

# *Easy Lab IO e AG*

## Manual do usuário

Revisão 1.7

17/12/2010



[www.dma.ind.br](http://www.dma.ind.br)



A DMA ELECTRONICS projeta e fabrica sistemas para aquisição e registro de dados com conexão a um computador do tipo PC. Embora seus produtos possuam mecanismos de proteção, descritos neste manual, é possível que, em casos de má utilização, cause danos aos sistemas de aquisição ou registro de dados e possivelmente ao PC a qual está conectado.

Exceto especificado, os produtos da DMA ELECTRONICS não são projetados para a utilização em componentes críticos no suporte a vida ou em sistemas onde o mau funcionamento pode se esperar no resultado de ferimentos de pessoas.

No caso da utilização de seus produtos como OEM ou incorporação em produtos de terceiros, a DMA ELECTRONICS não fornecerá suporte, instruções ou ser responsabilizada por qualquer dano ao consumidor pelo produto de terceiros.

### **Garantia:**

O Easy Lab está coberto por garantia limitada de um ano contra qualquer defeito de fabricação. Esta garantia não cobre mau uso ou conexões de sinais fora da especificação contida neste manual, nem possíveis danos ao PC conectado devido a tal.

No caso de uso da garantia, o produto deverá ser enviado para a DMA ELECTRONICS (frete por conta do consumidor). Caso constatado defeito de fabricação, os reparos ou substituição serão realizados e os custos de envio do produto ao consumidor serão da DMA ELECTRONICS.

Copyright © 2009, DMA Electronics

# Conteúdo

1.Introdução ao Produto.....	4
1.1 Bem vindo!.....	4
1.2 Especificações técnicas.....	4
2.Descrição do Produto.....	5
2.1 Conexões.....	5
2.2 Conexão de sinais digitais.....	6
2.3 Acionando cargas de potência.....	7
2.4 Conexão Sensores analógicos com saída 4-20mA.....	8
2.5 Conexão de Sinais analógicos com tensões maiores de 2.9v .....	9
3.Utilização do Produto.....	9
3.1 Instalação dos Drivers.....	9
3.2 Softwares.....	10
3.3 Programa Executável “Easy_Lab” .....	10
3.4 Planilha Excel.....	11
4.SDK.....	13
4.1 VB6.....	13
4.2 Excel.....	13
4.3 Delphi.....	13
4.4 Labview.....	13
4.5 Funções DLL.....	14

# 1. Introdução ao Produto

## 1.1 Bem Vindo!

O Easy Lab é um módulo USB com entradas/saídas digitais flexíveis, saída PWM, frequencímetro/contador de pulsos e entradas analógicas (somente versão AG) com conexão USB a um PC ou notebook.

Pode ser aplicado em análise de circuitos lógicos, frequencímetro, controlador de velocidade de motores elétricos, acionador de atuadores diversos, projetos de automação, CLP (controlador lógico programável), Data-Logger, etc.

O módulo acompanha software exclusivo e uma planilha em Excel, com código totalmente aberto com comentários, para leitura e envio de dados para o módulo. Também é fornecido uma biblioteca DLL com todas as funções do módulo, permitindo sua utilização com qualquer software de desenvolvimento na plataforma Windows, como: Delphi, VB, Excel, Labview, Mathcad, etc.

## 1.2 Especificações Técnicas

- Conexão USB v1.1/2.0, cabo USB integrado;
- 9 I/O's flexíveis compartilhadas: Entrada ,Saída digital, PWM, Frequencímetro e Contador de pulsos, (configuração via software);
- 1 PWM com resolução de 10bits, freqüência máxima de 11,7Khz (15 opções de freqüência);
- 1 Frequencímetro de pulsos (0,2Hz a 12Khz, 16bits de res.) + contador de pulsos com resolução de 16bits;
- Nível de sinais TTL;
- Conexão de múltiplos módulos em um mesmo PC;
- 7 entradas analógicas de precisão (14bits), 0 a +2.9v (somente versão AG).
- Sensor interno de temperatura ambiente (somente versão AG).

Parâmetro	Min	Típico	Máx	Unidade	Obs
Quantidade de canais	-	9	-	-	Compartilhadas: entrada digital / analógica, saída, PWM, Frequencímetro.
Impedância de entrada	-	100	-	MΩ	
Tensão Borne +5V	4.8	5.0	5.2	V	Tensão barramento USB.
Corrente saída Borne +5V	-	-	50	mA	
Tensão de entrada Digital	0	-	+5.5	V	Limites máximos.
Leitura Digital High	3.6	-	5.5	V	
Saída Digital High*	4.3	-	5.5	V	Corrente máx por I/O: 15mA
Taxa de amostragem / atualização	40	250	250	Hz	Variável conforme resolução analógica (AG)
Resolução A/D	10	12	14	bits	7 canais, 0.002 a +2.9V. Precisão 0.1% F.S. Somente versão AG
Sensor Temperatura**	10	-	50	°C	Res. 0.5°, precisão +/- 2°C, t. resp. 20min.

\*A soma das correntes de todas as I/Os não pode ultrapassar 50mA.

\*\* Precisão indicada quando todas as portas estão configuradas como “entrada”.

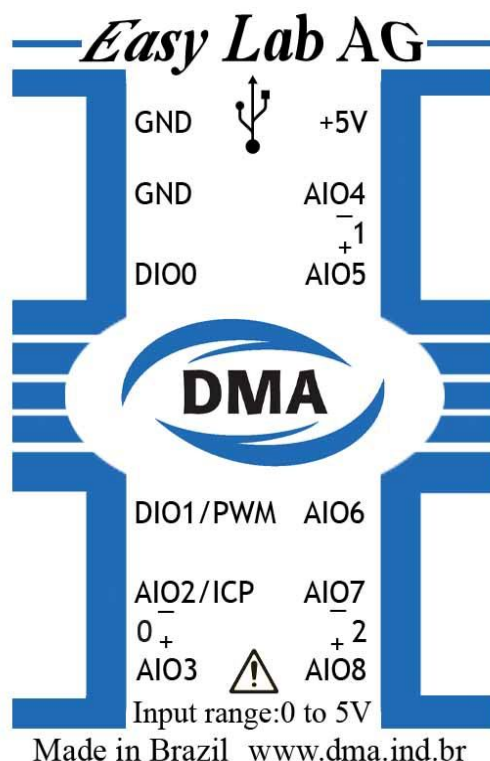
## 2. Descrição do Produto



**Nota:** Verifique o conteúdo do seu Kit:

- Interface com saída USB.
- CD de Instalação.
- 4 resistores de precisão 1%, 120Ω (somente versão AG, medição corrente)

### 2.1 Conexões



#### Atenção!



- Nunca conecte sinais fora de especificação, conforme seção 1.2;
  - Atenção para não curto circuitar os pinos GND com +5v, embora o Easy Lab não forneça proteção, o host, segundo especificação USB, possui fusíveis de proteção, que no caso de curto congelam ou reiniciam o PC.
- A não observância dos itens acima pode danificar irreversivelmente seu PC e/ou o módulo conectado.



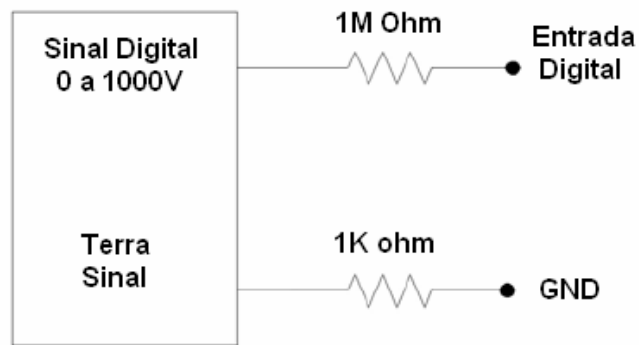
**Nota:** Por segurança, na ocorrência de curtos e/ou conexão de sinais fora da especificação o módulo cortará a comunicação com o computador a qual está conectado. Portanto se o módulo não responder, o usuário deve desconectar - lo imediatamente.

## 2.2 Conexão de sinais digitais

O Easy Lab trabalha com sinais TTL (0 a 5v) no entanto, sinais com amplitudes diferentes poderão ser conectados, desde que o usuário siga as instruções presentes nesta seção.

Todas as portas possuem diodos que fixam a tensão entre -0.5v a +5.5v, mas para tal é necessário o uso de resistor em série com o sinal para limitar a corrente em no máximo 1mA, caso a tensão de entrada não seja TTL.

O esquema abaixo pode ser utilizado com sinais de 5, 10, 12, 24V , etc. com proteção de até 1000V (pico, positivo ou negativo).

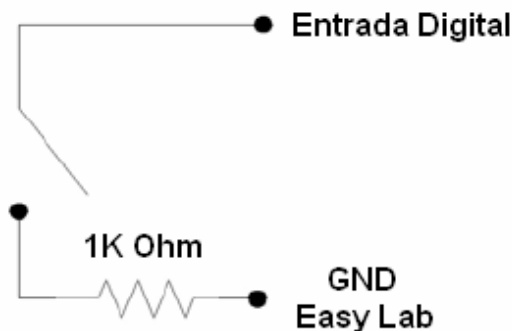


O resistor de  $1M\Omega$  em série com a entrada, limita a corrente a um valor seguro, caso houver uma sobre tensão de até 1000V.

O resistor de  $1K\Omega$  é usado como uma proteção contra curtos entre o terra e  $V_s$  (tensão positiva).

Apesar deste circuito fornecer uma proteção razoável ao Easy Lab, não protegerá contra surtos de tensão, alguns milhares de volts, em linhas da rede elétrica (geralmente devido a comutação de reles ou ligamento/desligamento de dispositivos na rede). Neste caso o usuário deve certificar que o sinal externo provenha de uma fonte de boa qualidade, com filtros ativos, de preferência uma fonte chaveada. Caso a origem e qualidade do sinal seja desconhecida, é recomendável o uso de acopladores ópticos, como por exemplo o acoplador: TIL-111.

O circuito abaixo descreve a conexão de sinal digital através de “contato seco”.



Nesta configuração, deverá ser acionado o “pull-up” da porta correspondente para que o valor da mesma não flutue quando o contato estiver aberto. O resistor de 1K $\Omega$  limita a corrente caso a porta for configurada erroneamente como uma saída.



**Nota:** Certifique-se que a porta esteja configurada como “ENTRADA”.

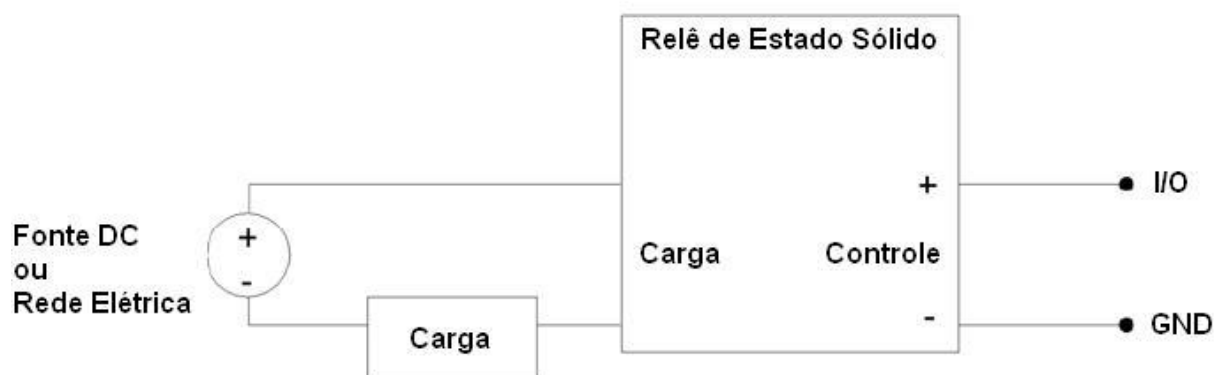


**Nota:** Os mesmos circuitos acima poderão ser utilizados para a entrada “ICP” (contagem e medição de frequência de pulsos), no entanto o sinal não deve ter frequência maior que 12Khz, pois poderá sobre carregar o processador do módulo e acarretar na perda de conexão. Alguns ruídos presentes nos sinais poderão ser de alta frequência, portanto é recomendado o uso de filtros. Geralmente um capacitor de 100nf em paralelo com o sinal é suficiente. Caso não utilizar a contagem de pulsos, desabilite esta função, pois mesmo sem conexão física na porta, os ruídos eletromagnéticos de alta frequência poderão sobre carregar o processador do módulo.

## 2.3 Acionando cargas de potência

O Easy Lab pode acionar cargas de potência como resistências, motores, etc. através de um relê. É recomendado o uso de um SSR (Relê de estado sólido) com isolamento óptica, pois o mesmo pode ser ligado diretamente a saída do Easy Lab.

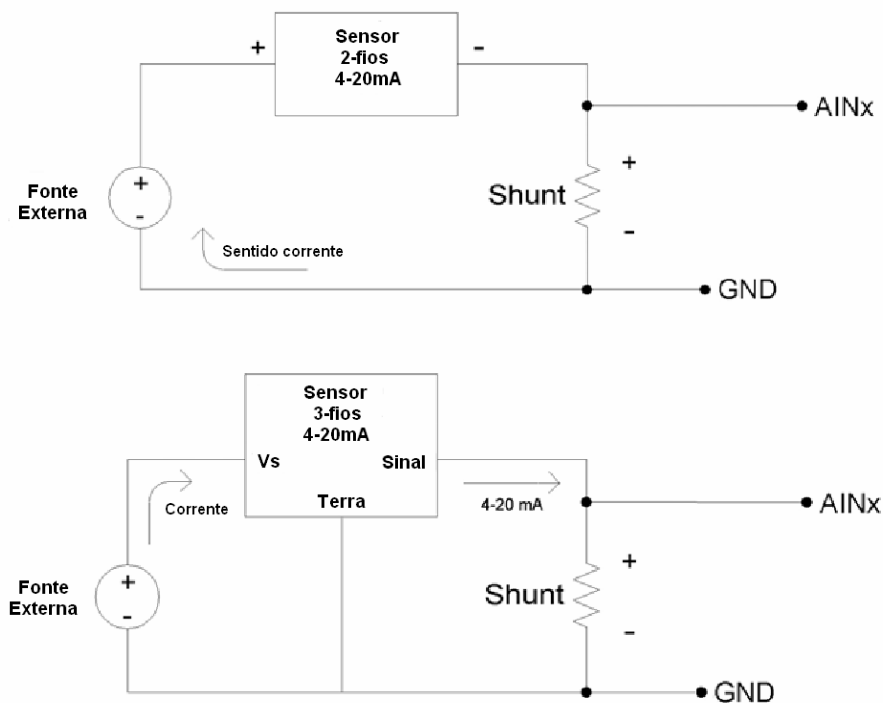
Esquema utilizando um relê de estado sólido:



**Nota:** Pela diversidade de relés no mercado, consulte o manual do fabricante. Verifique as especificações de controle conforme seção 1.2.

## 2.4 Conexão de Sensores analógicos com saída 4-20mA

Sensores e/ou transmissores com saída 4-20mA são muito disseminados devido a sua maior imunidade a ruídos, obtendo-se uma leitura mais precisa. No mercado existem dois tipos de sensores, nas configurações de 2 ou 3 fios. Abaixo segue esquema de conexão para o Easy Lab AG:



A saída do sensor deve ser conectada ao módulo utilizando-se um resistor para se obter uma tensão diretamente proporcional a corrente. Nos esquemas acima, caso fosse utilizado um resistor de  $120\Omega$  (resistor fornecido), a tensão de saída estaria entre 0.48v a 2.4v. É recomendado o uso de resistores de precisão (1% ou melhor) ou o usuário pode medir a resistência de um resistor de menor precisão, obtendo a corrente pela fórmula:

$$I_s = V_m / R_s$$

Onde:

- $I_s$ , corrente de saída do sensor
- $V_m$ , tensão medida pelo Easy Lab
- $R_s$ , resistência do Shunt

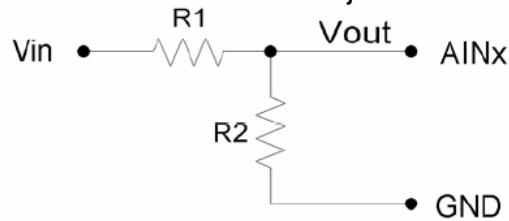


**Nota:** Antes conectar o sinal, verifique se a tensão está dentro dos limites do Easy Lab.



## 2.5 Conexão de Sinais analógicos com tensões maiores que 2.9 volts

Sinais com tensões acima da capacidade do Easy Lab podem ser conectados a partir de um simples divisor de tensão, para atenuação da tensão a ser lida. Abaixo segue circuito (deve se certificar que a tensão Vout não esteja acima 2.9 volts):



$$V_{out} = V_{in} * (R2 / (R1 + R2))$$



**Nota:** Ao conectar sinais analógicos, é recomendável a utilização de um resistor de 100R entre o terra do Easy Lab e o terra do sinal, para proteção.

## 3. Utilização do Produto

### 3.1 Instalação dos Drivers

Primeiro conecte o Easy Lab em uma das portas USB disponíveis. O Windows automaticamente irá pedir os drivers do dispositivo.



Escolha as opções: “Não, não agora” → “Instalar de um local específico” → “Não pesquisar. Escolherei o driver”. Na caixa de seleção localize a pasta “Driver 2.0” no CD de instalação. Caso mensagens apareçam, clique em “OK” ou “Continuar”.



**Nota:** Escolha a pasta 32 ou 64 bits conforme o sistema operacional. No Windows 64 bits as aplicações que acessam o módulo devem ser 32bits, inclusive o Excel.

## 3.2 Softwares

O módulo acompanha duas opções de software para comunicação e configuração. Um arquivo programa executável (desenvolvido em VB6) e uma planilha em Excel com macros, ambos em código aberto. Para instalação execute e siga os passos no arquivo “setup” presente no CD de instalação.

## 3.3 Programa Executável “Easy Lab”

Na inicialização do programa, o mesmo “lembra” das ultimas configurações e automaticamente configura o módulo.

Para configurar clique em “Configurar”, nesta tela se faz toda a configuração de todas as I/Os do módulo:

Na porta “DIO1” a opção PWM habilita a saída PWM nesta porta, é possível escolher também a frequência e Duty Cycle, de 0 a 1023 (100%), do PWM.

Na porta “DIO2” a opção “ICP” habilita o frequencímetro e contador de pulsos. Não há necessidade de marcar a opção “High” ao lado, esta opção é necessário para um outra versão do módulo.

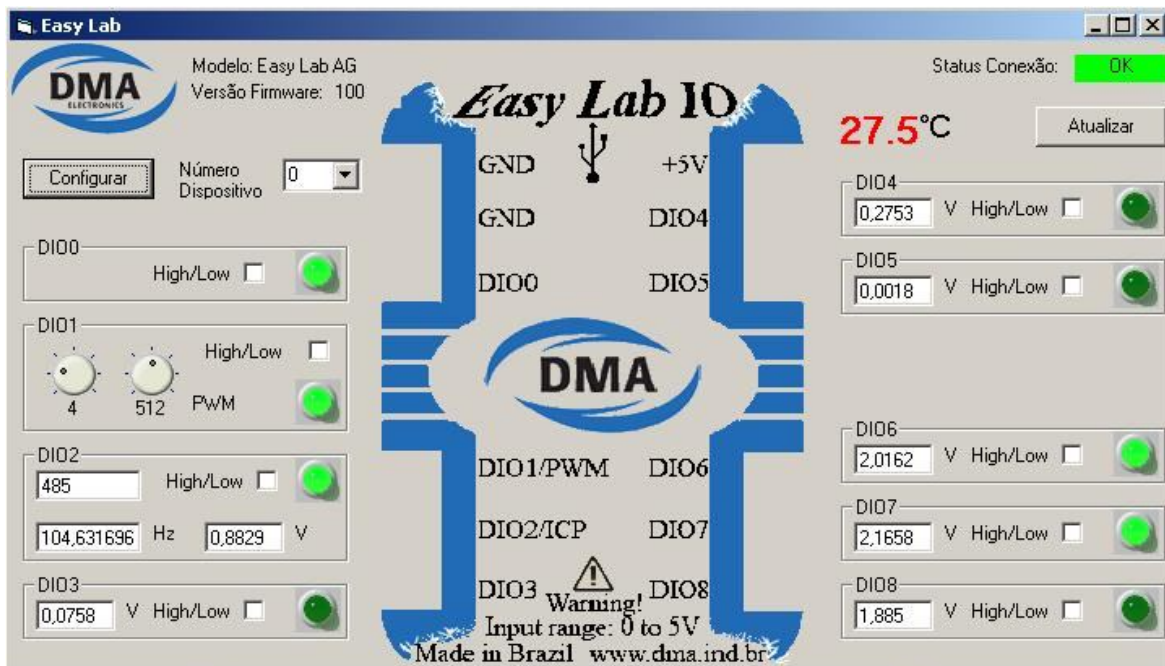
As opções “output” configuram as I/Os como “saída”. Caso uma porta seja configurada como “entrada” e seu estado colocado em “alto/high”, será acionado o “Pull-up” (resistor de 50Kohms ligado a 5V) correspondente da porta. A opção “Resolução ADC”, determina a resolução A/D de todas as portas.

Para salvar as configurações no módulo clique em “Salvar Configurações”.



**Nota:** O módulo mantém as configurações desde que não seja desconectado da porta USB (é permitido o reiniciamento do PC). Caso o PC seja desligado ou o módulo retirado, deverá ser gravada novamente a última configuração (feita automaticamente pelo programa, quando aberto).

Após a configuração do módulo, é possível ligar, desligar, ajustar PWM e ler a frequência e contagem dos pulsos, através da tela principal do programa.



### 3.4 Planilha Excel

Antes de abrir a planilha com macros, certifique-se de que a opção de executar macros em seu Excel esteja habilitada. Para habilitar acesse no menu do Excel: *Ferramentas* → *opções* → *segurança* → *segurança de macro*. Selecione o nível médio de segurança.

Duas versões de planilha estão disponíveis, dependendo da versão do Office instalada. A planilha “Easy\_lab\_daq\_vx.xO07” deve ser utilizada em versão do Office acima de 2003.

A arquivo Excel possui 4 planilhas: Home, Config, Logger, Data. Na planilha “Config” é possível configurar e enviar e ler todas as funções do módulo. Também é possível utilizar

formulas e gráficos conforme a aplicação. O botão “Configurar” é exatamente igual ao do programa executável. O botão “Run” envia e lê dados constantemente do módulo, e o botão “Stop” finaliza a comunicação com o módulo. As caixas de opção ao lado das portas acionam ou desligam a porta respectiva.

É possível utilizar apenas formulas em Excel para realizar operações lógicas entre as portas do módulo, sem que o usuário tenha conhecimento em programação. Os estados das portas estão disponíveis na coluna “E” e “J”. Inserindo um “x” nas colunas “D” e “K” acionam a respectiva porta.

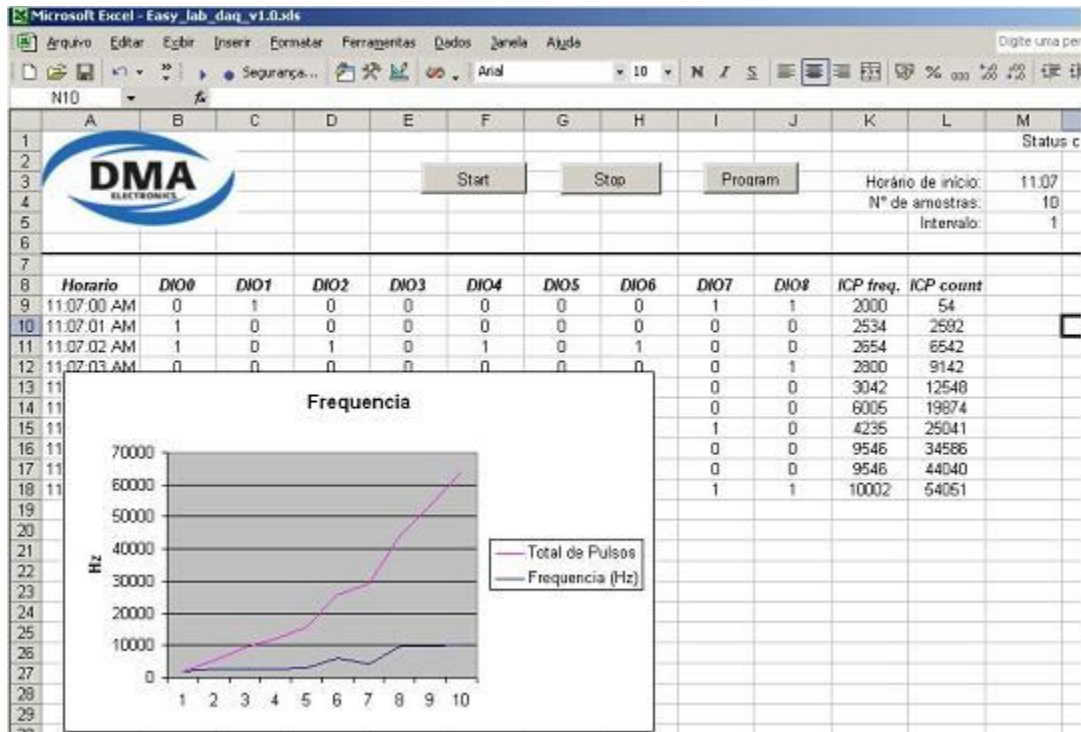
No exemplo da foto abaixo, é utilizado uma lógica “E” entre as portas DIO1 e DIO2 para acionar a porta DIO0. Neste caso poderia se utilizar dois botões e um LED ligados as portas do módulo, para a verificação do funcionamento da lógica.

Ex: a célula selecionada abaixo possui a seguinte formula: “=SE(E12\*E13=1;"x")” , caso E12 e E13 (portas DIO1/2) estejam em “alto” acione a porta DIO1 (através do x).



**Nota:** Para escrever as formulas o botão “Stop” deve ser acionado. Para rodar a lógica o botão “Run” deve ser acionado.

Na planilha “Logger” é possível programar um horário para início da captura dos sinais do módulo. Para programar, escolha o horário no formato 24h, o n° de amostrar e intervalo entre amostras (em segundos) depois clique em “Programar”. O botão “Start” inicia imediatamente o logger e “Stop” finaliza. Pode se criar gráficos específicos à aplicação.



## 4. SDK – Software development Kit

Qualquer plataforma de desenvolvimento Windows pode se comunicar com o Easy Lab através da biblioteca DLL (Ex: VB, Excel, Delphi, MathLab, etc.). Toda a comunicação de “baixo nível”, como o protocolo USB e funcionamento interno do módulo é tratado pela DLL, simplificando a comunicação. A DMA fornece exemplos e códigos fonte para VB6, Delphi, Excel-VBA e Labview.

### 4.1 VB6

Este exemplo foi desenvolvido em Visual Basic 6, realiza todas as configurações e leitura do módulo. Está localizado em “Setup files\1.4\vb”

### 4.2 Excel

Através da DLL o Excel acessa o módulo, a programação é bem semelhante ao do Visual Basic. O pacote office instalado deve ser 32 bits. Para acessar o código, com a planilha aberta, clique “ALT+F11”.

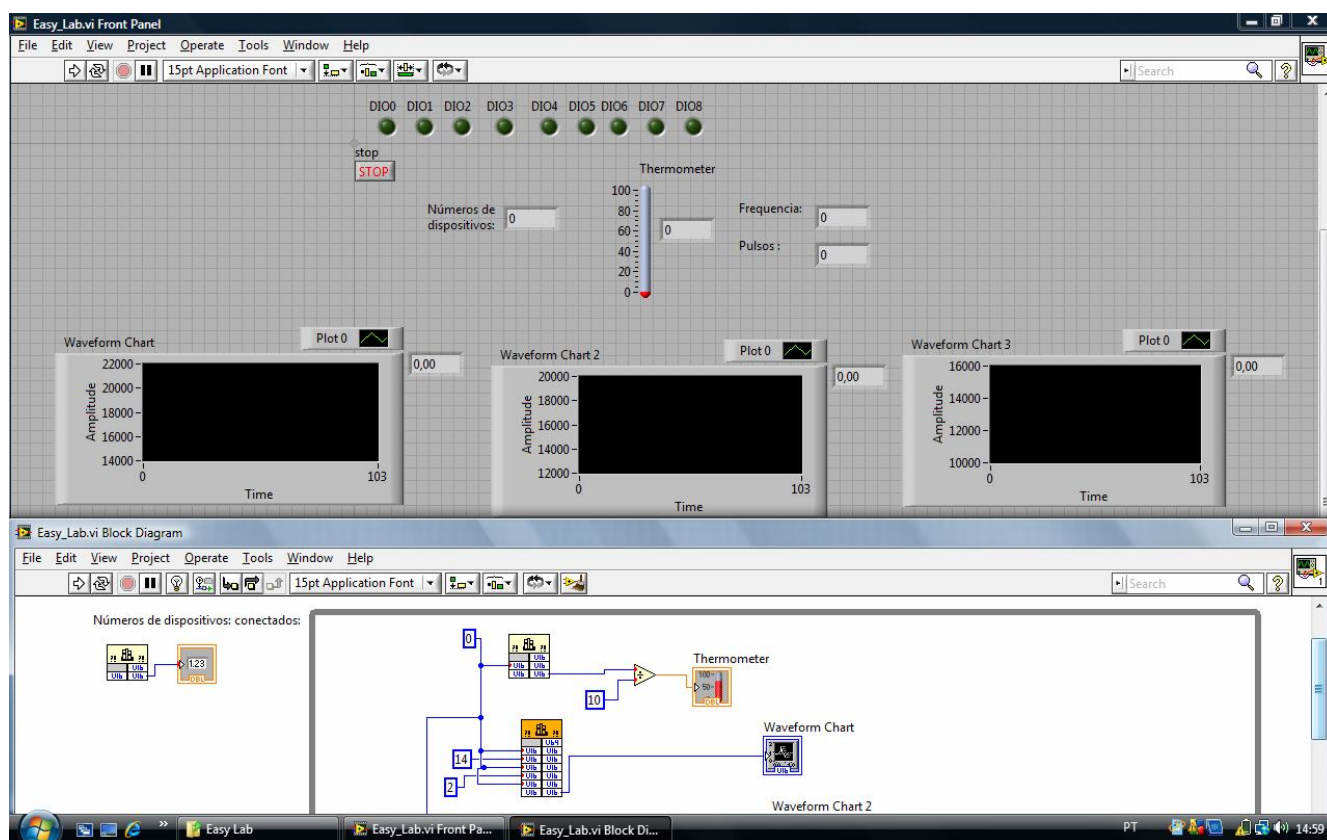
### 4.3 Delphi

Os exemplos do Delphi contemplam uma biblioteca PAS para a comunicação com a DLL (contribuição de Marcelo Maurin).



## 4.4 Labview

Dois exemplos em Labview estão disponíveis. Um exemplo realiza a leitura das variáveis e o segundo, aciona as portas (o usuário deve atentar que a configuração será mantida mesmo após fechar o Labview). Para a comunicação é utilizado o bloco “CALL LIBRARY FUNCTION NODE” onde o mesmo se comunica com a DLL do Easy Lab.



## 4.5 Detalhamento das funções da DLL.

Funções disponíveis:

- SearchDevices: Enumera e retorna o numero de dispositivos conectados.
- GetInPorts: Retorna o estado real (alto / baixo) das portas;
- GetOutPorts: Retorna o estado enviado (alto / baixo) das portas;
- SetPWM: Aciona / Desliga PWM com parâmetros de frequência e DutyCycle;
- SetPortDirections: Configura cada porta como entrada ou saída;
- GetPortDirections: Lê configuração da direção das portas;
- SetOutputPorts: Aciona os estados Alto ou Baixo de cada porta;
- Icen: Habilita frequencímetro e contador de pulsos;
- DoGetIcp: Retorna valor de frequência e pulsos.
- ReadAdc: Retorna valor em Volts do conversor A/D (somente versão AG);
- ReadAllAdc: Retorna todos os valores das 7 entradas analógicas (somente versão AG)
- GetTemp: Retorna o valor da temperatura ambiente em °C (somente versão AG)

Para as funções que configuram / alteram dados das portas, sempre é utilizado um parâmetro / variável / número inteiro de 0 a 511, que representa as 9 portas. Ex. Desligar todas as portas: "SetOutputPorts(0)", 0 decimal é igual ao binário: 000000000 (DIO9 a DIO0). Para ligar somente a porta DIO1: "SetOutputPorts(2)", 2 decimal é igual a: 000000010. No início da aplicação ou na conexão ou desconexão de qualquer dispositivo a função "SearchDevices" deve ser chamada para a correta enumeração e identificação dos módulos.

O parâmetro "dev" das funções abaixo representa o número do dispositivo conectado. Por ex: Para habilitar o PWM no módulo nº 0 : "SetPWM(0,15,511)"

As funções e parâmetros utilizadas por elas devem ser declaradas na plataforma utilizada. Abaixo segue como declarar as funções e seus parâmetros (exemplo de declaração em VB):

### **1. *Public Declare Function SearchDevices Lib "Easy\_lab.dll" (ByRef ndevs as byte) As Integer***

- Função: SearchDevices(ndevs)
- Ndevs: Total de dispositivos conectados

### **2. *Public Declare Function SetPWM Lib "Easy\_lab.dll" (ByVal dev As Byte, ByVal Divisor As Byte, ByVal Duty As Long) As Integer***

- Função: SetPWM(dev,Divisor,Duty)
- Divisor: valor de 0 a 15, divide a frequência máxima do PWM.  $F_{pwm} = 11718 / \text{divisor}$ , divisor = 0 desabilita PWM.
- Duty, valor de 0 a 1023, representando o duty cycle 0 a 100%.

**3. Public Declare Function SetPortDirections Lib "Easy\_lab.dll" (ByVal dev As Byte, ByVal ddr As Long) As Integer**

- Função: SetPortDirections(dev,ddr);
- Ddr: valor de 0 a 511, que representa a direção das 9 portas. Ex: se ddr = 511, todas as portas estão como saída.

**4. Public Declare Function GetPortDirections Lib "Easy\_lab.dll" (ByVal dev As Byte, ByRef DDRin As Integer) As Integer**

- Função: GetPortDirections(dev,ddrin)
- Função retorna valor de 0 a 511 na variável ddrin. Le a configuração realizada pela função "SetPortDirections".

**5. Public Declare Function GetInPorts Lib "Easy\_lab.dll" (ByVal dev As Byte, ByRef DataIn As Integer) As Integer**

- Função: GetInPorts(dev,Datain);
- Função retorna um valor de 0 a 511 na variável Datain representando o estado real das portas. Ex: Todas as portas em alto, Datain = 511.

**6. Public Declare Function GetOutPorts Lib "Easy\_lab.dll" (ByVal dev As Byte, ByRef DataOut As Integer) As Integer**

- Função: GetOutPorts(dev,Dataout)
- Função retorna um valor de 0 a 511 na variável Dataout a configuração realizada pelo função "SetOutputPorts"

**7. Public Declare Function SetOutputPorts Lib "Easy\_lab.dll" (ByVal dev As Byte, ByVal DataOut As Long) As Integer**

- Função: SetOutputPorts(dev,Dataout)
- Dataout: valor entre 0 a 511 que ativa ou não os estados das portas. Ex. Dataout = 511, todas as portas ativarão em "Alto".

**8. Public Declare Function Icen Lib "Easy\_lab.dll" (ByVal dev As Byte, ByVal fr As Long) As Long**

- Função: Icen(dev,fr)
- Fr: valor entre 0 e 2, Icen=2 desabilita ICP (frequencimetro), Icen=0 ativa ICP, Icen=1 (não utilizar).

**9. Public Declare Function DoGetIcp Lib "Easy\_lab.dll" (ByVal dev As Byte, (ByRef Freq As Long, ByRef counter As Long) As Long**

- Função: DogetIcp(dev,freq,counter)
- A função retorna em "freq" a frequência dos pulsos e em "counter" a quantidade de pulsos medidos. Para se ter o valor da frequência em hz, o valor armazenado em "freq" deve ser transformado conforme:  $F(hz) = 1500000/freq$ .

Obs: Para utilizar esta função o ICP deve estar habilitado (chamar função: "Icen(dev,0)).



**10. Public Declare Function ReadAdc Lib "Easy\_lab.dll" (ByVal dev As Byte, ByVal Res As Byte, ByVal Modo As Byte, ByVal Porta As Byte, ByVal Ganho As Byte, ByRef Adc As Long) As Long**

- Função: ReadAdc(dev,Res,Modo,Porta,Ganho,Adc)
- Res: 10, 12 ou 14. Define a resolução A/D.
- Modo igual a 0 e Ganho sempre igual a 1 (versão futura: AG +)
- Porta: número da porta A/D, valores: 2 a 8.
- Adc: A função retorna no parâmetro o valor da conversão A/D em volts\*10000.

**11. Public Declare Function ReadAllAdc Lib "Easy\_lab.dll" (ByVal dev As Byte, ByRef Adc As Variant) As Long**

- Função: ReadAllAdc(dev,Adc)
- Retorna em Adc (variável vetor de tamanho 7) os valores das 7 entradas analógicas. A conversão é feita em 12 bits.

**12. Public Declare Function GetTemp Lib "Easy\_lab.dll" (ByVal dev As Byte, ByRef Temp As Integer) As Long**

- Função: GetTemp(dev,Temp)
- Retorna em "Temp" o valor da temperatura ambiente em décimos de °C.

Deve se atentar para declarar as variáveis utilizadas pelas funções no mesmo formato (ou equivalente) como apresentado acima. Caso encontre problemas declare a variável como "Variant" (comum em todas as linguagens de programação).

Todas as funções retornam "0" quando realizada com sucesso, mas caso retorne um número diferente de "0" a função encontrou um erro, ou o parâmetro repassado está fora de especificação ou o módulo não foi encontrado ou ainda ocorreu um curto ou conexão de sinal fora da especificação.

Exemplos de declaração e outras linguagens para a função "SetPortsDirection":

**Delphi:**

**Function SetPortsDirection(DDR:byte):integer; stdcall external Easy\_Lab.dll name 'SetPortsDirection';**

**C++:**

**int \_\_stdcall SetPortsDirections(uchar DDR);**

## **Quadro de revisões:**

- 07/07/2009 – v1.0 Emissão manual
- 16/09/2009 – V1.1 Revisão Geral conforme novo release Drivers 2.0
- 01/10/2009 – V1.2 Inclusão especificações versão AG
- 10/10/2009 – V1.3 Sessões 2.2 e 2.3
- 19/11/2009 – V1.4 Revisão geral com nova função de temperatura
- 25/01/2010 – V1.5 Inclusão seções 2.2, e 2.3
- 25/01/2010 – V1.5 Inclusão seções 2.2, e 2.3
- 15/03/2010 – V1.6 Nota sobre contagem/frequência de pulsos
- 17/12/2010 – V1.7 SDK