



Blue Chip Technology ISA-PIO Device Driver

**Windows 2000 and Windows XP device driver
for the ISA-PIO (PIO-48)**

User and Developers Manual

Document Reference
Document Issue Level

ISA-PIO Device Driver User Manual
1.0

**Blue Chip Technology Ltd.
Chowley Oak
Tattenhall
Chester
CH3 9EX
UK**

**Telephone: +44 01829 772000
Facsimile: +44 01829 772001
www.bluechiptechnology.co.uk
support@bluechiptechnology.co.uk**

Contents

CONTENTS	2
TABLE OF FIGURES	ERROR! BOOKMARK NOT DEFINED.
DISCLAIMER	3
INTRODUCTION	3
SYSTEM HEALTH MONITOR APPLICATION.....	ERROR! BOOKMARK NOT DEFINED.
INTRODUCTION	ERROR! BOOKMARK NOT DEFINED.
INSTALLING THE SYSTEM HEALTH MONITOR	ERROR! BOOKMARK NOT DEFINED.
CONFIGURING THE SYSTEM HEALTH MONITOR.....	ERROR! BOOKMARK NOT DEFINED.
SYSTEM HEALTH MONITOR STATUS DISPLAY	ERROR! BOOKMARK NOT DEFINED.
REMOTE MONITOR APPLICATION.....	ERROR! BOOKMARK NOT DEFINED.
INTRODUCTION	ERROR! BOOKMARK NOT DEFINED.
DATA LOGGING WITH THE REMOTE MONITOR SOFTWARE.....	ERROR! BOOKMARK NOT DEFINED.
CONFIGURING THE REMOTE MONITOR SOFTWARE.....	ERROR! BOOKMARK NOT DEFINED.
API FUNCTIONS.....	5
BCTENABLEHWMONITOR	ERROR! BOOKMARK NOT DEFINED.
BCTDISABLEHWMONITOR	ERROR! BOOKMARK NOT DEFINED.
BCTREADVOLTAGE.....	6
BCTREADTEMP	7
BCTREADFANSPEED	ERROR! BOOKMARK NOT DEFINED.
API FUNCTION ERROR CODES.....	11
BCTERR_OK.....	11
BCTERR_MONITOR_ALREADY_ENABLED.....	11
BCTERR_UNABLE_TO_CONNECT_WITH_DRIVER.....	11
BCTERR_UNABLE_TO_FIND_HW_MONITOR.....	11
BCTERR_UNABLE_TO_DISCONNECT_DRIVER.....	11
BCTERR_MONITOR_NOT_ENABLED.....	11
BCTERR_INVALID_VOLTAGE_SOURCE_SPECIFIED	11
BCTERR_INVALID_FAN_SPECIFIED.....	11
BCTERR_NO_FAN_CONNECTED	12
BCTERR_INVALID_TEMP_SPECIFIED.....	12
BCTERR_UNSUPPORTED_HW_MONITOR	12
BCTERR_UNKNOWN_HW_MONITOR.....	12
BCTERR_UNABLE_TO_CLOSE_BOARD_HANDLE.....	12
BCTERR_UNABLE_TO_OPEN_BOARD_HANDLE.....	ERROR! BOOKMARK NOT DEFINED.
BCTERR_UNABLE_TO_READ_VALUES.....	ERROR! BOOKMARK NOT DEFINED.
APPENDIX A - SUPPORTED PLATFORMS	ERROR! BOOKMARK NOT DEFINED.
APPENDIX B - TCP/IP PORT SUMMARY	ERROR! BOOKMARK NOT DEFINED.

Disclaimer

All rights reserved. No part of this publication may be reproduced, stored in any retrieval system, or transmitted, in any form or by any means, electronic, mechanical photocopied, recorded or otherwise, without the prior permission, in writing, from the publisher. For written permission in the UK contact Blue Chip Technology.

Information offered in this manual is believed to be correct at the time of printing. Blue Chip Technology accepts no responsibility for any inaccuracies. The information contained herein is subject to change without notice. There are no express or implied licences granted herein to any intellectual property rights of Blue Chip Technology Ltd.

All trademarks and registered names acknowledged.

Introduction

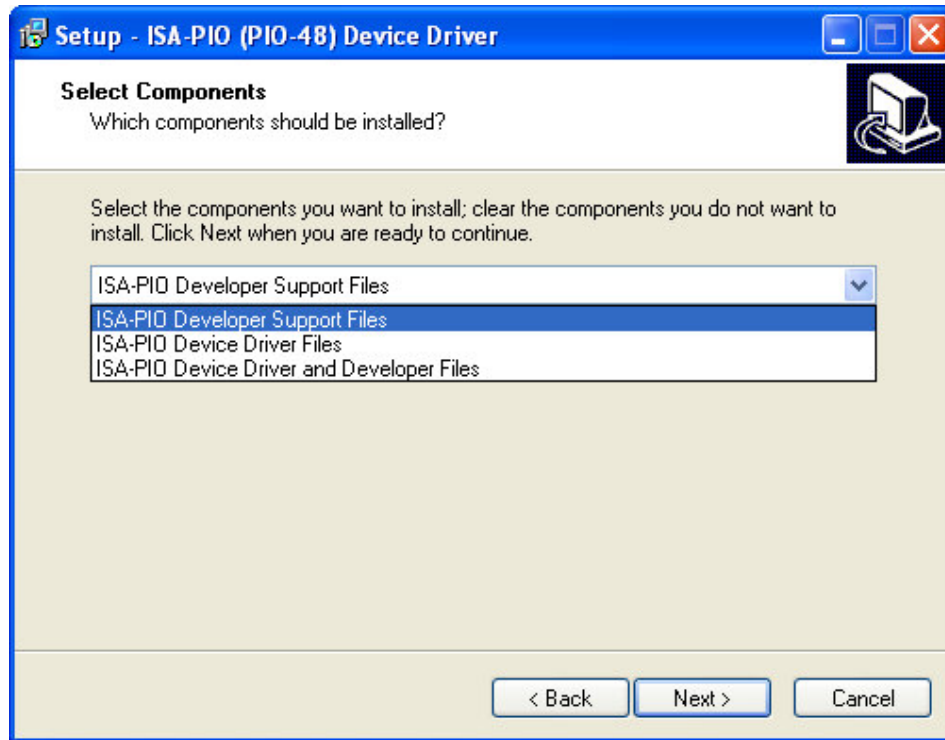
This user guide provides the details of the device driver software provided for accessing the ISA-PIO (PIO-48) from Blue Chip Technology, how to install the drivers onto a target system and how to develop applications that access the ISA-PIO using the API calls provided.

The software provided consists of three runtime files that have to be installed on the system that contains the ISA-PIO and a number of application support files used by developers writing applications to use the ISA-PIO. The development PC and the target PC do not have to be the same computer.

The ISA-PIO card consists of 48 programmable digital IO lines through the use of two 8255 (or equivalent) devices decoded through 8 IO port addresses on the ISA bus. At the present time only mode 0 of the 8255 is supported by the device driver. If other modes are required then please contact Technical Services at Blue Chip Technology.

Installation

The ISA-PIO device driver is provided as a single executable file with the name “*ISA-PIO Device Driver vX.XX.XX*” where X.XX.XX is the version number of the driver. To install the device driver run the file and after accepting the license agreement and specifying the installation folder a drop down menu is shown that gives the various installation options as shown below.



The three installation options are as follows...

ISA-PIO Developer Support Files

Selecting this option will install only the header and library files needed by application developers who wish to write applications that will communicate with the ISA-PIO card. These files, along with the user manual, are installed into the folder specified during the installation process.

ISA-PIO Device Driver Files

Selecting this option will install only the device driver and API library files needed by a system that includes the ISA-PIO cards. These files are always installed in the system32 and drivers folders of the Windows installation.

ISA-PIO Device Driver and Developer Files

Selecting this option installs all of the files from the two other installation options and allows both the application development and testing to take place on the same computer.

API Functions

In order to simplify the use of the ISA-PIO a number of easy to use API functions have been developed. Each of these functions is described below and returns an error code and these should be checked each time that a function is called. If a function returns an error code other than BCTERR_OK then any data returned by the function is invalid and should be ignored.

BCTRegisterBoard

As the ISA-PIO includes no plug and play functionality it can not be automatically be detected by the operating system and so must be registered with the device driver by the application programmer. The BCTRegisterBoard function registers the board with the driver and can support up to 10 ISA-PIO boards in the system.

Function prototype:

```
DWORD BCTRegisterBoard(DWORD wBaseAddress, PHANDLE phBoard);
```

Parameters:

wBaseAddress	Specifies the base address for the ISA-PIO card as set by the hardware links on the card. If multiple cards are installed into the same system then they should each be set so that there are no address conflicts.
phBoard	This is a pointer to a windows handle that is returned on successful registration of the board. Each registered board will have a unique handle and is used to identify the board in calls to other functions.

Return value:

BCTRegisterBoard returns one of the error codes defined in the PIO48XP.H, or equivalent, header file. Please see the section headed "API Function Error Codes" for a description of the error and any corrective action required.

BCTDeregisterBoard

BCTDeregisterBoard is used to notify the driver that a previously registered board is no longer required so that the driver can release the resources used internally. This function should be called for each registered board as the application terminates.

Function prototype:

```
DWORD BCTDeregisterBoard(HANDLE hBoard);
```

Parameters:

hBoard	The handle returned by a successful call to BCTRegisterBoard used to identify the board to deregister.
--------	--

Return value:

BCTDeregisterBoard returns one of the error codes defined in the PIO48XP.H, or equivalent, header file. Please see the section headed "API Function Error Codes" for a description of the error and any corrective action required.

BCTInit8255

Before an 8255 can be used then it must be initialised to specify which ports are to be used and inputs and which are to be used as outputs. BCTInit8255 is used to specify the mode of operation and the port directions for the 8255. A successful call to BCTRegisterBoard must be made before this function will operate successfully and this function needs to be called twice for each ISA-PIO board.

Function prototype:

```
DWORD BCTInit8255(HANDLE hBoard, BYTE bDevice, BYTE bMode,  
                 BYTE bPortA, BYTE bPortB, BYTE bPortC);
```

Parameters:

hBoard	The handle returned by a successful call to BCTRegisterBoard used to identify the board to deregister.
bDevice	Specifies the 8255 device within the ISA-PIO to initialise. Valid options are: PIO_DEVICE0 for 8255 device 0 on the ISA-PIO PIO_DEVICE1 for 8255 device 1 on the ISA-PIO
bMode	Specifies the mode of operation for the 8255. Valid options are: PIO_MODE0 for mode 0 operation. This is currently the only supported mode.
bPortA	Specifies the required port direction for port A on the specified PIO device. Valid options are: INPUT to specify the port as an input port. OUTPUT to specify the port as an output port.
bPortB	Specifies the required port direction for port B on the specified PIO device. Valid options are: INPUT to specify the port as an input port. OUTPUT to specify the port as an output port.
bPortC	Specifies the required port direction for port C on the specified PIO device. Valid options are: INPUT to specify the port as an input port. OUTPUT to specify the port as an output port. INOUT to specify the lower four bits as input and the upper four bits as output. OUTIN to specify the upper four bits as input and the lower four bits as output.

Return value:

BCTInit8255 returns one of the error codes defined in the PIO48XP.H, or equivalent, header file. Please see the section headed "API Function Error Codes" for a description of the error and any corrective action required.

BCTWritePort

BCTWritePort is used to write a value to a port that has been configured as an output port. A successful call to BCTRegisterBoard must be made before this function will operate successfully.

Function prototype:

```
DWORD BCTWritePort (HANDLE hBoard, BYTE bDevice, BYTE bPort,  
                   BYTE bValue);
```

Parameters:

hBoard	The handle returned by a successful call to BCTRegisterBoard used to identify the board to containing the port to be written.
bDevice	Specifies the 8255 device within the ISA-PIO to initialise. Valid options are: PIO_DEVICE0 for 8255 device 0 on the ISA-PIO PIO_DEVICE1 for 8255 device 1 on the ISA-PIO
bPort	Specifies the port within the specified 8225 to write a value to. Valid options are: PIO_PORTA for port A PIO_PORTB for port B PIO_PORTC for port C
bValue	Specifies the value to be written to the port.

Return value:

BCTWritePort returns one of the error codes defined in the PIO48XP.H, or equivalent, header file. Please see the section headed "API Function Error Codes" for a description of the error and any corrective action required.

BCTReadPort

BCTReadPort is used to read a value from a port that has been configured as an input port. A successful call to BCTRegisterBoard must be made before this function will operate successfully.

Function prototype:

```
DWORD BCTReadPort (HANDLE hBoard, BYTE bDevice, BYTE bPort,  
                  PBYTE pbValue);
```

Parameters:

hBoard	The handle returned by a successful call to BCTRegisterBoard used to identify the board to containing the port to be read.
bDevice	Specifies the 8255 device within the ISA-PIO to initialise. Valid options are: PIO_DEVICE0 for 8255 device 0 on the ISA-PIO PIO_DEVICE1 for 8255 device 1 on the ISA-PIO

<code>bPort</code>	Specifies the port within the specified 8255 to read a value from. Valid options are: <code>PIO_PORTA</code> for port A <code>PIO_PORTB</code> for port B <code>PIO_PORTC</code> for port C
<code>pbValue</code>	Specifies a pointer to an 8 bit variable to hold the value read from the port.

Return value:

BCTReadPort returns one of the error codes defined in the PIO48XP.H, or equivalent, header file. Please see the section headed "API Function Error Codes" for a description of the error and any corrective action required.

BCTSetBit

BCTSetBit is used to write a value to a single bit within a port that has been configured as an output port. A successful call to BCTRegisterBoard must be made before this function will operate successfully.

Function prototype:

```
DWORD BCTSetBit(HANDLE hBoard, BYTE bDevice, BYTE bPort, BYTE bBit,
                BYTE bState);
```

Parameters:

<code>hBoard</code>	The handle returned by a successful call to BCTRegisterBoard used to identify the board to containing the port that contains the bit to be written.
<code>bDevice</code>	Specifies the 8255 device within the ISA-PIO to initialise. Valid options are: <code>PIO_DEVICE0</code> for 8255 device 0 on the ISA-PIO <code>PIO_DEVICE1</code> for 8255 device 1 on the ISA-PIO
<code>bPort</code>	Specifies the port within the specified 8255 to write a value to. Valid options are: <code>PIO_PORTA</code> for port A <code>PIO_PORTB</code> for port B <code>PIO_PORTC</code> for port C
<code>bBit</code>	Specifies the bit within the port to be written.
<code>bState</code>	Specifies the logic state to set the bit.

Return value:

BCTSetBit returns one of the error codes defined in the PIO48XP.H, or equivalent, header file. Please see the section headed "API Function Error Codes" for a description of the error and any corrective action required.

BCTGetBit

BCTGetBit is used to read a single bit from within a port that has been configured as an input port. A successful call to BCTRegisterBoard must be made before this function will operate successfully.

Function prototype:

```
DWORD BCTGetBit(HANDLE hBoard, BYTE bDevice, BYTE bPort, BYTE bBit,
                PBYTE pbState);
```

Parameters:

hBoard	The handle returned by a successful call to BCTRegisterBoard used to identify the board to containing the port that contains the bit to be read.
bDevice	Specifies the 8255 device within the ISA-PIO to initialise. Valid options are: PIO_DEVICE0 for 8255 device 0 on the ISA-PIO PIO_DEVICE1 for 8255 device 1 on the ISA-PIO
bPort	Specifies the port within the specified 8255 to read a value from. Valid options are: PIO_PORTA for port A PIO_PORTB for port B PIO_PORTC for port C
bBit	Specifies the bit within the port to be read.
pbState	Specifies a pointer to an 8 bit variable to hold the logic state of the bit read.

Return value:

BCTGetBit returns one of the error codes defined in the PIO48XP.H, or equivalent, header file. Please see the section headed "API Function Error Codes" for a description of the error and any corrective action required.

API Function Error Codes

BCTERR_OK

The function has returned successfully and with no errors. Any data returned by the function is valid.

BCTERR_HANDLE_NOT_FOUND

The handle specified is not a valid handle for a registered board. Check that the board has been registered first and that the correct handle is being used.

BCTERR_UNABLE_TO_OPEN_DRIVER

The API layer has been unable to connect to the kernel mode driver. Check that the runtime files have been correctly installed on the target system.

BCTERR_INVALID_HANDLE_SPECIFIED

The handle specified is not a valid handle for a registered board. Check that the board has been registered first and that the correct handle is being used.

BCTERR_INVALID_DEVICE_SPECIFIED

The device identifier used is not valid for the ISA-PIO. Check that the device being specified is either `PIO_DEVICE0` or `PIO_DEVICE1`.

BCTERR_INVALID_PORT_SPECIFIED

The port identifier being used is not valid for the 8255 on the ISA-PIO. Check that the port being specified is either `PIO_PORTA`, `PIO_PORTB` or `PIO_PORTC`.

BCTERR_HANDLE_NOT_NULL

The handle being passed to a call to `BCTRegisterBoard` is not `NULL`. Check that the board isn't trying to be registered for a second time and that the variable being used has been initialised correctly.

BCTERR_TOO_MANY_BOARDS

A call to `BCTRegisterBoard` has failed because the maximum number of supported boards has been exceeded. If support for more boards is required then please contact Technical Services at Blue Chip Technology.

BCTERR_INVALID_BASE_ADDRESS

The base address specified in a call to `BCTRegisterBoard` is invalid as it conflicts with a board that has already been registered. Check that the value being specified is correct and check that the hardware links on the boards aren't set to conflicting addresses.

BCTERR_INVALID_PORTA DIRECTIONS

The direction specified for port A is invalid. Check that the direction being specified is either `INPUT` or `OUTPUT`.

BCTERR_INVALID_PORTB DIRECTIONS

The direction specified for port B is invalid. Check that the direction being specified is either `INPUT` or `OUTPUT`.

BCTERR_INVALID_PORTC DIRECTIONS

The direction specified for port C is invalid. Check that the direction being specified is `INPUT`, `OUTPUT`, `INOUT` or `OUTIN`.

BCTERR_INVALID_MODE_SPECIFIED

The mode specified for the 8255 is invalid. Check that the mode being specified is `PIO_MODE0`.