

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

ACCES I/O PRODUCTS, INC.
10623 Roselle Street, San Diego, CA 92121 • (858) 550-9559 • Fax (858) 550-7322
contactus@accesio.com • www.accesio.com

USB 16-CHANNEL HIGH-SPEED DIGITAL INPUT/OUTPUT MODULE

MODELS
USB-DIO-16H
USB-DIO-16A
USB-DI16A
USB-DO16A

USER MANUAL

MUSB-DIO-16H.B1d

Notice

The information in this document is provided for reference only. ACCES does not assume any liability arising out of the application or use of the information or products described herein. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of ACCES, nor the rights of others.

IBM PC, PC/XT, and PC/AT are registered trademarks of the International Business Machines Corporation.

Printed in USA. Copyright 2008 by ACCES I/O Products, Inc. 10623 Roselle Street, San Diego, CA 92121. All rights reserved.

WARNING!!

ALWAYS CONNECT AND DISCONNECT YOUR FIELD CABLING WITH THE COMPUTER POWER OFF. ALWAYS TURN COMPUTER POWER OFF BEFORE INSTALLING A BOARD. CONNECTING AND DISCONNECTING CABLES, OR INSTALLING BOARDS INTO A SYSTEM WITH THE COMPUTER OR FIELD POWER ON MAY CAUSE DAMAGE TO THE I/O BOARD AND WILL VOID ALL WARRANTIES, IMPLIED OR EXPRESSED.

Warranty

Prior to shipment, ACCES equipment is thoroughly inspected and tested to applicable specifications. However, should equipment failure occur, ACCES assures its customers that prompt service and support will be available. All equipment originally manufactured by ACCES which is found to be defective will be repaired or replaced subject to the following considerations.

Terms and Conditions

If a unit is suspected of failure, contact ACCES' Customer Service department. Be prepared to give the unit model number, serial number, and a description of the failure symptom(s). We may suggest some simple tests to confirm the failure. We will assign a Return Material Authorization (RMA) number which must appear on the outer label of the return package. All units/components should be properly packed for handling and returned with freight prepaid to the ACCES designated Service Center, and will be returned to the customer's/user's site freight prepaid and invoiced.

Coverage

First Three Years: Returned unit/part will be repaired and/or replaced at ACCES option with no charge for labor or parts not excluded by warranty. Warranty commences with equipment shipment.

Following Years: Throughout your equipment's lifetime, ACCES stands ready to provide on-site or in-plant service at reasonable rates similar to those of other manufacturers in the industry.

Equipment Not Manufactured by ACCES

Equipment provided but not manufactured by ACCES is warranted and will be repaired according to the terms and conditions of the respective equipment manufacturer's warranty.

General

Under this Warranty, liability of ACCES is limited to replacing, repairing or issuing credit (at ACCES discretion) for any products which are proved to be defective during the warranty period. In no case is ACCES liable for consequential or special damage arriving from use or misuse of our product. The customer is responsible for all charges caused by modifications or additions to ACCES equipment not approved in writing by ACCES or, if in ACCES opinion the equipment has been subjected to abnormal use. "Abnormal use" for purposes of this warranty is defined as any use to which the equipment is exposed other than that use specified or intended as evidenced by purchase or sales representation. Other than the above, no other warranty, expressed or implied, shall apply to any and all such equipment furnished or sold by ACCES.

TABLE OF CONTENTS

Chapter 1: Introduction 5
- **Features** 5
- **Applications** 5
- **Functional Description** 5
  - **Figure 1-1:** Block Diagram 6
  - **High-Speed Digital I/O** 6
  - **Digital I/O** 6
- **Ordering Guide** 7
  - **Model Options** 7
  - **Included with your board** 7
  - **Optional accessories** 7

Chapter 2: Installation 8
- **Software CD Installation** 8
- **Hardware Installation** 8
- **NOTES ABOUT HAND-SHAKING:** 8

Chapter 3: Hardware Details 9
- **Option Selection** 9
  - **Figure 3-1:** Option Selection Map 9
- **USB Connector (P1)** 10
- **Embedded USB Connector (P4)** 10
- **LED** 10
- **DC Power Jack (Optional)** 10
- **68-PIN I/O Connector (J1)** 10

Chapter 4: USB Address Information 11

Chapter 5: Programming 12

Chapter 6: Connector Pin Assignments 13
  - **TABLE 6-1:** Connector Pin Assignments 13
  - **TABLE 6-2:** Detailed Pin Descriptions 14

Chapter 7: Specifications 15
- **TTL High-Speed Digital I/O's** (LVTTL Optional) 15
- **TTL Standard Digital I/O's** (LVTTL Optional) 15
- **Internal Clock** 15
- **Data FIFOs** 15
- **Environmental** 15
- **Power** 15

Appendix A: Theory of Operation 17
- **Output** 17
- **Input** 18
- **Paired Operation**: 18
- **Warning about bus contention** 19

Appendix B: Proposed I/O Negotiation Algorithms 20
- **Hard Coded (Embedded) Configuration** 21
- **Handshaking Used (Known Slave)** 23
- **Handshaking Used (Known Master)** 25

Customer Comments 26

Chapter 1: Introduction

This USB based Data Acquisition Module is an ideal solution for adding portable, easy-to-install high-speed digital input/output to any computer with a USB port. The unit is a USB 2.0 high-speed device and requires a USB 2.0 port to function.

Features

- 16 high-speed digital I/O lines featuring continuous throughput of 16MB/s
- Capable of 80MB/s bursts
- Flexible internal or external synchronous clocking and handshaking capabilities
- 18 additional digital I/O lines
- All output lines buffered with 24mA sink/source current capabilities
- Standard 68-pin High Density connector
- Custom high-speed function driver
- PC/104 module size and mounting compatibility
- Small (4" x 4" x 1.25") rugged industrial enclosure
- Miniature USB connector for embedded applications

Applications

- Digital Imaging
- Embedded data acquisition
- Data Transfer
- Education/Laboratory

Functional Description

This product is a USB based digital board with16 high-speed digital inputs/outputs capable of up to 8MHz (16MegaByte) continuous scanning and up to 40MHz (80MegaByte) bursts. Four model versions are available; an input/output unit (USB-DIO-16A), an input only unit (USB-DI16A), an output only unit (USB-DO16A) all with 4k word FIFOs, and an input/output unit with expanded 64k word FIFO (USB-DIO-16H).

The 16 high-speed lines are TTL-compatible digital inputs/outputs with high-current capabilities. The programmable clock facilitates capability of 1k-40MHz transfers or the user may choose to use their own external clock. All clock signals are provided on the connector and are synchronous. In addition to the 16 high-speed inputs/outputs there are 18 standard digital I/O channels configurable as 4 separate ports for input or output. Power is supplied to the board via the USB cable; for higher current capabilities, external 9V AC/DC power adapter may be used. The I/O wiring connections for this unit are via an industry standard 68-pin SCSI connector. For external circuits, fused +5VDC (see note) power is available at the connector based on a hardware jumper selection made by the user. This resettable fuse is rated at 0.5A.

> *Note: Anywhere +5VDC is referred to in this manual with the exception of USB power required can be substituted with 3.3VDC for the LVTTL option.*

Pull-ups to +5VDC on the board assure that I/O lines are at a known state at power-up until the board is initialized by system software.

A type B USB connector is used on all models which features a high retention design that complies with the class 1, Div II minimum withdrawal requirement of over 3 pounds of force.

This connector has an orange color-coded insulator to differentiate it from standard USB connectors.

The complete unit is designed for use in rugged industrial environments but is compact enough to fit nicely onto any desk or testing station. The board is PC/104 sized (3.550 by 3.775 inches) with mounting capability and ships standard inside a steel powder-coated enclosure with an anti-skid bottom.

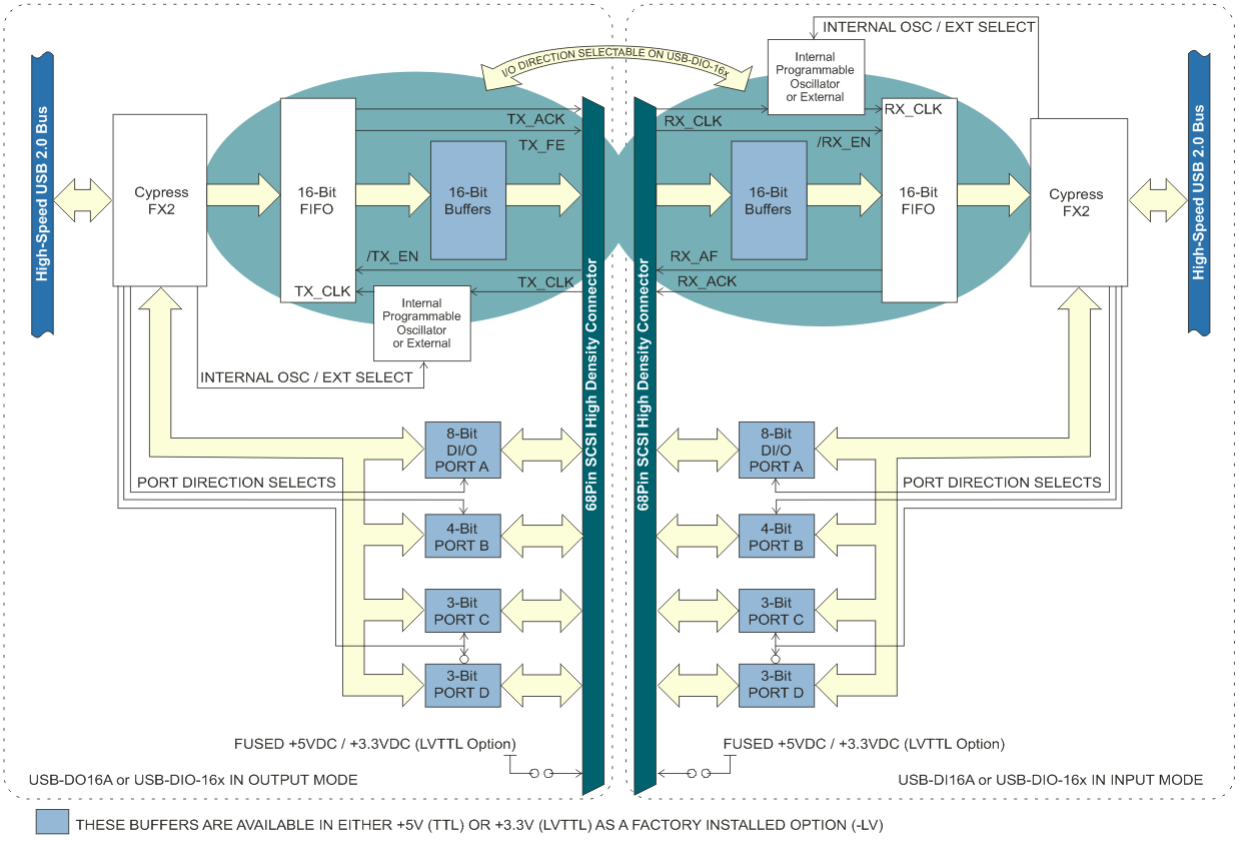


Figure 1-1: Block Diagram

High-Speed Digital I/O

There is one 16-bit high-speed Digital I/O bus (DATA I/O0 to DATA I/O15) available on this board. This bus can be configured for either inputs/outputs depending on the model ordered. These ports can also be tri-stated. Each DIO line is buffered and capable of sourcing 24mA or sinking 24mA. Be sure to consult the Power section for total power limitations before operation. By default the DIO lines are pulled up with a 10KΩ resistor to 5V.

Digital I/O

There are 18 additional Digital I/O lines (Port A, Port B, Port C and Port D) available on this board. Each port can be configured for either input or output however Port C and Port D are always configured in opposite directions of each other. These ports can also all be tri-stated. Port A is an 8-bit Port, Port B is a 4-bit Port, and Ports C and D are each 3-bits. Each DIO line is buffered and capable of sourcing/sinking 24mA. Be sure to consult the Power section for total power limitations before operation. By default the DIO lines are pulled up with a 10KΩ resistor to 5V.

Ordering Guide

USB-DI16A - 16 High-Speed Digital Inputs ONLY w/4K word FIFO and 18 standard I/Os
USB-DO16A - 16 High-Speed Digital Outputs ONLY w/4K word FIFO and 18 standard I/Os
USB-DIO-16A - 16 High-Speed Digital Inputs OR Outputs w/4K word FIFO and 18 standard I/Os
USB-DIO-16H - 16 High-Speed Digital Inputs OR Outputs with 64K word FIFO and 18 standard I/Os

Model Options

- -LV LVTTL for 3.3V applications
- -P External AC/DC adapter (power jack/regulator installed)
- -OEM Board only (no enclosure)
- -DIN DIN rail mounting provision
- -T Extended Temperature Operation (-40° to +85°C)
- -S0x "x" = special number designator
  examples include 16K FIFO, 32K FIFO, 64K FIFO, conformal coating, etc.

Included with your board

The following components are included with your shipment. Please take time now to ensure that no items are damaged or missing.

1. USB Module (unit installed in labeled enclosure)
2. 6' USB cable
3. Software Master CD (PDF user manual installed with product package)
4. Printed USB I/O Quick-Start Guide

Optional accessories

- C68PS18L - 68-Pin SCSI 18" shielded cable with one-touch latches
- STB-68 - Screw Terminal Board (mounted on standoffs)
- CUSB-EMB-6 - 6' USB Cable with type A connector to mini USB connector for embedded applications (mates with P4 on the board)
- MP104-DIN - DIN rail mounting provision

Chapter 2: Installation

Software CD Installation

The software provided with this board is contained on one CD and ***must be installed onto your hard disk prior to use.*** To do this, perform the following steps as appropriate for your operating system. Substitute the appropriate drive letter for your drive where you see D: in the examples below.

WIN98/Me/2000/XP/2003

a. Place the CD into your CD-ROM drive.
b. The CD should automatically run the install program. If the install program does not run, click START | RUN and type D:\INSTALL, click OK or press Enter.
c. Follow the on-screen prompts to install the software for this board.

LINUX

a. Please refer to linux.htm on the CD-ROM for information on installing under Linux.

Hardware Installation

Please install the software package ***before*** plugging the hardware into the system. The board can be installed in any USB 2.0 port. Please refer to the USB I/O Quick Start Guide which can be found on the CD, for specific, quick steps to complete the hardware and software installation.

NOTES ABOUT HAND-SHAKING:

In order to transmit data you MUST ground the pulled-up /TX_EN (Transmit Enable) signal at PIN 19.
In order to receive data you MUST ground the pulled-up /RX_EN (Receive Enable) signal at PIN 17.
Please note: if connecting two USB-DIO-16H family devices together with a standard 1-to-1 cable these signals are correctly connected to RX_AF (Receive FIFO Almost Full) and TX_FE (Transmit FIFO Empty), respectively. If your application supports this kind of handshaking, connect those signals instead of simply grounding these pins.

Chapter 3: Hardware Details

Option Selection

You may also refer to the setup program installed from the CD provided with the board. There are two user selectable hardware options available. One is VUSB vs. VEXT which selects between USB powered or externally powered. The other is fused +5V output on I/O connector pin 34 or NC (no connect).

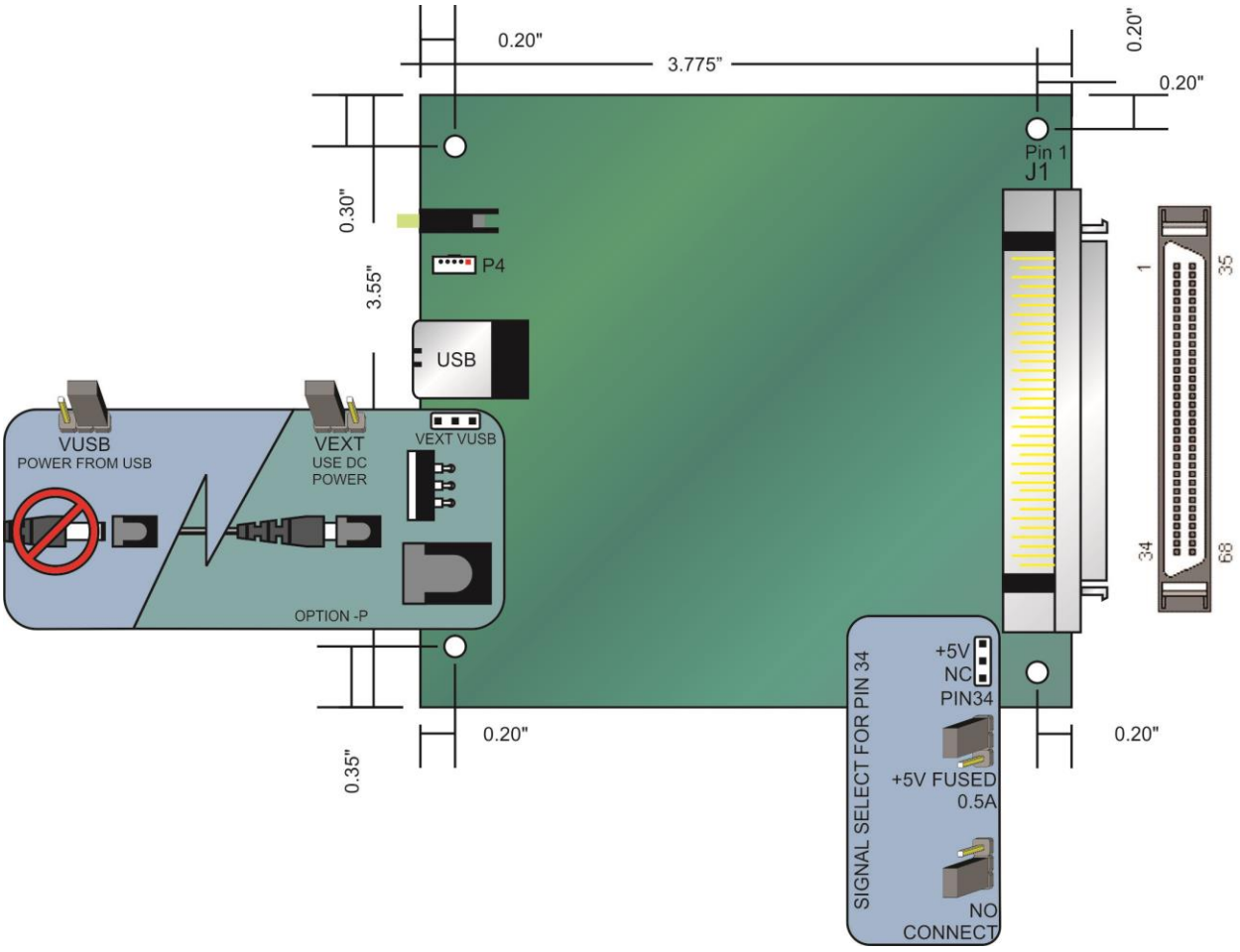


Figure 3-1: Option Selection Map

USB Connector (P1)

The USB connector available via the outside of the enclosure is a Type B and mates with the six foot cable provided. The USB port provides communication signals along with +5 VDC power. The board can be powered from the USB port or an external power supply can be used.

Embedded USB Connector (P4)

In applications where the OEM (board only, no enclosure) version of this board is used, it may be desirable to use the miniature USB connector onboard, which is next to and in parallel with the Type B connector.

LED

The LED on the front of the enclosure is used to indicate power and data transmissions. When the LED is in an illuminated steady green state, this signifies that the board is successfully connected to the computer and has been detected and configured by the operating system. When the LED flashes continuously, this signifies that there is data being transmitted over the USB bus.

FUSED +5V

The fused +5V jumper is located near J1. When installed in the +5V position the fused +5V will become available on pin 34 of J1. The fuse is rated at 500mA and is self resetting. If the jumper is installed in the NC position or is not installed, pin 34 is a no connect.

DC Power Jack (Optional)

Please note, not all boards will contain this option. This is an option for high current applications when more current is needed than what your computer can provide on the USB port (typically 500 mA). The DC jack has a 2.00mm post on board and is designed to be used with the 9 VDC AC/DC external power supply that ships with this option. The voltage regulator on board regulates the 9 VDC and provides 5 VDC to the onboard circuitry. When using external power, switch the jumper located near the USB connector to VEXT, otherwise when the jumper is in the VUSB position current is drawn from the USB port (please consult the option selection map for a visual reference).

If the board will be ordered as an OEM-P option (board only with external power) then the power input will be reconfigured (by removing the voltage regulator to prevent damage during shipment and while handling) such that the external power supply should then provide a regulated 5 VDC to the power jack instead of the standard 9 VDC.

68-PIN I/O Connector (J1)

The I/O is accessed via a 68-pin female SCSI Pin in Socket type connector with one-touch lock latches. Detailed pin assignments are listed in chapter 6, as well as a handy reference of pin functions printed on the enclosure label.

Chapter 4: USB Address Information

Use the provided driver to access the USB board. This driver will allow you to determine how many supported USB devices are currently installed, and each device's type. This information is returned as a Vendor ID (VID), Product ID (PID) and Device Index.

The VID for all four models is "0x1605", and the PID is unique for each model as follows:

USB-DIO-16H	PID = "0x800C",
USB-DI16A	PID = "0x800D",
USB-DO16A	PID = "0x800E",
USB-DIO-16A	PID = "0x800F"

Using the VID and PID as demonstrated in our sample programs will allow you to determine how many of these devices are attached to your system and the system-assigned "Device Index" for each of those devices. If you have more than one of any given PID you may encounter the need to "tell them apart" reliably, as the system-assigned Device Index is a plug-and-play value that changes as devices are plugged into and removed from your system.

In order to enable you to reliably tell devices with the same VID/PID apart we have implemented a user-accessible region of the onboard EEPROM. Refer to the Software Reference Manual for more information about the "CustomEEPROMRead" and "CustomEEPROMWrite" function for details. In general, you can simply write a unique value to a byte of EEPROM, then your software can track changes to the Device Index units by this value.

In addition, the user-accessible EEPROM is hundreds of bytes long, enabling numerous optional functionality of your choice.

See the Software Reference Manual, installed on your system along with the board support package, for more information.

Chapter 5: Programming

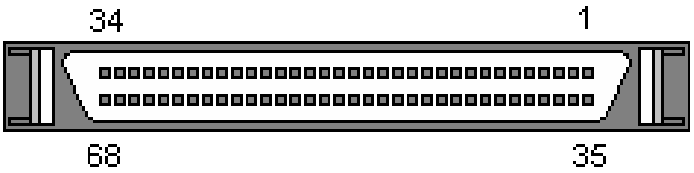
The driver software provided with the board uses a 32-bit .dll front end compatible with any Windows programming language. Samples provided in Borland C++Builder, Borland Delphi, Microsoft Visual Basic, and Microsoft Visual C++ demonstrate use of the driver.

Many functions are provided by the driver in Windows.

For detailed information on each function refer to the PDF Software Reference Manual located in the Win32 directory for this board.

Chapter 6: Connector Pin Assignments

The I/O connector is a 68-pin Type 2 SCSI female, right-angle type with quick-release “one-touch” locking latches, which protrudes through a cutout in the module’s enclosure.



Pin	Signal Name	Pin	Signal Name
1	DATA I/O 0	35	GND
2	DATA I/O 1	36	GND
3	DATA I/O 2	37	GND
4	DATA I/O 3	38	GND
5	DATA I/O 4	39	GND
6	DATA I/O 5	40	GND
7	DATA I/O 6	41	GND
8	DATA I/O 7	42	GND
9	DATA I/O 8	43	GND
10	DATA I/O 9	44	GND
11	DATA I/O 10	45	GND
12	DATA I/O 11	46	GND
13	DATA I/O 12	47	GND
14	DATA I/O 13	48	GND
15	DATA I/O 14	49	GND
16	DATA I/O 15	50	GND
17	Input Mode: /RX_EN Output Mode: TX_FE	51	GND
18	Input Mode: RX_CLK Output Mode: TX_ACK	52	GND
19	Input Mode: RX_AF Output Mode: /TX_EN	53	GND
20	Input Mode: RX_ACK Output Mode: TX_CLK	54	GND
21	DOI D0	55	GND
22	DIO C0	56	GND
23	DIO C1	57	GND
24	DIO C2	58	GND
25	NC	59	GND
26	DIO B0	60	DIO A0
27	DIO B1	61	DIO A1
28	DIO B2	62	DIO A2
29	DIO B3	63	DIO A3
30	DOI D1	64	DIO A4
31	DOI D2	65	DIO A5
32	NC	66	DIO A6
33	NC	67	DIO A7
34	FUSED +5V (jumper selected)	68	GND

TABLE 6-1: Connector Pin Assignments

Name	I/O	Description
DATA I/O 0 through 15	I/O	**16-Bit Data Bus** Input/Output Bit 0 through 15 (all pulled up through 10K ohms to 5V).
/RX_EN	I	**Receive Enable (active-low):** When /RX_EN is LOW, data is written into the FIFO on every LOW-to-HIGH transition of either RX_CLK or from the internal clock on board. When /RX_EN is HIGH, the FIFO holds the previous data. Data will not be written into the FIFO if it is full.
TX_FE	O	**Transmit FIFO Empty (active-high):** When TX_FE is HIGH, the FIFO is empty and further data reads from the output are inhibited. When TX_FE is low, the FIFO is not empty. TX_FE is synchronized to TX_ACK.
RX_CLK	I	**External Receive Clock:** When /RX_EN is LOW, data is written into the FIFO on a LOW-to-HIGH transition of RX_CLK, if the FIFO is not full. This signal is used when providing an external clock and not using the internal clock on board which is programmable from 1K-40MHz.
TX_ACK	O	**Transmit Acknowledge Clock:** When TX_ACK transitions from LOW-to-HIGH data can be read from the board. TX_ACK is the delayed clock used to read data out of the FIFO.
RX_AF	O	**Receive Almost Full Flag (active-high)**: When RX_AF is HIGH, the FIFO is 127 words from full.
/TX_EN	I	**Transmit Enable (active-low):** When /TX_EN is LOW, data is read from the FIFO on every LOW-to-HIGH transition of TX_CLK or from the internal clock on board. When /TX_EN is HIGH, the output register holds the previous data. Data will not be read from the FIFO if the TX_FE is HIGH.
RX_ACK	O	**Receive Acknowledge Clock:** When RX_ACK transitions from LOW-to-HIGH data has been read into the board. RX_ACK is the delayed clock used to read data into the FIFO.
TX_CLK	I	**External Transmit Clock:** When /TX_EN is LOW, data is read from the FIFO on a LOW-to-HIGH transition of TX_CLK, if the FIFO is not empty. This signal is used when providing an external clock and not using the internal clock on board which is programmable from 1K-40MHz.
GND	N/A	**Ground**
DIO A0 through A7	I/O	**Port A Digital I/O** Bit 0 through Bit 7 (all pulled up through 10K ohms to 5V)
DIO B0 through B3	I/O	**Port B Digital I/O** Bit 0 through Bit 3 (all pulled up through 10K ohms to 5V)
DIO C0 through C2	I/O	**Port C Digital I/O** Bit 0 through Bit 2 (all pulled up through 10K ohms to 5V) The direction of this port is the inverse of Port D's direction.
DOI D0 through D2	O/I	**Port D Digital O/I** Bit 0 through Bit 2 (all pulled up through 10K ohms to 5V) The direction of this port is the inverse of Port C's direction, which is why we named them O/I Bits.
FREQUENCY OUT	O	Optional Signal / Reserved
FUSED +5V	O	**Jumper Selectable 500mA Fused +5V**. This is an output ONLY! Do not apply +5V to this pin.
N/C	X	**Not Connected**

TABLE 6-2: Detailed Pin Descriptions

Chapter 7: Specifications

TTL High-Speed Digital I/O's (LVTTL Optional)

Lines	16, programmable as all inputs or all outputs ("16A" or "16H" only)	
Continuous Throughput	8MHz or 16MB/s maximum	
Input voltage	Logic low:	-0.5 to +0.8 VDC
	Logic high:	2.0 to 5.0 VDC
Input current	±20µA	(max)
Output voltage	Logic low:	0.44 VDC (max)
	Logic high:	4.4 VDC (min)
Output current	Logic low:	24 mA (max) sink
	Logic high:	24 mA (max) source

TTL Standard Digital I/O's (LVTTL Optional)

Lines	18, programmable as inputs or outputs as 4 Ports (A-D)	
Input voltage	Logic low:	-0.5 to 0.8 VDC
	Logic high:	2.0 to 5.0 VDC
Input current	±20µA	(max)
Output voltage	Logic low:	0.44 VDC (max)
	Logic high:	4.4 VDC (min)
Output current	Logic low:	24mA (max) sink
	Logic high:	24mA (max) source

Internal Clock

Frequency Range	1k - 40MHz

External Clock

Frequency	40 MHz maximum

Data FIFOs

Width:	16-bit	
Depth:	USB-DIO-16H:	128 kByte
	USB-DIO-16A:	8 kByte
	USB-DI16A:	8 kByte
	USB-DO16A:	8 kByte

Environmental

Operating Temperature	0° to +70°C, optional -40° to +85°C
Storage Temperature	-40° to +105°C
Humidity	5% to 95% RH, without condensation
Board Dimensions	PC/104 format, 3.550" by 3.775" and mounting holes
Box Dimension:	4.00 x 4.00 x 1.25 inches

Power

Power required	+5V at 100mA with no load typical[1]
Auxiliary power output	+5VDC via 0.5A resettable fuse at connector (jumper selectable)
	+3.3VDC via 0.5A resettable fuse (with LVTTL Option)

Power Modes

USB port	+5VDC provided via USB bus up to 500mA**
External supply	Optional external power adaptor @ 9VDC through DC power jack and on-board regulator to produce +5VDC

Factors that affect the amount of current required by this module include any power drawn from the auxiliary +5VDC through the resettable fuse, as well as any digital outputs that are wired to source current to external circuits (where the other side of the load is tied to ground / return). Any digital outputs wired to sink current to external loads will not necessarily increase the current drawn by the module, unless the positive side of the load is fed from the auxiliary +5VDC previously mentioned.

** If using more than a total of 500mA, use optional 9 VDC (on board voltage regulator outputs +5 VDC to card) external power supply and **remove** VUSB jumper and place jumper on VEXT. Then plug in external power before plugging into USB port. This option will give you a total of 1000mA available. *If using the OEM board with the power option (OEM-P) then use a regulated 5 VDC external supply as the on-board regulator will not be installed.*

[1] USB 2.0 spec defines a device in terms of a unit load. A unit load is defined to be 100mA. Devices drawing an absolute maximum of one unit load are considered to be low-powered and devices drawing an absolute maximum of five unit loads are considered to be high-powered. Because this spec is not strictly adhered to, it is best to verify the USB port's power capabilities before operation. This card, according to the USB 2.0 spec, is a high powered device. An optional external power supply can be ordered if the USB port cannot support high powered devices.

Appendix A: Theory of Operation

Unlike common 8255 based digital I/O, this unit provides a true synchronous, high-speed, buffered digital bus.

What does this mean for you? Well, you should understand the available, and sometimes necessary, handshaking signals on the connector associated with all that power.

You may wish to refer to the Block Diagram while reading this appendix.

In general, the entire high speed bus on any given unit can be treated as a single FIFO. Let's discuss only the output board (or the I/O board being used in output mode) for a moment:

Output
Four signals accompany the data on the output board: TX_ACK, TX_FE, /TX_EN, and TX_CLK.

It is convenient to think of these four signals in two pairs, "block" and "per word" handshaking lines.

TX_FE and /TX_EN are the "block" handshaking lines. TX_FE indicates the FIFO is Empty (it goes high when the FIFO is empty). /TX_EN indicates the receiving device wants to stop a moment; we'll get into *why* it might want to stop when we discuss the receive mode. Note, pulling /TX_EN LOW indicates GO. You need to raise the line HIGH to STOP transmission.

TX_CLK and TX_ACK are the "per word" handshaking lines. TX_CLK tells the FIFO *when* to transmit the next data word. Specifically, each rising edge of TX_CLK will assert the next word of data out of the FIFO. TX_ACK tells the receiving board that the FIFO has successfully asserted a word of data and it is ready to be read. These two signals are *almost* the same; TX_ACK is simply a slightly delayed version of TX_CLK, to ensure the data out of the FIFO has had time to stabilize, before the receiver tries to read it.

TX_CLK, as shown on the block diagram, can come from two sources: TX_CLK provided by the user to pin 20 on the connector, or the internal oscillator. If the internal oscillator is selected TX_ACK takes on even more importance, as the external device *must* synchronize to this signal. Theoretically the outside device could generate its own TX_ACK equivalent by simply waiting a moment after sending TX_CLK before reading the data, however, TX_ACK will not occur if /TX_EN is disabled (high). Because of these complexities it is best to design your device to use TX_ACK as the "read data now" signal.

To recap: if TX_FE is LOW data is available. If /TX_EN is LOW subsequently causing a TX_CLK will assert data and cause a TX_ACK, indicating the data has finished being asserted and is safe to read. This process will continue until TX_FE goes high, or /TX_EN is asserted high, indicating no more data is available, or the receiver needs to stop for some reason.

The receiver unit has a similar and functionally correlated set of handshaking signals associated with its high-speed bus.

Input
The four handshaking lines on the input board (or the I/O board in input mode) are: RX_ACK, RX_AF, RX_CLK, and /RX_EN.

It is convenient to think of these four signals in two pairs, “block”, and “per word” handshaking.

RX_CLK and RX_ACK are the “per word” handshaking lines. RX_CLK tells the receiving FIFO to latch in the data currently on the bus pins. RX_ACK tells the transmitting board that the data has been successfully latched. Again, RX_ACK is simply a slightly delayed form of RX_CLK, to ensure the FIFO has had time to digest the information.

RX_AF and /RX_EN are the “block” handshaking lines. RX_AF indicates the receive board’s FIFO is almost full. This line will usually be low, but if the digital bus data rate exceeds the USB bus’ and/or computer system’s ability to drain the FIFO, the FIFO may begin to fill. This line will go HIGH to indicate “slow down!”. *This* is *why* the receiving board may need to indicate “Stop” to the transmitting board. Conversely, /RX_EN indicates to the receiver board that the source of data needs to pause a moment. If the transmitter were supplying the clock, it could simply pause the clock, but if the receive card is using an internal clock, /RX_EN tells the receiver *when* to pause that clock. Please note, RX_AF is asserted well before the FIFO actually fills, giving your circuitry time to respond to the signal without losing data.

As mentioned briefly, and in identical fashion to the TX_CLK, RX_CLK can be supplied either from a pin on the connector or from onboard oscillator.

To recap: if /RX_EN is LOW the receiver will pay attention to RX_CLK rising edges by latching in data, then indicating success by asserting RX_ACK. If the receive FIFO begins to fill, the receiver will assert RX_AF.

Paired Operation:
We’ve carefully designed these units to work together using an off-the-shelf one-to-one cable, so no “gender changer” or “null-modem” interface is necessary. The I/O unit contains all the circuitry needed to adapt its handshaking pinout to the current direction of its I/O bus, eliminating all the hassles of common equipment.

The block diagram shows a typical application example, and indicates the pairing of handshaking signals.

TX_ACK can be used to supply the RX_CLK on pin 18 of the connector if desired; or you can use the internal RX_CLK source.

TX_FE supplies the /RX_EN signal; when the transmit FIFO is empty, the receiver card stops paying attention to its RX_CLK, even if it’s using internal.

Similarly, RX_AF is used as /TX_EN; whereby the transmit board ignores its TX_CLK signal when the receiver FIFO is almost full.

And last, the RX_ACK signal can be used to supply TX_CLK on pin 20 of the connector, or TX_CLK can be generated internally.

Warning about bus contention
Please note you should never connect an output unit to an output unit, nor an input unit to an input unit, to avoid bus contention and potentially destroying your equipment. (Even when used as inputs, the bus has handshaking pins configured as output, so these handshaking pins would be in a contention situation.) Similarly, if you're using the selectable I/O unit, avoid ever having both sides of the cable configured as input or output at the same time.

The same warning applies to the slow-speed digital I/O ports shown on the block diagram; the next section will discuss methods of avoiding this kind of contention.

Appendix B: Proposed I/O Negotiation Algorithms

NOTICE:

For most customers there will be no need to allow the end-user to select I/O directions; you will simply write your software to hard-code your digital ports into specific directions at initialization and know confidently that the software written for the connecting device will safely configure *its* ports in a compatible manner.

However, during development in the lab, or for those customer's that cannot know how the end-user will desire the ports to be configured, it may be useful to explore implementing a handshaking protocol to minimize the opportunity for error.

In any application where two flexible devices are going to be connected together it is important that both devices are configured such that no output drivers are attempting to drive another devices output pins. That is, outputs on one device must only be connected to inputs on the other device.

Because this module is capable of flexibly selecting the input-vs-output direction of its various digital ports at run-time, some sort of "handshaking" protocol may be useful, depending on how thoroughly hard-coded you wish to make your software application.

This USB module includes hardware to support such a protocol, without requiring its use; this provides the maximum flexibility and device I/O capability. Digital Port "A" can be used to provide basic handshaking, or it can be used as an 8-bit output or 8-bit input port, as your specific application requires.

The following pages will describe several possible ways to initialize the ports. We'll first cover the typical and most common "embedded" case, where Port A is not being used as a handshaking port, and all configuration is hard-coded into the controlling software. Then we'll consider three alternate situations, and algorithms for handling them.

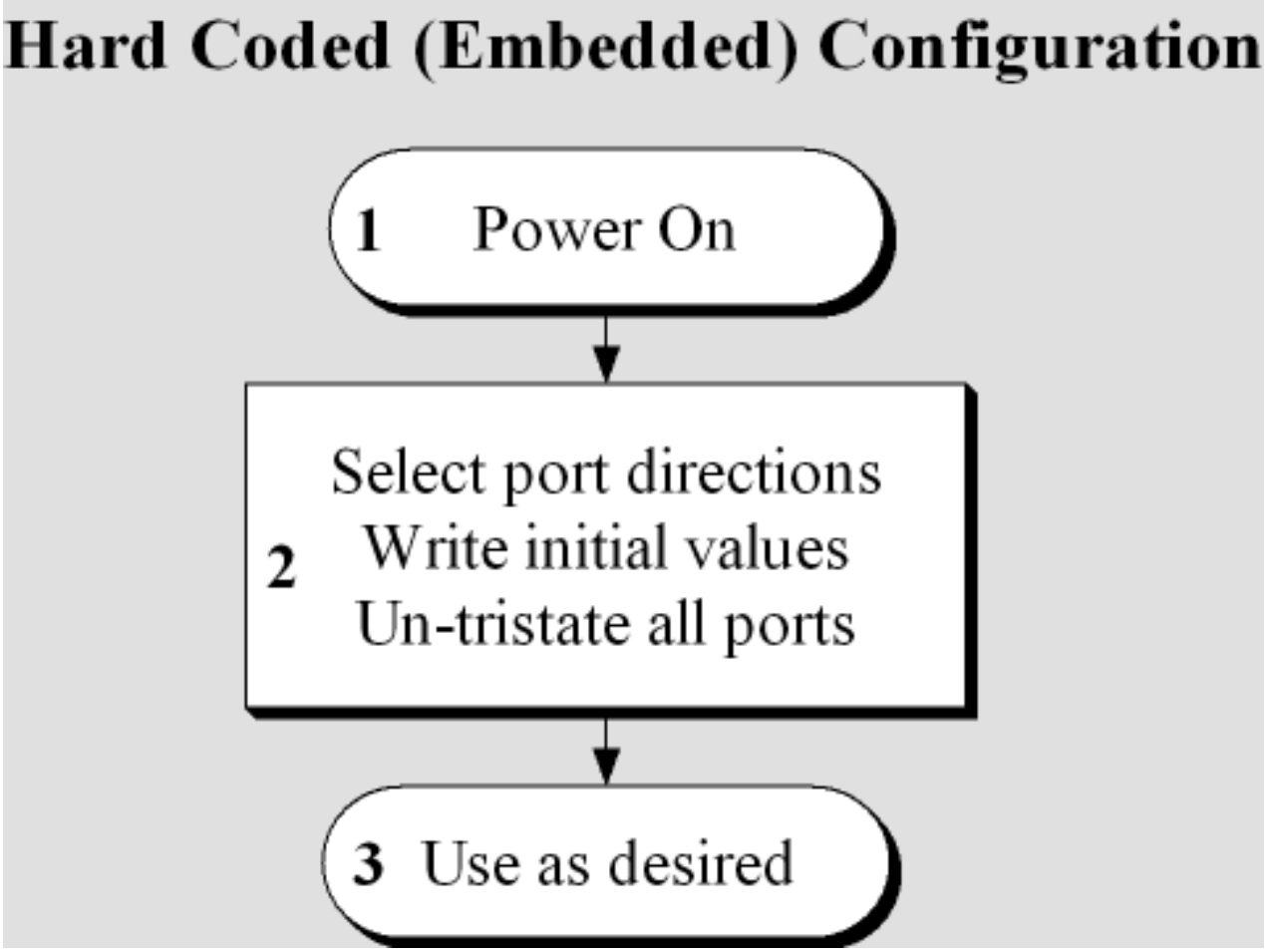
When reading the following flowcharts, please note the small number located in each box, and refer to the numbered paragraphs following the four flowcharts for detailed information about each step.

Several important concepts will be useful before reading these flowcharts:

"Master" means a unit that has been designated as controlling the direction of the ports on both itself and any device connected to it.

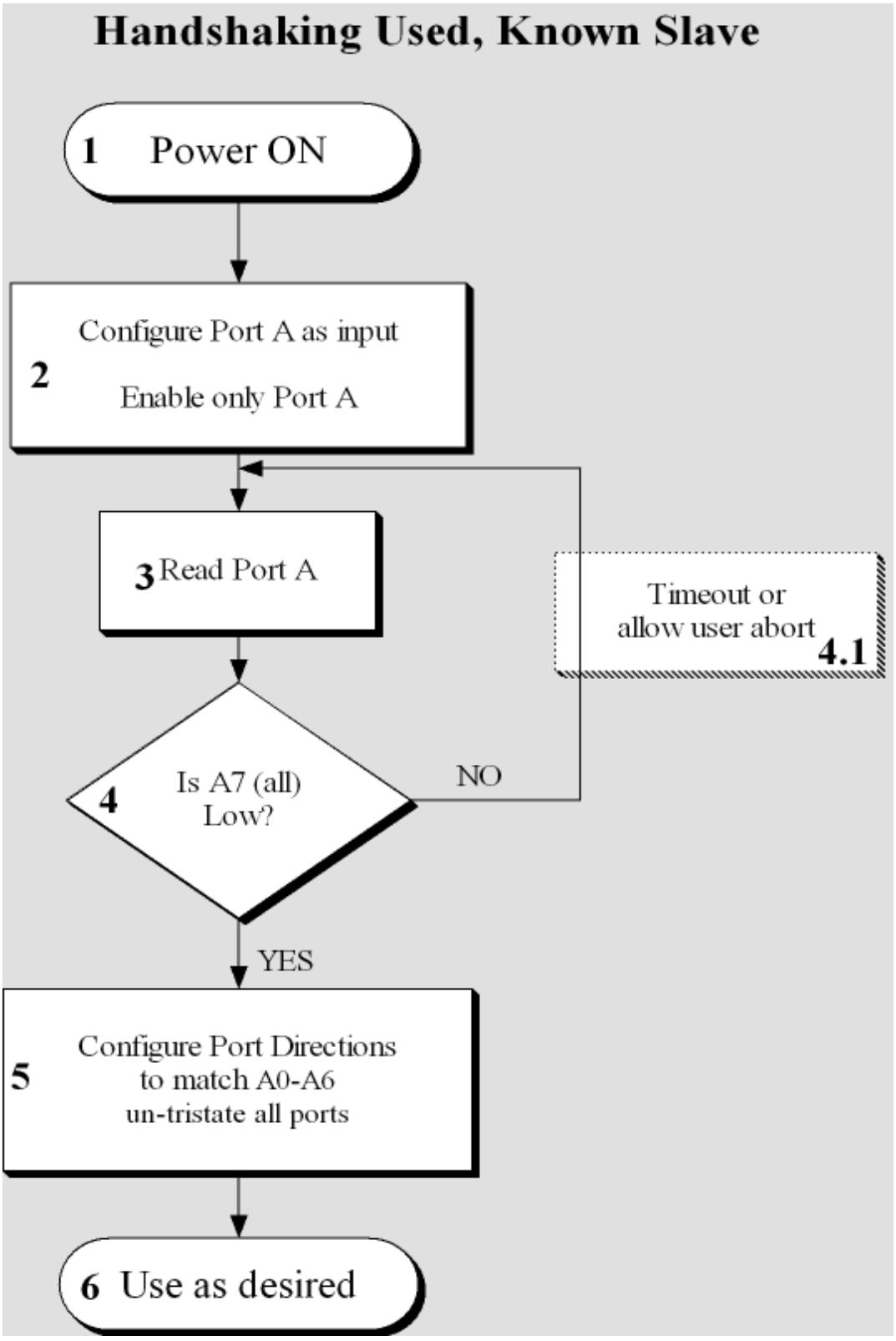
"Slave" means a unit that only sets the direction of its ports based on the directions of a connected "Master" unit.

"Port A" is the 8-bit digital input/output port we're using to communicate this handshaking information.
Why port A? Because only it can be enabled while leaving all the other ports tristated.



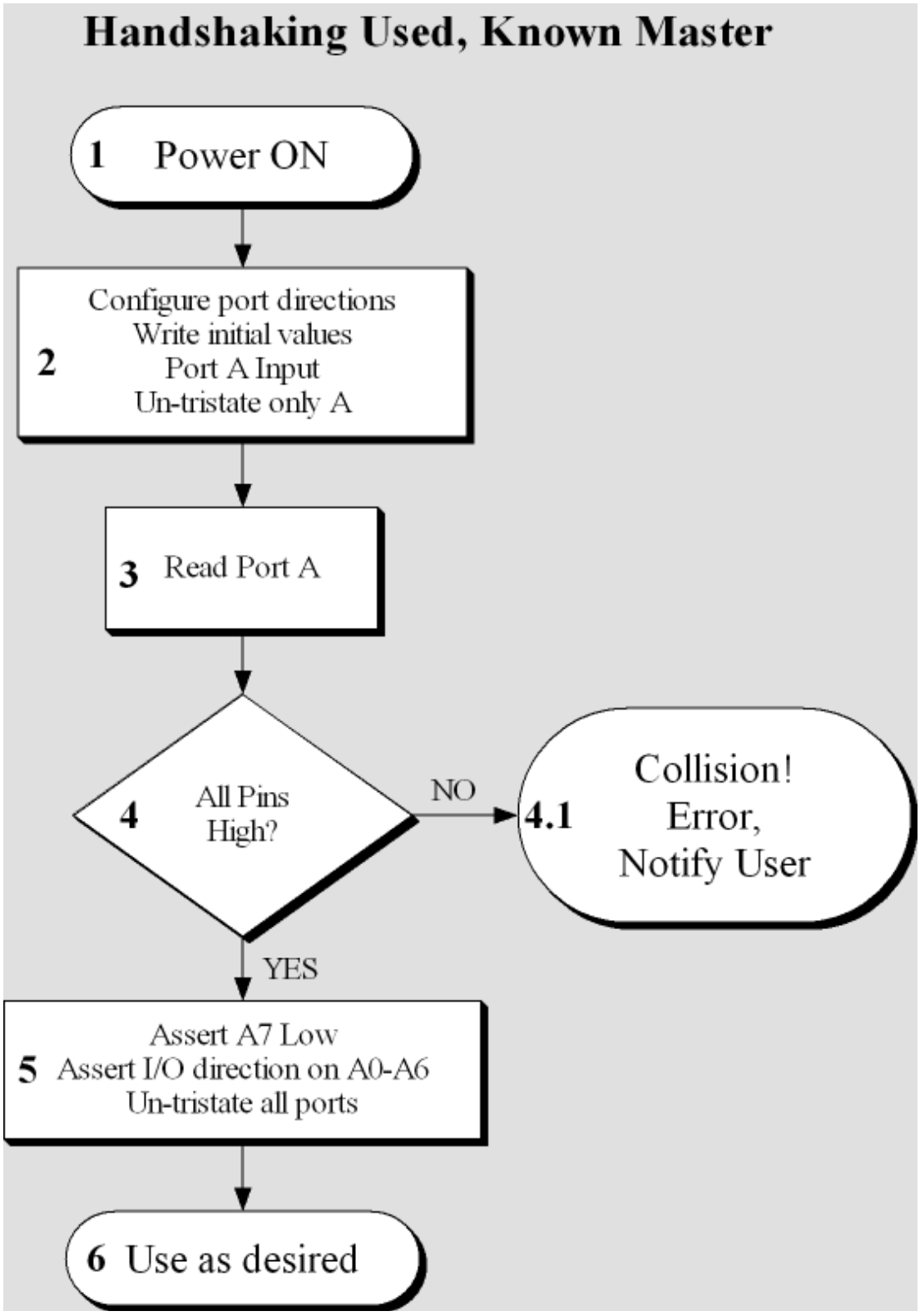
Hard Coded (Embedded) Configuration

1) Power On - at this point the unit will be tristated. All I/O lines at the connector will have +5V from 10k ohm pullups.
2) Using DIO_Configure, set PORT A, B, C, and D for their desired directions (remembering that PORT C and D must always be opposed); select the initial values for any ports configured as outputs, and untristate all the ports. Remember this mode requires either that the software for the connected device be written to match these port directions, that the connected device is simply hardwired to match these port directions, or that you're using a cable that doesn't make these connections. If you connect an output on this device to an output from your connected device, you may destroy one or both devices and will certainly void your warranty.
3) Done. When you're embedded or hardcoded there aren't many steps needed. The unit is now configured and ready for use. Any combination of DIO_Readxxx and DIO_Writexxx functions may be used to access the slow digital bits, and the DIO_Streamxxx functions can be used to perform fast, buffered I/O.


Handshaking Used, Known Slave
1 Power ON
Configure Port A as input
2
Enable only Port A
3 Read Port A
Timeout or
allow user abort
4.1
4
Is A7 (all)
Low?
NO
YES
Configure Port Directions
5
to match A0-A6
un-tristate all ports
6 Use as desired

Handshaking Used (Known Slave)

1) At this point the unit will be tristated. All I/O lines at the connector will have +5V from 10k ohm pull-ups.
2) Using DIO_Configure, set PORT A as an input, and untristate ONLY port A. The direction settings for ports B-D don't matter, as they will remain tristated. Similarly, the initial output values are also irrelevant.
3) Use DIO_Read8, or DIO_ReadAll, to read the bits from Input Port A. Remember the state of all the bits, as they will be used later.
4) Check the value you read from Port A to see if bit 7 is "low", or zero. Using bit 7 here is arbitrary – you could write your software to use any bit, or even several bits. This algorithm uses bit 7 to indicate "a master has asserted itself". If bit 7 is NOT LOW, the connected master has not yet gotten around to presenting its handshaking information on these pins. Follow the "NO" branch of the flowchart. (Please note, for enhanced error detection it would be useful to detect a situation where Bit 7 was high, but one or more bits on the port were low, and flag this as an error condition.) If bit 7 IS LOW, it indicates the connected master device has presented its handshaking configuration on the pins, and we can continue to the "YES" branch of the flowchart.
    a. In a real-world environment it is unwise to code loops that may never exit. This loop is one such case. If the cable has become disconnected your software would forever wait for Port A7 to go low, and it never would. We therefore recommend either timing-out from this loop, or giving the user the option to abort the detection process–or both!
5) Using the DIO_Configure API you can select the direction for all the I/O ports to match the value you read on Ports A0-A4, write initial values to all of them (any port selected for input will ignore its configured output value), and untristate all the ports. A word about matching the value: the master will assert one bit for each port. Port "A" is an output, by definition, because the master is using it to communicate the other bits to us. This leaves Port B, Port C, Port D, and the fast DI/O Port as possible direction information the master needs to communicate to us. Because Port C and Port D are guaranteed to always be opposite direction, you could skip using a pin for Port D. If you are not using the unit with the fast port having selectable direction you could skip that, as well. For the sake of this algorithm, we recommend not to skip any ports. Port A0 indicates the direction of Port A, bit A1 is associated with Port B's direction, bit A2 will indicate Port C, bit A3 is for Port D, and bit A4 is the direction of the fast DI/O bus. Doing so allows you to code more error detection into your algorithm: if bit A2 is not the opposite of A3, there's an error. If A0 does not indicate the master is using it as an output, its another error, etc. It is also for this reason we recommend using "low" as "output". Doing so guarantees that bit A0 should be low, as well as bit A7 - furthering your ability to detect cable-breaks, etc. So, to finish: if your reading of Port A tells you what inputs and outputs the master is configuring, simply configure your ports to be the opposite; if bit A2 is low indicating Port C is an output, make sure you configure the slave's Port C as an input.
6) The unit is now configured and ready for use. Any combination of DIO_Readxxx and DIO_Writexxx functions may be used to access the slow digital bits, and the DIO_Streamxxx functions can be used to perform fast, buffered I/O.


Handshaking Used, Known Master
1 Power ON
Configure port directions
Write initial values
2 Port A Input
Un-tristate only A
3 Read Port A
4 All Pins High?
NO
4.1 Collision! Error, Notify User
YES
Assert A7 Low
5 Assert I/O direction on A0-A6
Un-tristate all ports
6 Use as desired

Handshaking Used (Known Master)

1) At this point the unit will be tristated. All I/O lines at the connector will have +5V from 10k ohm pull-ups.
2) Using DIO_Configure, set PORT A as an input, and untristate ONLY port A. The direction settings for ports B-D don't matter, as they will remain tristated. Similarly, the initial output values are also irrelevant.
3) Use DIO_Read8, or DIO_ReadAll, to read the bits from Input Port A. Remember the state of all the bits, as they will be used later.
4) Check the value you read from Port A to see if bit 7 is "low", or zero. Using bit 7 here is arbitrary – you could write your software to use any bit, or even several bits. This algorithm uses bit 7 to indicate "a master has asserted itself". If any bit is LOW, move to the "NO" branch of the flowchart; if all bits on input port A read HIGH, move to the YES branch of the flowchart.

a) Since this device is running as a dedicated master, seeing bit 7 low means something has gone horribly wrong, and you need to abort the process before you set this port as an output. For better error detection, you should check all the bits to make sure none of them are asserted low; any bits that read low are either grounded at the connector or broken, in either case you should abort before using the unit.

5) Now that you've detected no collision on Port A you should change Port A to output, and write the handshaking pattern. We're using the same bit pattern from the previous flowchart; Port A, bit 0 indicates the direction of port A, and LOW means output. So, bit 0 should always be asserted low under this algorithm. (Your software can do it however you'd like). Bit A1 will reflect port B's direction, A2 Port C, A3 Port D, and A4 the fast DI/O bus. Remember that port C and D must always be configured for opposite directions (and if you attempt to set C and D the same way, "C" will be set to the opposite of "D" – that is, the configuration for port C is ignored.) Additionally set the bit 7 "Master" bit low to indicate you've completed asserting your handshaking pattern. And, since it is all in one DIO_Configure command, its most efficient to select the direction of the ports, write initial values to all of them, and untristate all of them in this single step.
6) The unit is now configured and ready for use. Any combination of DIO_Readxxx and DIO_Writexxx functions may be used to access the slow digital bits, and the DIO_Streamxxx functions can be used to perform fast, buffered I/O.

Customer Comments

If you experience any problems with this manual or just want to give us some feedback, please email us at: ***manuals@accesio.com***. Please detail any errors you find and include your mailing address so that we can send you any manual updates.

ACCES I/O PRODUCTS, INC.
10623 Roselle Street, San Diego CA 92121
Tel. (858)550-9559 FAX (858)550-7322
www.accesio.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP