_dc (  
    unsigned short channel,  
    unsigned short dc)
```

Cette procédure contrôle la position des interrupteurs AC/DC.

channel 0 = voie A, 1 = voie B

dc 1 = DC, 0 = AC

## **adc200\_set\_range**

```
unsigned short adc200_set_range (  
    unsigned short channel,  
    A200_GAIN gain)
```

Cette procédure sélectionne la gamme de tension pour une voie. Si les paramètres sont valides, la valeur de retour est la gamme de tension en millivolts. Sinon, elle est zéro.

Si l'on souhaite déterminer toutes les gammes disponibles, on peut appeler cette procédure à plusieurs reprises, en augmentant le gain chaque fois.

channel 0 - voie A, 1 - voie B

gain une valeur entre 0 et 10

Les gammes suivantes sont disponibles:

0 10mV (212/3 et 216 seulement)  
1 20mV (212/3 et 216 seulement)

2	50mV
3	100mV
4	200mV
5	500mV
6	1V
7	2V
8	5V
9	10V
10	20V

## adc200\_set\_channels

```
unsigned short adc200_set_channels (  
    A200_MODE mode)
```

Cette procédure définit si l'ADC-200 doit recueillir des données d'une ou des deux voies. Si l'opération est réussie, la valeur de retour est le nombre de voies (1 ou 2), sinon, elle est zéro. Sur les dispositifs plus rapides, l'ADC-200 peut doubler le taux d'échantillonnage lorsqu'il recueille des données d'une seule voie.

```
mode          0 - A200_CHANNEL_A - voie A seulement  
              1 - A200_CHANNEL_B - voie B seulement  
              2 - A200_BOTH_CHANNELS - les deux voies
```

## adc200\_set\_oversample

```
unsigned short adc200_set_oversample (  
    unsigned short factor)
```

Cette procédure précise le nombre de mesures à prendre pour chaque échantillon. Un échantillon est la moyenne de ce nombre de mesures.

Ce nombre doit être entre 1 et 16. Plus le facteur de suréchantillonnage augmente, plus le taux d'échantillonnage maximum et le nombre d'échantillons maximum par bloc diminue.

Cette procédure doit être appelée avant les procédures [adc200\\_set\\_timebase](#) et [adc200\\_max\\_samples](#).

## adc200\_set\_timebase

```
unsigned short adc200_set_timebase (  
    unsigned long * ns,  
    unsigned char * is_slow,  
    A200_TIME timebase)
```

Cette procédure précise l'intervalle de temps entre les échantillons. Si `timebase` est valide, la valeur de retour est TRUE et les paramètres `ns` et `is_slow` sont réglés.

```
ns          adc200_set_timebase règle ce paramètre sur l'intervalle de temps, en  
            nanosecondes, entre les échantillons.  
is_slow     adc200_set_timebase règle ce paramètre sur TRUE si l'ADC200 fonctionne  
            en slow mode (mode lent) quand cette base de temps est sélectionnée (c'est à dire  
            qu'on peut transférer les échantillons de l'ADC200 sans l'arrêter).  
timebase    ce paramètre est un nombre entre 0 et 19 (tous les nombres ne sont pas valables pour  
            tous les dispositifs- vérifier la valeur de retour).  
            timebase=0 est la base de temps la plus rapide,  
            timebase=1 est deux fois plus lente  
            timebase=2 est quatre fois plus lente... et cetera
```

L'intervalle par échantillon est normalement  $ns\_min * 2^{timebase} * oversample$ .

Pour l'ADC200-50 ( $ns\_min=20ns$ ) avec `oversample` réglé sur 1, les intervalles par échantillon sont les suivants:

0	20ns
1	40ns
2	80ns
3	160ns
.....	
18	5242880ns
19	10485760ns

Pour l'ADC212 (ns\_min=333ns) avec oversample réglé sur 6, les intervalles par échantillon sont les suivants:

0	2000 ns
1	4000 ns
2	8000ns
....	
15	65536000ns
16	131072000ns

## adc200\_set\_trigger

```
unsigned short adc200_set_trigger (  
    unsigned char enabled,  
    A200_TSOURCE source,  
    A200_TDIR direction,  
    A200_TDELAY delay_percent,  
    short threshold)
```

Cette procédure règle les paramètres pour un événement de lancement, et précise également la position de celui-ci dans le bloc de données.

enabled	Régler sur TRUE si l'ADC200 doit attendre un événement de lancement, et sur FALSE si l'ADC200 doit commencer le recueil de données immédiatement.
source	0 - A200_TSOURCE_A 1 - A200_TSOURCE_B 2 - A200_TSOURCE_E - utiliser l'entrée de lancement
direction	0 - A200_RISING - montant 1 - A200_FALLING - descendant
delay_percent	Ce paramètre règle le délai, exprimé en pourcentage du nombre d'échantillons dans le bloc, entre l'événement de lancement et le début du bloc. Il doit se situer entre -100% et +100%. Ainsi, 0% indique que l'événement de lancement sera le premier échantillon du bloc: -50% indique que l'événement de lancement sera au milieu du bloc.
threshold	Ce paramètre est le seuil de lancement (voies A ou B seulement- l'entrée de lancement externe utilise un seuil TTL... 2,5V). Il est <b>échelonné</b> en valeurs ADC, par exemple entre -2047 et 2048 pour un ADC200.

## adc200\_set\_rapid

```
unsigned short adc200_set_rapid (  
    unsigned short enabled)
```

Cette procédure sert à activer le mode de renouvellement rapide. Dans ce mode, le pilote initialise le dispositif seulement la première fois: les blocs suivants peuvent être recueillis avec un délai minimum entre eux. Les taux de renouvellement sont possibles jusqu'à 200 par seconde.

enabled Régler sur TRUE pour activer le renouvellement rapide, FALSE pour le désactiver. L'exemple suivant montre comment recueillir 50 blocs de 100 échantillons. Noter que le premier appel à adc200\_run prendra 50-100ms de plus que les appels suivants:

```
adc200_set_rapid (TRUE);  
for (i = 1; i < 50; i++)  
{  
    adc200_run (100);  
    while (!adc200_ready ())  
        {};  
    adc200_stop ();  
}
```

```
    adc200_get_values (buffer, buffer, 100);
}
adc200_set_rapid (FALSE);
```

## adc200\_max\_samples

```
unsigned long adc200_max_samples (void)
```

Cette procédure indique le nombre maximum d'échantillons qu'on peut demander avec les réglages actuels. Ce nombre dépend de plusieurs facteurs:

- type d'ADC2xx
- réglage de voies (une/deux voies)
- facteur de suréchantillonnage
- délai de lancement

Il faut donc appeler cette procédure après avoir réglé ces paramètres.

L'ADC-2xx est si rapide qu'il faut plusieurs centaines d'échantillons pour démarrer et arrêter le recueil: le nombre maximum d'échantillons est donc inférieur à la taille du tampon.

Si le lancement est activé avec un délai supérieur à 0%, les échantillons entre l'événement de lancement et le premier échantillon du bloc ne sont pas disponibles: avec un délai de lancement de 100%, la moitié du tampon est inutilisée.

Pour les versions à 8 bits, le dispositif peut acheminer une voie vers les deux tampons: la taille de la mémoire est donc doublée.

Le nombre maximum d'échantillons peut être estimé à l'aide de la formule suivante:

$$\text{max sample} = (\text{buffer size} - 1000) / \text{oversample}$$

## adc200\_run

```
unsigned short adc200_run (unsigned long no_of_values)
```

Cette procédure démarre, soit l'attente d'un événement de lancement, soit le recueil de données. La valeur de retour est TRUE si l'opération a réussi.

no\_of\_values            Ce paramètre règle le nombre d'échantillons à recueillir

## adc200\_ready

```
unsigned short adc200_ready (void)
```

Cette procédure indique si le recueil de données est terminé.

La valeur de retour sera TRUE quand l'ADC-200 aura recueilli le bloc entier de données.

## adc200\_stop

```
void adc200_stop (void)
```

Pour arrêter l'ADC200 (si aucun événement de lancement ne se produit, par exemple), appeler cette procédure. Les données de l'ADC-200 ne seront alors pas valides.

## adc200\_get\_values

```
unsigned long adc200_get_values (  
    short huge *      buffer_a,  
    short huge *      buffer_b,  
    unsigned long     no_of_values)
```

Cette procédure transfère les données de l'ADC200 vers l'ordinateur. Elle s'utilise quand le recueil de données est terminé (voir [adc200\\_ready](#)).

La valeur de retour correspond au nombre de valeurs transférées.

**buffer\_a**      Ce paramètre est l'adresse du tampon destiné aux données de la voie A. Il n'est pas utilisé si la voie A est désactivée. La taille du tampon doit être au moins `no_of_values`.

**buffer\_b**      Ce paramètre est l'adresse du tampon destiné aux données de la voie B. Il n'est pas utilisé si la voie B est désactivée. La taille du tampon doit être au moins `no_of_values`.

**no\_of\_values**      Ce paramètre précise le nombre d'échantillons à transférer.

Pour l'ADC200 et l'ADC-212, chaque valeur sera entre 2047 et -2047: zéro correspond à 0 volts, 2047 à tension maximale de la gamme actuelle.

Pour l'ADC216, chaque valeur sera entre 32767 et -32767: zéro correspond à 0 volts, 32767 à tension maximale de la gamme actuelle.

## adc200\_get\_overflow

```
short adc200_get_overflow (  
    short channel)
```

Cette procédure indique si un dépassement de tension s'est produit au cours du recueil précédent.

Dans l'affirmative, la valeur de retour est TRUE.

**channel**      0 - voie A  
              1 - voie B

## adc200\_get\_single

```
void adc200_get_single (  
    short far *      buffer)
```

Cette procédure démarre l'ADC200, recueille un petit nombre d'échantillons et fournit ensuite la moyenne de ces échantillons. Elle est utilisée en attendant la mise en service du **mode lent**.

**buffer**      Ce paramètre est l'adresse d'un tampon pour deux valeurs. La première valeur sera la moyenne pour la voie A: la deuxième pour la voie B.

## adc200\_set\_frequency

```
long adc200_set_frequency (  
    long frequency)
```

Cette procédure contrôle le générateur de signaux. Si la fréquence est zéro, le générateur de signaux est arrêté. Si elle est entre 1 et 250 000, le pilote démarre le générateur de signaux à la fréquence disponible la plus proche. La valeur de retour correspond à la fréquence réelle.  
frequency    fréquence requise en Hz

**Remarque:**      Le générateur de signaux s'arrêtera si une procédure autre que `adc200_ready` est appelée.

## adc200\_has\_relays

```
short adc200_has_relays (void)
```

Cette procédure indique si le dispositif est doté de relais permettant de contrôler la sélection AC/DC. Si la valeur de retour est TRUE, la procédure `adc200_set_dc` peut être utilisée pour régler la sélection AC/DC.

## adc200\_get\_status

```
short adc200_get_status (void)
```

La valeur de retour de cette procédure indique le statut de l'appel le plus récent de la procédure `adc200_open_unit`. Les codes suivants sont définis dans `adc200.h`:

- 0 - A200\_OK
- 1 - A200\_INVALID\_PORT
- 2 - A200\_INVALID\_HW\_VERSION
- 3 - A200\_INVALID\_SW\_VERSION
- 4 - A200\_CONFIG\_FAILED,
- 5 - A200\_ADDR\_READ\_FAILED,
- 6 - A200\_NVR\_FAIL,
- 7 - A200\_UNIT\_NOT\_FOUND,
- 8 - A200\_INVALID\_LENGTH,
- 9 - A200\_DRIVER\_NOT\_FOUND,
- 10 - A200\_OLD\_DRIVER\_VERSION

Si le statut est autre que zéro, veuillez contacter le support technique de Pico Technology.

## adc200\_get\_product

```
short adc200_get_product (void)
```

La valeur de retour de cette procédure indique le type de dispositif, qui sera une des valeurs suivantes:

- 200 - ADC-200 8-bits, mais le pilote fournira des valeurs de 12 bits
- 212 - ADC-212 12-bits
- 216 - ADC-216 16-bits

## Pilote en mode standard pour DOS

Ce pilote est fourni en format « object », `adc200.obj`. Il fonctionne avec C

Il existe également une version en **mode protégé** pour DOS.

## Pilote en mode protégé pour DOS

Ce pilote, ADC200PM.DLL, fonctionne avec les programmes en C (Borland C V4.52 à l'aide de Borland Powerpack) et en Borland Pascal V7. Ce pilote est disponible sur demande auprès de Pico.

## Pilote à 16 bits Windows

Ce pilote, ADC20016.DLL, fonctionne avec Windows 3.x et avec les applications à 16 bits sous Windows 95 et 98.

Le pilote a été testé avec les applications suivantes:

- Borland C version 4.52
- Delphi version 1
- Visual Basic version 3
- Excel version 5

Le pilote pico.386 est également requis. Ce fichier est normalement chargé au moment de l'installation des logiciels. Pour vérifier que l'installation est réussie, consulter le fichier system.ini, qui devra contenir le texte suivant:

```
[386Enh]
:
:
device=pico.386
:
:
```

## Pilote à 32 bits Windows 95/98

Il existe deux versions du pilote, ADC20032.DLL: l'une fonctionne avec Windows 95/98: l'autre avec Windows NT/2000. Une application à 32 bits fonctionnera sans modifications avec les deux versions. Il faut cependant noter que la version du pilote doit correspondre au système d'exploitation.

Le pilote pico.VXD est également requis. Ce fichier est normalement chargé au moment de l'installation des logiciels. Pour vérifier que l'installation est réussie, consulter le fichier system.ini, qui devra contenir le texte suivant:

```
[386Enh]
:
:
device=pico.VXD
:
:
```

## Windows NT/2000 driver

Il existe deux versions du pilote, ADC20032.DLL: l'une fonctionne avec Windows 95/98: l'autre avec Windows NT/2000. Une application à 32 bits fonctionnera sans modifications avec les deux versions. Il faut cependant noter que la version du pilote doit correspondre au système d'exploitation.

Le pilote adc200.sys est également requis. Il doit être installé dans c:\winnt\system32\drivers. Ce fichier est normalement chargé au moment de l'installation des logiciels. Pour vérifier que l'installation est réussie:

- Cliquer sur Start
- Sélectionner Settings
- Sélectionner Control panel
- Sélectionner Devices
- Vérifier que adc200 existe et est started.

Sinon, vérifier que le fichier existe en c:\winnt\system32\drivers, puis utiliser le programme regdrive.exe, qui se trouve dans le répertoire PICO. Taper:

```
regdrive adc200
```

Remarque: le pilote NT ne peut pas identifier les adresses réelles des ports imprimantes. Il présume qu'elles sont comme suit:

```
LPT1  0x278  
LPT2  0x378  
LPT3  0x3BC
```

Si votre ordinateur n'est pas conforme à cette norme, utilisez le numéro du port correspondant à l'adresse réelle du port.

## Pascal

Il n'est plus possible d'utiliser le pilote DOS en mode standard avec Turbo Pascal V6.  
Voir le [pilote en mode protégé](#) qui fonctionne avec Turbo Pascal V7

## C (DOS)

Pour relier le pilote au programme, suivre les étapes suivantes:

```
#include le fichier adc200.h dans le programme.  
Si un IDE est utilisé, inclure le fichier adc200.obj dans le projet.  
Si un compilateur de commande de ligne est utilisé, inclure le fichier  
adc200.obj dans le fichier liaison
```

Voir aussi a200.c comme exemple de programme DOS simple .

Le pilote DOS a été testé avec les compilateurs suivants:

```
Borland C++ V4.5  
Microsoft Visual C V 1.5  
Watcom C V10.0
```

## C (Windows)

Pour les compilateurs Borland et Watcom, et Microsoft Visual C V1.5, utiliser le programme implib fourni avec le compilateur pour créer un fichier d'imports adc200xx.lib du pilote adc200xx.dll. Utiliser la commande

```
Implib adc200xx.lib adc200xx.dll
```

Pour Microsoft Visual C version 2 et supérieure, Microsoft ne fournit plus le programme implib. Il faut à la place créer un fichier .lib à l'aide d'un [fichier .DEF](#) ( un fichier adc20032.def est inclus) comme suit:

```
Lib /def:adc20032.def
```

Une fois le fichier obtenu:

```
inclure le fichier adc200xx.lib dans le projet
inclure le fichier adc200.h dans le(s) fichier(s) source C du programme.
```

Voir les fichiers a200test.c and a200test.rc comme exemples de programme Windows simple.

## Fichiers .DEF et Visual C

Les noms des procédures en Visual C sont 'décorés'- un suffixe indique le nombre d'octets de paramètres requis. Une procédure ma\_proc qui accepte 2 paramètres de 4 octets sera convertie en ma\_proc@8. Ceci réduit le risque de l'utilisation erronée d'une procédure avec un nombre de paramètres incorrect.

En conséquence, les noms des procédures en Visual C ne correspondent pas aux noms dans les DLLs conçus pour une utilisation avec d'autres applications, telles que Excel, Visual Basic, Delphi, Labview, HP-Vee etc.....

Le programme LIB, fourni avec Visual C, est censé créer un fichier .lib fournissant, à la fois une interface au .dll, et une conversion des noms Visual C en noms .dll. Malheureusement, le programme LIB semble avoir un défaut important: il interprète mal le caractère '@' placé en tête du suffixe. Il est donc nécessaire d'utiliser la liaison par nombre (ordinal) plutôt que par nom.

Si l'on souhaite créer un fichier .DEF pour un DLL sans noms décorés, On peut d On peut déterminer les nombres ordinaux à l'aide du programme DUMPBIN, qui s'utilise comme suit:

```
DUMPBIN /exports xxxxxx.dll
```

Pour déterminer le suffixe pour une procédure, calculer le nombre d'octets de paramètres en multipliant le nombre de paramètres par quatre, puis ajouter un signe @ entre le nom et le nombre d'octets. Une procédure ma\_proc qui accepte 2 paramètres de 4 octets sera convertie en ma\_proc@8.

Voir le fichier adc20032.def pour le format à utiliser pour votre fichier xxxxxx.def. La commande suivante produira un fichier .lib:

```
Lib /def:xxxxxx.def
```

## Delphi

Le programme adc200.dpr montre comment utiliser l'ADC200. Le fichier adc200.inc contient des prototypes de procédure, pouvant être inclus dans un programme personnel. Ceci a été testé avec les versions 1, 2 et 3 de Delphi.

## Excel

### Excel 5

Charger le fichier tableur adc20016.xls  
Sélectionner Tools | Macro  
Sélectionner getadc200  
Sélectionner Run

### Excel 7 (Office 95 etc)

Charger le fichier tableur adc20032.xls

Sélectionner Tools | Macro  
Sélectionner getadc200  
Sélectionner Run

**Remarque:** le langage macro d'Excel est semblable à Visual Basic. Pour les procédures dont la valeur de retour est TRUE ou FALSE, la valeur de retour réelle est 0 pour FALSE et 1 pour TRUE. Visual Basic utilisant 65535 pour TRUE, il est conseillé d'utiliser le test '>0' plutôt que '= TRUE'.

## Visual Basic

### Version 3

Le répertoire WIN16 renferme les fichiers suivants:

ADC200.mak  
adc20016.frm - formulaire, prototypes et programme

### Version 4 and 5

Le répertoire WIN32 renferme les fichiers suivants:

adc20032.vbp - fichier projet  
adc20032.bas - prototypes de procédure  
adc20032.frm - formulaire et programme

**Remarque:** Pour les procédures dont la valeur de retour est TRUE ou FALSE, la valeur de retour réelle est 0 pour FALSE et 1 pour TRUE. Visual Basic utilisant 65535 pour TRUE, il est conseillé d'utiliser le test '>0' plutôt que '= TRUE'.

## LabVIEW

Le fichier adc200.vi dans le répertoire DRIVERS\WIN32 indique comment accéder aux procédures dans le pilote. Cet exemple a été testé avec LabVIEW pour Windows 95 version 4.0. Pour l'utiliser, procéder comme suit:

- copier adc200.vi et adc20032.dll sur le répertoire user.lib de Labview
- Lancer LabVIEW et charger adc200.vi
- Sélectionner le port imprimante auquel est connecté l'adc200
- Cliquer sur RUN

Remarque: avec l'ADC216, il sera nécessaire de modifier l'échelonnage dans cadre 7 en remplaçant 2048 par 32768.

## HP-Vee

Le programme d'exemple adc200.vee montre comment recueillir un bloc de données de l'ADC200. Le fichier adc200.vh contient les prototypes des procédures. Cet exemple a été testé avec HP-Vee version 5 sous Windows 98.

## Driver Linux

Voir les informations **man** dans le fichier adc200.tar pour tout renseignement complémentaire.

## LabWindows

LabWindows est un programme en mode protégé DOS qui utilise les conventions de liaison C. Les fichiers adc200lw.obj et adc200.lvh sont disponibles sur demande auprès de Pico.

## Introduction

Le dispositif ADC-2xx est un convertisseur analogique-numérique à grande vitesse comportant deux voies d'entrée. Les gammes de tension, le couplage (AC/DC), le taux d'échantillonnage, les paramètres de lancement etc sont tous sous contrôle des logiciels. Il existe trois versions de l'ADC200, ainsi que de l'ADC212 et 216. Voir les **caractéristiques** pour des informations sur les modèles disponibles.

On peut utiliser ce dispositif comme Oscilloscope virtuel à l'aide des logiciels PicoScope fournis, ou comme enregistreur de données à l'aide des logiciels PicoLog. On peut également utiliser les pilotes ADC200 fournis pour développer ses propres logiciels de recueil et d'analyse de données.

Le coffret ADC-2xx contient les éléments suivants:

- Dispositif ADC-2xx
- câble D25 (parallèle)
- Bloc d'alimentation (12V @ 500mA) - disponible en format GB/UE/US
- Logiciels (soit sur trois disquettes, soit sur un CD)
- Guide d'installation

## Connexion vers PC

Pour utiliser l'ADC-2xx:

- connecter le câble D25 entre l'ADC2xx et le port imprimante de son ordinateur
- brancher le bloc d'alimentation sur une prise secteur
- brancher la broche d'alimentation DC sur l'ADC2xx
- vérifier que le voyant rouge s'allume.

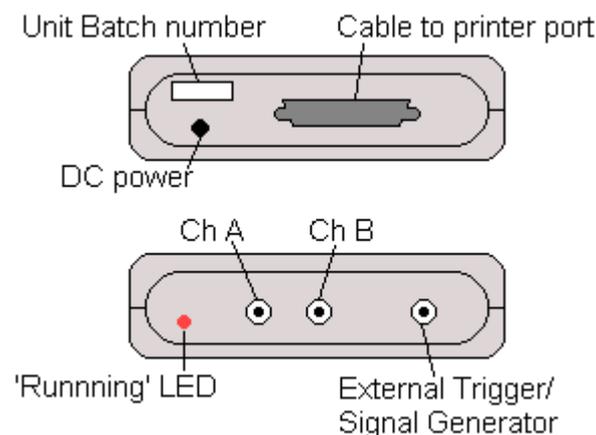
**Remarque:** après le lancement des logiciels, le voyant ne sera allumé que lorsque le recueil de données sera en cours.

L'ADC2xx est doté des même connecteurs qu'un oscilloscope: on peut donc utiliser les sondes d'oscilloscope standards. L'impédance d'entrée est la même, donc la fonction x10 marche normalement.

Le connecteur BNC marqué 'E' a deux fonctions:

- entrée de lancement pour les signaux TTL
- générateur de signaux carrés

Le générateur de signaux doit être désactivé si l'on souhaite utiliser l'entrée de lancement.



Pour vérifier que le dispositif fonctionne bien, démarrer PicoScope. PicoScope affichera la tension sur le connecteur voie A. Si l'on utilise les sondes d'oscilloscope, on peut toucher le bout de la sonde pour produire un signal de 50Hz secteur.

