

## DigiTools I/O

### Caractéristiques

- \* Fonctions Reset et Initialisation des périphériques.
- \* Fonctions d'acquisition de signaux analogiques et logiques.
- \* Fonctions de restitution de signaux analogiques et logiques.
- \* Fonctions de comptage d'impulsions.
- \* Fonctions de génération de signaux.
- \* Fonctions de gestion d'interruptions.
- \* Renvoi d'une erreur sur chaque fonction.

### Applications

- \* Acquisition et traitement de données.
- \* Commande et régulation de dispositifs analogiques ou logiques.
- \* Comptage d'évènements
- \* Toute application interfacée par des cartes Digimétrie

### Introduction

DigiTools est un ensemble de routines réalisant une interface standard d'application indépendante de l'interface matérielle utilisée. Il s'agit d'une boîte à outils logicielle d'acquisition et de contrôle de processus à la disposition du programmeur sous DOS ou Windows (DLL).

Ces outils logiciels permettent l'accès à la majorité des cartes Digimétrie qu'elles aient des fonctions analogiques ou logiques. Les routines sont développées en C, les sources sont fournies afin de pouvoir recompiler dans une version ou modèle différent. DigiTools I/O s'adresse au programmeur qui doit réaliser une application spécifique sans entrer dans la programmation des registres de la carte.

Le DigiTools I/O de chaque carte est livré avec celle-ci (fichiers .OBJ, .C, .H) ainsi que la .DLL correspondante pour Windows. Si l'utilisateur désire plus d'informations que ce qui est indiqué dans le .H pour utiliser la DLL sous windows, il peut alors commander la documentation complète.

## FONCTION RESET:

**Reset\_carte** (mode,ext,carte,adresse) **Fonction:** Permet le test des différents modules de la carte AD-DA-TOR-TIM et une remise à l'état initialisation de tous les registres.

## FONCTIONS ENTREES/SORTIES Tout Ou Rien:

**Dinit** (portx, mode, no\_composant, carte, adresse) **Fonction:**Initialiser le sens des lignes logiques .

**Din** (\*resultat, portx, no\_composant, carte, adresse) **Fonction:** Lire un port (8bits) d'entrées/sorties logiques.

**Dout** (data, portx, no\_composant, carte, adresse) **Fonction:** Ecrire un port (8bits) d'entrées/sorties logiques .

**Dinscan** (port\_tampon, port, n\_ports\_max, n\_ports, n\_composant, n\_val, \*fréquence, mode, \*tampon, exts, carte, adresse)

**Fonction:** Remplir un tampon à partir d'un port logique en programmant le fréquence d'acquisition et le nombre de ports.

**Doutscan**(port\_tampon, port, n\_ports\_max, n\_ports, n\_composant, n\_val, \*fréquence, mode, \*tampon, exts, carte, adresse)

**Fonction:** Restituer un tampon sur un port logique en programmant le fréquence d'acquisition et le nombre de ports.

### **Description:**

**port\_tampon, n\_ports\_max** numéro du premier port et nombre de port max du tampon

**port, n\_ports** correspondent aux ports A, B ou C (0, 1, 2) et au nombre.

**mode** sert à définir le sens de transfert des données.

Un bit à 1 signifie voie en sortie, un bit à 0 = voie en entrée

**no\_composant** correspond au numero (De 0 à N-1) du composant sur la carte quand il y a plusieurs composants par carte

(ex: 2 groupes de 24 lignes (ports A, B, C) pour une carte

PC\_TOR48 (équivalent à 2 PIO 8255)==>no\_composant= 0 ou 1)

```
/* program demo din.c */
#include <stdio.h>
#include <digitool.h>
main ()
{
    WORD data;          /* Valeur du port */
    BYTE portx=PORTA;  /* Initialisation port A */
    WORD entree=0;     /* PORT en entrée */
    WORD erreur;
    /* port A en entrée du PIO 0 carte*/
    /* PC_TOR48 adresse 320 hexa */
    erreur=Dinit(portx,entree,0,PC_TOR48,0x320);
    erreur=Din(&data,portx,0,PC_TOR48,0x320);
    printf(« PORT A=%X\n»,data);
    getch();
}
```

## FONCTIONS ENTREES/SORTIES ANALOGIQUES

**Ain** (\*data, voie, gain, mode, exts, carte, adresse)

**Fonction:** Lire un port d'entrée analogique.

**Ainif** (\*data, voie, gain, mode, exts, carte, adresse)

**Fonction:** Lire un port d'entrée analogique pour écriture dans un tableur..

**Ainscan** (voie\_tampon, voie, n\_voies\_max, n\_voies, \*gain, n\_val, \*fréquence, mode, \*tampon, exts, carte, adresse)

**Fonction:** Remplir un tampon à partir d'un port d'entrée analogique en programmant la fréquence d'acquisition et le nombre de voies.

**Aout** (data, voie, mode, ext, carte, adresse)

**Fonction:** Ecrire sur un port de sortie analogique.

**Aoutf** (data, voie, mode, ext, carte, adresse)

**Fonction:** Ecrire sur un port de sortie analogique à partir d'un tableur.

**Aoutscan** (voie\_tampon, voie, nvoies\_max, n\_voies, n\_val, \*fréquence, tint, mode, \*tampon, ext, carte, adresse)

**Fonction:** Restituer un tampon sur un port analogique.

### Description:

**voie\_tampon, n\_voies\_max** sont la première voie et le nombre de voie max sur le tampon.

**voie, n\_voies** sont la première voie et le nombre sélectionnées sur la carte.

**gain** est la gain programmé pour les cartes possédant un amplificateur à gain programmable.

**mode** détermine les différents modes disponibles

sur les cartes (monopolaire/bipolaire-simple différentiel.....).

**tampon** pointeur de départ pour le stockage des données.

**fréquence** est la fréquence d'échantillonnage.

```
/* demo program ain */
#include <stdio.h>
#include <digitool.h>
main ()
{
    WORD data;
    WORD voie=0;          /* voie choisie          */
    WORD gain=0;         /* tableau des gains par voie */
    WORD mode=0;
    WORD carte= AT_LAB12B; /* nom de la carte          */
    WORD adresse=0x320; /* Adresse de la carte      */
    mode=mode|DT_AD_SIMP;
    do
    {
        erreur= Ain(&data,voie,gain,mode,0,carte,adresse);
        printf(" Data=%i \n",data);
    }while(!kbhit());
}
```

## FONCTIONS

### Gestion d'interruption sous DOS

**install\_dos\_irq** (irq, source, interrup, carte, adresse)

**Fonction:** Initialiser sous **DOS** un nouveau vecteur d'interruption et valide son déclenchement.

**desinstall\_dos\_irq** (irq, carte, adresse)

**Fonction:** dévalide et restaure sous DOS une gestion d'interruption initialisée par la fonction install\_dos\_irq().

**test\_valid\_eoi** (irq)

**Fonction:** permet d'acquiescer sous DOS une interruption matérielle déclenchée.

### Description:

**source:** définit la source matérielle de l'interruption.

## FONCTIONS GENERATEUR et COMPTEUR

### GENERATEUR:

**Ginit** (\*fréquence, no\_composant, mode, carte, adresse)

**Fonction:** Initialiser un générateur 16 ou 32 bits sur un timer (ex:8254 intel).

**Gstart** (no\_composant, mode, carte, adresse)

**Fonction:** Démarrer un générateur 16 ou 32 bits.

**Gstop** (no\_composant, mode, carte, adresse)

**Fonction:** Arrêter un générateur 16 ou 32 bits.

### Description:

**fréquence** est la fréquence du générateur en Hz.

**no\_composant** donne le n° compteur et le n° timer pour le générateur.

**mode** donne le type de générateur 16bits ou 32bits.

### COMPTEUR:

**Cinit** (valeur,no\_composant, mode, carte, adresse)

**Fonction:** Initialiser un compteur 16 bits.

**Cstart** (no\_composant, carte, adresse)

**Fonction:** Démarrer un compteur 16 bits.

**Cstop** (no\_composant, carte, adresse)

**Fonction:** arrêter un compteur 16 bits.

**Cread** (\*result, no\_composant, carte, adresse)

### Description:

**valeur** est le nombre de clocks à compter.

**no\_composant** donne le n° compteur (0,1,2 pour un 8254) et le n° timer pour le compteur.

**mode** donne le type de compteur 16bits et le mode de comptage.

### Informations pour commander

Les fichiers : .H, DLL et sources en C pour DOS (avec détails des fonctions et paramètres dans le fichier .H) sont livrés gratuitement avec la carte commandée sur demande.

Pour obtenir la documentation détaillée des fonctions et un exemple pour chacune d'elle, l'utilisateur peut commander les références suivantes:

Fonctionne en 16 bits et 32 bits, Windows 3.0, 95, 98, NT.

- DigiTools I/O-C** (C,C++:MS Visual C 1,3. Borland).
- DigiTools I/O-VB** (visual Basic 4).
- DigiTools I/O-DELPHI** (Delphi 3,4).

REV 2.0/02.97