

AQB11/12

Núcleo das funções para controle da placa de aquisição de dados

Sumário

Descrição geral	1
Funções dedicadas	2
Declare Sub SelPlaca Lib "aqb1032.dll" (ByVal ender%).....	2
Dispositivo PIO	2
Declare Sub ConfPIO Lib "aqb1032.dll" (ByVal modo%)	2
Declare Sub ConfCnPIO Lib "aqb1032.dll" (ByVal canal%, ByVal modo%).....	2
Declare Sub EscCnPIO Lib "aqb1032.dll" (ByVal canal%, ByVal dado%).....	3
Declare Function LeCnPIO Lib "aqb1032.dll" (ByVal canal%) As Integer3	
Declare Function LeBitCnPIO Lib "aqb1032.dll" (ByVal cnPIO%, ByVal bitPIO%) As Integer	3
Declare Sub EscBitCnPIO Lib "aqb1032.dll" (ByVal cnPIO%, ByVal bitPIO%, ByVal bit%)	3
Dispositivo ADC	4
Declare Sub SelCnADC Lib "aqb1032.dll" (ByVal canal%)	4
Declare Function LeADC Lib "aqb1032.dll" () As Integer	4
Declare Function LeCnADC Lib "aqb1032.dll" (ByVal canal%) As Integer.....	4
Declare Function LeCnADCmed Lib "aqb1032.dll" (ByVal canal%, ByVal nmed%) As Integer	5
Dispositivo DAC	5
Declare Sub EscCnDAC Lib "aqb1032.dll" (ByVal canal%, ByVal dado%).....	5
Dispositivo CTD	5

```

    Declare Sub ConfCnCTD Lib "aqb1032.dll" (ByVal canal%, ByVal
modo%).....5
    Declare Function LeCnCTD Lib "aqb1032.dll" (ByVal canal%) As Long6
    Declare Sub EscCnCTD Lib "aqb1032.dll" (ByVal canal%, ByVal
dado&) .....6
    Declare Function LeOutCnCTD Lib "aqb1032.dll" (ByVal canal%) As
Integer.....6
    Declare Function LeStsCnCTD Lib "aqb1032.dll" (ByVal canal%) As
Integer.....6
    Declare Function EspTrans Lib "aqb1032.dll" (ByVal cnPIO%, ByVal
bitPIO%) As Integer .....6
    Declare Sub LeVetCnADCbt Lib "aqb1032.dll" (ByVal canal%, ByVal
cnCTD%, buf%, ByVal n%, ByVal nmed%) .....7
    Declare Sub LeVetADC Lib "aqb1032.dll" (canais%, ByVal cnPIO%,
ByVal bitPIO%, buf%, ByVal n%, ByVal med%, ByVal evn%).....7
    Declare Sub LeVetCnADC Lib "aqb1032.dll" (ByVal canal%, ByVal
cnPIO%, ByVal bitPIO%, buf%, ByVal n%, ByVal nmed%, ByVal evn%).....7

```

Descrição geral

Para acessar as funções descritas abaixo, deve-se seguir o procedimento descrito à seguir:

- copiar a biblioteca AQB1032.dll para o subdiretório C:\WINDOWS\SYSTEM
- adicionar o módulo AQB1032.BAS no seu programa em Visual Basic 5.0

A placa de aquisição possui DISPOSITIVOS e cada um deles pode possuir um ou mais CANAIS:

Os dispositivos são:

PIO: Entrada/Saída paralela, implementada com um 8255.

ADC: Entrada analógica, implementada com um multiplexador analógico MPC506 (Burr-Brown) e um conversor A/D ADS7804 (Burr-Brown).

DAC: Saída analógica, implementada com um DAC7545 (Burr-Brown).

CTD: Contador, implementado com um 8254.

Cada um deles possui os seguintes números de canais:

PIO: 3 canais 0-2

ADC: 16 canais 0-15

DAC: 2 canais 0-1

CTD: 3 canais 0-2

Os dispositivos devem ser configurados genericamente por meio de funções de configuração do tipo:

Conf**XXX** (**m**) Configura dispositivo **XXX** no modo **m**

Os canais normalmente possuem as seguintes funções:

ConfCn**XXX** (**c,m**) Configura o canal **c** no modo **m**

EscCn**XXX** (**c,d**) Escreve o dado **d** no canal **c**

LeCn**XXX** (**c,d**) Le o dado **d** do canal **c**. **d** pode ser dispensado

Funções dedicadas

Alguns dispositivos apresentam funções e subrotinas dedicadas. As funções e subrotinas relacionadas abaixo fazem parte da biblioteca AQB1032.dll. O primeiro parágrafo refere-se à declaração da rotina existente no módulo AQB1032.bas e logo a seguir encontram-se um comentário e a sintaxe de utilização em VB5.0.

Declare Sub SelPlaca Lib "aqb1032.dll" (ByVal ender%)

Estabelece o endereço base da placa de aquisição de dados. Este endereço deve respeitar o que foi definido nos *jumpers* de configuração do *hardware*.

Sintaxe: SelPlaca &H300

Configura a AQB11/12 no endereço 300 hexadecimal.

Este comando deve ser efetuado antes do primeiro acesso a qualquer função da placa de aquisição.

Dispositivo PIO

Declare Sub ConfPIO Lib "aqb1032.dll" (ByVal modo%)

Configura o modo de operação da PIO (porta de entrada e saída digital). Verifique o manual do fabricante do 8255 para maiores detalhes.

Sintaxe: ConfPio m

onde:

m=0 configura a PIO no modo simples

m=1 configura a PIO no modo de handshake 1

m=2 configura a PIO no modo de handshake 2

Declare Sub ConfCnPIO Lib "aqb1032.dll" (ByVal canal%, ByVal modo%)

Configura um canal da PIO como entrada ou saída.

Sintaxe: ConfCnPIO c,m

onde:

c canal a ser configurado

m=0 configura o canal c da PIO como entrada

- m=1 configura o canal c da PIO como saída
- m=2 configura os 4 bits menos significativo como entrada e os mais significativos como saída. **Este modo é válido somente para o canal 2.**

Declare Sub EscCnPIO Lib "aqb1032.dll" (ByVal canal%, ByVal dado%)

Escreve um dado de 8 bits diretamente num canal da PIO. Este canal já deverá estar configurado anteriormente com saída.

Sintaxe: EscCnPIO c,dado

onde:

- c o canal onde o dado será escrito
- dado valor de 8 bits (0 à 255) que será escrito no canal c da PIO

Declare Function LeCnPIO Lib "aqb1032.dll" (ByVal canal%) As Integer

Realiza a leitura de um dado de 8 bits que está presente no canal da PIO. Este canal já deverá estar configurado anteriormente com entrada.

Sintaxe: X = LeCnPIO (c)

onde:

- c o canal onde o dado está disponível
- X é a variável do seu programa em VB que deverá receber o dado e, preferencialmente, deve ser do tipo **INTEIRA**.

Declare Function LeBitCnPIO Lib "aqb1032.dll" (ByVal cnPIO%, ByVal bitPIO%) As Integer

Realiza a leitura de um bit que está presente no canal da PIO. Este canal já deverá estar configurado anteriormente com entrada.

Sintaxe: X = LeBitCnPIO (c,bit)

onde:

- c o canal onde o dado está disponível
- bit número do bit desejado (0 à 7).
- X é a variável do seu programa em VB que deverá receber o dado (0 ou 1) e, preferencialmente, deve ser do tipo **INTEIRA**.

Declare Sub EscBitCnPIO Lib "aqb1032.dll" (ByVal cnPIO%, ByVal bitPIO%, ByVal bit%)

Escreve um dado (0 ou 1) no bit especificado, sem alterar os outros bits, do canal da PIO. Este canal já deverá estar configurado anteriormente com saída.

Sintaxe: EscBitCnPIO c, bit, dado

onde:

c o canal onde se deseja escrever o bit
 bit número do bit desejado (0 à 7).
 dado o valor (0 ou 1) que se deseja que a saída assuma.

Dispositivo ADC

Declare Sub SelCnADC Lib "aqb1032.dll" (ByVal canal%)

Seleciona um canal do conversor A/D. Uma vez selecionado o canal não é mais necessário efetuar esta operação, até que quando desejar muda o canal a ser lido. Este comando é indicado quando deseja-se ler somente um canal com elevada taxa de aquisição juntamente com a função LeADC.

Sintaxe: SelCnADC c

onde:

c o canal onde deseja-se ler a tensão

Declare Function LeADC Lib "aqb1032.dll" () As Integer

Realiza uma conversão A/D do sinal presente na entrada analógica pré-selecionada. Mais veloz que a rotina LeCnADC, pois não efetua a seleção do canal.

Sintaxe: X = LeADC

onde:

c o canal onde deseja-se ler a tensão

X é a variável do seu programa em VB5.0 que deverá receber o dado de 12 bits (0 à 4095) e, preferencialmente, deve ser do tipo **INTEIRA**.

Declare Function LeCnADC Lib "aqb1032.dll" (ByVal canal%) As Integer

Realiza uma conversão A/D do sinal presente na entrada analógica especificada.

Sintaxe: X = LeCnADC (c)

onde:

c o canal onde deseja-se ler a tensão

X é a variável do seu programa em VB que deverá receber o dado de 12 bits (0 à 4095) e, preferencialmente, deve ser do tipo **INTEIRA**.

Declare Function LeCnADCmed Lib "aqb1032.dll" (ByVal canal%, ByVal nmed%) As Integer

Realiza uma conversão A/D do sinal presente na entrada analógica especificada. Retorna o valor resultante da média do número de leituras solicitado.

Sintaxe: X = LeCnADCmed (c, med)

onde:

c o canal onde deseja-se ler a tensão
 med número de leituras consecutivas para cálculo da média
 X é a variável do seu programa em VB que deverá receber o dado de 12 bits resultante (0 à 4095) e preferencialmente deve ser do tipo **INTEIRA**.

Dispositivo DAC

Declare Sub EscCnDAC Lib "aqb1032.dll" (ByVal canal%, ByVal dado%)

Converte o dado para uma tensão analógica na saída analógica

Sintaxe: EscCnDAC c,dado

onde:

c o canal do DAC onde deseja-se disponibilizar uma tensão analógica
 dado o dado de 12 bits (0 à 4095) que se pretende converter em um sinal analógico. Deve ser um valor **INTEIRO**.

Dispositivo CTD

Declare Sub ConfCnCTD Lib "aqb1032.dll" (ByVal canal%, ByVal modo%)

Configura o modo de operação de um canal do dispositivo CTD.

Sintaxe: ConfCnCTD c,n

onde:

c é o canal a ser configurado
 n=0 configura o canal c do CTD como atraso progr + pulso
 n=2 configura o canal c da CTD como um monoestável
 n=3 configura o canal c da CTD como divisor de frequência
 n=4 configura o canal c da CTD como gerador de gate

Declare Function LeCnCTD Lib "aqb1032.dll" (ByVal canal%) As Long

Le o valor de contagem de um canal do dispositivo CTD. Esta contagem se inicia no valor programado (no máximo 65535) no canal e é decrementada a cada evento contado, até atingir zero.

Sintaxe: X = LeCnCTD (c)

onde:

c é o canal a ser lido

X é a variável do seu programa em VB que receberá a contagem. Ela deverá ser do tipo **LONG**.

Declare Sub EscCnCTD Lib "aqb1032.dll" (ByVal canal%, ByVal dado&)

Escreve o valor de contagem inicial (no máximo 65536) de um canal do dispositivo CTD. Esta contagem apenas é carregada efetivamente no contador se houver um evento a ser contado.

Sintaxe: EscCnCTD c, dado

onde:

c é o canal a ser lido

dado é a contagem inicial a ser escrita. Ela deverá ser do tipo **LONG**.

Declare Function LeOutCnCTD Lib "aqb1032.dll" (ByVal canal%) As Integer

Le o valor presente no bit de saída do contador especificado (pino OUT da 8254). Retorna 0 ou 1.

Sintaxe: X = LeOutCnCTD (c)

onde:

c é o canal cuja saída será lida

X é a variável do seu programa em VB que receberá o bit. Ela deverá ser do tipo **INTEGER**.

Declare Function LeStsCnCTD Lib "aqb1032.dll" (ByVal canal%) As Integer

**** aguardando alterações

Declare Function EspTrans Lib "aqb1032.dll" (ByVal cnPIO%, ByVal bitPIO%) As Integer

**** aguardando alterações

Declare Sub LeVetCnADCbt Lib "aqb1032.dll" (ByVal canal%, ByVal cnCTD%, buf%, ByVal n%, ByVal nmed%)

Lê um conjunto de pontos em um determinado canal do ADC a uma taxa determinada pelo contador especificado. Retorna o conjunto de pontos, cada um deles representando uma medida (com o número de médias especificado). O contador deve estar programado no modo de geração de onda quadrada, e **deve haver uma onda quadrada na sua saída quando executada esta função**, sob risco de travar o computador.

Sintaxe: LeVetCnADCbt(cADC, cCTD, buf, n, nMed)

onde:

cADC é o canal do ADC a ser lido

cCTD é o canal do CTD que comandará a aquisição

buf é o vetor do tipo INTEGER que receberá os dados lidos

n é o número de pontos a serem lidos

nMed é o número de médias feitas em cada ponto

Obs: Quando invocada a função, o vetor buf deve ser passado por meio do seu primeiro elemento, e não de maneira genérica. Se isto não for feito, o computador pode **travar completamente**, necessitando de *reset*.

Correto: buf(0), ou buf(1) se o início do vetor for 1

Errado: buf() ou buf

Declare Sub LeVetADC Lib "aqb1032.dll" (canais%, ByVal cnPIO%, ByVal bitPIO%, buf%, ByVal n%, ByVal med%, ByVal evn%)

Le um conjunto de pontos em vários canais do ADC a uma taxa determinada pela transição de um bit externo lido pela PIO. Retorna o conjunto de pontos, cada um deles representando uma medida (com o número de médias especificado) em um canal.

Sintaxe: LeVetADC(cADC, cPIO, bPIO, buf, n, nMed, tran)

onde:

cADC é um vetor de **INTEGER** de 16 posições (0 a 15) que especifica quais canais serão lidos. Por exemplo, um vetor com estes valores {1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0} indica que devem ser lidos os canais 0, 4, 8 e 12.

cPIO é o canal da PIO ligado ao sinal que comandará a aquisição

bPIO é o bit da PIO ligado ao sinal que comandará a aquisição

buf é o vetor do tipo **INTEGER** que receberá os dados lidos

n é o número de pontos a serem lidos

nMed é o número de médias feitas em cada ponto.

tran é o tipo de transição onde será feita a leitura de todos os canais do ADC. 0 é positiva, 1 é negativa e 2 são em ambas

Obs: Quando invocada a função, os vetores buf e cADC devem ser passados por meio do seu primeiro elemento, e não de maneira genérica. Se

isto não for feito, o computador pode **travar completamente**, necessitando de *reset*.

Correto: buf(0), ou buf(1) se o início do vetor for 1

Errado: buf() ou buf

Declare Sub LeVetADCbt Lib "aqb1032.dll" (canais%, ByVal cnCTD%, buf%, ByVal n%, ByVal med%)

Le um conjunto de pontos em vários canais do ADC a uma taxa determinada pelo contador especificado. Retorna o conjunto de pontos, cada um deles representando uma medida (com o número de médias especificado). O contador deve estar programado no modo de geração de onda quadrada, e **deve haver uma onda quadrada na sua saída quando executada esta função**, sob risco de travar o computador. Deve ser observado que as aquisições ocorrerão tanto na subida quanto na descida da onda quadrada gerada pelo CTD especificado.

Sintaxe: LeVetADCbt(cADC, cCTD, buf, n, nMed)

onde:

cADC é um vetor de **INTEGER** de 16 posições (0 a 15) que especifica quais canais serão lidos. Por exemplo, um vetor com estes valores {1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0} indica que devem ser lidos os canais 0, 4, 8 e 12.

cCTD é o canal do CTD que comandará a aquisição

buf é o vetor do tipo **INTEGER** que receberá os dados lidos

n é o número de pontos a serem lidos

nMed é o número de médias feitas em cada ponto.

Obs: Quando invocada a função, os vetores buf e cADC devem ser passados por meio do seu primeiro elemento, e não de maneira genérica. Se isto não for feito, o computador pode **travar completamente**, necessitando de *reset*.

Correto: buf(0), ou buf(1) se o início do vetor for 1

Errado: buf() ou buf