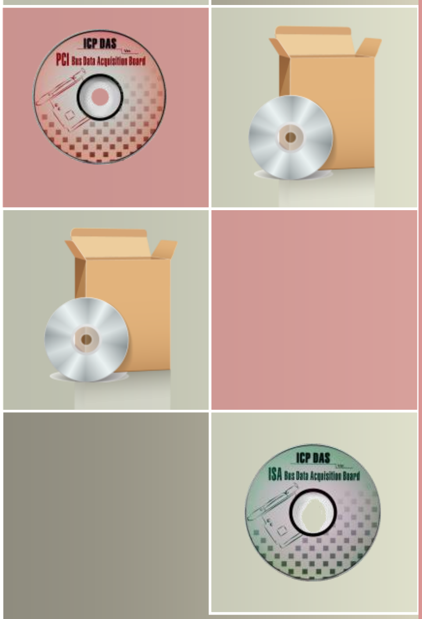
PCI-M512 Series Classic Driver DLL Software Manual

Version 1.0, Sep. 2015


ICP DAS
PCI Bus Data Acquisition Board
ICP DAS
ISA Bus Data Acquisition Board

SUPPORTS

Board includes PCI-M512 and PCI-M512U.

WARRANTY

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

WARNING

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

COPYRIGHT

TRADEMARKS

Names are used for identification only and may be registered trademarks of their respective companies.

CONTACT US

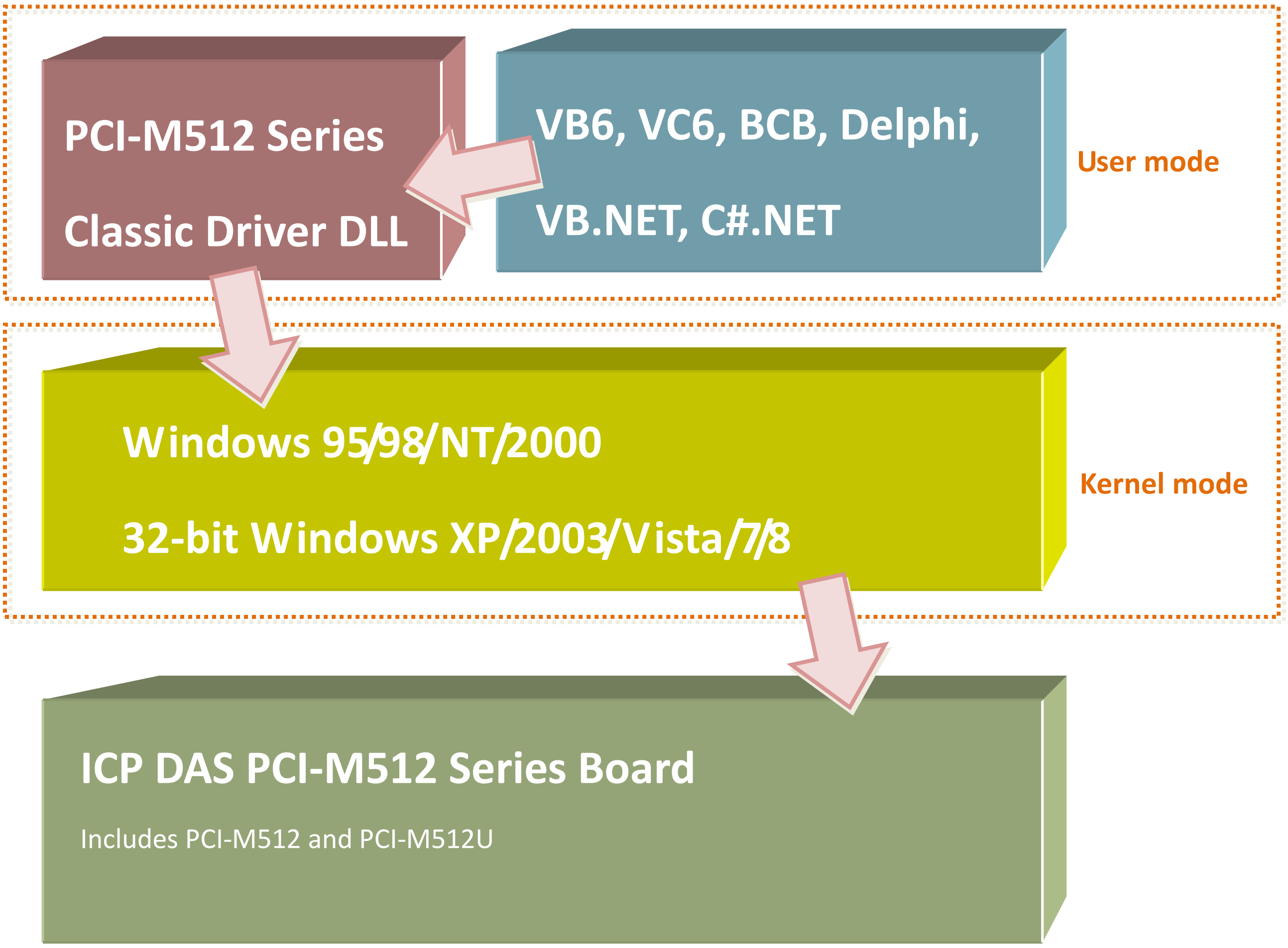
If you have any question, feel to contact us by email at:
Email: service@icpdas.com or service.icpdas@gmail.com
We will respond to you within 2 working days.

TABLE OF CONTENTS

1. Introduction

The included DLL driver software contains a collection of functions specifically designed for PCI-M512/M512U cards running applications in a Windows 98/NT/2000 and 32-bit Windows XP/2003/2008/7/8 environment. These functions are written using the C language, and can used to perform a variety of Digital I/O operations.

The functions contained in the PCIM512.DLL are easy to understand and provide a powerful, user-friendly approach to developing custom data acquisition applications. These DLL functions can easily by called from within an application using **VC++, VB, Delphi, BORLAND C++ Builder, etc**. The source code for a range of demonstration programs is also provided to enable application developers to accelerate their development process.



For more details, refer to the following user manuals, which can be found in the NAPDOS\PCI\Manual folder on the companion CD.

- PCI_ISA_PnP_Driver_Installation_in_Win9x_2K_XP.pdf

This document describes the process for installing the Plug and Play driver for the PCI card operating in a Windows 95/98.

- Software_Installation_Guide_in_Win32.pdf

This document describes the process for installing the software package in a Windows 95/98/NT/XP environment.

- Calling_DLL_functions_in_VB_VC_Delphi_BCB.pdf

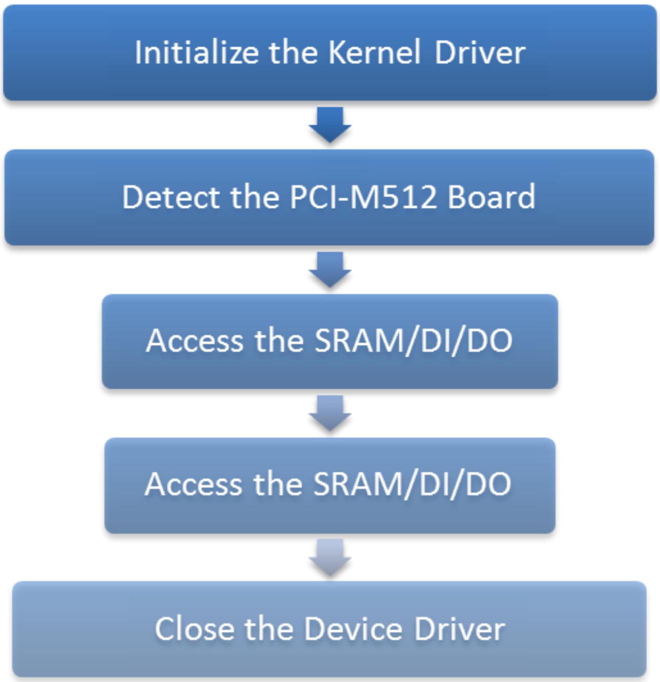
This document describes the processes required for calling the DLL functions using VC++6, VB6, Delphi3 and Borland C++ Builder 3.

- TroubleShooting_PCI_ISA_in_Win32_Resource_Conflict.pdf

This document provides a guide to checking resolving resource conflicts associated with the I/O Port address, IRQ number and DMA number for add-on cards operating in a Windows 95/98/NT environment.

1.1 Program Architecture

The following diagram is an illustration of the program architecture for PCI-M512/M512U.



PCIM512_DriverInit()

PCIM512_DetectCards()

………………

PCIM512_ReadSramDword(…)

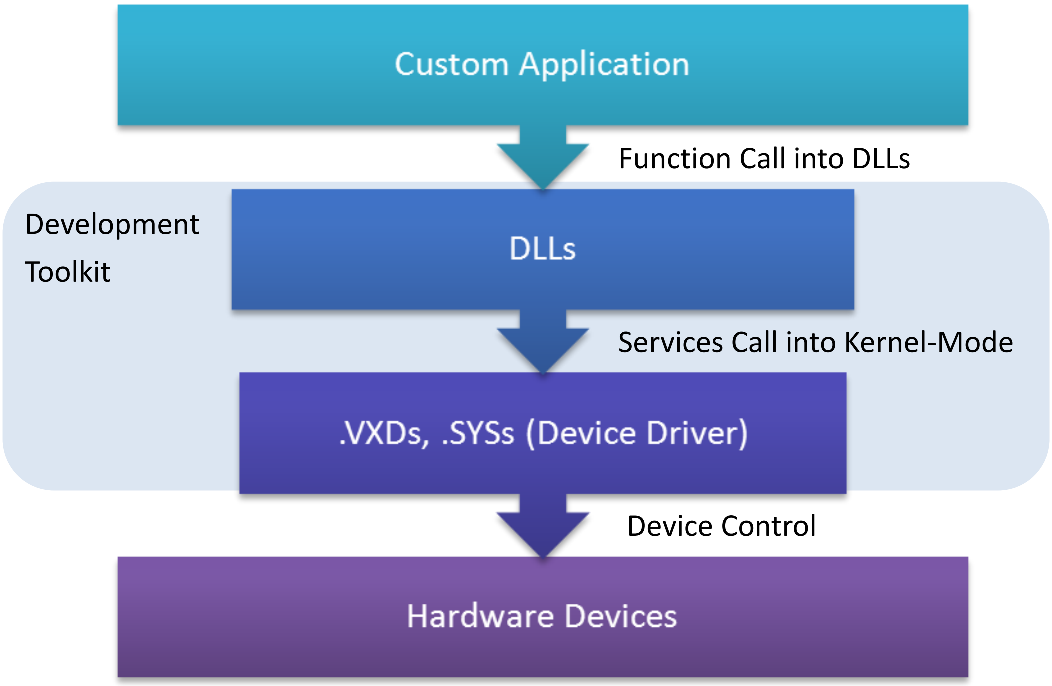
………………

………………

PCIM512_WriteSramDword(…)

……………..

PCIM512_DriverClose()



2. Software Installation

2.1 Obtaining the Driver Installer Package

PCI-M512 series card can be used on Linux and Windows 98/NT/2000 and 32-bit XP/2003/Vista/7/8 based systems, and the drivers are fully Plug and Play (PnP) compliant for easy installation.

The driver installer package for the PCI-M512 series can be found on the supplied CD-ROM, or can be obtained from the ICP DAS FTP web site. The location and addresses are indicated in the table below:

	CD:\\NAPDOS\PCI\PCI-M512\DLL\Driver\
FTP	ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-m512/dll/driver/
	http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-m512/dll/driver/

Install the appropriate driver for your operating system, as follows:

Name	OS
PCI-Memory_Win_Setup_xxx.exe	For Windows 95, Windows 98, Windows NT, Windows 2000, 32-bit Windows XP, 32-bit Windows 2003, 32-bit Windows Vista, 32-bit Windows 7 and 32-bit Windows 8 .

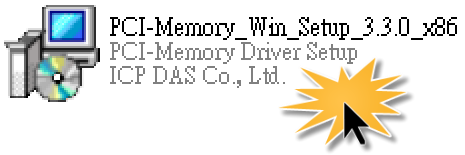
2.2 Driver Installing Procedure

Before the driver installation, you must complete the hardware installation. For detailed information about the hardware installation, please refer to appropriate hardware user manual for your PCI-M512 series card. The hardware user manual is contained in:

CD:\NAPDOS\PCI\PCI-M512 \Manual\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-m512/manual/

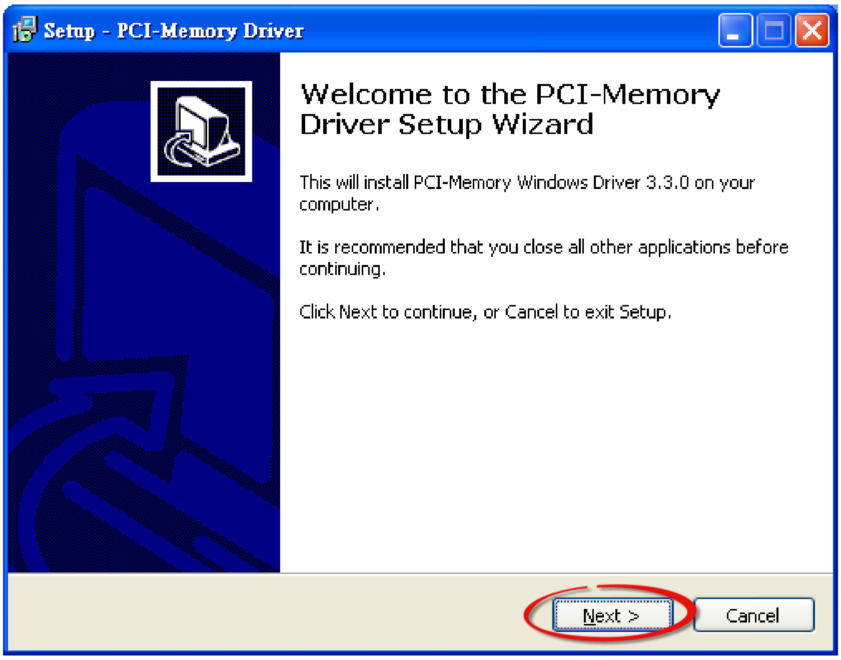
To install the PCI-M512 series classic drivers, follow the procedure described below:



PCI-Memory_Win_Setup_3.3.0_x86
PCI-Memory Driver Setup
ICP DAS Co., Ltd.

Step 1: Double-Click **"PCI-Memroy_Win_Setup_xxxx.exe"** to install driver.

Step 2: Click the **"Next>"** button to start the installation on the **"Setup – PCI-Memory Driver"** window.



Setup - PCI-Memory Driver

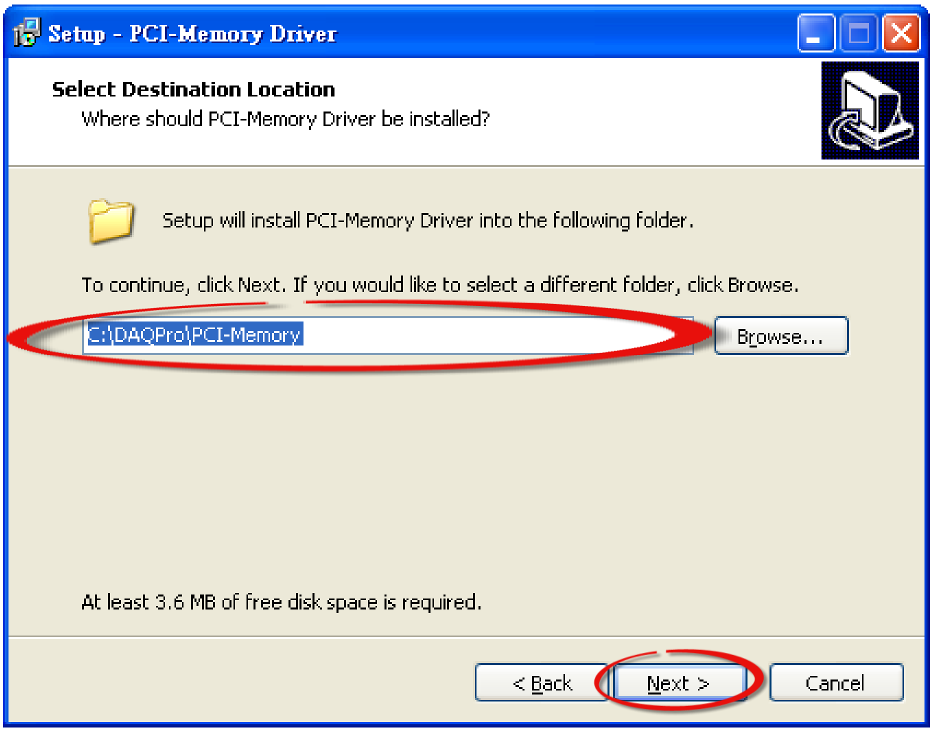
Welcome to the PCI-Memory Driver Setup Wizard

This will install PCI-Memory Windows Driver 3.3.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Step 3: Click the **"Next>"** button to install the driver into the **default** folder.



Setup - PCI-Memory Driver

Select Destination Location
Where should PCI-Memory Driver be installed?

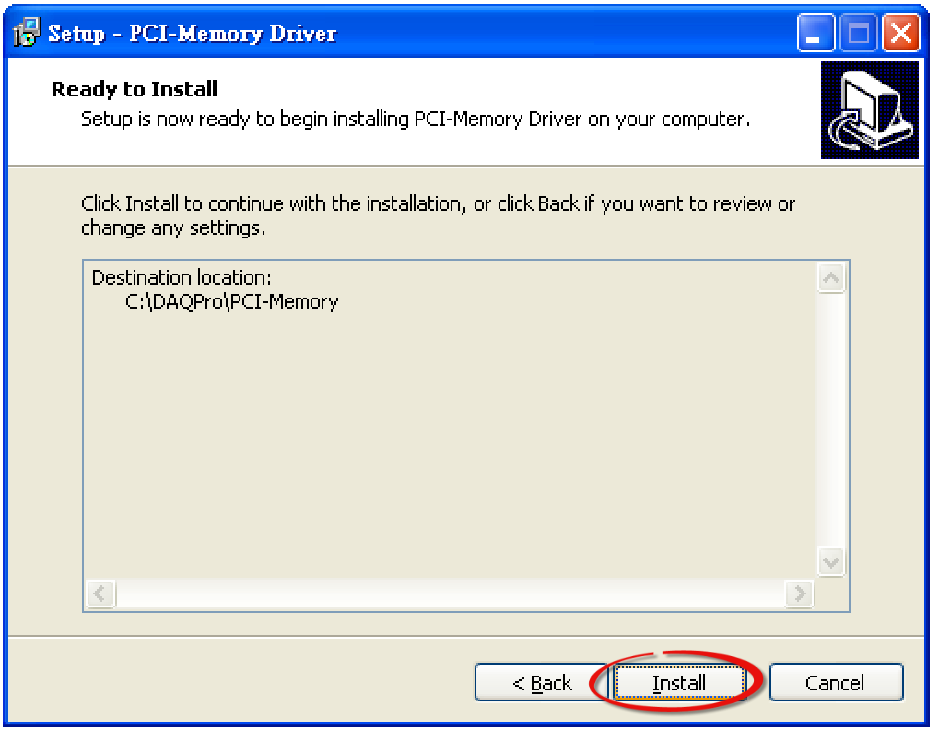
Setup will install PCI-Memory Driver into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\DAQPro\PCI-Memory

At least 3.6 MB of free disk space is required.

Step 4: Click the **"Install"** button to continue.



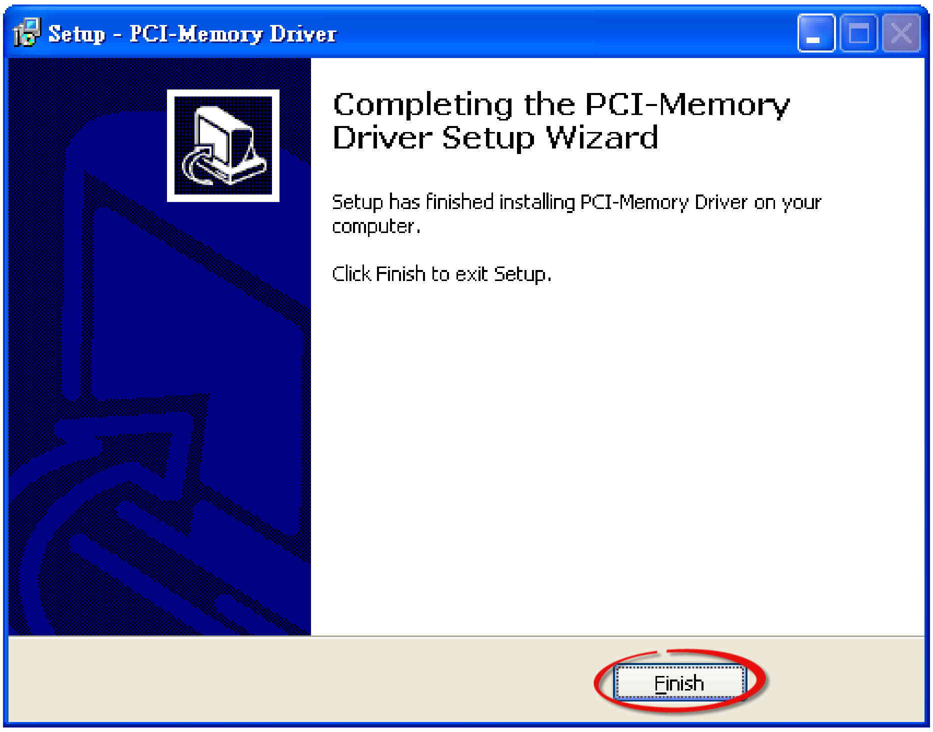
Setup - PCI-Memory Driver

Ready to Install
Setup is now ready to begin installing PCI-Memory Driver on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\DAQPro\PCI-Memory

Step 5: Click the **"Finish"** button.



Completing the PCI-Memory Driver Setup Wizard

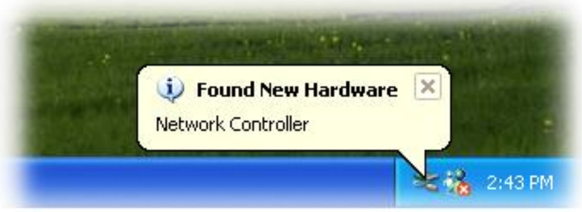
Setup has finished installing PCI-Memory Driver on your computer.

Click Finish to exit Setup.

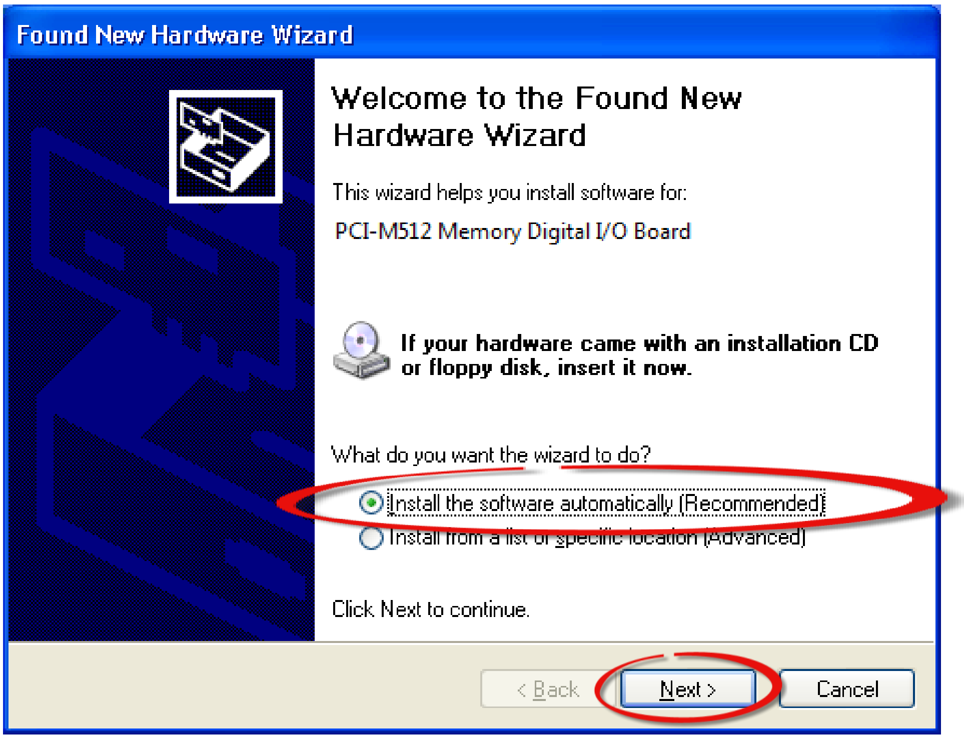
2.3 PnP Driver Installation

Step 1: The system should find the new card and then continue to finish the Plug&Play steps.

Note: More recent operating systems, such as Windows 7/8 will automatically detect the new hardware and install the necessary drivers etc., so Steps 2 to 4 can be skipped.


Found New Hardware
Network Controller

Step 2: Select **"Install the software automatically [Recommended]"** and click the **"Next>"** button.


Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

PCI-M512 Memory Digital I/O Board

If your hardware came with an installation CD or floppy disk, insert it now.

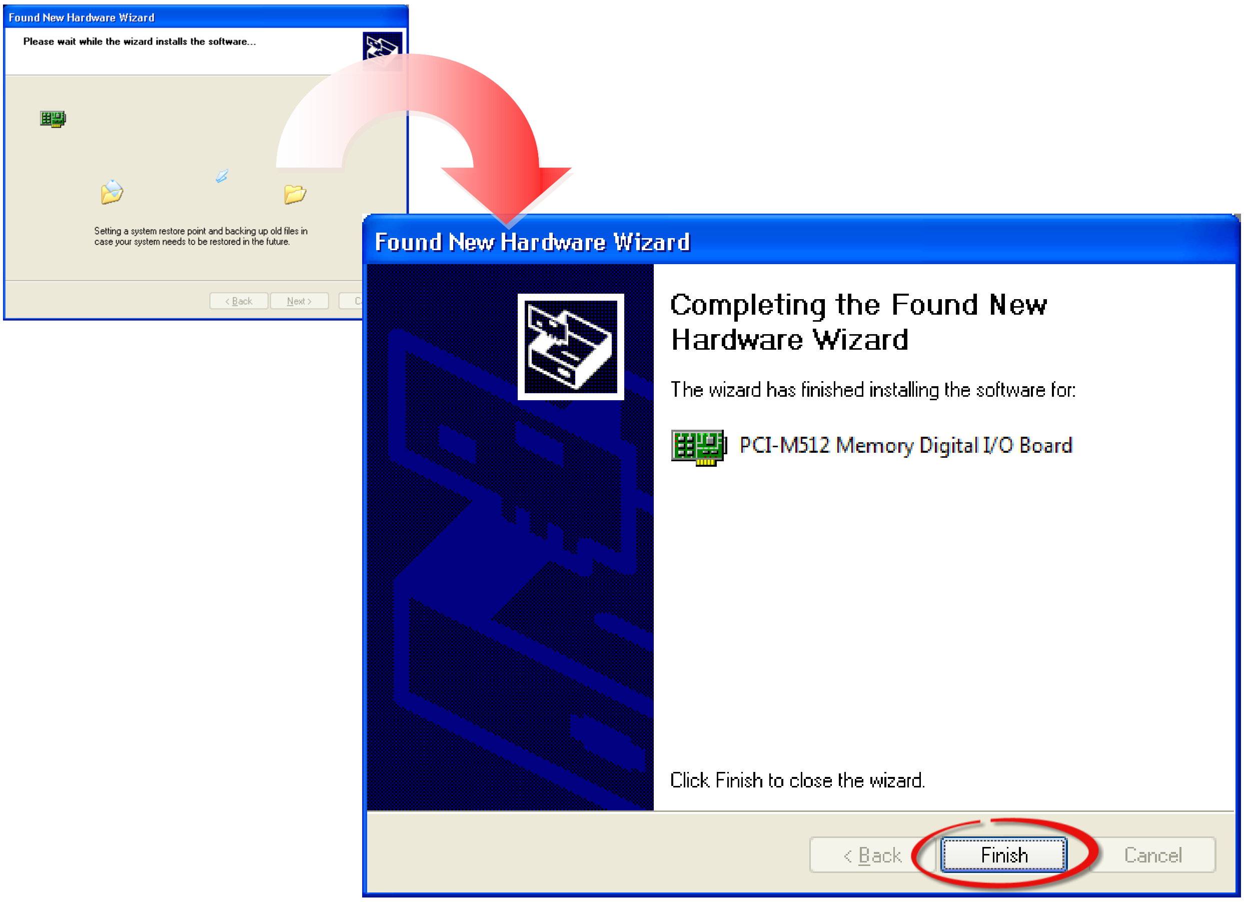
What do you want the wizard to do?

● Install the software automatically (Recommended)

○ Install from a list or specific location (Advanced)

Click Next to continue.

Step 3: Click the **"Finish"** button.



Please wait while the wizard installs the software...

Setting a system restore point and backing up old files in case your system needs to be restored in the future.

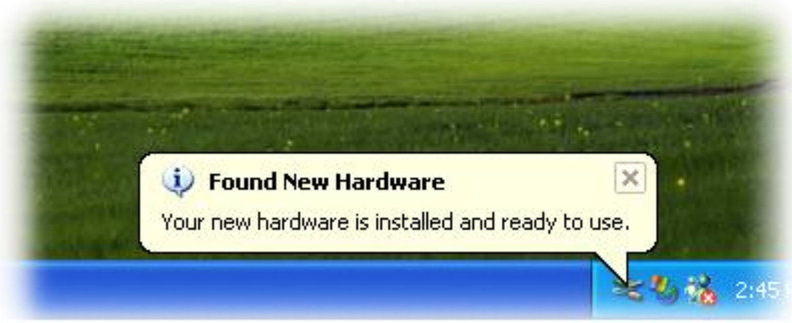
Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

PCI-M512 Memory Digital I/O Board

Click Finish to close the wizard.

Step 4: Windows pops up **"Found New Hardware"** dialog box again.



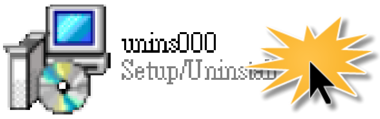
Found New Hardware

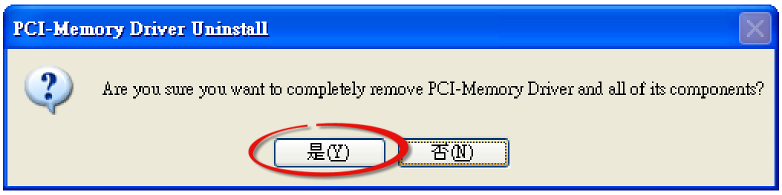
Your new hardware is installed and ready to use.

2.4 Uninstalling the PCI-M512 Series Classic Driver

The ICP DAS PCI-M512 series classic driver includes an uninstallation utility that allows you remove the software from your computer. To uninstall the software, follow the procedure described below:

Step 1: Double clock the **unins000.exe** uninstaller application, which can be found in the following folder: **C:\DAQPro\PCI-Memory**.

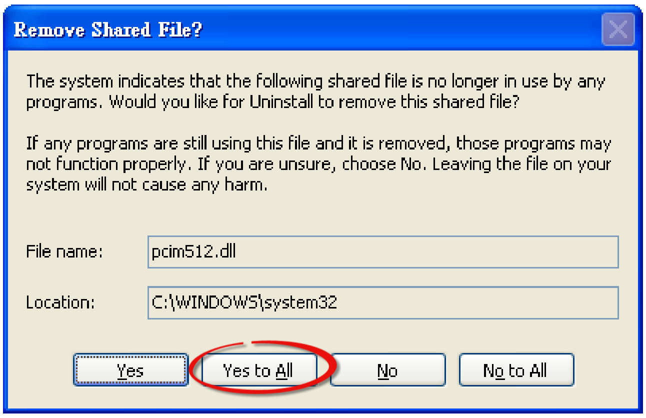



PCI-Memory Driver Uninstall

Are you sure you want to completely remove PCI-Memory Driver and all of its components?

Step 2: A dialog box will be displayed asking you to confirm that you want to remove the utility program. Click the “**Yes**” button to continue.

Step 3: The **“Remove Shared File?”** dialog box will then be displayed to confirm whether you want to remove the share files. Click the **“Yes to All”** button to continue.

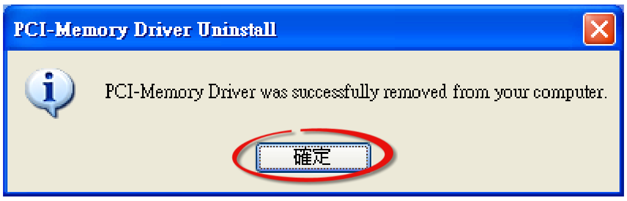

Remove Shared File?

The system indicates that the following shared file is no longer in use by any programs. Would you like for Uninstall to remove this shared file?

If any programs are still using this file and it is removed, those programs may not function properly. If you are unsure, choose No. Leaving the file on your system will not cause any harm.

File name: pcim512.dll

Location: C:\WINDOWS\system32


PCI-Memory Driver Uninstall

PCI-Memory Driver was successfully removed from your computer.

Step 4: After the uninstallation process is complete, a dialog box will be displayed to you that the driver was successfully removed. Click the **“OK”** button to finish the uninstallation process.

3. DLL Function Descriptions

This list of functions is expanded on in the text that follows. However, in order to make a clear and simplified description of the functions, the attributes of the input and output parameters for every function is indicated as [input] and [output] respectively, as shown in following table. Furthermore, the error code of all functions supported by PCI-M512 series card is also listed in Section 3.1.

Keyword	The parameter must be set by the user **before** calling the function	The data/value returned by the parameter **after** calling the function
[Input]	**Yes**	No
[Output]	No	**Yes**
[Input, Output]	**Yes**	**Yes**

Note: The memory space required by the parameters must first be allocated by the application.

The following is an overview of the defined DLL functions:

- **Test Functions (Refer to Section 3.2 for more details)**

float	CALLBACK **PCIM512_FloatSub**(float fA, float fB);
short	CALLBACK **PCIM512_ShortSub**(short nA, short nB);
int	CALLBACK **PCIM512_IntSub**(int iA,int iB);
DWORD	CALLBACK **PCIM512_GetDllVersion**(void);

- **Driver Initialization Functions (Refer to Section 3.3 for more details)**

DWORD	CALLBACK **PCIM512_DriverInit**(void);
DWORD	CALLBACK **PCIM512_CloseBoard**(DWORD dwBoardNo);
DWORD	CALLBACK **PCIM512_DetectBoards**(void);
DWORD	CALLBACK **PCIM512_OpenBoard**(DWORD dwBoardNo, DWORD dwIntEnable);
DWORD	CALLBACK **PCIM512_ReadBoardStatus**(DWORD dwBoardNo);
DWORD	CALLBACK **PCIM512_CloseAll**(void);

➢ SRAM Read/Write Functions (Refer to Section 3.4 for more details)

DWORD CALLBACK **PCIM512_WriteSramByte**(DWORD dwBoardNo, DWORD dwOffset, BYTE Data);

DWORD CALLBACK **PCIM512_WriteSramWord**(DWORD dwBoardNo, DWORD dwOffset, WORD Data);

DWORD CALLBACK **PCIM512_WriteSramDword**(DWORD dwBoardNo, DWORD dwOffset, DWORD Data);

DWORD CALLBACK **PCIM512_ReadSramByte**(DWORD dwBoardNo, DWORD dwOffset, BYTE *Data);

DWORD CALLBACK **PCIM512_ReadSramWord**(DWORD dwBoardNo, DWORD dwOffset, WORD *Data);

DWORD CALLBACK **PCIM512_ReadSramDword**(DWORD dwBoardNo, DWORD dwOffset, DWORD *Data);

➢ DIO Read/Write Functions (Refer to Section 3.5 for more details)

DWORD CALLBACK **PCIM512_WriteToDo**(DWORD dwBoardNo, WORD Data);

DWORD CALLBACK **PCIM512_ReadFromDi**(DWORD dwBoardNo, WORD *Data);

3.1 Error Code Table

For the most errors, it is recommended to check:

1. Does the device driver installs successful?
2. Does the card have plugged?
3. Does the card conflicts with other device?
4. Close other applications to free the system resources.
5. Try to use another slot to plug the card.
6. Restart your system to try again.

Error Code	Error ID
0	PCI_NoError
1	PCI_DriverOpenError
2	PCI_DriverNoOpen
3	PCI_GetDriverVersionError
4	PCI_InstallIrqError
5	PCI_ClearIntCountError
6	PCI_GetIntCountError
7	PCI_RegisterApcError
8	PCI_RemoveIrqError
9	PCI_FindBoardError
10	PCI_ExceedBoardNumber
11	PCI_ResetError
12	PCI_IrqMaskError
13	PCI_ActiveModeError
14	PCI_GetActiveFlagError
15	PCI_ActiveFlagEndOfQueue
16	PCI_BoardNoIsZero
17	PCI_BoardNoExceedFindBoards

3.2 Test Functions

As part of the software installation provided for the PCI-M512/M512U, ICPDAS also includes a range of functions that can be used to test the functionality of the board. The following sections provide a detailed description of these functions.

PCIM512_FloatSub

This function is used to perform the subtraction **fA - fB** as a float data type, and is provided for DLL linkage testing purposes.

- **Syntax:**
  float **PCIM512_FloatSub**(float **fA**, float **fB**);

- **Parameters:**
  fA
  [Input] A 4 byte floating point value

  fB
  [Input] 4 byte floating point value

- **Returns:**
  The value of fA – fB

PCIM512_ShortSub

This function is used to perform the subtraction **nA - nB** as a short data type, and is provided for DLL linkage testing purposess.

- **Syntax:**

  short **PCIM512_ShortSub**(short **nA**, short **nB**);

- **Parameters:**

  nA

  [Input] A 2 byte short data type value

  nB

  [Input] A 2 byte short data type value

- **Returns:**

  The value of nA - nB

PCIM512_IntSub

This function is used to perform the subtraction iA - iB as an int data type, and is provided for DLL linkage testing purposes.

- **Syntax:**

  Int **PCIM512_IntSub**(int **iA**, int **iB**);

- **Parameters:**

  iA

  [Input] A 4 byte int data type value

  iB

  [Input] A 4 byte int data type value

- **Returns:**

  The value of iA - iB

PCIM512_GetDllVersion

This function is used to read the version number information for the PCIM512.DLL.

- **Syntax:**
  DWORD **PCIM512_GetDllVersion**(void);

- **Parameters:**
  This function does not require any parameters

- **Returns:**
  The version umber information for the PCI-M512.DLL
  For example, 102(in hexadecimal format) denotes Version 1.02

3.3 Driver Initialization Functions

PCIM512_DriverInit

This function is used to allocate resources for the Windows Driver, and must be called before using any of the DLL functions described in Sections 3.4 to 3.5.

- **Syntax:**
  DWORD **PCIM512_DriverInit**();

- **Parameters:**
  This function does not require any parameters

- **Returns:**
  PCI_NoError: OK. The command was successful.
  PCI_DriverOpenError: The Windows Driver kernel was not found.

PCIM512_OpenBoard

This function is used to initialize the PCI-M512 kernel driver on the PCI-M512 board and allocate resources for the device, and must be called before using other I/O functions.

- **Syntax:**
  void **PCIM512_OpenBoard**(DWORD **dwBoardNo**, DWORD **dwIntEnable**);

- **Parameters:**
  dwBoardNo
  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  dwIntEnable
  [Input] Enable or disable the interrupt on the PCI-M512 board.

dwIntEnable	Description
0	Disabled
1	Enabled

- **Returns:**

  PCI_NoError: OK. The command was successful.

  PCI_DriverOpenError: An error occurred when attempting to initialize the kernel driver for the Board.

  PCI_BoardNoExceedFindBoards: The Board could not be found.

PCIM512_DetectBoards

This subroutine will detect all installed PCI-M512/M512U boards. This function must be called before using other I/O functions given in Sections 3.4 to 3.5.

- **Syntax:**

  void **PCIM512_DetectBoards**();

- **Parameters:**

  This function does not require any parameters

- **Returns:**

  0: There are no PCI-M512 boards is installed in this system

  1: There is only one PCI-M512/M512U board installed in this system (board number = 1)

  2: There are two PCI-M512/M512U boards installed in this system (board number =1 and 2)

  N: The number of PC-M512/M512U boards installed in this system

- **Note:**

  1. The **PCIM512_DriverInit()** function must be called before calling this function

  2. The **PCIM512_OpenBoard()** function must be called before calling this function

  3. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system.

  4. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

PCIM512_ReadBoardId

This function is used to display the ID information for the PCI-M512/M512U boards detected in the system.

- **Syntax:**

  DWORD **PCIM512_ReadBoardId**(dwBoardNo, *dwVendorId, *dwDeviceId, *dwSubVendorId, *dwSubdeviceId);

- **Parameters:**

  <u>*dwBoardNo*</u>

  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N. A value of 1 indicates the first board.

  <u>*dwVendorID*</u>

  [Output] The vendor ID for this board

  <u>*dwDeviceID*</u>

  [Output] The device ID for this board

  <u>*dwSubVendorID*</u>

  [Output] The sub-vendor ID for this board

  <u>*dwSubDeviceID*</u>

  [Output] The sub-device ID for this board

- **Returns:**

  0: This is a valid board number, so all returned ID values are valid.
  Others: This is not a valid board number, so all returned ID values are invalid.

- **Note:**

  1. The **PCIM512_DriverInit()** function must be called before calling this function.
  2. The **PCIM512_OpenBoard()** function must be called before calling this function
  3. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system.
  4. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

PCIM512_ReadBoardStatus

This function is used to detect whether or not the DLL of PCI-M512/M512U boards is open status.

- **Syntax:**
  DWORD **PCIM512_ReadBoardStatus**(DWORD dwBoardNo);

- **Parameters:**
  <u>*dwBoardNo*</u>
  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

- **Returns:**
  0: The DLL for the board dwBoardNo is not open.
  1: The DLL for the board dwBoardNo is open.

- **Note:**

1. The **PCIM512_DriverInit()** function must be called before calling this function.

PCIM512_CloseBoard

This function is used to close the PCI-M512/M512U kernel driver and release the resources for the device.

- **Syntax:**
  DWORD **PCIM512_CloseBoard**(DWORD dwBoardNo);

- **Parameters:**
  <u>*dwBoardNo*</u>
  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

- **Returns:**
  PCI_NoError : OK. The command was successful.
  PCI_BoardIsNotOpen: An error occurred because this board is not open.
  PCI_BoardNoExceedFindBoards: The board could not be found.

PCIM512_CloseAll

This function is used to close all PCI-M512/M512U kernel drivers currently in use and release the resources for the device.

- **Syntax:**
  DWORD **PCIM512_CloseAll**();

- **Parameters:**
  This function does not require any parameters

- **Returns:**
  PCI_NoError : OK. The command was successful.

3.4 SRAM Read/Write Functions

PCIM512_WriteSramByte

This function is used to write one byte of data (8-bit) to the SRAM on the PCI-M512/M512U board.

- **Syntax:**

  DWORD **PCIM512_WriteSramByte**(dwBoardNo, dwOffset, Data);

- **Parameters:**

  dwBoardNo

  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  dwOffset

  [Input] The offset address of the SRAM in the range from 0 to 0x7ffff.

  Data

  [Input] One byte of data (8-bit).

- **Returns:**

  0: Write OK. The command was successful.

  PCI_DriverNoOpen: An error occurred because the kernel driver was not found.

  PCI_BoardNoIsZero: The **dwBoardNo** value is 0. It must be in the range of 1 to N.

  PCI_BoardNoExceedFindBoards: The board could not be found because the **dwBoardNo** value is > N.

- **Note:**

  1. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system before calling this function.
  2. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

PCIM512_WriteSramWord

This function is used to write one word of data (16-bit) to the SRAM on the PCI-M512/M512U board.

- **Syntax:**

  DWORD **PCIM512_WriteSramWord**(dwBoardNo, dwOffset, Data);

- **Parameters:**

  dwBoardNo

  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  dwOffset

  [Input] The offset address of the SRAM in the range from 0 to 0x7ffff.

  Data

  [Input] One word of data (16-bit).

- **Returns:**

  0: Write OK. The command was successful.

  PCI_DriverNoOpen: An error occurred because the kernel driver was not found.

  PCI_BoardNoIsZero: The **dwBoardNo** value is 0. It must be in the range of 1 to N.

  PCI_BoardNoExceedFindBoards: The board could not be found because the **dwBoardNo** value is > N.

- **Note:**

  1. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system before calling this function.
  2. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

PCIM512_WriteSramDword

This function is used to write one DWORD of data (32-bit) to the SRAM on the PCI-M512/M512U board.

- **Syntax:**
  DWORD **PCIM512_WriteSramDword**(dwBoardNo, dwOffset, Data);

- **Parameters:**
  dwBoardNo
  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  dwOffset
  [Input] The offset address of the SRAM in the range from 0 to 0x7ffff.

  Data
  [Input] One DWORD of data (32-bit).

- **Returns:**
  0: Write OK. The command was successful.
  PCI_DriverNoOpen: An error occurred because the kernel driver was not found.
  PCI_BoardNoIsZero: The **dwBoardNo** value is 0. It must be in the range of 1 to N.
  PCI_BoardNoExceedFindBoards: The board could not be found because the **dwBoardNo** value is > N.

- **Note:**
  1. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system before calling this function.
  2. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

PCIM512_ReadSramByte

This function is used to read one byte of data (8-bit) from the SRAM on the PCI-M512/M512U board.

- **Syntax:**
  DWORD **PCIM512_ReadSramByte**(dwBoardNo, dwOffset, *Data);

- **Parameters:**
  dwBoardNo
  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  dwOffset
  [Input] The offset address of the SRAM in the range from 0 to 0x7ffff.

  Data
  [Output] One byte of data (8-bit).

- **Returns:**
  0: Write OK. The command was successful.
  PCI_DriverNoOpen: An error occurred because the kernel driver was not found.
  PCI_BoardNoIsZero: The **dwBoardNo** value is 0. It must be in the range of 1 to N.
  PCI_BoardNoExceedFindBoards: The board could not be found because the **dwBoardNo** value is > N.

- **Note:**
  1. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system before calling this function.
  2. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

PCIM512_ReadSramWord

This function is used to read one word of data (16-bit) from the SRAM on the PCI-M512/M512U board.

- **Syntax:**

  DWORD **PCIM512_ReadSramWord**(dwBoardNo, dwOffset, *Data);

- **Parameters:**

  dwBoardNo

  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  dwOffset

  [Input] The offset address of the SRAM in the range from 0 to 0x7ffff.

  Data

  [Output] One word of data (16-bit).

- **Returns:**

  0: Write OK. The command was successful.

  PCI_DriverNoOpen: An error occurred because the kernel driver was not found.

  PCI_BoardNoIsZero: The **dwBoardNo** value is 0. It must be in the range of 1 to N.

  PCI_BoardNoExceedFindBoards: The board could not be found because the **dwBoardNo** value is > N.

- **Note:**

  1. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system before calling this function.
  2. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

PCIM512_ReadSramDword

This function is used to read one word of data (32-bit) from the SRAM on the PCI-M512/M512U board.

- **Syntax:**

  DWORD **PCIM512_ReadSramDword**(dwBoardNo, dwOffset, *Data);

- **Parameters:**

  dwBoardNo

  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  dwOffset

  [Input] The offset address of the SRAM in the range from 0 to 0x7ffff.

  Data

  [Output] One DWORD of data (32-bit).

- **Returns:**

  0: Write OK. The command was successful.

  PCI_DriverNoOpen: An error occurred because the kernel driver was not found.

  PCI_BoardNoIsZero: The **dwBoardNo** value is 0. It must be in the range of 1 to N.

  PCI_BoardNoExceedFindBoards: The board could not be found because the **dwBoardNo** value is > N.

- **Note:**

  1. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system before calling this function.
  2. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

3.5 DIO Read/Write Functions

PCIM512_WriteToDo

This function is used to write one word of data (16-bit) to the Digital Output of the PCI-M512/M512U board.

- **Syntax:**
  DWORD **PCIM512_WriteToDo**(dwBoardNo, Data);

- **Parameters:**
  dwBoardNo
  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  Data
  [Input] One word of data (16-bit).

- **Returns:**
  0: Write OK. The command was successful.
  PCI_DriverNoOpen: An error occurred because the kernel driver was not found.
  PCI_BoardNoIsZero: The **dwBoardNo** value is 0. It must be in the range of 1 to N.
  PCI_BoardNoExceedFindBoards: The board could not be found because the **dwBoardNo** value is > N.

- **Note:**
  1. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system before calling this function.
  2. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

PCIM512_ReadFromDi

This function is used to read one word of data (16-bit) from Digital Input and battery status bits of the PCI-M512/M512U board.

- **Syntax:**

  DWORD **PCIM512_ReadFrom**(dwBoardNo, *Data);

- **Parameters:**

  dwBoardNo

  [Input] The board number for the detected PCI-M512/M512U board in the range from 1 to N.

  Data

  [Output] One word of data (16-bit), Bit0 to 3 are battery status bits and Bit4 to 15 are external Digital Input bits as follows:

Bit0 = 1	**BT1 is low battery**
Bit1 = 1	BT1 is bad battery
Bit2 = 1	BT2 is low battery
Bit3 = 1	BT3 is bad battery

  Refer to Section 3.1 for more information

- **Returns:**

  0: Write OK. The command was successful.

  PCI_DriverNoOpen: An error occurred because the kernel driver was not found.

  PCI_BoardNoIsZero: The **dwBoardNo** value is 0. It must be in the range of 1 to N.

  PCI_BoardNoExceedFindBoards: The board could not be found because the **dwBoardNo** value is > N.

- **Note:**

  1. Use the **PCIM512_DetectBoards()** function to detect all PCI-M512/M512U boards installed in the system before calling this function.
  2. Use the **PCIM512_ReadBoardId(…)** function to identify the detected PCI-M512/M512U boards. Refer to Section 3.1 for more information.

4. Demo Programs

4.1 For Microsoft Windows

ICP DAS PCI-M512 Series Classic Driver DLL contains a set of functions. It can be used in various application programs for PCI-M512 series card. The API functions supports many development environments and programming languages, including Microsoft Visual C++，Visual Basic，Borland Delphi，Borland C Builder++，Microsoft Visual C#.NET，Microsoft Visual VB.NET.

The demo programs of Windows OS for the PCI-M512 series can be found on the supplied CD-ROM, or can be obtained from the ICP DAS FTP web site. The location and addresses are indicated in the table below:

CD:\NAPDOS\PCI\PCI-M512\DLL\Demo\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-m512/dll/demo/

⌖ BCB4 → for Borland C^{++} Builder 4
PCIM512.H → Header files
PCIM512.LIB → Linkage library for BCB only

⌖ Delphi4 → for Delphi 4
PCIM512.PAS → Declaration files

⌖ VC6 → for Visual C^{++} 6
PCIM512.H → Header files
PCIM512.LIB → Linkage library for VC only

⌖ VB6 → for Visual Basic 6
PCIM512.BAS → Declaration files

⌖ VB.NET2005 → for VB.NET2005
PCIM512.vb → Visual Basic Source files

⌖ CSharp2005 → for C#.NET2005
PCIM512.cs → Visual C# Source files

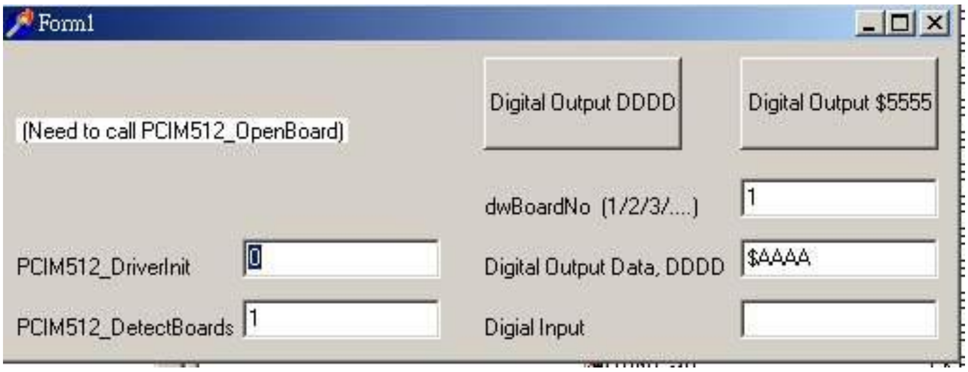
4.1.1 Determining the Board Number

During the boot process, the Plug and Play BIOS will assign an appropriate base address to PCI-M512/M512U board. If there is only one PCI-M512 board installed in the system, the board will be identified as board_1. If two or more PCI-M512 boards are installed in the system, it becomes more difficult to identify which board is board_1. The ICP DAS software driver can support a maximum of 20 boards, meaning that up to 20 PCI-M512 boards can be installed in a single host system.

The easiest way to determine the correct board number is to use the "DioTest" application that can be found in the Delphi4 demo program folder. This demo program can be used to send a value to the Digital Output, and then read it from the Digital Input. The low 4 bits of the Digital Input are the battery status bits, so they can be used as an indicator in the following manner:

1. Insert a piece of paper between the terminals of BT1 on one of the PCI-M512 boards
2. Install all PCI-M512 boards into the host system
3. Power-on the system
4. Once the system boots, the LED1 & LED2 indicators for only one of the PCI-M512 boards will be ON
5. Run the **"DioTest" application from the Delphi4 demo program folder**
6. Enter a value of 1 in the **dwBoardNo** field, as illustrated in the image below
7. Click the **Digital Output 0xDD** button
8. Check the value displayed in the **Digital Input** field. If the LSB is displayed as 1, the target PCI-M512 has been identified and can be assumed to be board 1.

DioTest



dwBoardNo (1/2/3/....) 1

PCIM512_DriverInit 0

Digital Output Data, DDDD $AAAA

PCIM512_DetectBoards 1

Digial Input

4.2 For DOS

The demo program is contained in:

CD:\NAPDOS\PCI\PISO-DIO\DOS\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-dio/dos/

- \TC*.* → for Turbo C 2.xx or above
- \MSC*.* → for MSC 5.xx or above
- \BC*.* → for BC 3.xx or above

- \TC\LIB\*.* → for TC Library
- \TC\DEMO\*.* → for TC demo program
- \TC\DIAG\*.* → for TC diagnostic program

- \TC\LIB\Large\*.* → TC Large Model Library
- \TC\LIB\Huge\*.* → TC Huge Model Library File
- \TC\LIB\Large\PCIM512.H → TC Declaration File
- \TC\LIB\Large\IOPORT_L.LIB → TC Large Model Library File

- \MSC\LIB\Large\PCIM512.H → MSC Declaration File
- \MSC\LIB\Large\IOPORTL.LIB → MSC Large Model Library File

- \BC\LIB\Large\PCIM512.H → BC Declaration File
- \BC\LIB\Large\IOPORT_L_L.LIB → BC Large Model Library File

5. Problems Report

Technical support is available at no charge as described below. The best way to report problems is to send electronic mail to Service@icpdas.com or Service.icpdas@ gmail.com on the Internet.

When reporting problems, please include the following information:

1. Is the problem reproducible? If so, how?
2. What kind and version of **platform** that you using? For example, Windows 98, Windows 2000 or 32-bit Windows XP/2003/Vista/7/8.
3. What kinds of our **products** that you using? Please see the product's manual.
4. If a dialog box with an **error message** was displayed, please include the full test of the dialog box, including the text in the title bar.
5. If the problem involves **other programs** or **hardware devices**, what devices or version of the failing programs that you using?
6. **Other comments** relative to this problem or **any suggestions** will be welcomed.

After we had received your comments, we will take about two business days to test the problems that you said. And then reply as soon as possible to you. Please check that if we had received you comments? And please keeps contact with us.