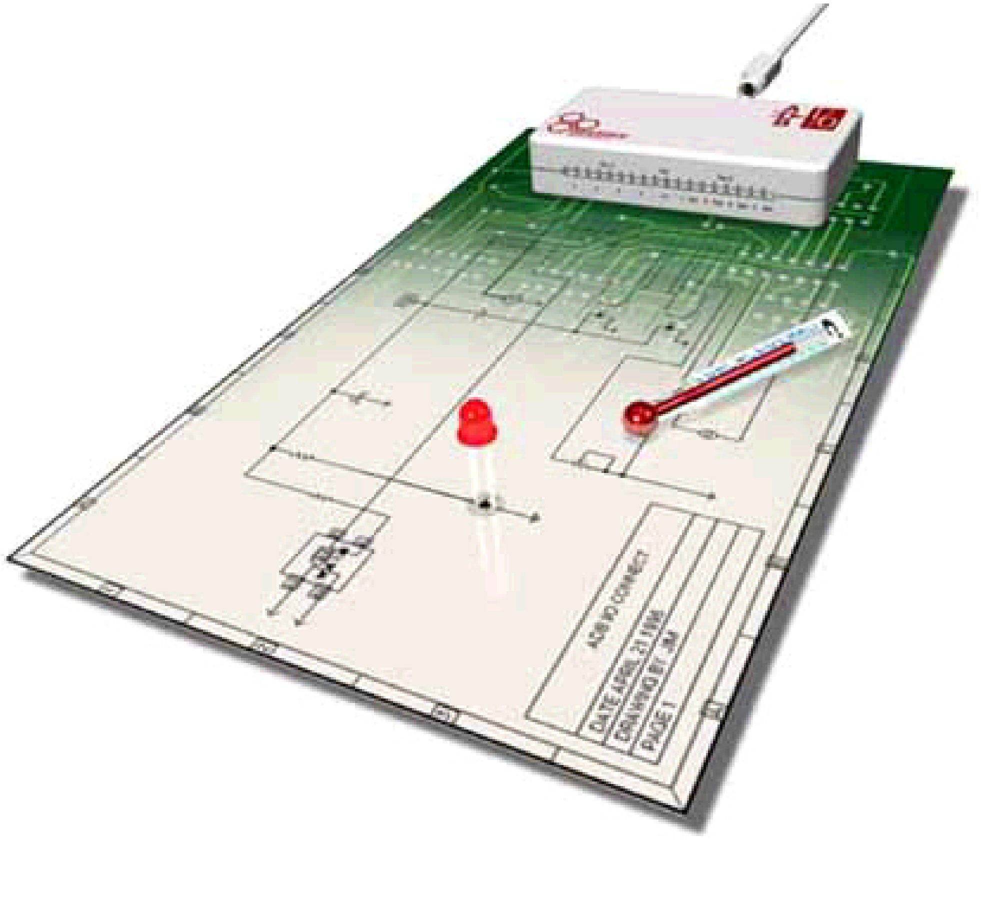
ADB I/O Manual



BEEHIVE
TECHNOLOGIES INC.

First Edition June 1996

LIMITED WARRANTY

BeeHive Technologies, Inc. warrants ADB I/O against defects in materials and workmanship for a period of ninety (90) days from the date of original purchase. If you discover a manufacturing defect, BeeHive Technologies, Inc. will repair your product with either new or refurbished replacement parts at no charge. If BeeHive Technologies, Inc. is unable to restore your ADB I/O to good working order after a reasonable number of attempts, BeeHive Technologies, Inc. will, at its option, replace or refund the purchase price of the ADB I/O. To obtain service under warranty you must obtain a Return Merchandise Authorization (RMA) number from BeeHive Technologies, Inc. or its service provider. Products must be returned to BeeHive Technologies, Inc. or its service provider with shipping charges prepaid and must be accompanied with a brief description of the problem as well as proof of purchase.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, WHETHER ORAL OR WRITTEN, EXPRESS OR IMPLIED. BEEHIVE TECHNOLOGIES, INC.SPECIFICALLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATIONS, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

BEEHIVE TECHNOLOGIES, INC. IS NOT RESPONSIBLE FOR DIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY BREACH OF WARRANTY, OR UNDER ANY OTHER LEGAL THEORY, INCLUDING BUT NOT LIMITED TO LOST PROFITS, DOWNTIME, GOODWILL, DAMAGE TO OR REPLACEMENT OF EQUIPMENT AND PROPERTY.

FEDERAL COMMUNICATIONS COMMISSION RADIO AND TELEVISION INTERFERENCE STATEMENT FOR A CLASS "B" DEVICE

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications.

User Instructions:

If the equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

Reorient or relocate the receiving antenna.

Increase the separation between the equipment and receiver.

Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

Consult the dealer or an experienced radio/TV technician for help.

Changes or modifications not expressly approved by BeeHive Technologies, Inc. could void the user's authority to operate the equipment.

NOTE: This product was FCC certified under test conditions that included the use of shielded I/O cables and connectors between system components. To be in compliance with FCC regulations, the user must use shielded cables and connectors and install them properly.

This device complies with Part 15, of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received including interference that may cause undesired operation.
FCC ID: MSL001ADBIOV001

The ADB I/O was tested in accordance with the International Electrotechnical Committee (IEC) publications 801 and 1000-4 series standards for information technology equipment (ITE), as required by the European Union harmonized document EN 50082-1 (1992).

CONTENTS

ABOUT THIS MANUAL

This manual contains the information you need in order to install and use the ADB I/O.

You can always find the most current information regarding ADB I/O updates as well as application examples on the web. To get to the ADB I/O pages point your web browser at the BeeHive Technologies, Inc. home page:

http://www.bzzzzzz.com

and follow the links.

If you have questions or problems you can contact BeeHive Technologies, Inc. or the store where you bought it.

BeeHive Technologies, Inc.
E-mail: Contact@mail.bzzzzzz.com
URL: http://www.bzzzzzz.com
Phone: USA (818) 304-0664
Facsimile: USA (818) 568-1530

You can also get direct technical support by sending e-mail to:

ADB_IO_Support@mail.bzzzzzz.com

and we maintain an ADB I/O users’ discussion list at:

ADB_IO_Discussion@mach2.bzzzzzz.com

You can join the discussion list by just sending a message to the list with

subscribe

as the subject.

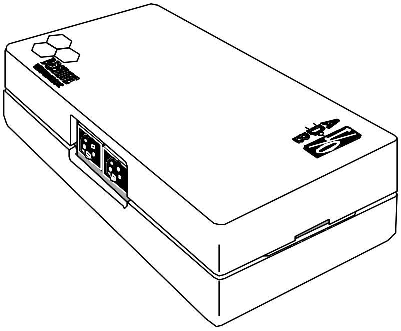
Finally, we encourage third party developers and will be happy to work with you during the development, marketing and sale of add-on products. Just send us an e-mail addressed to contact@mail.bzzzzzz.com and let us know what you have in mind.

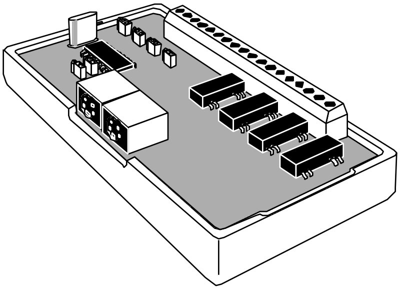
INTRODUCTION

Welcome to ADB I/O

ADB I/O is a peripheral for the Macintosh which allows you to control electronic devices and read sensor data. ADB I/O attaches to your Macintosh via the ADB (Apple Desktop Bus). You control the ADB I/O either by writing your own software or by using some of the simple interface software included with this product.

The possible applications of ADB I/O are almost endless, but to list just a few: Computer controlled science experiments in schools or at science fairs; control of electronic devices in your home; access control in office buildings; control of spotlights, audio equipment or props in multi-media presentations; construction of complete weather stations by attaching wind, temperature, humidity, light sensors and barometric sensors.





The included software will quickly get you up and running. You can use HyperCard, AppleScript, C, ACI 4th Dimension, MacroMedia Director to develop your own custom control software.

- ADB I/O can be used with all Macintosh computers equipped with the Apple Desktop Bus and which are running system 7 or higher.

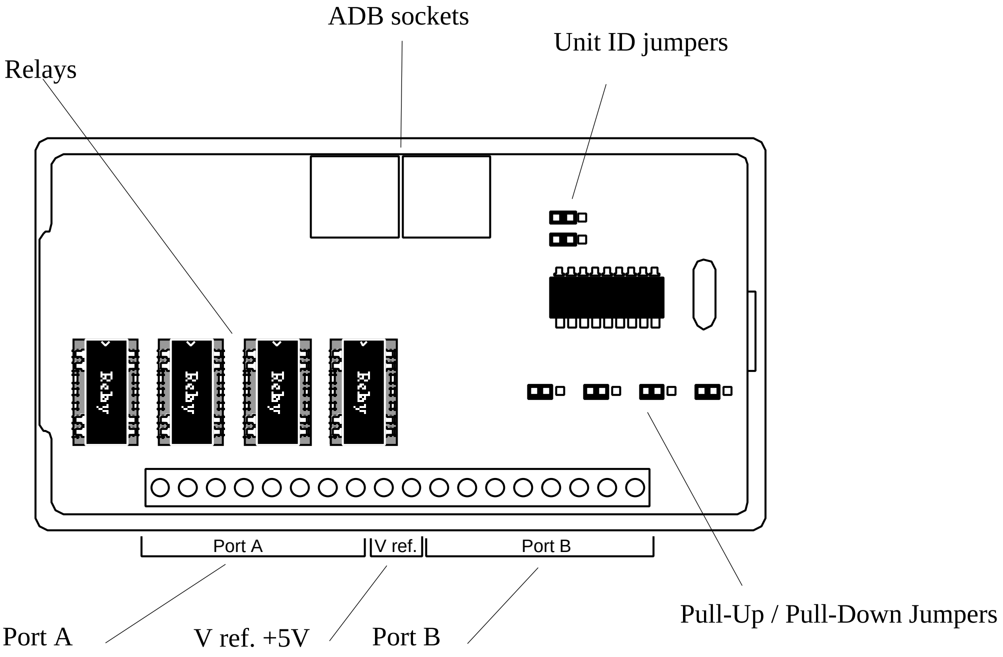
- ADB I/O attaches to your Macintosh via an ADB cable (a 6’ or 1.8m cable is included with the unit) .

- No external power supply is needed as ADB I/O gets its power from the ADB.

- You can attach electronic devices to the ADB I/O using the screw terminals which are made accessible by prying off the ADB I/O cover.

- If you need more ports than one ADB I/O unit provides, you have the option of attaching up to four units on any Macintosh.

ADB I/O Overview



Relays
The ADB I/O is equipped with four socket mounted relays. See chapter 2 for more information.

ADB Ports
Each ADB I/O has two ADB sockets which lets you daisy chain ADB devices. See chapter 1 for more information about hooking up the ADB I/O to your Macintosh.

Unit ID Jumpers
If you are going to attach more than one ADB I/O to a single Macintosh you need to set the unit ID using the Unit ID jumpers. See chapter 1 for detailed information.

Port A
Port A is a screw terminal block with four pairs of connectors. These connectors are hooked into the relays which each can carry 10W, 100VDC at a maximum of 0,5A. The relays are socket mounted and can be substituted by optoisolators if needed. The relays can be programmed to incorporate a delay of from 0 to 25.5 seconds in 0.1 second increments. See chapter 2 for more information.

Vref +5V
This is a two-pole screw terminal block hooked into a current limiting resistor. Vref is current limited to 18mA. See chapter 2 for additional information.

Port B

Port B has a screw terminal block with four pairs of connectors. Each pair of connectors consist of a signal and a ground terminal. The four channels can be configured as either four TTL compatible inputs or outputs or they can be configured to measure analog voltages between 0V and +5V or a reference voltage between +3 and +5V. The reading is accurate to 8 bits of precision or 256 data values. See chapter 2 for additional information.

Pull-Up / Pull-Down Jumpers

The Pull-Up / Pull-Down Jumpers are used to connect a 10kΩ resistor on each of the channels of port B to +5V, 0V or an open circuit. This is useful when you want to connect push-buttons or contacts to the channels. The jumper should be in the open circuit position when the channel is configured as analog input. See Chapter 2 for additional information.

1. HOOKUP AND INSTALLATION

Hooking up ADB I/O

WARNING!
Never connect ADB I/O to a running Macintosh. Such hot-swapping of ADB components can cause serious damage to your computer and will result in unpredictable ADB I/O performance.

This is how you attach an ADB I/O unit to your Macintosh

1. Shut down the Macintosh.

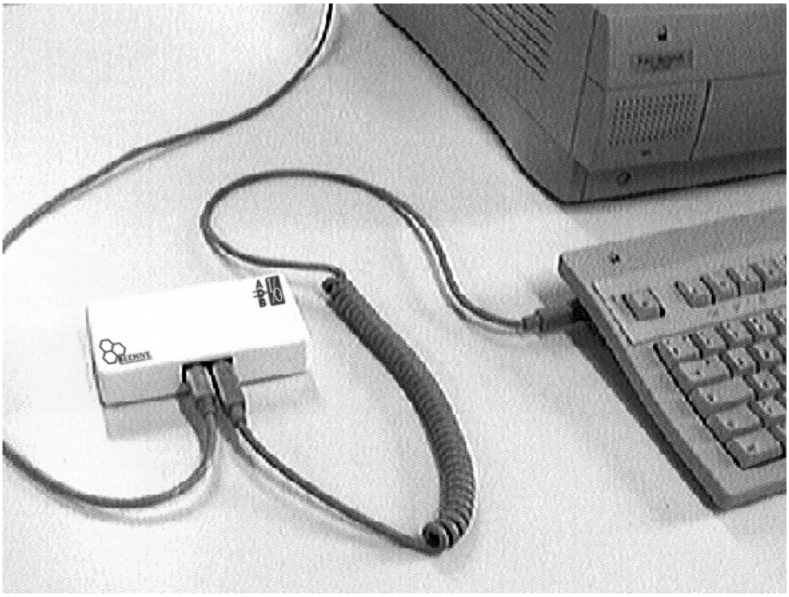
2. Disconnect the keyboard cable from the ADB port at the back of the computer.

3. Connect the keyboard cable to one of the two ADB ports in the ADB I/O.

4. Connect one end of the ADB cable supplied with the ADB I/O to the other ADB port in the ADB I/O.

5. Connect the other end of the supplied cable to the ADB port at the back of the computer where the keyboard was connected.

The ADB I/O is now ready for use.



ADB I/O software installation

1. Turn on your Macintosh.

2. Insert the *ADB I/O Software* disk in the disk drive.

3. Double-click the *ADB I/O Software Installer* icon.

4. Use the dialog which pops up to specify where you want the software installed.

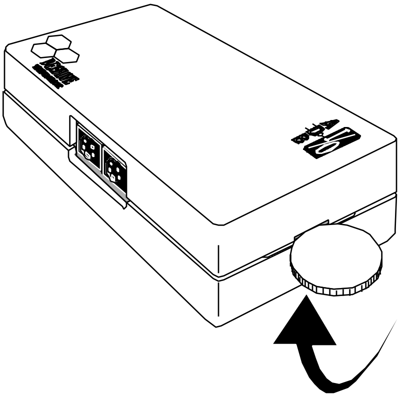
5. Open the *ADB I/O Software* folder and double-click the *Read Me* file. This file contains additional information about the software.

ADB I/O Software

ADB I/O Software Installer

Opening ADB I/O

You need to open the ADB I/O when attaching or disconnecting devices. To open the case simply insert a coin or wide screwdriver blade into the slot at either end of the case and give it a twist.



Attaching multiple ADB I/O units to one ADB chain

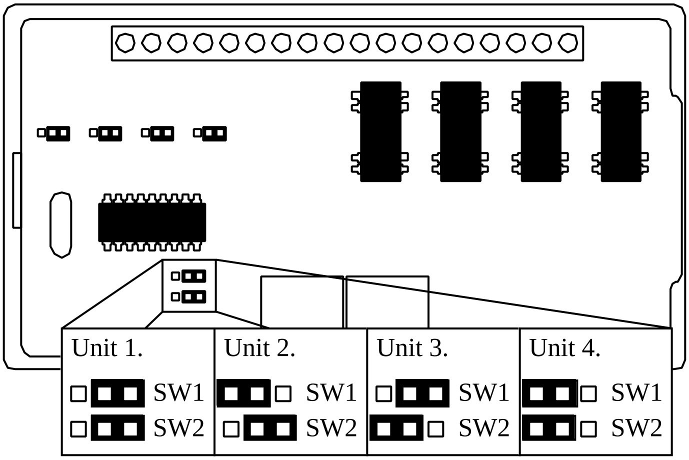
You can attach up to four ADB I/O units to a single Macintosh. The units attach in series on the ADB chain. The unit ID jumpers are used to give each ADB I/O unit a unique ID so that they can be individually addressed by the computer. Each unit is delivered with the unit ID preset to 1, but you can change that to any number from 1 - 4. To change the unit ID carry out the following steps:

1. Turn off the Macintosh.

2. Open the ADB I/O case.

3. Use a pencil or a similar device to pry up the unit ID jumpers. Replace the jumpers so that a unique ID is assigned to each ADB I/O unit on the ADB chain. The order of the unit IDs on the chain makes no difference, but each unit must have a different ID. The picture below shows how jumper positions correspond to ID numbers.

4. You need to remember which ID is assigned to each ADB I/O and you may want to attach a label with the ID to the bottom of each unit.



This figure shows how different jumper settings correspond to unit ID assignments.

The unit ID is used when your software addresses an ADB I/O to control a device or to make a measurement. The following is an example of how to set ADB I/O unit 2, port A, channel 1 to ”high”. That causes the relay on channel 1 to close.

Example in AppleScript: `set ADBIO unit 2 port A channel 1 to high`
Example in HyperTalk: `setADBIO 2, A, 1, high`

If two or more ADB I/O units on the ADB chain have the same unit ID, the software will return an error when any ADB I/O unit is addressed.

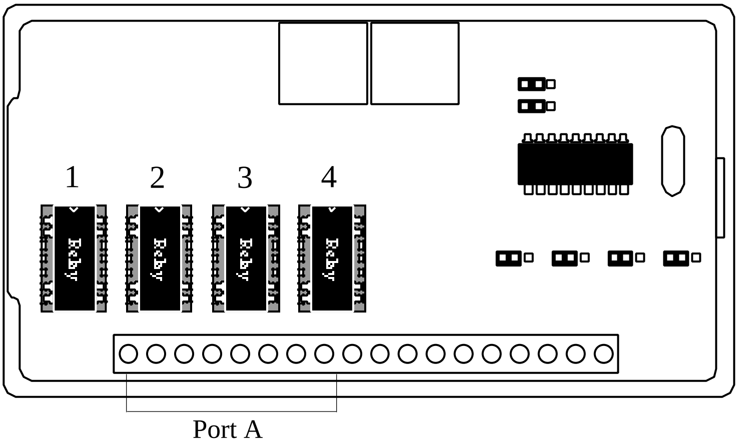
2. ATTACHING DEVICES TO ADB I/O

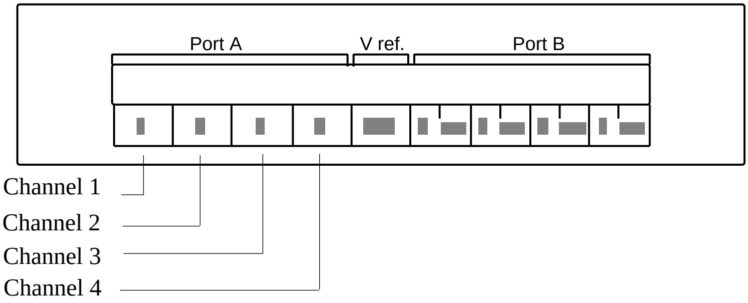
This section describes how to attach devices to the ADB I/O.

WARNING!
Always make sure that your Macintosh is turned off when attaching or removing devices from the ADB I/O.

Attaching a device to ADB I/O port A.

ADB I/O port A is a digital input-output port intended for devices which need to be electrically isolated from the Macintosh. Electrically isolated means that there is no electrical connection between the devices attached to port A and the Macintosh. Electrical isolation is required when attaching ADB I/O to devices which use higher voltages than +5 V. Electrical isolation may also be needed when long cables are attached and there is risk for a floating ground. Electrical motors often induce electrical noise and such noise is prevented from reaching your Macintosh by the electrical isolation.





Port A has four socket mounted relays. You can use the sockets to replace the relays with optoisolators. The socket is connected to two of the terminals in the screw terminal blocks. ADB I/O is delivered with 4 relays. Upon startup each channel in port A is set for output and the output is set for ”low”.

- The relays can be used to control electrical devices, turning them on and off. You use software to set the channels in port A to ”high” or ”low”. To close the relay you set the channel to ”high”. To open the relay you set it to ”low”. Each channel can optionally be set ”high” or ”low” with a time delay. The time delay can be set anywhere from 0 to 25.5 seconds in increments of 0.1 second.

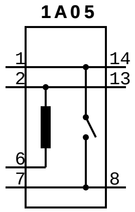
- An optoisolator can be used to detect the on-off state of a signal. You use the ADB I/O software to read the ”high” or ”low” state of the channels. When a channel is ”high” it means that the optoisolator LED is turned on, when it reads ”low” it is off.

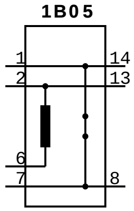
The Relays

Each relay is connected to two terminals in the screw terminal block. The relays are reed-relays and are normally open. Reed relays can not carry high voltages or current.

Maximum power: 10VA
Maximum voltage: 100V direct current
Maximum current: 0.5A

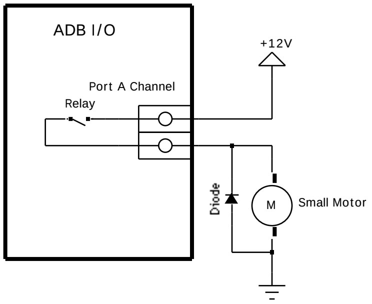
You can replace the relays with normally closed relays if needed.





WARNING!
Never attach 110V devices to the relays. The relays are not designed to directly control household voltage/current devices or appliances. If you want to control household appliances you should do so by having the ADB I/O control an external relay designed for such loads.

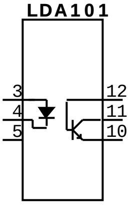
One example of how to use the relays in ADB I/O is in the control of an electric motor as shown in the diagram to the right. Note the use of a diode mounted in parallel with the motor. High current spikes are often produced when switching inductive loads such as motors, relay coils etc.. These spikes can damage the relays in ADB I/O. With a diode connected across the load as shown, these spikes are short circuited, thereby protecting the relay. A wide variety of diodes can be used, but one example is the 1N4148.



The following code example puts ADB I/O unit 1, port A, channel 1 to high for 10 seconds. The result is that the relay on that channel is closed for 10 seconds and then opens again.

Example in AppleScript: `set ADBIO unit 1 port A channel 1 to high for seconds 10`
Example in HyperTalk: `setADBIO 1, A, 1, high, 10`

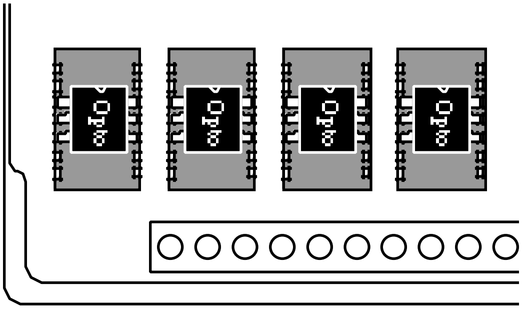
Optoisolators



When you want to detect "on" or "off" at voltages above +5V you may want to use an optoisolator in place of the standard relay.

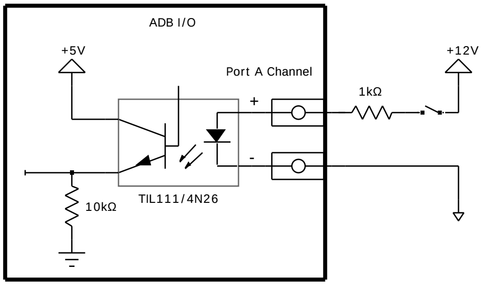
ADB I/O is only delivered with relays so you will have to get optoisolators if you plan to use them. A few suggested optoisolators which are compatible with ADB I/O are TIL111, TIL112, 4N26, 4N28, LDA101. There are many more. These items are readily available at electronic supply stores.

Install the optoisolators by removing a relay and replacing it with the optoisolator. Note that the optoisolators are smaller than the relays and that they should be mounted in the middle of the socket, oriented as shown below.

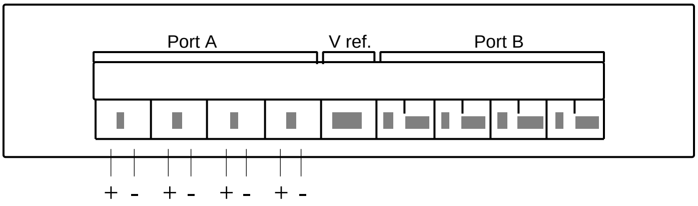


You can mix relays and optoisolators in port A. Each channel in port A is individually configurable as either input or output.

An optoisolator consists of a LED and a photosensitive transistor. The use of the light from the LED ensures electronic isolation as there is no electrical connection between the input and the output. The diode only lights up in one current direction so it is important to get the polarity of the connection correct. LEDs are sensitive to high currents (check the specifications of the unit you are using). For that reason it is important to use a current limiting resistor which limits the current to 10-20 mA.



An example of optoisolator use



Polarity of channels with optoisolators

The following code example reads from ADB I/O unit 1, port A after first having set channel 1 of port A as input.

Example in AppleScript:

```
configure ADBIO unit 1 port A as {digital in, digital out, digital out, digital
out}
get ADBIO unit 1 port A channel 1
```

The result will be a list with the data from port A, channel 1.

```
{channeltype:digital in, value:low}
```

Example in HyperTalk:

```
configureADBIO 1, A, ”digital in”, ”digital out”, ”digital out”, ”digital out”
put getADBIO(1, A) into result
```

The result will be four lines with the data from each of the channels in port A.

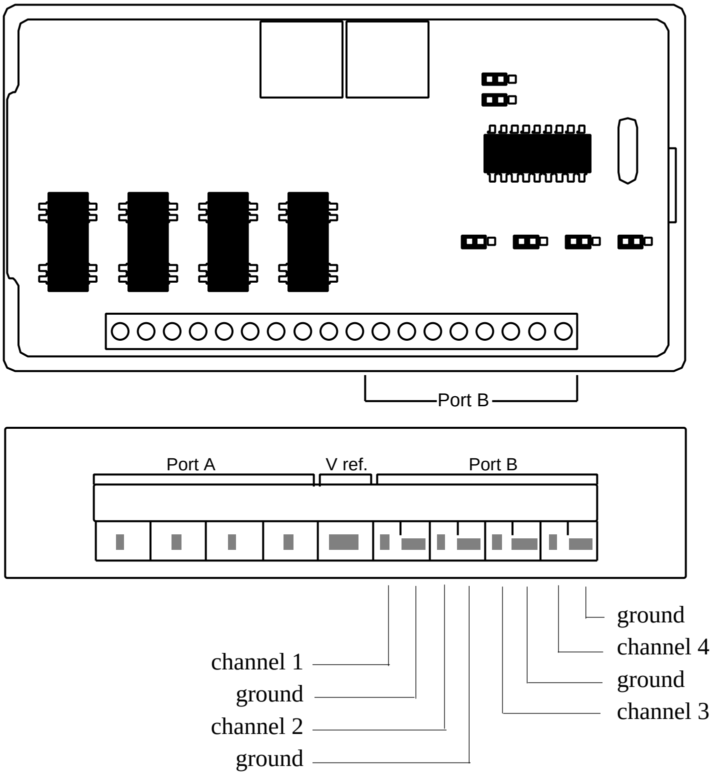
```
digital in, low
digital out, low
digital out, low
digital out, low
```

Attaching electronic devices to ADB I/O port B.

WARNING!
Always make sure that your Macintosh is turned off before attaching or disconnecting electrical devices from your ADB I/O.

Port B is designed for equipment which doesn’t need to be electrically isolated from the Macintosh. Such devices can be simple contacts, temperature sensors, or other sensors which operate at no more than +5V.

WARNING!
Since there is no electrical isolation in port B it is critical to use great care when attaching devices to the port. Although the channels on port B have internal voltage surge protection, incorrect hookup of electronic devices can result in a destroyed ADB I/O or even in severe damage to your Macintosh.



Port B has four channels, each connected to the screw terminal block. Each channel is equipped with both a signal and a ground in the terminal block.

The maximum voltage between signal and ground is +5V.

Each channel can be set to measure analog voltages or be set for digital input or output. You control these settings through software. When ADB I/O first boots up all channels are set to digital input.

Port B can be configured as follows, using software:

- Channel 1-4: Analog voltage measurement from 0V to +5V.

- Channel 1-3: Analog voltage measurement from 0V to reference voltage,
  Channel 4: Reference voltage input from +3V to +5V.

- Channel 1-2: Analog voltage measurement from 0V to +5V,
  Channel 3-4: digital input or output in any combination.

- Channel 1-4: Digital input or output in any combination.

The following sample code puts ADB I/O unit 1, port B, channels 1-4 to analog voltage measurement from 0V to +5V.

Example in AppleScript:

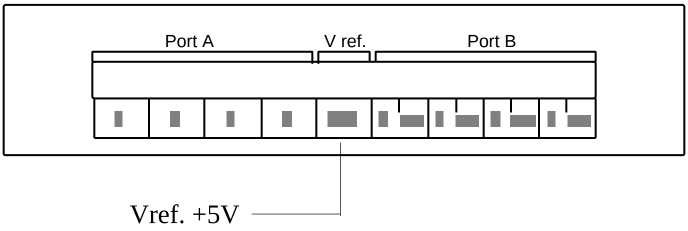
```
Configure ADBIO unit 1 port B as {analog in, analog in, analog in, analog in}
```

Example in HyperTalk:

```
ConfigureADBIO 1, B, "analog in", "analog in", "analog in", "analog in"
```

Vref +5V

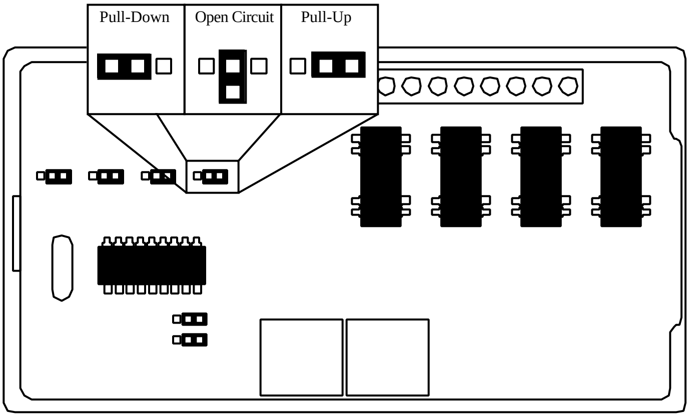
Two of the screw terminals are designated Vref. These two provide +5V through a 270Ω current limiting resistor. This voltage is suitable for powering voltage reference circuits, sensors or other small loads. The maximum total current from Vref can not exceed 18mA.



Pull-Up / Pull-Down Jumpers

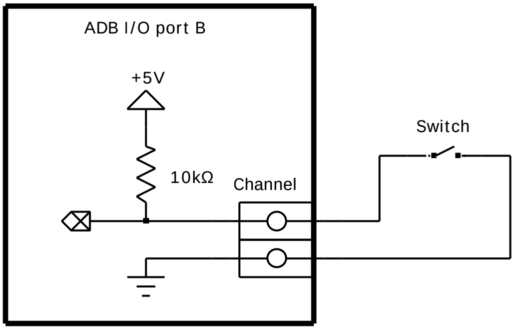
Each channel in port B has a jumper which is used to switch a 10kΩ resistor between the channel and +5V, 0V or an open circuit. This is useful when you want to connect contacts or push buttons to the channels and detect their closure state from software. When the channel is configured as an analog input the jumpers must be in the open circuit position for accurate analog input readings.

- If the resistor is connected to +5V you get a pull-up effect. This means that when the channel is open (not externally connected to GND) the software will read it as ”high”.
- If the resistor is connected to 0V you get a pull-down effect. This means that when the channel is open (not externally connected to a +5V source) the software will read it as ”low”.
- If the resistor is disabled the channel is open. Now you can use the channel for analog input.



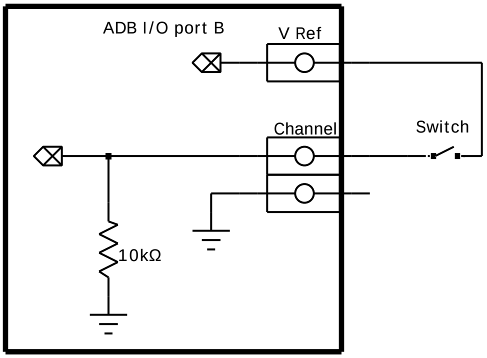
Pull-Up

This is the configuration at delivery. In this configuration you can detect the closure state of a contact. When the contact is open the channel will read ”high”. When the contact is closed the channel will read ”low”.



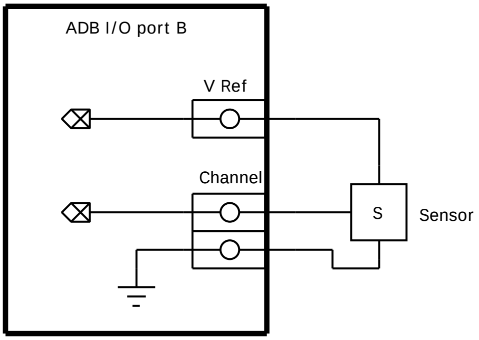
Pull-Down

You can also use pull-down as an alternative for contact state monitoring. When the contact is open the channel will read ”low”, when it is closed it will read ”high”.



Open Circuit

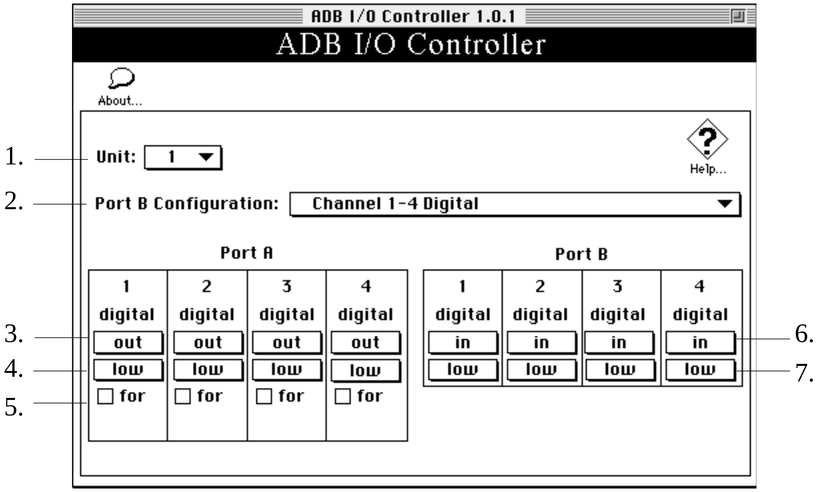
When measuring analog voltages you need to disable the resistor or you will get measurement errors. The diagram shows a sensor powered by Vref.



3. CONTROLLING DEVICES WITH ADB I/O

This section describes the included ADB I/O software and tells you how to write your own ADB I/O control software.

Start the *ADB I/O Controller.*
This application is useful for simple testing of the ADB I/O ports. Click the help button for help.



ADB I/O Controller 1.0.1

ADB I/O Controller

Unit: 1

Port B Configuration: Channel 1-4 Digital

1.This menu lets you specify which ADB I/O unit to control.

2.This menu lets you choose the port B configuration.

3.This is where you choose if the port A channels are to be used for input or output. If you want to control a relay it should be set to ‘out’.

4.Use these buttons to set the channels ‘high’ or ‘low’.

5.Each channel in port A can be set with a timer. Click here to specify the delay.

6.This is where you choose between input and output for port B.

7.If the channel is set for output, the button allows you to set the channel ‘high’ or ‘low’. If it is set for input the value of the reading will be displayed here.

Sampling Frequency

ADB I/O, like all Apple Desktop Bus devices, is limited by the maximum transmission speed of the bus. The bandwidth of the bus has to be shared by all the connected ADB devices. The Macintosh is the *master* of the bus and polls each device for data. The polling is done 90 times per second under the control of the operating system. The software shipped with ADB I/O talks directly to the bus so that you could theoretically get 90 samples or commands per second if there were no other devices on the bus. However, since there is usually a mouse and keyboard connected to the Macintosh, the ADB will have to share the ADB bandwidth with those two devices or they won’t function properly.

To avoid having the mouse appear unresponsive and sluggish it is recommended that you use the following sampling frequency rules of thumb for each port.

When using AppleScript or the XCMDs for HyperCard, Director or 4D we suggest that any ADB I/O port is read 12 times per second or less and written to 6 times per second or less (the write operation is always preceded by a read operation).

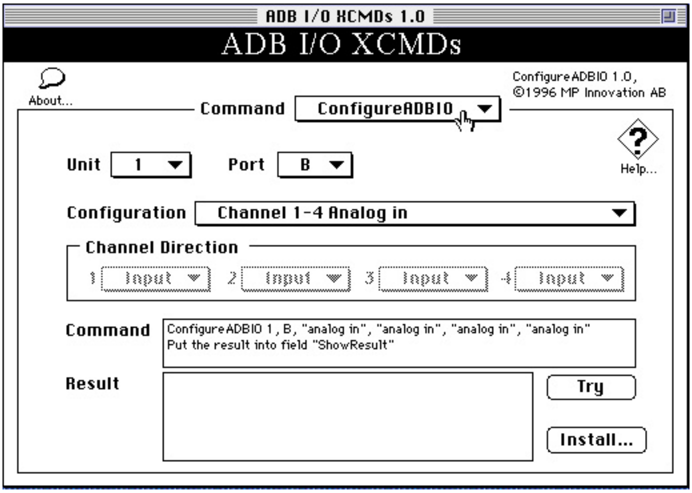
When using the C routines we suggest that any ADB I/O port is read 12 times per second or less and written to 12 times per second or less.

If the ADB I/O will be used only for control and data logging without user interaction, higher frequencies of access will work up to the overall limit of 90 samples per second.

Writing a program to control ADBI/O

HyperCard

1. Start up the application *ADB I/O XCMDs*
   This is a program to help you write software in HyperTalk for ADB I/O control. Basic familiarity with HyperCard and HyperTalk is required.



Command: ConfigureADBIO

Unit: 1

Port: B

Configuration: Channel 1-4 Analog in

Channel Direction: 1 Input, 2 Input, 3 Input, 4 Input

Command:

```
ConfigureADBIO 1, B, "analog in", "analog in", "analog in", "analog in"
Put the result into field "ShowResult"
```

Result:

2. You use the *Install…* button to install XCMDs for ADB I/O into your own HyperCard stack.

3. Use the pop-up menu *Command* to select the command you want to try. Specify the parameters using the various pop-up menus and buttons. By doing this you are automatically creating a command string in the *Command* window. You can test the command by clicking the try button and observing the result in the *result* window. Once you are satisfied just copy and paste the command into your own stack.

4. Click the Help button to learn more about each command.

DESCRIPTION

ConfigureADBIO configures the channels on a port of the ADB I/O unit as digital inputs, digital outputs, or analog inputs.

The default configuration for an ADB I/O unit is all channels on Port A configured as digital outputs, and all channels on Port B configured as digital inputs.

To get the configuration of a port on an ADB I/O unit use the **GetADBIO** XFCN.

SYNTAX

ConfigureADBIO <Unit>, <Port>, <Channel1Type>, <Channel2Type>, <Channel3Type>, <Channel4Type>

Using "!", "?" and "=" as first parameter will return on-line help.
"!" returns a Version String.
"?" returns a Syntax String.
"=" returns an Output Format String.

PARAMETERS

<Unit> The unit number of the ADB I/O to be configured.
This is used to address one of up to four ADB I/O units connected to the Apple Desktop Bus.
Each ADB I/O has "unit jumpers" inside the box, that is used to set the unit number for that ADB I/O unit.
The **<Unit>** parameter can be 1, 2, 3 or 4.

<Port> The port of the ADB I/O unit to be configured.
Each ADB I/O unit has two ports, A and B.
Port A is a digital input/output port with timer functions on the outputs.
Port B is a combined digital input/output port and analog input port.
The Port parameter can be A or B.

Channel Types

Each port of an ADB I/O unit has four channels. Each channel can be configured as a specific channel type. Only certain combinations of channel types on port B is allowed.

The possible channel types are:
"digital in" : a digital input channel.
"digital out" : a digital output channel.
"analog in" : an analog input channel.
"analog ref" : an analog reference input channel.

Channel1Type: The channel type of channel 1 of a port.
On port A this parameter can be:
"digital in"
"digital out"
On Port B this parameter can be (with limitations on combinations with other channels) :
"digital in"
"digital out"
"analog in"

Channel2Type: The channel type of channel 2 of a port.
On port A this parameter can be:
"digital in"
"digital out"
On Port B this parameter can be (with limitations on combinations with other channels) :
"digital in"
"digital out"
"analog in"

Channel3Type: The channel type of channel 3 of a port.
On port A this parameter can be:
"digital in"
"digital out"
On Port B this parameter can be (with limitations on combinations with other channels) :
"digital in"
"digital out"
"analog in"

Channel4Type: The channel type of channel 4 of a port.
On port A this parameter can be:
"digital in"
"digital out"
On Port B this parameter can be (with limitations on combinations with other channels) :
"digital in"
"digital out"
"analog in"
"analog ref"

USAGE

Use the ConfigureADBIO XCMD to configure an ADB I/O unit before controlling it with the SetADBIO XCMD and GetADBIO XFCN.
The first time an ADB I/O XCMD is called after startup, it will resolve the ADB addresses of all the ADB I/Os attached. It's a good practice to always use the ConfigureADBIO once before using the SetADBIO XCMD and GetADBIO XFCN.

RESULT

If the ConfigureADBIO XCMD executes successfully the result is empty.
If an error occurs, ConfigureADBIO sets the result to :

"Error : Wrong number of parameters"
"Error : Invalid Unit parameter"
"Error : Invalid Port parameter"
"Error : Invalid Channel1Type parameter"
"Error : Invalid Channel2Type parameter"
"Error : Invalid Channel3Type parameter"
"Error : Invalid Channel4Type parameter"
"Error : Wrong combination of ChannelType parameters"
"Error : No ADB devices were found"
"Error : No ADB I/O was found"
"Error : No ADB I/O data was received"
"Error : ADB I/O data had wrong length"
"Error : No ADB I/O with that unit number was found"
"Error : Two or more ADB I/O devices have the same unit number"
"Error : ADB Manager Error"

SetADBIO XCMD

DESCRIPTION

SetADBIO controls individual digital channels on the ports of the ADB I/O or sets a whole port to a number representing a binary bit pattern on that port. You can use the command to set channels high or low and also invoke the timers on Port A to set a channel high or low for a certain time from 0 to 25.5 seconds.

SYNTAX

SetADBIO <Unit>, <Port>, <Channel> | <Number> [,<"high"|"low">] [,<Time>]

Using "!", "?" and "=" as first parameter will return on-line help
"!" returns a Version String
"?" returns a Syntax String
"=" returns an Output Format String

PARAMETERS

<Unit> The unit number of the ADB I/O.
This is used to address one of up to four ADB I/O units connected to the Apple Desktop Bus.
Each ADB I/O unit has "unit jumpers" inside the box, which are used to set the unit number for that ADB I/O unit.
The **<Unit>** parameter can be 1, 2, 3 or 4.

<Port> The port of the ADB I/O unit to be set.
Each ADB I/O unit has two ports, A and B.
Port A is a digital input/output port with timer functions on the outputs.

Port B is a combined digital input/output port and analog input port.
The Port parameter can be A or B.

<Channel>
The channel to be set.
Which one of four channels on a port to set high or low.
The **<Channel>** parameter can be 1, 2, 3, or 4.

<Number>
The value to set the port to.
A decimal integer number between 0 and 15 will set the 4 channels of the port to the binary bit pattern representing that number. Channel 1 is the least significant bit in that bit pattern. Example: Setting the port to 1 will set channel 1 high and channel 2 to 4 low. All channels on the port must be configured as digital outputs when using this parameter.

"high"|"low"
The value to set the channel to.
On Port A, high means that the relay on that channel is activated. (closed).
On Port B, high means that the channel takes a logic high level (+5 V).

On Port A, low means that the relay on that channel is deactivated. (open).
On Port B, low means that the channel takes a logic low level (0 V).
If this parameter is present the previous parameter is interpreted as a **<Channel>.**
If this parameter is omitted the previous parameter is interpreted as a **<Number>.**

<Time>
This is the delay time in tenths of a second. It is used to set a channel on port A high or low for a certain period, The **<Time>** parameter works only on individual channels on port A and it is optional.
The **<Time>** parameter can have values between 0 and 255 meaning delays between 0.0s and 25.5s.

If a channel is set to a value (high or low) for a delay time, the channel will be set to the opposite value (low or high) when the time expires.

A channel delay can be extended by calling the SetADBIO command again with the **<Time>** parameter before the delay time of the previous call has expired.

USAGE
Use the SetADBIO XCMD to control individual channels that are configured as digital outputs. You can also output a binary bit pattern by setting the whole port to an integer that represents that bit pattern.

The SetADBIO XCMD will try to locate an ADB I/O unit corresponding to the unit number specified. If it does not find an ADB I/O unit it will set the result to an error string.

RESULT

If the SetADBIO XCMD executes successfully the result is empty.
If an error occurs, SetADBIO sets the result to :

"Error : Wrong number of parameters"
"Error : Invalid Unit parameter"
"Error : Invalid Port parameter"
"Error : Invalid Channel parameter"
"Error : Invalid 'high' | 'low' parameter"
"Error : The Time Parameter can not be used on port B"
"Error : Invalid Time parameter"
"Error : Invalid Number parameter"
"Error : Channel 1 is not an output"
"Error : Channel 2 is not an output"
"Error : Channel 3 is not an output"
"Error : Channel 4 is not an output"
"Error : All port channels are not digital outputs"
"Error : No ADB devices were found"
"Error : No ADB I/O was found"
"Error : No ADB I/O data was received"
"Error : ADB I/O data had wrong length"
"Error : No ADB I/O with that unit number was found"
"Error : Two or more ADB I/O devices have the same unit number."
"Error : ADB Manager Error"

GetADBIO XFCN

DESCRIPTION

GetADBIO gets information from a port of the ADB I/O unit. The function returns a data structure with channel type and digital or analog values of all channels of a port. The function can optionally return the whole port as a decimal integer number representing the bit pattern of the four digital channels of that port.

SYNTAX

getADBIO(<Unit>, <Port> [,<"Number">])

Using "!", "?" and "=" as first parameter will return on-line help.
"!" returns a Version String
"?" returns a Syntax String
"=" returns an Output Format String

PARAMETERS

<Unit> The unit number of the ADB I/O to get data from.
This is used to address one of up to four ADB I/O units connected to the Apple Desktop Bus. Each ADB I/O has "unit jumpers" inside the box, that is used to set the unit number for that ADB I/O unit.
The **<Unit>** parameter can be 1, 2, 3 or 4.

<Port> The port of the ADB I/O unit to get data from.
Each ADB I/O unit has two ports, A and B.
Port A is a digital input/output port with timer functions on the outputs. Port B is a combined digital input/output port and analog input port.
The Port parameter can be A or B.

<"Number">
Use the optional keyword "Number" to get the port as a number instead of a data structure. The number will be the decimal representation of the bit pattern of all 4 channels of the port. Channel 1 is the least significant bit in this bit pattern. Example: If channel 1 is high and channel 2, 3 and 4 is low, the number returned will be 1.

USAGE

Use the GetADBIO XFCN to get data from an ADB I/O unit.

RESULT

If the GetADBIO function executes successfully, it will return a PortData structure.
If the "Number" parameter is included in the call it will return a decimal number.

The PortData structure consists of four comma separated lists separated by return characters:

<Channel1Type>,<Channel1Value>
<Channel2Type>,<Channel2Value>
<Channel3Type>,<Channel3Value>
<Channel4Type>,<Channel4Value>

The channel types have the following meaning:
"digital in" : a digital input channel.
"digital out" : a digital output channel.
"analog in" : an analog input channel.
"analog ref" : an analog reference input channel.

The ChannelValue can be "high" or "low" for a digital channel.
The ChannelValue is an integer from 0 to 255 for an analog channel.

If the Number parameter is used the result is an integer from 0 to 15 which is the decimal representation of the bit pattern of all 4 channels of the port. Channel 1 is the least significant bit in this bit pattern.

ERRORS

If an error occurs, GetADBIO returns :
"Error : Wrong number of parameters"
"Error : Invalid Unit parameter"
"Error : Invalid Port parameter"
"Error : Invalid 'Number' parameter"
"Error : No ADB devices were found"
"Error : No ADB I/O was found"

"Error : No ADB I/O data was received"
"Error : ADB I/O data had wrong length"
"Error : No ADB I/O with that unit number was found"
"Error : Two or more ADB I/O devices have the same unit number"
"Error : ADB Manager Error"

AppleScript

You can use ADB I/O with many programs by using AppleScript. Claris FileMaker Pro and Microsoft Excel are just two examples of the multitude of programs which can be tied into ADB I/O through the use of AppleScript.

To use AppleScript for control of ADB I/O you need to install the ADB I/O AppleScript extension in the *Scripting Additions* folder in the *Extensions* folder in the System folder.

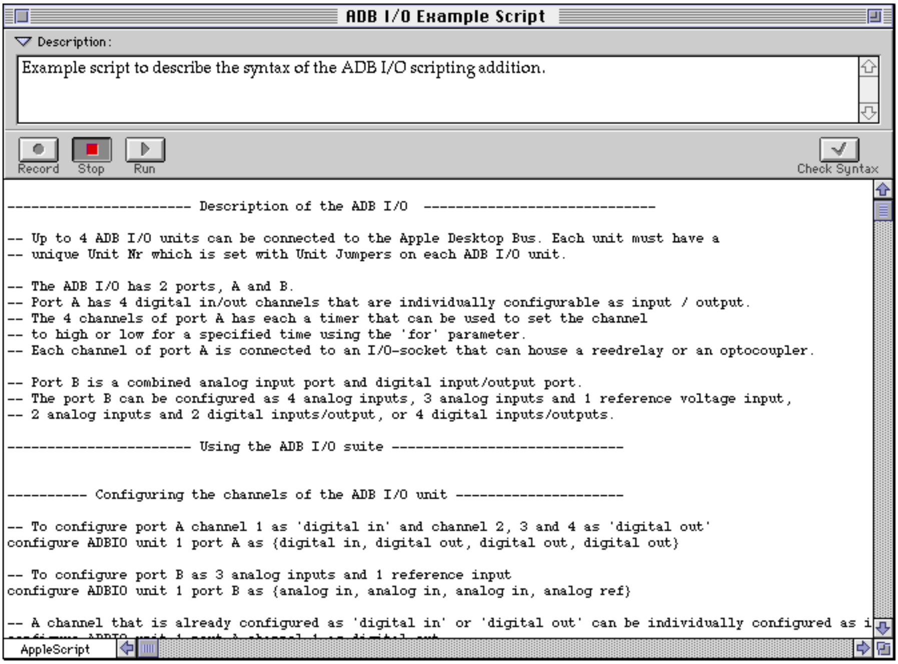
ADB I/O

You can use the *Script Editor* included with System 7.5 to write your AppleScripts. We recommend Scripter from Main Event Software for more extensive and complicated script editing and debugging. Scripter has many tools which help in both writing and debugging AppleScripts.

Script Editor

1. Open the document *ADB I/O Example Script.*
   This document shows several examples of ADB I/O command usage with AppleScript.

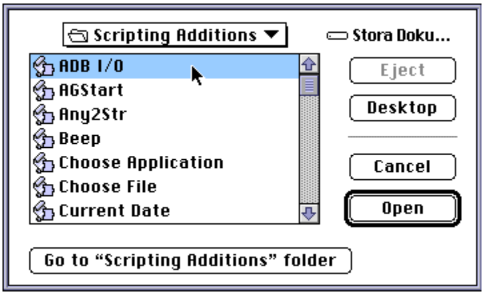
ADB I/O Example Script



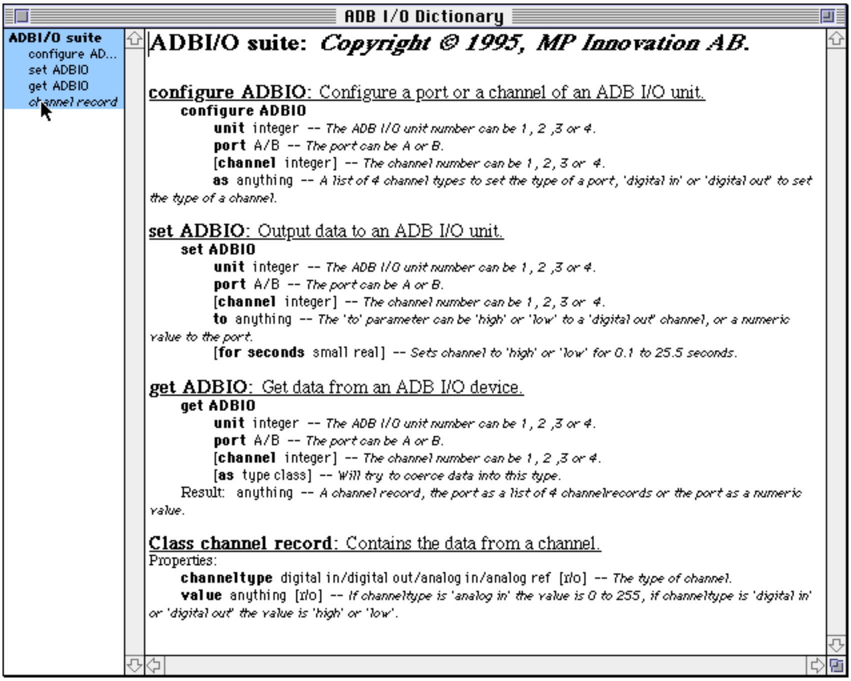
Example script to describe the syntax of the ADB I/O scripting addition.

```
----------------------- Description of the ADB I/O  -----------------------------

-- Up to 4 ADB I/O units can be connected to the Apple Desktop Bus. Each unit must have a
-- unique Unit Nr which is set with Unit Jumpers on each ADB I/O unit.

-- The ADB I/O has 2 ports, A and B.
-- Port A has 4 digital in/out channels that are individually configurable as input / output.
-- The 4 channels of port A has each a timer that can be used to set the channel
-- to high or low for a specified time using the 'for' parameter.
-- Each channel of port A is connected to an I/O-socket that can house a reedrelay or an optocoupler.

-- Port B is a combined analog input port and digital input/output port.
-- The port B can be configured as 4 analog inputs, 3 analog inputs and 1 reference voltage input,
-- 2 analog inputs and 2 digital inputs/output, or 4 digital inputs/outputs.

----------------------- Using the ADB I/O suite -----------------------------


---------- Configuring the channels of the ADB I/O unit ---------------------

-- To configure port A channel 1 as 'digital in' and channel 2, 3 and 4 as 'digital out'
configure ADBIO unit 1 port A as {digital in, digital out, digital out, digital out}

-- To configure port B as 3 analog inputs and 1 reference input
configure ADBIO unit 1 port B as {analog in, analog in, analog in, analog ref}

-- A channel that is already configured as 'digital in' or 'digital out' can be individually configured as i
configure ADBIO unit 1 port A channel 1 as digital out
```

2. Open the script dictionary to see the ADB I/O command set. To open the dictionary use the *Open Dictionary ...* menu selection in the File menu. Click on the *Go to ”Scripting Additions” folder* button. Select the ADB I/O entry and click the *Open* button.



3.A window with the ADB I/O scripting dictionary opens up. If you select all the items in the list on the left you will see the syntax for all the ADB I/O commands.



```
ADBI/O suite:  Copyright © 1995, MP Innovation AB.

configure ADBIO: Configure a port or a channel of an ADB I/O unit.
     configure ADBIO
          unit integer -- The ADB I/O unit number can be 1 , 2 ,3 or 4.
          port A/B -- The port can be A or B.
          [channel integer] -- The channel number can be 1 , 2, 3 or 4.
          as anything -- A list of 4 channel types to set the type of a port, 'digital in' or 'digital out' to set
the type of a channel.

set ADBIO: Output data to an ADB I/O unit.
     set ADBIO
          unit integer -- The ADB I/O unit number can be 1 , 2 ,3 or 4.
          port A/B -- The port can be A or B.
          [channel integer] -- The channel number can be 1 , 2, 3 or 4.
          to anything -- The 'to' parameter can be 'high' or 'low' to a 'digital out' channel, or a numeric
value to the port.
          [for seconds small real] -- Sets channel to 'high' or 'low' for 0.1 to 25.5 seconds.

get ADBIO: Get data from an ADB I/O device.
     get ADBIO
          unit integer -- The ADB I/O unit number can be 1 , 2 ,3 or 4.
          port A/B -- The port can be A or B.
          [channel integer] -- The channel number can be 1 , 2 ,3 or 4.
          [as type class] -- Will try to coerce data into this type.
     Result:  anything -- A channel record, the port as a list of 4 channelrecords or the port as a numeric
value.

Class channel record: Contains the data from a channel.
Properties:
     channeltype digital in/digital out/analog in/analog ref [r/o] -- The type of channel.
     value anything [r/o] -- If channeltype is 'analog in' the value is 0 to 255, if channeltype is 'digital in'
or 'digital out' the value is 'high' or 'low'.
```

Configure ADBIO

The Configure ADBIO command configures the channels of a port on the ADB I/O unit as digital inputs, digital outputs or analog inputs.

When the ADB I/O unit is first powered on, port A is configured as all digital outputs and port B is configured as all digital inputs.

The first time an ADB I/O command is called after startup, it will resolve the ADB addresses of all the ADB I/O units attached to the Macintosh. It is good practice to always use the Configure ADBIO command once before using the Set ADBIO or Get ADBIO commands.

SYNTAX

```
configure adbio ¬
      unit unitnumber ¬
      port ( A | B ) ¬
      [ channel channelnumber ] ¬
      as listOrchanneltype
```

PARAMETERS

unitnumber

The unit number of the ADB I/O unit to be configured. This is used to address one of a maximum of four ADB I/O units connected to the Macintosh. Each ADB I/O has “Unit Jumpers” inside the box that are used to set the unit number for that ADB I/O unit. The unitnumber can be 1, 2, 3 or 4.
Class: Integer

`( A | B )`

The port to be configured. Each ADB I/O has two ports, A and B. The port parameter is thus A or B.
Class: Enumerated

channelnumber

The channel to be configured. This parameter is used to individually change a digital channel from input to output or vice versa. Only digital channels can be individually configured. The channelnumber can be 1, 2, 3 or 4.
Class: Integer

listOrchanneltype

This parameter must be a single digital channel type when the channel parameter is present. With no channel parameter the whole port is configured with a list of four channel types, starting with channel 1. The channel types are:

`digital in`	: a digital input channel
`digital out`	: a digital output channel
`analog in`	: an analog input channel
`analog ref`	: an analog reference input channel

Class: List or Enumerated

RESULT

None

EXAMPLE

```
configure adbio unit 1 port B as ¬
{ analog in, analog in, analog in, analog in}
```

NOTES

Only certain combinations of channel types are valid for the channels on the ports. On the channels of port A, any combination of `digital in` and `digital out` are valid channel types. On port B the following combinations are possible:

- Channel 1, 2, 3 and 4: `digital in` or `digital out` in any combination.

- Channel 1 and 2: `analog in,` Channel 3 and 4: `digital in` or `digital out` in any combination.

- Channel 1, 2 and 3: `analog in,` channel 4: `analog ref.`

- Channel 1, 2, 3 and 4: `analog in.`

ERRORS

Error #	**Error message**
22210	No ADB devices at all can be found on the desktop bus. Make sure that the ADB I/O device, keyboard and mouse is connected to the Apple Desktop Bus of the Computer.
22211	No ADB I/O devices can be found on the desktop bus. Make sure that the ADB I/O device is connected properly to the Apple Desktop Bus port of the Computer. Always shut down your Macintosh before connecting the ADB I/O device.

22212 Could not get data from the ADB I/O device. Make sure that the ADB I/O device is connected properly to the Apple Desktop Bus port of the Computer. Always shut down your Macintosh before connecting the ADB I/O device.

22213 The data from the ADB I/O device has wrong format.

22214 An ADB I/O device has been found on the desktop bus, but it doesn't match the unit parameter. Make sure that the ‘unit jumpers’ on the ADB I/O device are set correctly.

22215 Two or more ADB I/O devices have the same unit number. Make sure that the 'unit jumpers' on the ADB I/O devices are set correctly.

22221 The 'unit' parameter is not valid. It should be 1, 2, 3 or 4.

22222 The 'channel' parameter is not valid. It should be 1, 2, 3 or 4.

22225 The 'as' parameter ('digital in' or 'digital out') is not valid for that channel. The channel type is not 'digital in' or 'digital out'. The port should be configured with a list of four channel types first.

22226 The 'as' parameter (a list of channel types) must have four items.

22227 The 'as' parameter (a list of channel types) is not valid for port A. Only 'digital in' or 'digital out' is valid items.

22228 The 'as' parameter (a list of channel types) is not valid. The items in the list can not be combined in that way.

22231 The 'port' parameter is not valid, it should be A or B.

22232 The 'as' parameter (a channel type) is not valid, it should be 'digital in' or 'digital out'.

Set ADBIO

The Set ADBIO command sets a channel of a port on the ADB I/O `high` or `low`. The channel must be configured to be `digital out` before using this command.

SYNTAX

```
          set adbio ¬
                unit unitnumber ¬
                port ( A | B ) ¬
                [ channel channelnumber ] ¬
                to valueOrNumber ¬
                [ for seconds delaytime ]
```

PARAMETERS

unitnumber

The unit number of the ADB I/O unit to be set. This is used to address one of a maximum of four ADB I/O units connected to the Macintosh. Each ADB I/O has “Unit Jumpers” inside the box that are used to set the unit number for that ADB I/O unit. The unitnumber can be 1, 2, 3 or 4.
Class: Integer

`( A | B )`

The port to be set. Each ADB I/O has two ports, A and B. The port parameter is thus A or B.
Class: Enumerated

channelnumber

The channel to be set. Only digital output channels can be set `high` or `low`. The channelnumber can be 1, 2, 3 or 4.
Class: Integer

valueOrNumber

This parameter must be `high` or `low` when the channel parameter is present. With no channel parameter the whole port can be set with a decimal integer number representing a binary bit pattern on the port. The least significant bit of that pattern will be channel 1 and the most significant bit will be channel 4.
The integer number can be 0 to 15.
Class: Enumerated or Integer

delaytime

This parameter can be used on any channel on port A and will invoke a delaytimer on that channel. The delaytime can be set between 0.0 and 25.5 seconds in 0.1 second increments. By setting a channel `high` for a delaytime, the channel will first go `high` and then go `low` after the delay

time has expired. If a channel is set `low` the channel will first go `low` and then go `high` after the delaytime. The timers are re-triggerable, meaning that consecutive set commands on a channel with a delaytime parameter, will keep the delay from expiring.
Class: Real or Integer

RESULT

None

EXAMPLE

```
set adbio unit 1 port A channel 1 ¬
to high for seconds 10.5
```

NOTES

On port A a relay will be closed when a channel is set `high` and open when the channel is set `low`. On port B the channel will have a logical high voltage level (+5 V) when a channel is set `high` , and a logical low voltage level (0 V) when the channel is set `low`. The delay timer on port A has a 0.1 second resolution and it can take up to 0.1 second for the channel to take the value set when the delaytime parameter is used.

ERRORS

Error #	**Error message**
22210	No ADB devices at all can be found on the desktop bus. Make sure that the ADB I/O device, keyboard and mouse is connected to the Apple Desktop Bus of the Computer.
22211	No ADB I/O devices can be found on the desktop bus. Make sure that the ADB I/O device is connected properly to the Apple Desktop Bus port of the Computer. Always shut down your Macintosh before connecting the ADB I/O device.
22212	Could not get data from the ADB I/O device. Make sure that the ADB I/O device is connected properly to the Apple Desktop Bus port of the Computer. Always shut down your Macintosh before connecting the ADB I/O device.
22213	The data from the ADB I/O device has wrong format.
22214	An ADB I/O device has been found on the desktop bus, but it doesn't match the unit parameter. Make sure that the ‘unit jumpers’ on the ADB I/O device are set correctly.
22215	Two or more ADB I/O devices have the same unit number. Make sure that the 'unit jumpers' on the ADB I/O devices are set correctly.

22221 The 'unit' parameter is not valid. It should be 1, 2, 3 or 4.

22222 The 'channel' parameter is not valid. It should be 1, 2, 3 or 4.

22223 The 'to' parameter is not valid. It should be 'high' or 'low'.

22224 The 'to' parameter is not valid for that channel. The channel should be configured as 'digital out' first.

22229 The 'for seconds' parameter (time in seconds) is not valid, it should be between 0.1 and 25.5.

22230 The 'for seconds' parameter must be used together with a 'channel' parameter.

22231 The 'port' parameter is not valid, it should be A or B.

22232 The 'as' parameter (a channel type) is not valid, it should be 'digital in' or 'digital out'.

22233 The 'to' parameter (an integer) is not valid for a port, it should be between 0 and 15.

22234 The port is not configured as all outputs.

22235 The 'for seconds' parameter can not be used on port B.

Get ADBIO

The Get ADBIO command gets the values of a port or a channel of an ADB I/O unit.

SYNTAX

```
get adbio ¬
      unit unitnumber ¬
      port ( A | B ) ¬
      [ channel channelnumber ] ¬
      [ as className ]
```

PARAMETERS

unitnumber

The unit number of the ADB I/O unit to get data from. This is used to address one out of a maximum of four ADB I/O units connected to the Macintosh. Each ADB I/O has “Unit Jumpers” inside the box that are used to set the unit number for that ADB I/O unit. The unitnumber can be 1, 2, 3 or 4.
Class: Integer

`( A | B )`

The port to get data from. Each ADB I/O has two ports, A and B. The port parameter is thus A or B.
Class: Enumerated

channelnumber

The channel to get data from. The channelnumber can be 1, 2, 3 or 4.
Class: Integer

className

To get the data from the ADB I/O coerced to another datatype like an integer or string this parameter can be used.
Class: Class

RESULT

A list of records of object class *channel record* (as defined by the ADB I/O scripting addition) when the *channelnumber* and *className* parameters are not used. The list will start with the channel record for channel 1.

When the *channelnumber* parameter is used the result is a single *channel record.*

When the *className* and *channelnumber* parameter is used the value of the channel is returned as a numeric type and coerced to *className*. A value of `high` will be returned as 1 and `low` as 0.

When the *className* but not the *channelnumber* parameter is used the whole port is returned as a numeric value representing a binary bit pattern with channel 1 as the least significant bit. This numeric value is then coerced to the *className.*

A *channel record* contains the following properties:

`channeltype`

The type of the channel can be one of the following Enumerations:

`digital in`	: a digital input channel
`digital out`	: a digital output channel
`analog in`	: an analog input channel
`analog ref`	: an analog reference input channel

Class: Enumerated

`value`

The value of the channel is `high` or `low` if the channel is digital and an integer between 0 and 255 if the channel is analog.

Class: Enumerated or Integer

EXAMPLE

```
get adbio unit 1 port B channel 1
```

NOTES

A port configured with all channels digital can be read as an integer representing the binary pattern of the port. By using `integer` as the *className* parameter and omitting the *channelnumber* parameter the whole port is read as an integer between 0 and 15.

ERRORS

Error #	**Error message**
22210	No ADB devices at all can be found on the desktop bus. Make sure that the ADB I/O device, keyboard and mouse is connected to the Apple Desktop Bus of the Computer.
22211	No ADB I/O devices can be found on the desktop bus. Make sure that the ADB I/O device is connected properly to the Apple Desktop Bus port of the Computer. Always shut down your Macintosh before connecting the ADB I/O device.
22212	Could not get data from the ADB I/O device. Make sure that the ADB I/O device is connected properly to the Apple Desktop Bus port of the Computer. Always shut down your Macintosh before connecting the ADB I/O device.

22213 The data from the ADB I/O device has wrong format.

22214 An ADB I/O device has been found on the desktop bus, but it doesn't match the unit parameter. Make sure that the ‘unit jumpers’ on the ADB I/O device are set correctly.

22215 Two or more ADB I/O devices have the same unit number. Make sure that the 'unit jumpers' on the ADB I/O devices are set correctly.

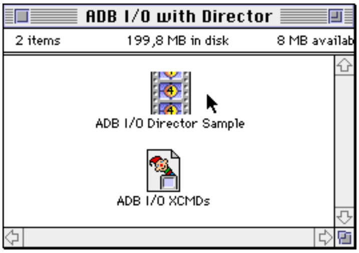
22221 The 'unit' parameter is not valid. It should be 1, 2, 3 or 4.

22222 The 'channel' parameter is not valid. It should be 1, 2, 3 or 4.

22231 The 'port' parameter is not valid, it should be A or B.

Macromedia Director

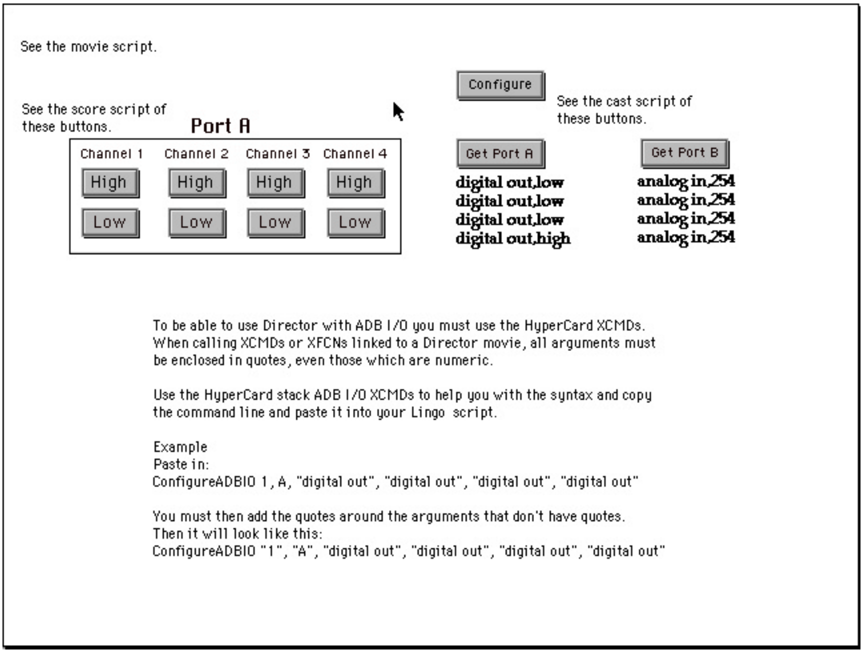
You can also use the included HyperCard XCMDs for ADB I/O together with Macromedia Director. The file *ADB I/O XCMDs* must be placed in the same folder as the Director document file you want to control ADB I/O. Use the program *ADB I/O XCMDs* to learn about using XCMDs before you try to use them with Director.



The file *ADB I/O Director Sample* contains examples of Director used with ADB I/O.

1. Open the file *ADB I/O Director Sample* .

2. Select *Play* from the Edit menu.



See the movie script.

See the score script of these buttons.

See the cast script of these buttons.

digital out,low
digital out,low
digital out,low
digital out,high

analog in,254
analog in,254
analog in,254
analog in,254

To be able to use Director with ADB I/O you must use the HyperCard XCMDs. When calling XCMDs or XFCNs linked to a Director movie, all arguments must be enclosed in quotes, even those which are numeric.

Use the HyperCard stack ADB I/O XCMDs to help you with the syntax and copy the command line and paste it into your Lingo script.

Example
Paste in:
ConfigureADBIO 1, A, "digital out", "digital out", "digital out", "digital out"

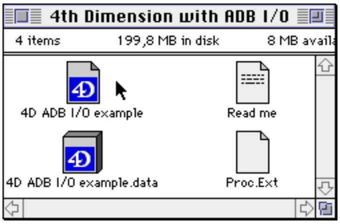
You must then add the quotes around the arguments that don't have quotes.
Then it will look like this:
ConfigureADBIO "1", "A", "digital out", "digital out", "digital out", "digital out"

4th Dimension

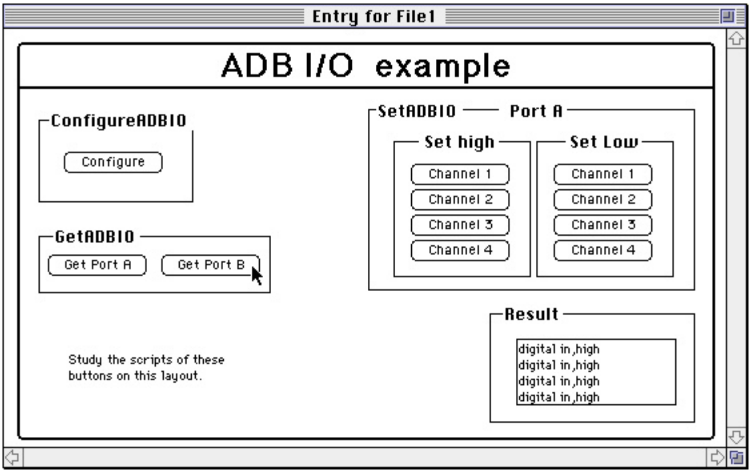
4th Dimension is a powerful relational database. By using ADB I/O 4D externals you can control ADB I/O from within 4D.

The document *4D ADB I/O example* contains simple examples of how to to control the ADB I/O. The document *Proc.Ext* has to be in the same folder as the 4D-document which contains the scripting code controlling the ADB I/O. You can also use the utility *4D mover* to install the ADB I/O externals in your 4D document. *4D mover* is included when you buy 4D. More recent versions of 4D have a folder named 4DEX which is where 4D externals should be stored. There is more detailed information in the 4D manual. The file *Read me* contains more information on the ADB I/O 4D externals.

1. Open the file *Read me* to learn more about 4D and ADB I/O.



2. Open the document *4D ADB I/O example* to try the examples.



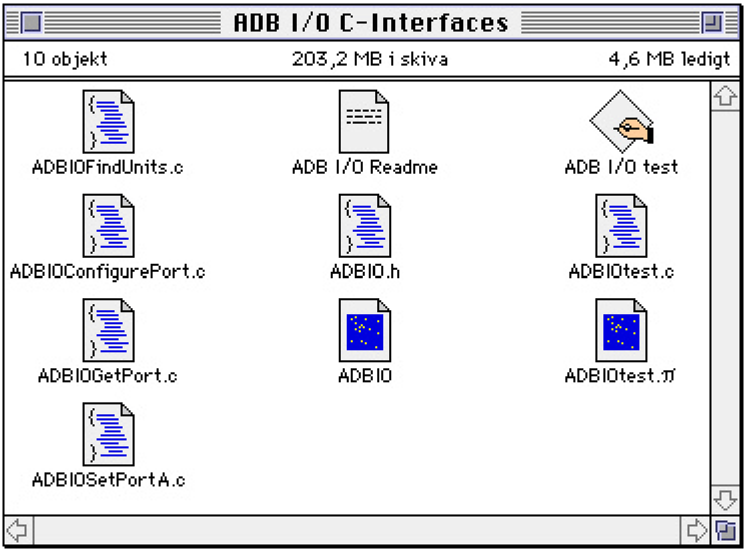
Study the scripts of these buttons on this layout.

```
digital in,high
digital in,high
digital in,high
digital in,high
```

C-Interfaces

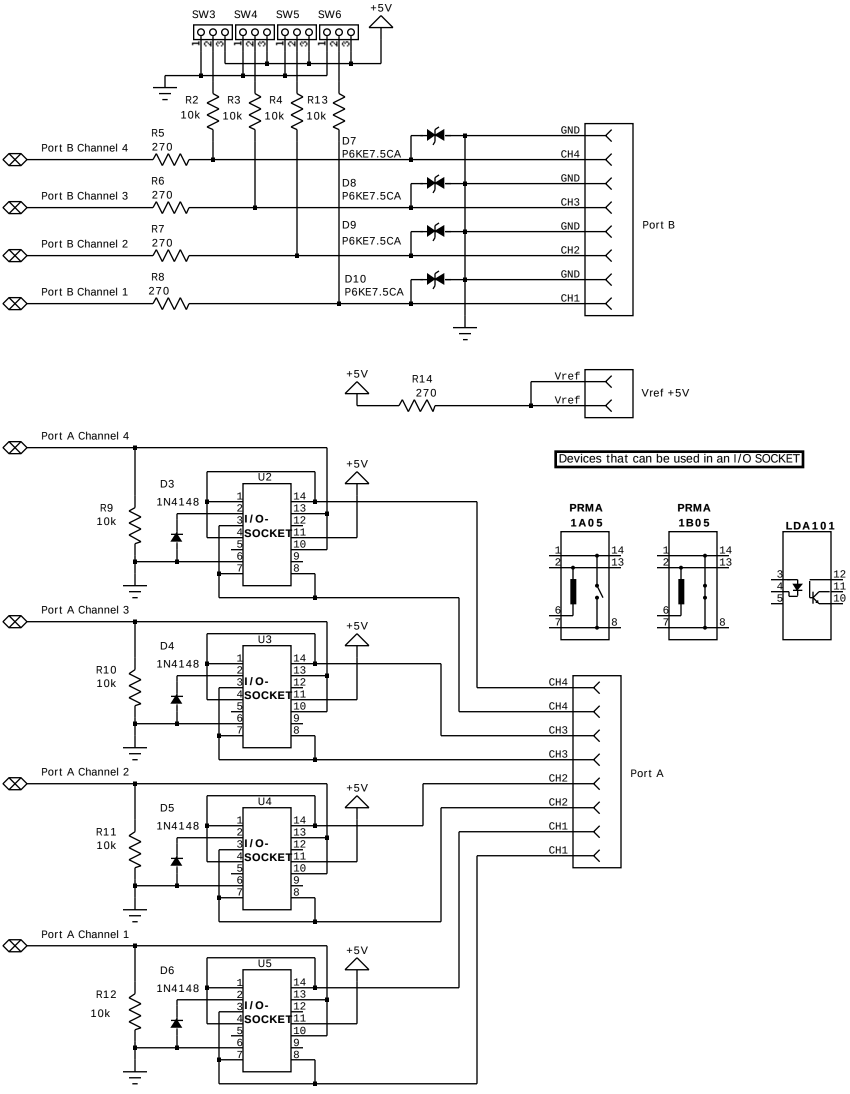
The *ADB I/O C-Interfaces* let you write software in C to control ADB I/O. A Symantec C library is provided. The source code for the routines is also provided. A small demonstration and test program shows how the routines can be incorporated in your own software.

Since the source code is provided you have the option of using other development environments such as Metrowerks or MPW.



APPENDIX A. ADB I/O SCHEMATIC

The schematic below shows the input and output sections of ADB I/O.



APPENDIX B. TECHNICAL SPECIFICATIONS

PortA relays

Max current:	0.5A dc
Max power:	10VA
Max voltage:	100V dc
Lifetime:	100 000 000 operations (10Vdc, 10mA)
Max contact resistance:	150mΩ

PortB digital channels

Input voltage ‘High’	Min: +2.0V	Max: +5.0V
Input voltage ‘Low’	Min: 0V	Max: +0.8V
Output voltage ‘High’	Min: +4.3V	Max: +5.0V (I_{OH} = 2mA)
Output voltage ‘Low’	-	Max: +0.6V (I_{OL} = 2mA)

PortB analog channels

Resolution:	8 bits	
Non-linearity:	less than ±1 LSb (Least significant bit)	
Differential non-linearity:	less than ±1 LSb (Least significant bit)	
Scaling error:	less than ±1 LSb (Least significant bit)	
Zero offset error:	less than ±1 LSb (Least significant bit)	
Monotonicity:	Guaranteed	
Reference voltage:	Min: +3.0V	Max: +5.0V
Recommended impedance in voltage source:	Max: 500Ω	

V ref. +5V

Max current:	18mA (270Ω resistor to +5V)

Connectors

ADB	2 MiniDIN 4, female
Port A	8 screw terminals with wire protectors
Port B	8 screw terminals with wire protectors
Vref	2 screw terminals with wire protectors

Power supply

Voltage:	+5Vdc (from Apple Desktop Bus)
Current:	Max:105mA (all relays active, Port B and Vref shorted to ground)
	Typ: 55mA (all relays active,Vref shorted to ground)
	Min: 4mA (no relays active, no devices attached)

Dimensions

Height:	29mm
Length:	122mm
Width:	65mm

Environmental requirements

Temperature during use:	+10°C to +40 °C
Humidity:	5% - 95% (non-condensing)

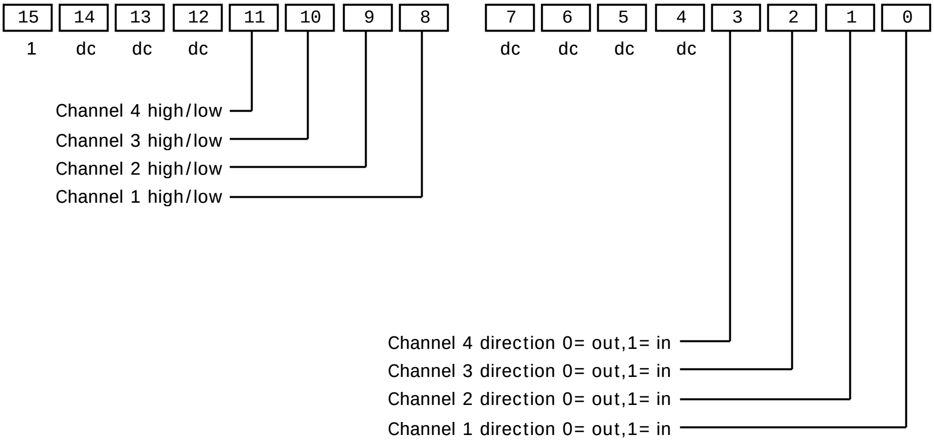
Specifications are subject to change without notice.

APPENDIX C. APPLE DESKTOP BUS COMMUNICATION

This section briefly overviews the communications protocol used to control the ADB I/O via the Apple Desktop Bus. You can use these protocols if you wish to write your own drivers from scratch. You will of course need to know how to develop software for the Macintosh using a language such as C or Pascal. You will also need to be familiar with the ADB Manager and assembly level programming for the Macintosh.

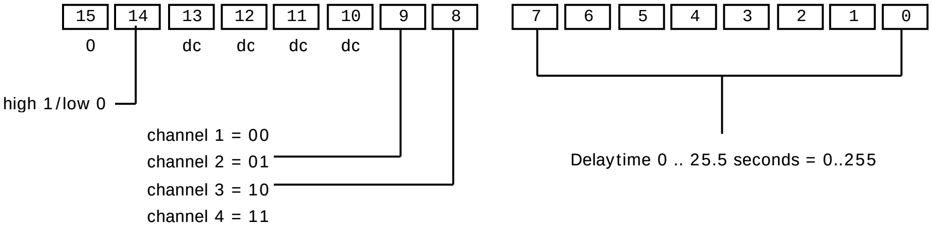
Configure port A

ADB command = Listen ADB Register 2


15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1
dc
dc
dc
dc
dc
dc
dc
Channel 4 high/low
Channel 3 high/low
Channel 2 high/low
Channel 1 high/low
Channel 4 direction 0= out,1= in
Channel 3 direction 0= out,1= in
Channel 2 direction 0= out,1= in
Channel 1 direction 0= out,1= in

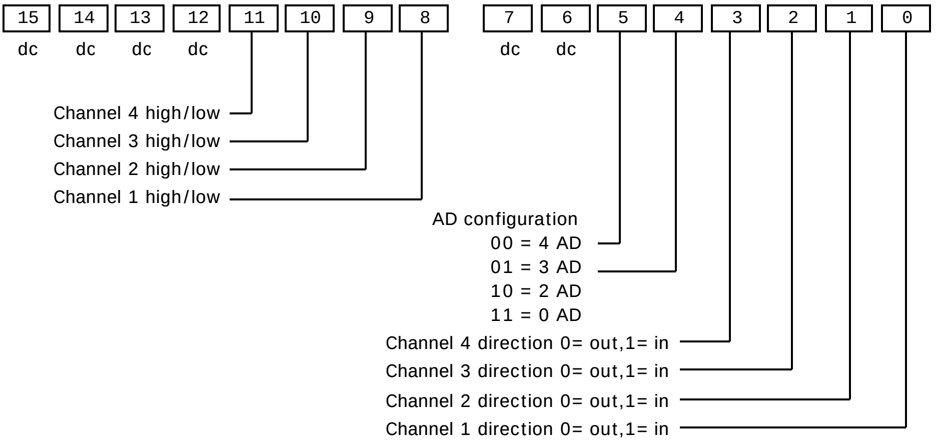
Control individual channels on port A

ADB command = Listen ADB Register 2



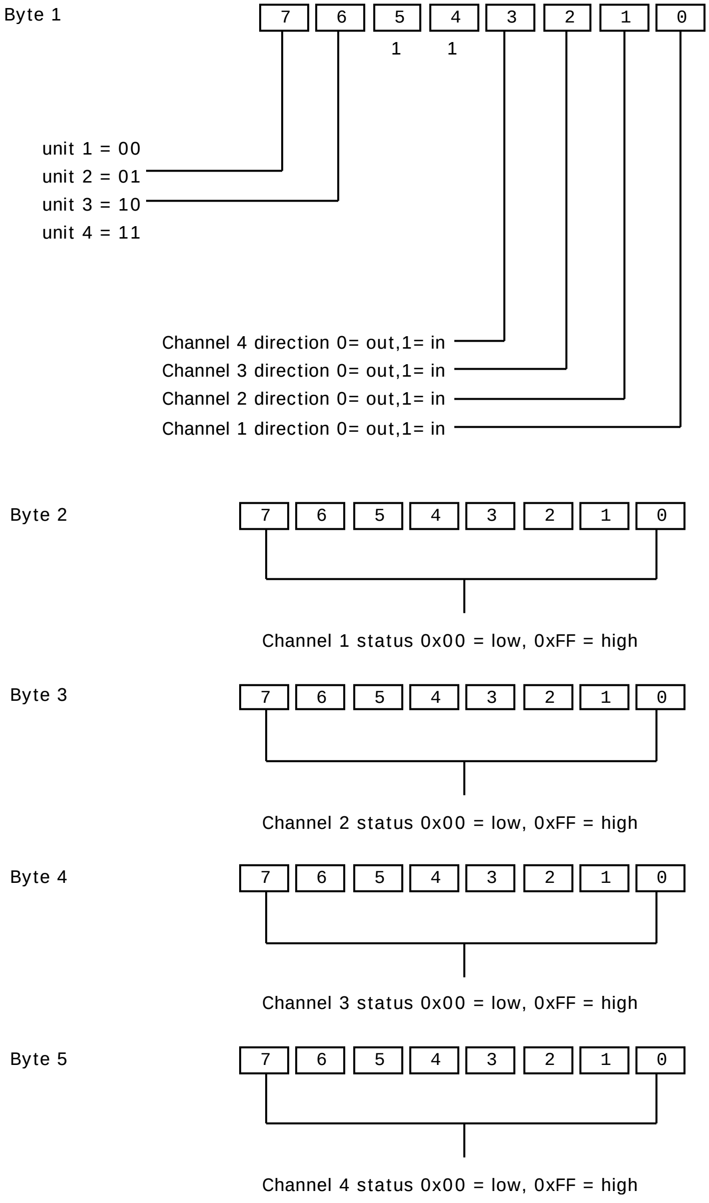
Configure port B

ADB command = Listen ADB Register 1



Get port A data

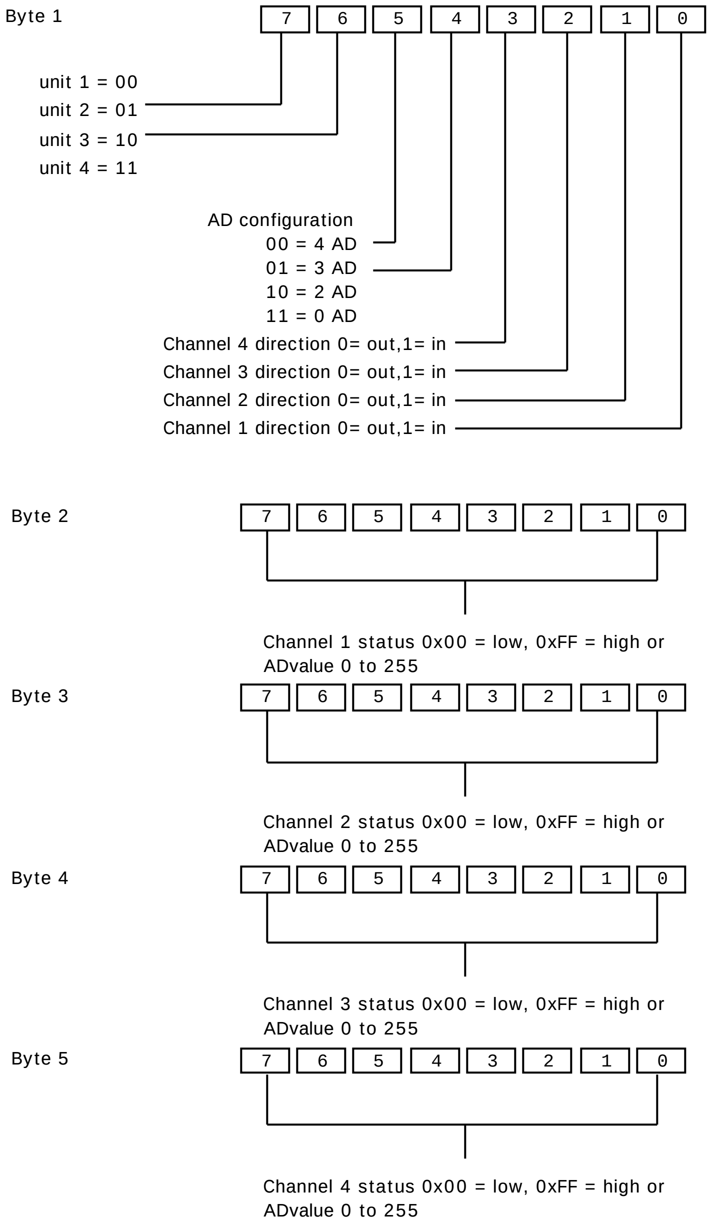
ADB command = Talk ADB Register 2


Byte 1
7
6
5
4
3
2
1
0
1
1
unit 1 = 00
unit 2 = 01
unit 3 = 10
unit 4 = 11
Channel 4 direction 0= out,1= in
Channel 3 direction 0= out,1= in
Channel 2 direction 0= out,1= in
Channel 1 direction 0= out,1= in
Byte 2
7
6
5
4
3
2
1
0
Channel 1 status 0x00 = low, 0xFF = high
Byte 3
7
6
5
4
3
2
1
0
Channel 2 status 0x00 = low, 0xFF = high
Byte 4
7
6
5
4
3
2
1
0
Channel 3 status 0x00 = low, 0xFF = high
Byte 5
7
6
5
4
3
2
1
0
Channel 4 status 0x00 = low, 0xFF = high

Get port B data

ADB command = Talk ADB Register 1

On digital channel change, a Service Request signal is sent. ADB I/O will then report Port B data on ADB command = Talk ADB Register 0



Byte 1

7	6	5	4	3	2	1	0

unit 1 = 00
unit 2 = 01
unit 3 = 10
unit 4 = 11

AD configuration
00 = 4 AD
01 = 3 AD
10 = 2 AD
11 = 0 AD

Channel 4 direction 0= out,1= in
Channel 3 direction 0= out,1= in
Channel 2 direction 0= out,1= in
Channel 1 direction 0= out,1= in

Byte 2

7	6	5	4	3	2	1	0

Channel 1 status 0x00 = low, 0xFF = high or ADvalue 0 to 255

Byte 3

7	6	5	4	3	2	1	0

Channel 2 status 0x00 = low, 0xFF = high or ADvalue 0 to 255

Byte 4

7	6	5	4	3	2	1	0

Channel 3 status 0x00 = low, 0xFF = high or ADvalue 0 to 255

Byte 5

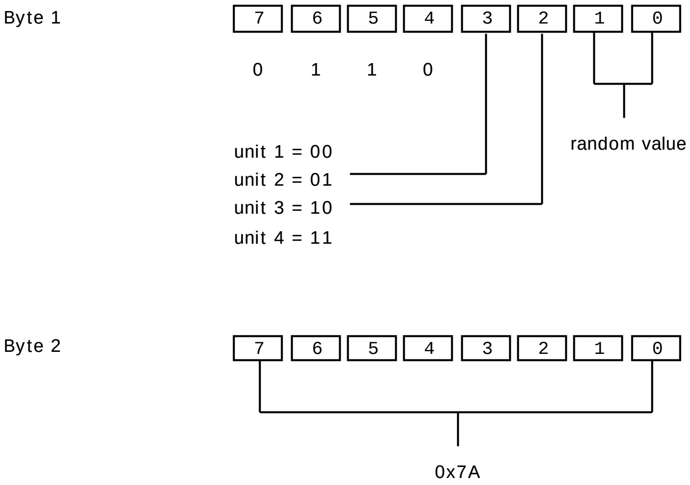
7	6	5	4	3	2	1	0

Channel 4 status 0x00 = low, 0xFF = high or ADvalue 0 to 255

Get ADB I/O status

ADB command = Talk ADB Register 3

Default ID: 0x7A
Default adress: 0x07



INDEX