PISOCANX_ActiveX

User's Manual

ActiveX Control (OCX) for
PISO-CAN200/200U/400/400U-D/T
PEX-CAN200i-D/T
PCM-CAN200-D/T

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 assume 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

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

Table of Contents

1 INTRODUCTION

The ActiveX Control (OCX) is designed for developing program quickly, easily and friendly for PISOCAN series module. The user can plug the ActiveX Control (OCX) component and use the properties, methods and events of the object to programming application very easily.

1.1 ARCHITECTURE

The following figure gives the programming system architecture for the PISOCANX OCX.

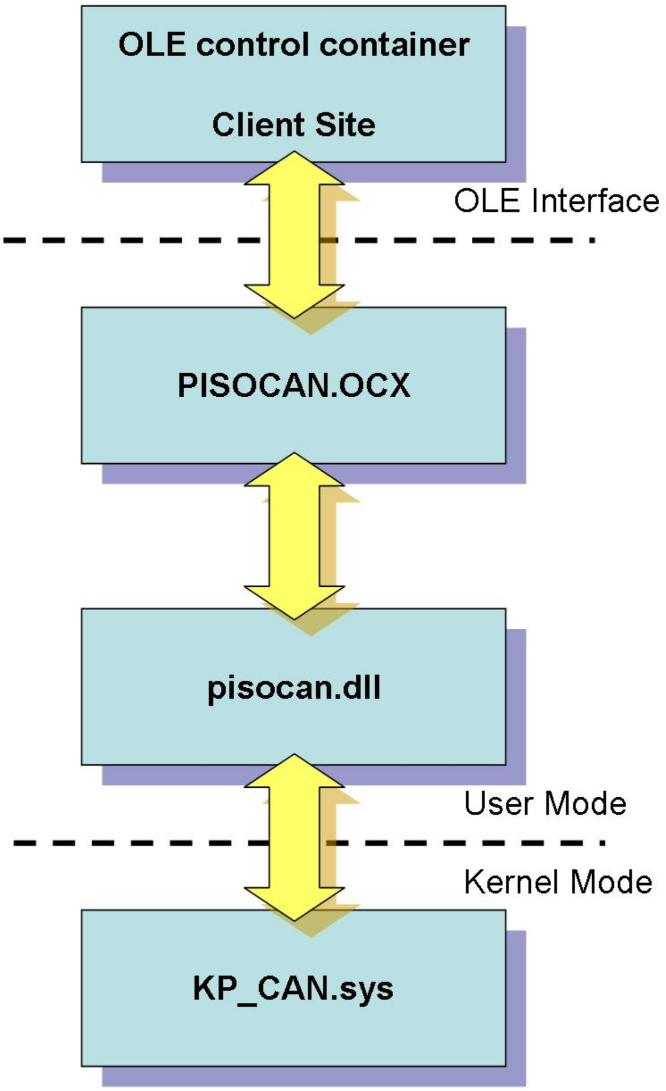


Fig 1.1 the programming system architecture for the PISOCANX OCX

2 INSTALLATION

After install the PISO-CAN driver version 2.2 later, the ActiveX Control (OCX) driver, PISOCANX.ocx, will be installed into your system and plug this control component in your development tool automatically. About the PISO-CAN driver installation, please refer to section 2.4 “Installation” of “piso-can_user_manual.pdf”.

The web path of manual:

ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/pci/piso-can200_400/manual/

The CD path of manual:

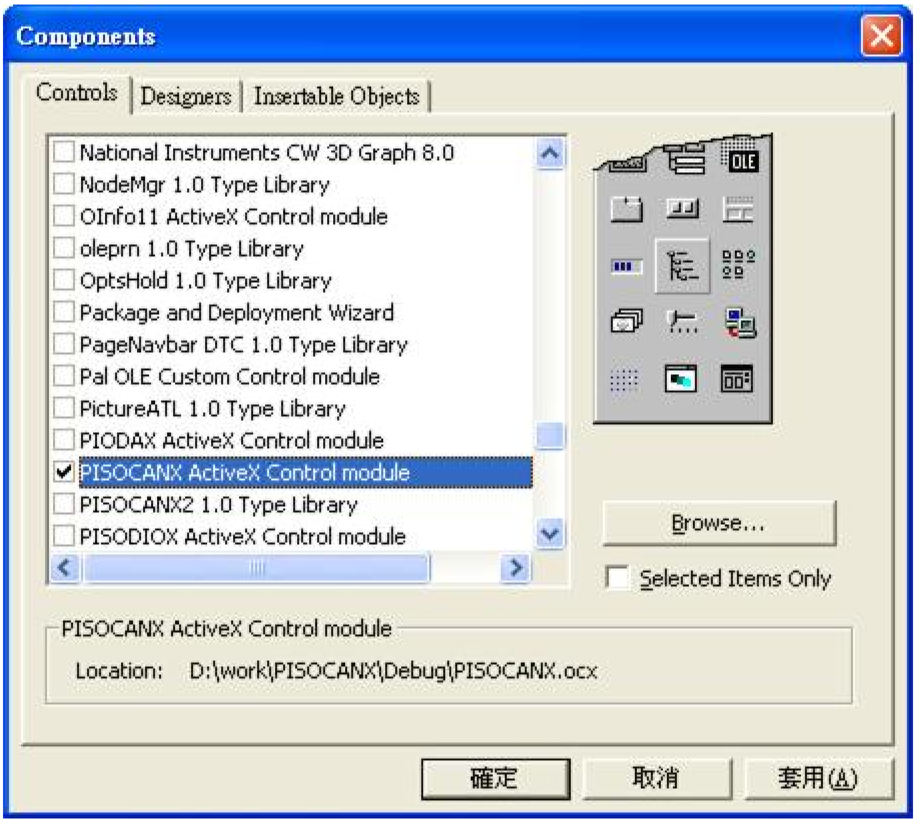
Fieldbus_CD:\\CAN\PCI\PISO-CAN200_400\Manual

Note that the ActiveX Control (OCX) supports the Windows 95/98 and Windows NT 4.0 or above. It doesn't support the Windows 3.1 or Windows NT 3.51(the format of type-lib doesn't the same between NT 3.51 and NT 4.0).

3 USING ACTIVEX CONTROL (OCX)

3.1 PISOCANX applied in Visual Basic 6.0

Step 1: Add the ActiveX Control (OCX) into the VB toolbox. Click the "Project/Components" in Visual Basic 6.0 menu as shown in Figure 3-1.



- □ National Instruments CW 3D Graph 8.0
- □ NodeMgr 1.0 Type Library
- □ OInfo11 ActiveX Control module
- □ oleprn 1.0 Type Library
- □ OptsHold 1.0 Type Library
- □ Package and Deployment Wizard
- □ PageNavbar DTC 1.0 Type Library
- □ Pal OLE Custom Control module
- □ PictureATL 1.0 Type Library
- □ PIODAX ActiveX Control module
- ☑ PISOCANX ActiveX Control module
- □ PISOCANX2 1.0 Type Library
- □ PISODIOX ActiveX Control module

□ Selected Items Only

PISOCANX ActiveX Control module

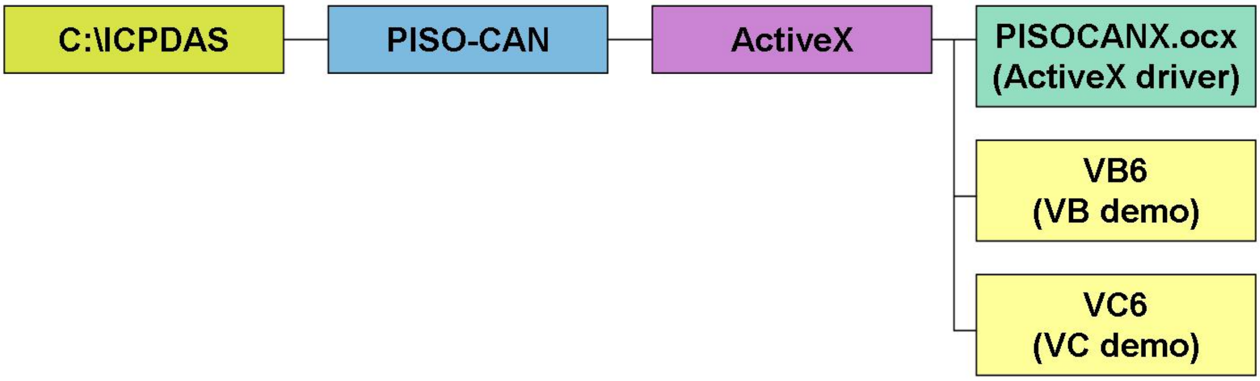
Location: D:\work\PISOCANX\Debug\PISOCANX.ocx

Fig 3.1 Component

Step 2: Check the item "PISOCANX ActiveX Control" in the Controls dialog box and click "Apply". Then the icon of "PISOCANX" will display in the ToolBox.

4 DEMO LIST

The path of the PISOCANX.ocx related files in users' PC after installed PISO-CAN driver are as below.



5 INTERFACE OF PISOCANX

5.1 Gereral Methods

5.1.1 TotalBoard

- ***Description***

  Obtain the total board number of PISO-CAN boards installed in the PCI bus.

- ***VB Syntax***

  Dim TotalBoard As Integer

  TotalBoard = PISOCANX1.**TotalBoard**

- ***Parameters***

  None

- ***Return Values***
  Return the total board number.

5.1.2 DllVersion

- ***Description***

  Obtain the version information of PISOCAN.dll driver.

- ***Visual Basic Syntax***

  Dim DllVersion As Integer

  DllVersion = PISOCANX1.**GetDllVersion**

- ***Parameters***

  None

- ***Return Values***

  DLL version information. For example: If 101(hex) is return, it means driver version **is 1.01.**

5.1.3 BoardInf

- ***Description***

  Obtain the information of PISO-CAN boards, which include vender ID, device ID and interrupt number.

- ***Visual Basic Syntax***

  PISOCANX1**.BoardInf**(0,VendorID,DeviceID,SubVendorID,SubDeviceID

  ,SubAuxiliaryID,InterruptNum)

- ***Parameters***

BoardNo:	[input] PISO-CAN board number (0~7).
VendorID :	[output] vendor ID of the specific board
DeviceID:	[output] device ID of the specific board
SubVendorID:	[output] sub-vendor ID of the specific board
SubDeviceID:	[output] sub-device ID of the specific board
SubAuxiliaryID:	[output] sub-auxiliary ID of the specific board
InterruptNum:	[output] logical interrupt number of the specific board

- ***Return Values***

  CAN_NoError: OK
  CAN_DriverError: Kernel driver can not be opened.
  CAN_BoardNumberError: BoardNo exceeds the current total board number.

5.1.4 CardPortNum

- ***Description***

  Call this function to Get CAN port number of the CAN card.

- ***Visual Basic Syntax***

  PISOCANX1**.CardPortNum**(0,PortNum)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).
  PortNum: [output] Port number of the PISO-CAN card.

- ***Return Values***

  CAN_NoError: OK
  CAN_DriverError: Kernel driver can not be opened.
  CAN_BoardNumberError: BoardNo exceeds the current total board number.

5.1.5 ActiveBoard

- ***Description***

  Activate the device. **It must be called once before using the other functions of PISO-CAN board.**

- ***Visual Basic Syntax***

  PISOCANX1.**ActiveBoard**(BoardNo)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).

- ***Return Values***

  CAN_NoError: OK
  CAN_BoardNumberError: BoardNo exceeds the current total board number.
  CAN_ActiveBoardError: The specific board can not be activated or kernel driver can not be found.

5.1.6 CloseBoard

- ***Description***

  Stop and close the kernel driver and release the device resource from computer device resource. This method must be called once before exiting the user's application program.

- ***Visual Basic Syntax***

  PISOCANX1.**CloseBoard**(BoardNo)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).

- ***Return Values***

  CAN_NoError: OK
  CAN_ActiveBoardError: The board is not activated
  CAN_BoardNumberError: BoardNo exceeds the current total board number.

5.1.7 BoardIsActive

- ***Description***

  Obtain the information about the specific board to check if the board is active or not.

  Visual Basic Syntax

  PISOCANX1.**BoardIsActive**(BoardNo)

- ***Parameter***

  BoardNo: [input] PISO-CAN board number (0~7).

- ***Return Values***

  FALSE: The specific board is inactive.

  TRUE: The specific board is active.

5.1.8 Reset

- ***Description***

  Reset the specific hardware of PISO-CAN CAN controller.

- ***Visual Basic Syntax***

  PISOCANX1.**Reset**(BoardNo, Port)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).

  Port: [input] PISO-CAN port number (1~4 for CAN400 series, 1~2 for CAN200 series)

- ***Return Values***

  CAN_NoError: OK

  CAN_DriverError: Kernel driver can't be opened.

  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.

  CAN_PortNumberError: Port number is not correct.

  CAN_ActiveBoardError: The specific board is not activated.

5.1.9 CANInitConfig

- ***Description***

Initiate and configure the specific CAN controller of PISO-CAN. After calling this function, the CAN controller will enter operating mode.

- ***Visual Basic Syntax***

PISOCANX1.**CANInitConfig**(BoardNo, Port, AccCode, AccMask, BaudRate, BT1, BT0);

- ***Parameters***

BoardNo: [input] PISO-CAN board number (0~7).
Port: [input] PISO-CAN port number (1~4 for CAN400 series, 1~2 for CAN200 series)
AccCode[4]: [input] Acceptance code for PISO-CAN controller.
AccMask[4]: [input] Acceptance mask for PISO-CAN controller.
BaudRate: [input] 0→user-defined (must to set BT0, BT1), 1→10Kbps, 2→20Kbps, 3→50Kbps, 4→125Kbps, 5→250Kbps, 6→500Kbps, 7→800Kbps, 8→1Mbps.
BT0, BT1: [input] user-defined baud rate (used only if BaudRate=0)). For example, BT0=0x04, BT1=0x1C, then baud rate setting for the PISO-CAN controller is 100Kbps. For more detail baud rate setting, please refer to manual of SJA1000 CAN controller.

- ***Return Values***

CAN_NoError: OK
CAN_DriverError: Kernel driver can't be opened.
CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
CAN_PortNumberError: Port number is not correct.
CAN_ActiveBoardError: The specific board is not activated.
CAN_SoftResetError: CAN controller software reset error.
CAN_SetACRError: Set Acceptance code to CAN controller error
CAN_SetAMRError: Set Acceptance mask to CAN controller error
CAN_SetBaudRateError: Set baud rate to CAN controller error
CAN_ConfigError: CAN controller enter operating mode failed.

5.1.10 EnableRxIrq

- ***Description***

  Enable receive interrupt for PISO-CAN Board.

- ***Visual Basic Syntax***

  PISOCANX1.**EnableRxIrq**(BoardNo, Port)

- ***Parameters***

BoardNo:	[input] PISO-CAN board number (0~7).
Port:	[input] PISO-CAN port number.

- ***Return Values***

  CAN_NoError: OK
  CAN_DriverError: Kernel driver can't be opened.
  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
  CAN_PortNumberError: Port number is not correct.
  CAN_ActiveBoardError: The specific board is not activated.
  CAN_EnableRxIrqFailure: Enable receives interrupt failed.

5.1.11 DisableRxIrq

- ***Description***

  Disable receive interrupt of the CAN controller.

- ***Visual Basic Syntax***

  PISOCANX1.**DisableRxIrq**(BoardNo, Port)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7)
  Port: [input] PISO-CAN port number.

- ***Return Values***

  CAN_NoError: OK
  CAN_DriverError: Kernel driver can't be opened.
  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
  CAN_PortNumberError: Port number is not correct.
  CAN_ActiveBoardError: The specific board is not activated.
  CAN_DisableRxIrqFailure: Disable receives interrupt failed.

5.1.12 InstallIrq

- ***Description***

  Enable or start IRQ for PISO-CAN Board. Before calling this function, **EnableRxIrq** must to be called first.

- ***Visual Basic Syntax***

  PISOCANX1.**InstallIrq**(BoardNo)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).

- ***Return Values***

  CAN_NoError: OK
  CAN_DriverError: Kernel driver can't be opened.
  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
  CAN_ActiveBoardError: The specific board is not activated.
  CAN_DisableRxIrqFailure: Disable receives interrupt failed.

5.1.13 RemoveRxIrq

- ***Description***

  Disable or stop IRQ for PISO-CAN board. After calling this function, the interrupts for all CAN controllers on board will be disabled.

- ***Visual Basic Syntax***

  PISOCANX1.**RemoveRxIrq**(BoardNo, Port)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).

- ***Return Values***

  CAN_NoError: OK
  CAN_DriverError: Kernel driver can't be opened.
  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
  CAN_ActiveBoardError: The specific board is not activated.
  CAN_RemoveIrqFailure: Disable or stop IRQ failed.

5.1.14 RxIrqStatus

- ***Description***

  Obtain receive interrupt status of the CAN controller.

- ***Visual Basic Syntax***

  PISOCANX1.**RxIrqStatus**(BoardNo, Port, Status)

- ***Parameters***

BoardNo:	[input] PISO-CAN board number (0~7).
Port:	[input] PISO-CAN port number.
Status:	[output] 0→receive interrupt disable; 1→ receive interrupt enable.

- ***Return Values***

  CAN_NoError: OK CAN_DriverError: Kernel driver can't be opened.
  CAN_BoardNumberError: BoardNo is not correct or exceeds total board number.
  CAN_PortNumberError: Port number is not correct.
  CAN_ActiveBoardError: The specific board is not activated.

5.1.15 IrqStatus

- ***Description***

  Obtain IRQ status of the PISO-CAN board.

- ***Visual Basic Syntax***

  PISOCANX1.**IrqStatus**(BoardNo, Status)

- ***Parameters***

BoardNo:	[input] PISO-CAN board number (0~7).
Status:	[output] 0→IRQ disable; 1→ IRQ enable.

- ***Return Values***

  CAN_NoError: OK CAN_DriverError: Kernel driver can't be opened.
  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
  CAN_ActiveBoardError: The specific board is not activated.

5.1.16 CANStatus

- ***Description***

  Obtain the status of CAN controller for PISO-CAN board.

- ***Visual Basic Syntax***

  PISOCANX1.**Status**(BoardNo, Port, Status)

- ***Parameters***

BoardNo:	[input] PISO-CAN board number (0~7).
Port:	[input] PISO-CAN port number.
Status:	[output] Status value of CAN controller.

Table 5.1 Bit interpretation of the “Status” parameter.

BIT	***NAME***	***VALUES***	***STATUS***
Bit7	Bus Status	1	bus-off
		0	bus-on
Bit6	Bus Status	1	error
		0	ok
Bit5	Transmit Status	1	transmit
		0	idle
Bit4	Receive Status	1	receive
		0	idle
Bit3	Transmission Complete Status	1	complete
		0	incomplete
Bit2	Transmit Buffer Status	1	release
		0	locked
Bit1	Data Overrun Status	1	overrun
		0	absent
Bit0	Receive Buffer Status	1	full/not empty
		0	empty

- ***Return Values***

  CAN_NoError: OK
  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
  CAN_PortNumberError: Port number is not correct.
  CAN_ActiveBoardError: The specific board is not activated.

5.1.17 SendMsg

- ***Description***

Send a CAN message immediately.

- ***Visual Basic Syntax***

PISOCANX1.**SendMsg**(BoardNo, Port, mode, id, rtr, len, data)

- ***Parameters***

BoardNo:	[input] PISO-CAN board number (0~7).
Port:	[input] PISO-CAN port number.
mode:	[input] 0→ 11-bit identifier, 1 → 29-bit identifier.
id:	[input] Identifier.
rtr:	[input] Remote transmission request.
len:	[input] Data length.
data[8]:	[input] data byte.

- ***Return Values***

CAN_NoError: OK
CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
CAN_PortNumberError: Port number is not correct.
CAN_ActiveBoardError: The specific board is not activated.
CAN_TransmitBufferLocked: Transmit buffer in CAN chip is locked.
CAN_TransmitIncomplete: Transmission is not yet completed.
CAN_ConfigError: Port has not been configured successfully.

5.1.18 RxMsgCount

- ***Description***

  Obtain the number of messages available within the CAN controller's RXFIFO or the software buffer (4KBytes). After calling the functions CAN_EnableRxIrq and CAN_InstallIrq, the number of messages is within the software buffer; otherwise it is within the CAN controller's RXFIFO.

- ***Visual Basic Syntax***

  PISOCANX1.**RxMsgCount**(BoardNo, Port)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).
  Port: [input] PISO-CAN port number.

- ***Return Values***

  The number of CAN messages.

  Note：If the parameter for BoardNo or Port isn't correct, the return value will always be 0.

5.1.19 ReceiveMsg

- ***Description***

  Obtain receive message from CAN controller’s RXFIFO or software buffer. After calling the functions **CAN_EnableRxIrq** and **CAN_InstallIrq**, the messages is within the software buffer, otherwise it is within the CAN controller’s RXFIFO.

- ***Visual Basic Syntax***

  PISOCANX1.**ReceiveMsg**(BoardNo, Port, MsgTimeStamps, mode, id, rtr, len , data)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).
  Port: [input] PISO-CAN port number.
  MsgTimeStamps: [output] This parameter will record the time with system clock counter when the CAN message is received from SJA1000. The system clock counter starts to count after the PC boots up. If more than one CAN messages are received and stored in the 64-byte SJA1000 FIFO, the time stamps of these CAN messages may be closed.
  mode: [output] 0→ 11-bit identifier, 1 → 29-bit identifier.
  id: [output] Identifier
  rtr: [output] Remote transmission request
  len: [output] Data length
  data[8]: [output] Data byte

- ***Return Values***

  CAN_NoError: OK
  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
  CAN_PortNumberError: Port number is not correct.
  CAN_ActiveBoardError: This board is not activated.
  CAN_ConfigError: Port has not been configured successfully.
  CAN_ReceiveBufferEmpty: CAN controller’s RXFIFO is empty.
  CAN_SoftBufferIsEmpty: Software RX Buffer Is empty.
  CAN_SoftBufferIsFull: Software RX Buffer Is full.

5.1.20 ClearSoftBuffer

- ***Description***

Clear the software buffer of the PISOCAN.DLL driver.

- ***Visual Basic Syntax***

PISOCANX1.**ClearSoftBuffer**(BoardNo, Port)

- ***Parameters***

BoardNo: [input] PISO-CAN board number (0~7).
Port: [input] PISO-CAN port number.

- ***Return Values***

CAN_NoError: OK
CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
CAN_PortNumberError: Port number is not correct.

5.1.21 ClearDataOverrun

- ***Description***

  Clear the data overrun status bit for the CAN controller.

- ***Visual Basic Syntax***

  PISOCANX1.**ClearDataOverrun**(BoardNo, Port)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).
  Port: [input] PISO-CAN port number.

- ***Return Values***

  CAN_NoError: OK
  CAN_BoardNumberError: BoardNo is not correct or exceeds the current total board number.
  CAN_PortNumberError: Port number is not correct.
  CAN_ActiveBoardError: The specific board is not activated.
  CAN_ConfigError: CAN controller enter operating mode failure.

5.1.22 OutputByte

- ***Description***

Write data to CAN chip (SJA1000).

- ***Visual Basic Syntax***

PISOCANX1.**OutputByte**(BoardNo, Port, Offset, Value)

- ***Parameters***

BoardNo:	[input] PISO-CAN board number (0~7).
Port:	[input] PISO-CAN port number.
Offset:	[input] Address offset from base address
Value:	[input] Data byte

- ***Return Values***

None.

5.1.23 InputByte

- ***Description***

  Read data from CAN chip (SJA1000).

- ***Visual Basic Syntax***

  PISOCANX1.**InputByte**(BoardNo, Port, Offset)

- ***Parameters***

  BoardNo: [input] PISO-CAN board number (0~7).
  Port: [input] PISO-CAN port number.
  Offset: [input] Address offset from base address.

- ***Return Values***

  Data Byte of CAN chip.