xPC TargetBox™

For Use with Real-Time Workshop®

- Modeling
- Simulation
- Implementation

User's Guide

Version 2

The MathWorks

How to Contact The MathWorks:

	`www.mathworks.com`	Web
	`comp.soft-sys.matlab`	Newsgroup
@	`support@mathworks.com`	Technical support
	`suggest@mathworks.com`	Product enhancement suggestions
	`bugs@mathworks.com`	Bug reports
	`doc@mathworks.com`	Documentation error reports
	`service@mathworks.com`	Order status, license renewals, passcodes
	`info@mathworks.com`	Sales, pricing, and general information
☎	508-647-7000	Phone
	508-647-7001	Fax
✉	The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098	Mail

For contact information about worldwide offices, see the MathWorks Web site.

Getting Started with xPC TargetBox™

Printing History: November 2002 First printing New for Version 2 (Release 13)

Contents

Preface

1 Introduction

2 Initial Loop-Back Test

3 Installation and Configuration

4 xPC TargetBox I/O Options

5 Using xPC TargetBox

Preface

xPC TargetBox™ is a hardware product you use with xPC Target software to build, test, validate, and deploy real-time systems. This chapter includes the following sections:

What Is xPC TargetBox?

xPC Target is a solution for prototyping, testing, and deploying real-time systems using standard PC hardware. It is an environment that uses a target PC, separate from a host PC, for running real-time applications.

xPC TargetBox is an industrial target PC optimized for executing real-time code generated with xPC Target, Real-Time Workshop, and a C/C++ compiler. With additional I/O options, you can connect to a hardware environment using analog input (A/D), analog output (D/A), digital I/O, counter/timers, interrupts, and connecting to a CAN field bus. This product allows

- **Rapid prototyping** — Prototype a controller running on an xPC TargetBox, and validate your design without the need for custom target hardware.
- **Hardware-in-the-loop simulation** — Test a real controller with a simulated plant running on an xPC TargetBox.
- **Limited deployment** — Using the stand-alone mode, connect an xPC TargetBox to your plant and run the controller software.

On a host PC, using xPC Target, Real-Time Workshop, and a C/C++ compiler, you can generate executable code from your Simulink and Stateflow models. With the xPC Target Embedded Option, you can use an external floppy disk drive or network connection to

- **Boot from flash disk**— Transfer the xPC Target kernel to a flash disk, boot the xPC TargetBox with the kernel, and then, from the host PC, download and run a real-time application
- **Run stand-alone applications** — Transfer the xPC Target kernel and a stand-alone application to a flash disk, disconnect the host PC, and then boot and run the real-time application.

These key features and more make the xPC TargetBox the ideal solution for applications requiring strong processing capability, low power consumption, flexibility, small size, wide range of temperatures, and ruggedness.

Quick Start Guide for xPC TargetBox

xPC TargetBox is an industrial PC that lets you run target applications generated by Real-Time Workshop and xPC Target. Use the following process to get your xPC TargetBox up and running quickly.

1. Unpack the xPC TargetBox. When you receive your xPC TargetBox, it is delivered in a rugged shipping case. Place the case on a table with the top of the box up, and push the lower end of the latches to open the case.

   You will see the xPC TargetBox, a floppy disk drive, screw terminal boards, the AC adapter, and an AC power cord.

   See “Unpacking the Shipping Box” on page 2-2.

2. Remove the xPC TargetBox from the shipping case.

   Depending on the I/O options you selected, test dongles are shipped already attached to the xPC TargetBox I/O connectors 1 through 6. The CAN dongle, for I/O option 308, is packed with the power cord.

3. Attach the AC adapter and the AC power cord. If your xPC TargetBox includes I/O option 308, attach the CAN dongle to the CAN 1 and CAN 2 connectors.

4. Plug in the AC power cord. When you do this, the internal self-test program for your xPC TargetBox will start. The two USER LEDs will turn on, turn off, and then, if the self-test program is successful, turn back on.

   See “Running the Self-Test Without a Monitor” on page 2-4.

5. After you have tried the self-test program, you can connect a keyboard and SVGA video monitor to the xPC TargetBox. You can then run the self-test program again to see the results using scope displays.

   See “Running the Self-Test with a Monitor” on page 2-9.

6 Set up xPC TargetBox for use with xPC Target.

- See “xPC TargetBox Installation for Regular Use” on page 2-15.
- See “xPC Target Installation” on page 3-2.
- See “Serial Communication” on page 3-3 or “Network Communication” on page 3-6.

7 Create an xPC Target boot disk for `BootFloppy` mode.

See “Target Boot Disk” on page 3-10.

8 Boot the xPC TargetBox and test the connections to the host PC.

See “Testing xPC Target Installation” on page 3-13.

Using This Guide

To help you read and use this guide effectively, this section provides a brief description of the chapters and a suggested reading path. This section includes the following topics:

- **Expected Background** — Proficiency with using MATLAB and Simulink, and familiarity with Real-Time Workshop
- **Organization** — Table with a list of chapters in the xPC TargetBox Getting Started documentation

Expected Background

Users who read this book should be familiar with

- Using Simulink and Stateflow to create models as block diagrams, and simulating those models in Simulink
- The concepts and use of Real-Time Workshop to generate executable code

When using Real-Time Workshop and xPC Target, you do not need to program in C or other programming languages to create, test, and deploy real-time systems.

If you are a new xPC Target user — Read *Getting Started with xPC TargetBox* and *Getting Started with xPC Target*.

Begin in this manual with Chapter 1, “Introduction,” and then read Chapter 1, “Introduction,” in the xPC Target Getting Started documentation. These chapters will give you an overview of the xPC TargetBox features and xPC Target environment.

Next, read and complete the installation and testing tasks in Chapter 3, “Installation and Configuration.”

If you are an experienced xPC Target user — Read and complete the installation and testing tasks in Chapter 3, “Installation and Configuration.”

Organization

The following table lists the organization of the xPC TargetBox Getting Started documentation.

Chapter	Description
Preface	List of required products and organization of this book
Chapter 1, “Introduction”	Overview of the functions and features of xPC TargetBox
Chapter 2, “Initial Loop-Back Test”	Installing and testing the xPC TargetBox
Chapter 3, “Installation and Configuration”	Connecting the xPC TargetBox to a host computer, and configuring the host computer for communication
Chapter 4, “xPC TargetBox I/O Options”	Reference information for the xPC TargetBox I/O options and a description of the loop-back testing procedures
Chapter 5, “Using xPC TargetBox”	Using the DOSLoader and StandAlone modes with xPC TargetBox, and transferring files from a host PC to an xPC TargetBox with FTP

MPL Warranty For xPC TargetBox

xPC TargetBox comes with a one-year warranty provided by MPL (xPC TargetBox is designed by MPL for The MathWorks exclusively). During this time, MPL will repair defects in material and workmanship under the terms of the MPL warranty. This section includes the following topics:

- **Repairing an xPC TargetBox Under Warranty** — Contact The MathWorks for instructions to ship your xPC TargetBox directly to MPL.
- **Limited Warranty from MPL** — Statement of warranty, exclusive remedy, and limitation of liability.
- **Contacting The MathWorks for Technical Support** — Use e-mail, the Internet, or telephone to get help with your problem.
- **xPC TargetBox Certification** — The xPC TargetBox has been certified by passing emission tests required for use in the US, Europe, Japan, Canada, and Australia/New Zealand.

Repairing an xPC TargetBox Under Warranty

Your first point of contact for all xPC Target and xPC TargetBox questions is The MathWorks. See “Contacting The MathWorks for Technical Support”.

If a MathWorks Technical Support Representative determines that your xPC TargetBox hardware needs repair, follow the steps below:

1. A MathWorks Customer Service Representative will give you a Return Material Authorization (RMA) number and provide you with shipping documentation and instructions. The MathWorks will also inform MPL of your problem.

2. Using the shipping instructions provided by The MathWorks, package and return your xPC TargetBox to MPL. For information on correct packaging, refer to “Unpacking the Shipping Box” on page 2-2.

3. MPL will repair your xPC TargetBox and return it to you within the shortest possible time.

Limited Warranty from MPL

WARRANTY: MPL WARRANTS THAT THE PRODUCTS DELIVERED HEREUNDER SHALL BE FREE FROM DEFECTS IN WORKMANSHIP AND MATERIAL FOR A PERIOD OF TWELVE (12) MONTHS FROM DATE OF DELIVERY TO THE BUYER, INCLUDING COMPONENT PARTS OF PRODUCTS SOLD AS SPARE, REPLACEMENT, MAINTENANCE OR STORAGE PARTS, WHICH ARE ALSO WARRANTED FOR TWELVE (12) MONTHS FROM DATE OF DELIVERY, PROVIDED, HOWEVER, IN EITHER CASE, THAT NOTICE OF ANY SUCH DEFECT IS PROVIDED TO MATHWORKS OR MPL WITHIN THIRTY (30) DAYS OF ITS DISCOVERY BY THE BUYER. NEITHER MATHWORKS NOR MPL MAKE ANY OTHER WARRANTY OF ANY KIND TO BUYER, AND HEREBY EXPRESSLY DISCLAIMS ANY SUCH OTHER WARRANTY, WHETHER EXPRESS OR IMPLIED, IN FACT OR BY LAW, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OR WARRANTY OF ANY KIND INCORPORATED OR REFERENCED IN BUYER'S SPECIFICATIONS OR PURCHASE ORDER, OR ANY OTHER WARRANTY ARISING BY STATUTE, OR OTHERWISE IN LAW, OR FROM A COURSE OF DEALINGS OR USE OF TRADE, ALL OF WHICH ARE HEREBY DISCLAIMED.

EXCLUSIVE REMEDY: IN ANY EVENT, THE BUYER'S EXCLUSIVE REMEDY HEREUNDER IS LIMITED TO THE FURNISHING OF REPLACEMENT PARTS ON AN EXCHANGE BASIS, OR, AT THE OPTION OF MPL, TO THE REPAIR OR REPLACEMENT OF DEFECTIVE PRODUCTS OR COMPONENT PARTS AT ITS PLANT, BUT IN EITHER CASE ONLY SO LONG AS AN EXAMINATION WITHIN THE PERIOD OF WARRANTY REVEALS THE PARTS TO BE DEFECTIVE, AND IN ALL CASES ALL COSTS OF SHIPPING AND PACKAGING SHALL BE BORNE BY THE BUYER.

IN ADDITION, AND NOTWITHSTANDING ANYTHING HEREIN TO THE CONTRARY, NEITHER MATHWORKS NOR MPL SHALL INCUR ANY OBLIGATION HEREUNDER WITH RESPECT TO PRODUCTS WHICH ARE MODIFIED IN ANY WAY BY THE BUYER WITHOUT THE PRIOR WRITTEN CONSENT OF MPL, AND IN NO EVENT SHALL MATHWORKS OR MPL INCUR ANY OBLIGATION TO REPAIR OR REPLACE PRODUCTS OR COMPONENT PARTS WHICH ARE DETERMINED BY MPL OR SELLER, IN ITS SOLE DISCRETION, TO BE DEFECTIVE DUE TO BUYER

MISUSE, USE OF UNAUTHORIZED REPAIR PARTS OR UNAUTHORIZED THIRD-PARTY SERVICE, OR BECAUSE OF A USE NOT IN ACCORDANCE WITH SPECIFIC MPL OR SELLER PRODUCT OPERATION AND MAINTENANCE INSTRUCTIONS. CUSTOMER ACCEPTS AND ACKNOWLEDGES THAT THE FOREGOING ALLOCATION OF RISK IS REFLECTED IN THE PURCHASE PRICE.

LIMITATION OF LIABILITY: IN NO EVENT SHALL MATHWORKS OR MPL BE LIABLE FOR SPECIAL, EXEMPLARY, INDIRECT, PUNITIVE, CONSEQUENTIAL OR INCIDENTAL DAMAGES INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS, LOSS OF BUSINESS OR GOODWILL, OR LOSS OF USE FOR ANY BREACH OF THESE TERMS AND CONDITIONS OR ARISING UNDER CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY, WARRANTY OR ANY OTHER THEORY OF LIABILITY OR ANY LIABILITY CLAIM OF ANY THIRD PARTY EVEN IF MATHWORKS OR MPL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

FOR ANY BREACH OF THESE TERMS AND CONDITIONS, MATHWORKS' AND MPL'S MAXIMUM LIABILITY SHALL NOT IN ANY EVENT EXCEED THE TOTAL PRICE OF THE PRODUCTS ORDERED BY CUSTOMER.

NEITHER MATHWORKS NOR MPL SHALL BE LIABLE IF IT IS UNABLE TO PERFORM ANY OF ITS OBLIGATIONS CONTAINED IN THESE TERMS AND CONDITIONS DUE TO ANY INDUSTRIAL DISPUTE, WAR, FLOOD, EXPLOSION, ACT OF GOD OR ANY OTHER EVENT BEYOND THE REASONABLE CONTROL OF MATHWORKS OR MPL.

Contacting The MathWorks for Technical Support

If you are having a problem with your xPC TargetBox that you cannot solve, contact The MathWorks directly for help.

Internet	`http://www.mathworks.com/support/`
E-mail	`mailto:support@mathworks.com`
Telephone	508-647-7000 Ask for Technical Support.

xPC TargetBox Certification

The xPC TargetBox has received the following certifications:

- FCC Part 15, Class A Emissions Testing (US)
- EN 55022 (Europe)
- VCCI (Japan)
- ICES-033 (Canada)
- AS/NZS 3548 (Australia/New Zealand)

1

Introduction

This chapter is an overview of the functions and features of xPC TargetBox. An introduction to these features and the xPC Target software environment will help you develop a model for working with the xPC TargetBox. This chapter includes the following sections:

Features of xPC TargetBox

xPC TargetBox includes many features to help you prototype, test, and deploy real-time systems using xPC Target software. This section includes the following topics:

- **PC Compatibility** — Test your prototype controllers on a fully PC-compatible computer
- **Hardware** — Designed for a wide range of applications and environments
- **Software** — QuickBoot BIOS (Pentium III based xPC TargetBox), FreeDOS, FTP server, and a target application for system testing preinstalled on the flash disk
- **xPC Target Embedded Option** — Create stand-alone target applications and GUI application that interface to the target application
- **Communication** — Communicate with the xPC TargetBox through standard peripherals, serial ports, and a network connection
- **I/O Options** — Add I/O options to the xPC TargetBox that satisfy the needs of most xPC Target applications

PC Compatibility

xPC TargetBox is a 100% PC-compatible, all-in-one embedded computer with the following characteristics:

CPU performance — Available systems include a Pentium II 266 MHz, Pentium III 400 MHz, or Pentium III 700 MHz processor. The performance of these systems is sufficient for 75% of all xPC Target applications found in office, laboratory, and mobile (field) environments.

System memory — A FlashRAM chip is mounted directly on the IDE connector, allowing you to boot the system without a floppy disk drive. This configuration requires no additional mechanical parts. However, you do need the external floppy drive or a network connection to transfer files to the flash disk.

PC/104+ expansion bus — I/O options are standard PC/104 boards.

Hardware

The xPC TargetBox hardware was carefully selected and designed to handle a wide range of applications and environments:

Small size — The overall design makes xPC TargetBox ideal for its use in mobile and field applications. Physical dimensions are the same for all units: 270 mm x 162 mm x 82 mm (6.8 in x 4 in x 2 in).

Rugged system —The xPC TargetBox has a fully aluminum enclosure that is anodized on the inside, equipped with all necessary I/O connectors, and has the ability to operate under normal or harsh conditions without the need for cooling fans or other moving parts.

Temperature range — A low power design allows fanless operation with no mechanically rotating parts. Cooling holes in the enclosure are not necessary. The standard temperature range is 0°C to 60°C. For harsh conditions, the xPC TargetBox is available with an extended temperature range that can run from -40°C to 75°C (models 206 and 207) or from -40°C to 65°C (model 208).

Internal power supply — Onboard power supply with DC power input for voltages between 8 and 28 volts.

External AC adapter— The external AC power adapter supplied with the xPC TargetBox allows worldwide AC operation with an input of 110 to 230 volts, 50 Hz to 60 Hz.

xPC TargetBox has capabilities that allow you to use it in a stand-alone mode of operation. Since the unit is powered by a DC voltage, you can power it with a number of battery types that range from 8 to 28 volts DC.

External floppy disk drive — An external floppy disk drive is provided and can be connected to the xPC TargetBox using a standard 25-pin shielded cable (additional 25-pin port on the back panel). The external floppy disk drive does not need additional power. Instead, the xPC TargetBox provides power to the disk drive.

With this configuration, you can use the external floppy disk drive to

- Boot the xPC Target kernel from a boot floppy disk using the standard setup for xPC Target. When using the xPC TargetBox for mobile applications, you can load the kernel/application onto the flash disk, and then remove the external disk drive.
- Restore files to flash memory and recover from a system failure.

Software

The xPC TargetBox is shipped with the following software on the flash disk:

FreeDOS — The xPC TargetBox uses FreeDOS to transfer stand-alone kernel/target application files to the flash disk, and to boot the kernel for downloading target applications from a host PC.

xPC TargetBox does not include Microsoft Windows.

FTP server — If you have a target application that is larger than 1 MByte (xPC Target allows applications up to 16 MB) you cannot use the external floppy disk drive to transfer stand-alone applications to an xPC TargetBox. Or if your application requires a rugged environment, using a mechanical drive would not be suitable.

In these cases, you can connect an xPC TargetBox to a LAN or directly to a host PC with an Ethernet crossover cable, and use the FTP server on the flash disk to transfer files.

Quick-booting the xPC TargetBox — Normally the BIOS of an xPC TargetBox takes about 20 to 30 seconds to boot. With the QuickBoot capability, the BIOS boot time is reduced to about 2 seconds.

The QuickBoot capability provides a BIOS optimized for a particular system configuration. The optimized BIOS (QuickBoot binary) is preinstalled in the xPC TargetBox EPROM.

Note The QuickBoot capability is available only with an xPC TargetBox using a Pentium III CPU (xPC TargetBox models 107, 108, 207, and 208).

Target application for system testing — The flash disk in the xPC TargetBox contains an xPC Target stand-alone application that self-tests the hardware and all I/O options installed in the xPC TargetBox. One of your first tasks is to run this self-test application and verify that the system is functioning correctly.

xPC Target Embedded Option

The xPC Target Embedded Option is an extension to xPC Target. The xPC TargetBox currently includes a promotional copy of the xPC Target Embedded Option.

BootFloppy mode — Without the xPC Target Embedded Option, you can boot the xPC TargetBox from a boot floppy disk created by xPC Target. When you boot the xPC TargetBox from that disk, the xPC Target kernel uses the resources on the target PC (CPU, RAM, and serial port or network adapter) without changing the files already stored on the flash disk.

DOSLoader mode — With the xPC Target Embedded Option and DOSLoader mode, you can copy the xPC Target kernel to the flash disk using a a floppy disk or FTP. After transferring the files, boot the kernel from the flash disk, and then download a target application from the host PC.

(DOSLoader)/StandAlone mode — With xPC Target Embedded Option and StandAlone mode, you can copy the xPC Target kernel with a self-contained target application using a floppy disk or FTP. After transferring the files, you can remove the floppy drive and reset the xPC TargetBox. The xPC TargetBox automatically boots the kernel and executes your target application without the need to communicate with the host PC.

Communication

You can communicate with the xPC TargetBox through standard peripherals, serial ports, and a network connection.

Standard PC peripherals — There is support for standard peripherals accessible through standard connectors: SVGA interface up to 1280 x 1024, PS/2 mouse, PS/2 keyboard, four RS-232 ports, and a standard parallel port.

With the addition of standard peripherals, you can use the xPC Target kernel command-line interface to enter commands for control, signal acquisition, and parameter tuning directly from the xPC TargetBox without the need of a host PC.

Network or serial connection to the host PC— Connect your xPC TargetBox to your host computer using either an RS-232 port or the onboard Ethernet controller (10BASE-T/100BASE-TX, Intel 82559ER).

You use the connection to the host PC to download a target application, or to FTP files to the flash disk.

I/O Options

You can add up to three I/O options to an xPC TargetBox.

I/O expandability — The xPC TargetBox provides the ability to expand I/O connections through a standard PC/104 (ISA) and PC/104+ (PCI) expansion bus. The MathWorks currently offers a set of seven I/O options you can purchase with your system.

If needed, you can add your own PC/104 I/O boards, but The MathWorks and MPL will not provide repair service for a system with your boards inside. If your system requires service, you must return it for repair in the same configuration in which it was shipped to you.

For more information about the available I/O options, see “xPC TargetBox I/O Options” on page 1-11.

I/O driver support — The xPC Target software contains driver blocks for all of the seven I/O options currently available for the xPC TargetBox. These options satisfy typical rapid prototyping requirements with A/D, D/A, digital I/O, CAN, counters, timers, encoders, and PWM. The drivers are represented by Simulink blocks in the xPC Target driver library. In addition, the library contains driver blocks for the user LEDs and the watchdog circuitry. Your interaction with the drivers is through these Simulink blocks and their parameter dialog boxes.

You drag and drop blocks from the I/O library and connect I/O drivers to your Simulink model the same way as you would connect any standard Simulink block.

Internal connectors and cables — Shielded connectors and internal cables bring the PC/104 I/O board signals to the outside of the box.

External connectors and cables — Shielded connectors are mounted onto the front and rear panels. The connection layout is the same for all configurations. I/O options include shielded cables with screw terminal boards.

Onboard CAN controller — The Intel 82527 CAN controller on the CPU board is not officially supported by the xPC TargetBox driver library. For applications that need CAN connections, add I/O option 308 to the xPC TargetBox.

System Requirements

The host PC is your desktop or notebook computer where you install MATLAB, Simulink, Stateflow and Stateflow Coder (optional), Real-Time Workshop, xPC Target, and xPC Target Embedded Option. This section includes the following topics:

- **Software Requirements for the Host PC** — MATLAB, Simulink, Real-Time Workshop, xPC Target, and xPC Target Embedded Option
- **Hardware Requirements for the Host PC** — Desktop or notebook computer
- **xPC TargetBox Software** —FreeDOS, QuickBoot capability for Pentium III based xPC TargetBox systems, and a custom target application to test your system
- **xPC TargetBox System Hardware** —Pentium CPU, flash disk, standard PC connections, I/O slots for three boards, and an external floppy drive
- **xPC TargetBox I/O Options** —Analog input, analog output, digital I/O, encoders, counter/timers, and interrupt inputs

Software Requirements for the Host PC

The following table lists the minimum software xPC Target requires on your host PC.

Software	Description
Operating system	A Microsoft Windows platform supported by The MathWorks
MATLAB	Version 6.5
Simulink	Version 5.0
Real-Time Workshop	Version 5.0
C language compiler	Microsoft Visual C/C++ Version 5.0, 6.0, or 7.0 Watcom C/C++ Version 10.6 or 11.0

Software	Description
xPC Target	Version 2.0 with most recent xPC Target library update. Each xPC TargetBox requires an active software license for xPC Target
xPC Target Embedded Option	Version 2.0. (optional)

Hardware Requirements for the Host PC

The following table lists the minimum resources xPC Target requires on the host PC.

Hardware	Description
Communication	One free serial port (COM1 or COM2) with a 9-pin or 25-pin D-sub connector, or an Ethernet card connected to a network
CPU	Pentium, Athlon, or higher
Peripherals	Hard disk drive with 50 MB of free space, 23 MB for xPC Target, 20 MB for xPC Target Embedded Option, and 7 MB for HTML documentation One 3.5-inch floppy disk drive CD-ROM drive
RAM	128 MB or more

xPC TargetBox Software

The following table lists the software included with your xPC TargetBox.

Software	Description
BIOS	PC compatible. Standard BIOS with Pentium II systems (models 106 and 206). QuickBoot BIOS with Pentium III systems (models 107, 108, 207, 208).
Operating system	xPC TargetBox includes a copy of FreeDOS on the flash disk.
FTP server	Copy stand-alone kernel/target applications to an xPC TargetBox using a network connection. You do not need a floppy disk drive.
Self-test target application	Customized target application for testing the specific I/O options installed in your xPC TargetBox

xPC TargetBox System Hardware

xPC TargetBox is designed and manufactured by MPL exclusively for The MathWorks. For a detailed hardware specification, refer to the *xPC TargetBox Reference Guide* provided in the shipping case from MPL. The following table summarizes the hardware.

Hardware	Description
Communication	Serial ports (COM1, COM2) with a 9-pin D-sub connector, and network connection with an Ethernet J45 connector
CPU	Pentium
Peripherals	You need to supply a VGA monitor, PS/2 keyboard, and PS/2 mouse.
RAM	128 MB

Hardware	Description
Nonvolatile memory	32 MB FlashRAM module
I/O expansion slots	Room for three PC/104 expansion boards. The xPC TargetBox is not expandable beyond adding I/O boards.
External connectors	Front and back panels with shielded I/O connectors for I/O board options CAN bus interface for CAN specification 2.0 Connectors for standard PC peripherals (monitor, mouse, keyboard, two RS-232 ports, one parallel port, and RJ-45 Ethernet)
External power source	External AC adaptor with an output voltage of +19 volts DC for worldwide operation. The type of power cord provided with an xPC TargetBox is dependent on the shipping location.
External floppy disk drive	One external 3.5-inch floppy disk drive with a dedicated parallel port connector

The following table lists available xPC TargetBox base systems.

Standard Unit Product Name	Description
xPC TargetBox 106	266 MHz Pentium II, 128 MB RAM, 32 MB FlashRAM. Standard temperature from 0°C to 60°C.
xPC TargetBox 107	400 MHz Pentium III, 128 MB RAM, 32 MB FlashRAM with QuickBoot capability. Standard temperature from 0°C to 60°C.
xPC TargetBox 108	700 MHz Pentium III, 128 MB RAM, 32 MB FlashRAM with QuickBoot capability. Standard temperature from 0°C to 60°C.

The following table lists available xPC TargetBox base systems with extended temperature ranges.

Extended Temperature Unit Product Name	Description
xPC TargetBox 206	266 MHz Pentium II, 128 MB RAM, 32 MB FlashRAM. Extended temperature from -40°C to 75°C.
xPC TargetBox 207	400 MHz Pentium III, 128 MB RAM, 32 MB FlashRAM with QuickBoot capability. Extended temperature from -40°C to 75°C.
xPC TargetBox 208	700 MHz Pentium III, 128 MB RAM, 32 MB FlashRAM with QuickBoot capability. Extended temperature from -40°C to 65°C.

xPC TargetBox I/O Options

The following table lists the available I/O options for the xPC TargetBox. You can use up to three I/O options in a single xPC TargetBox.

Each I/O option includes an installed PC/104 board, one or two internal cables connected to either the front or back panels, one or two external SCSI-II cables, one or two screw terminal boards (except for the CAN I/O option), one or two test dongles, and I/O board reference documentation.

The label on the bottom of the xPC TargetBox indicates the I/O boards in your system, their base addresses, and the ports they are connected to.

I/O Board Product Name	Description
xPC TargetBox IO 301 Diamond MM-32-AT Extended temperature -40 to 85°C	**Analog input (A/D)** — 32 single-ended, 16 differential, or 16 single-ended/8 differential (16 bit) channels with a maximum sample rate of 200 kHz (5 µs). Channel coupling selected with jumpers on board. If you do not specify a channel coupling when purchasing an xPC TargetBox, the coupling is set to 16 differential channels. Unipolar and bipolar input ranges of 0-10 V, 0-5 V, 0-2.5 V, 0-1.25 V, 0-0.625 V, ±10 V, ±5 V, ±2.5 V, ±1.25 V, and ±0.625 V selected by software.
	Analog output (D/A) — 4 (12 bit) channels with a 6 µs settling time. Output range set to 0-10 V, 0-5 V, ±10 V, or ±5 V with jumpers on the board. If you do not specify an output range when purchasing an xPC TargetBox, the range is set to ±10 V.
	Digital I/O — 24 digital I/O lines divided into three groups with 8 bits each. xPC Target does not support the counters on this board.

I/O Board Product Name	Description (Continued)
xPC TargetBox IO 302 Diamond Ruby-MM-1612 Extended temperature -40 to 85°C	**Analog output (D/A)** — 16 (12 bit) channels with a 6 µs settling time. Output range for each group of 8 channels set to 0-10 V, 0-5 V, 0-2.5 V, ±10 V, ±5 V, or ±2.5 V with jumpers on the board. If you do not specify a range when purchasing an xPC TargetBox, the range is set to ±10 V. **Digital I/O** — 24 TTL digital I/O lines divided into three groups with 8 bits each xPC Target does not support the external trigger on this board.
xPC TargetBox IO 303 Diamond Ruby-MM-416 Standard temperature 0 to 70°C	**Analog output (D/A)** — 4 (16 bit) channels with a 10 µs settling time. Output range for each channel set independently to 0-10 V, ±10 V, or ±5 V with jumpers on the board. If you do not specify a range when purchasing an xPC TargetBox, the range is set to ±10 V. **Digital I/O** — 24 TTL digital I/O lines divided into three groups with 8 bits each. xPC Target does not support the external trigger on this board.
xPC TargetBox IO 304 Diamond Onyx-MM Extended temperature -40 to 85°C	**Digital I/O** —48 TTL digital I/O lines divided into six groups with 8 bits each. Each group set to either input or output. xPC TargetBox does not support the counter/timers and external interrupts on this board.

I/O Board Product Name	Description (Continued)
xPC TargetBox IO 305 Diamond Quartz-MM-10 Standard temperature 0 to 60°C	**Counters** — 10 (16 bit) general purpose counters for pulse-train generation and pulse-width measurements. Maximum clock frequency 4 MHz. **Digital I/O** — 8 TTL digital input lines and 8 TTL digital output lines. **Interrupt** — 1 digital interrupt line.
xPC TargetBox IO 306 RealTime Devices DM6814 Standard temperature 0 to 70°C	**Encoders** — 3 up/down (16 bit) counters with a maximum input rate of 1 MHz. **Digital I/O** — 18 TTL digital input-only lines and 6 TTL input or output lines. **Interrupts** — 2 digital interrupt lines. xPC TargetBox does not support the counter/timers on this board.
xPC TargetBox IO 308 Softing CAN-AC2-104 Extended temperature -40 to 85°C	**CAN field bus** — 2 CAN channels. SJA 1000 controller for CAN 2.0A (11 bit standard frames) and CAN 2.0B (29 bit extended frames). Both CAN channels are set to be nonterminated. If you locate one or both of the CAN channels at the end of a CAN bus, you must provide termination by placing the correct resistor directly at the connector pins.

2

Initial Loop-Back Test

The software environment for xPC Target uses two separate computers. Because of this added complexity, installation and configuration are more involved. This chapter includes the following sections:

xPC TargetBox Installation for Loop-Back Testing

When you receive your xPC TargetBox, your first task is to run a self-test target application to test your particular box with your selected I/O options. This section includes the following topics:

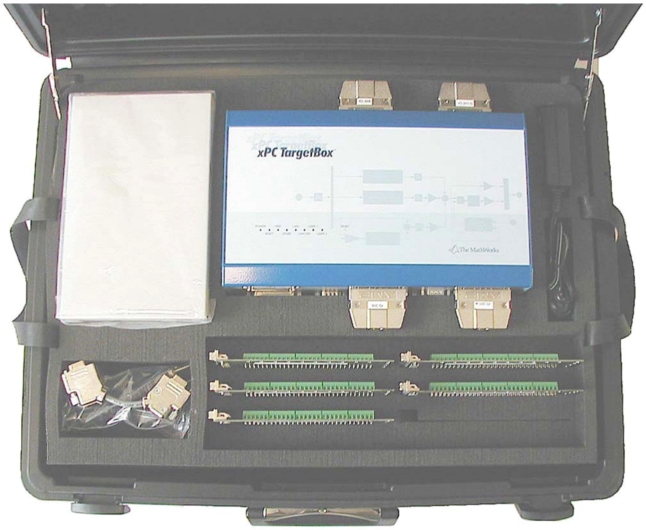
- **Unpacking the Shipping Box** — Remove the xPC TargetBox from the plastic shipping case
- **Running the Self-Test Without a Monitor** — Connect the AC power adaptor and run the custom target application
- **Running the Self-Test with a Monitor** — Connect a monitor to the xPC TargetBox and use the self-test to isolate problems
- **Troubleshooting the Loop-Back Self-Test** —Use information from the self-test to solve problems identified with this test
- **Removing Test Dongles** — Remove the test dongles for normal use with xPC Target

Unpacking the Shipping Box

The xPC TargetBox is delivered in a plastic shipping case. This shipping case includes everything you need to power up the xPC TargetBox and run the self-test.

1 Place the shipping case on a flat surface so you can read the labels on either side of the shipping case handle.

2 Press the bottom of both latches and open the top lid.



Contents of the top layer — xPC TargetBox with attached test dongles, Read First document, external floppy drive, external AC power adaptor, AC power cord, and screw terminal boards.

3 Remove the external floppy drive to uncover the floppy drive cable.

Contents of floppy drive compartment — External floppy drive cable.

4 Remove the upper layer by its handles.

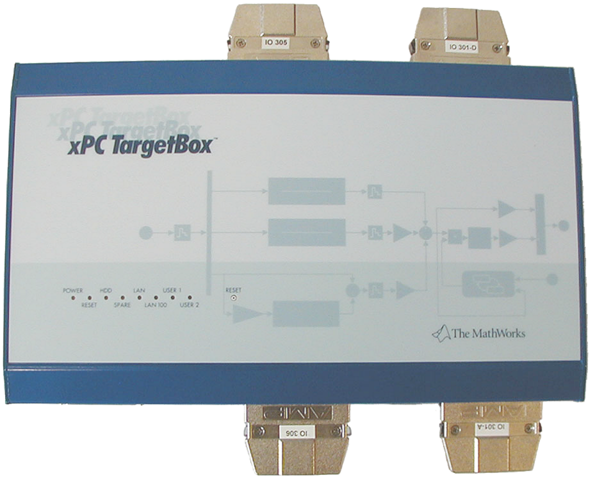
Contents of the lower layer — I/O cables, *Getting Started with xPC TargetBox* manual, MPL hardware reference manual, and I/O reference manuals.

5 If any parts seem to be missing, contact your MathWorks representative.

Running the Self-Test Without a Monitor

A self-test is preinstalled on the xPC TargetBox flash disk. This test is a custom xPC Target application to exercise the I/O options in your system. Run this application to initially test your system, and run this application any time you are not sure about the performance of your xPC TargetBox.

1 From the upper layer, remove the xPC TargetBox. Check visually for any physical damage.



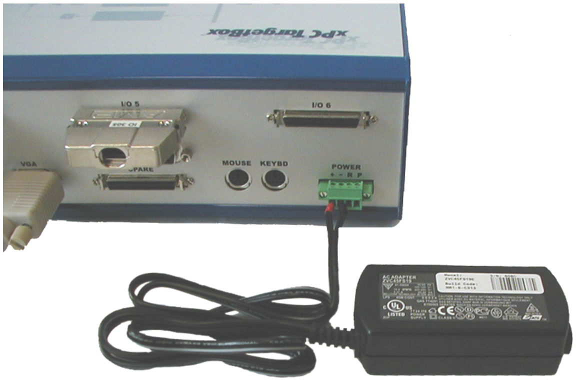
2 Turn the box over, and on the bottom, find the configuration label. Check for the correct xPC TargetBox model and I/O options that you ordered. Also, notice for which industrial standards your xPC TargetBox is certified.

3 Turn the box back over and place on a desk with the top label up. Make sure the test dongles are tightly plugged in.

4 If your xPC TargetBox includes the I/O option 308 (CAN-AC2-104), remove the CAN dongle from the upper compartment, and attach to the CAN 1 and CAN 2 connectors.

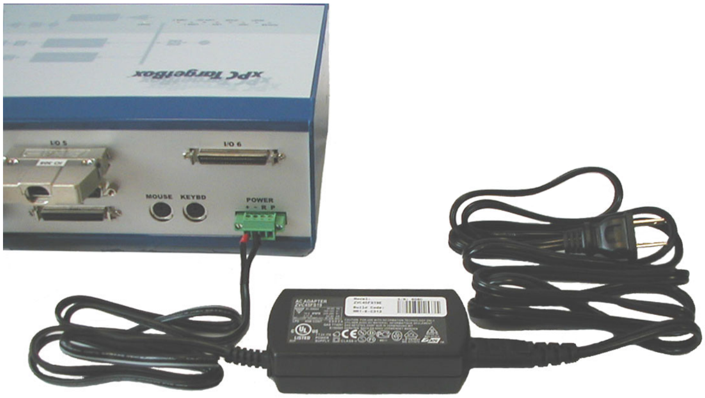
5 From the top layer of the case, remove the external AC power adapter. Check visually for any physical damage.

6 Place the AC power adapter near the xPC TargetBox. Connect the loose end, with the green connector block, to the power input on the xPC TargetBox.



7 From the top compartment, remove the AC power cord.

8 Plug the socket end of the cable into the external AC power adapter.



You are now ready to start the xPC TargetBox for self-testing. Because the xPC TargetBox does not have a power switch, the system starts to operate as soon as you plug the AC power cord into a wall outlet.

9 Plug in the power cord and observe the LEDs on the top of the box. The LEDs light with the following sequence:

- Power on — Red Reset LED turns on momentarily, and then the yellow Power LED turns on and stays on. If you have a network connection, the green LAN and LAN100 LEDs turn on. The green HDD LED turns on when the BIOS accesses the FlashRAM disk.
- DOS test — Both USER 1 and USER 2 LEDs turn on.
- xPC Target self-test — Kernel loads (both USER LEDs off), self-test starts (USER 1 on), self-test is successful (both USER LEDs on), or self-test is unsuccessful (USER 2 on).

● = LED turned on

USER 1	USER 2	
●	●	1. Initial LED test
○	○	1. Self-test not running
●	○	2. Self-test starts
●	●	3a. Self-test finishes successfully
○	●	3b. Self-test finishes unsuccessfully

10 Observe the sequence of LEDs turning on and off.

- If none of the LEDs turns on, immediately unplug the power cord. Check for loose connectors (AC power adapter to xPC TargetBox connector, AC adapter power cord).

  Plug in the power cable again. If the system does not boot up as described above, unplug and contact a MathWorks representative for help.
- If the self-test fails (USER 1 LED does not turn on), then try to isolate the problem using a monitor connected to the xPC TargetBox. See “Running the Self-Test with a Monitor” on page 2-9.

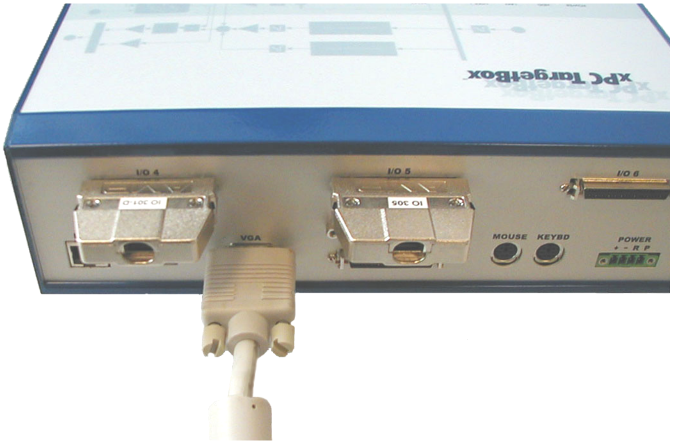
Running the Self-Test with a Monitor

A self-test is preinstalled on the xPC TargetBox flash disk. This test is a custom xPC Target application to exercise the I/O options in your system. Run this application to test your system initially, and run this application any time you are not sure about the performance of your xPC TargetBox.

After you run the self-test without a monitor, you can try running the self-test with a monitor.

1 Set up the xPC TargetBox as described in “Running the Self-Test Without a Monitor” on page 2-4. The AC power cord should be unplugged.

   Find a monitor with a resolution of at least 640x480 pixels, 16 colors, and an SVGA 15-pin (3 rows) connector. Connect the VGA cable of the monitor to the VGA connector on the back panel of the xPC TargetBox. Plug in the monitor power cable and turn on the monitor.

   You are now ready to start the xPC TargetBox for self-testing. Because the xPC TargetBox does not have a power switch, the system starts to operate as soon as you plug the AC power cord into a wall outlet.

2 Plug in the AC power cord and observe the LEDs on the top of the box and the monitor screen.

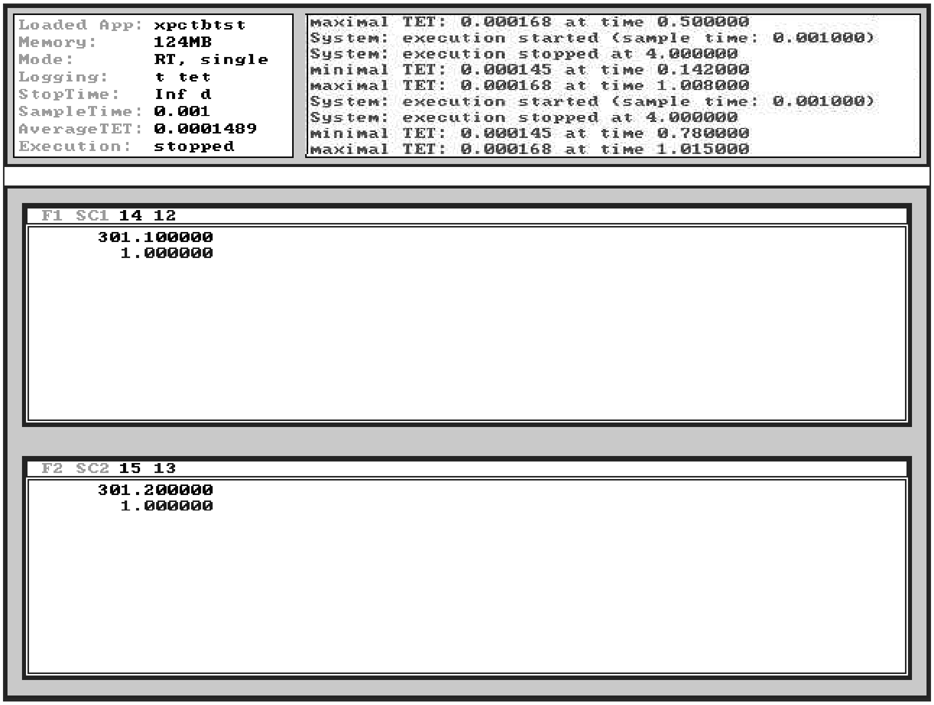
   The sequence of LEDs turning on and off should follow the sequence described in “Running the Self-Test Without a Monitor” on page 2-4, and you should see a display on the monitor for the first time. The initial content you see on the screen depends on the model of your xPC TargetBox.

   xPC TargetBox 106/206 — The monitor shows the typical system (BIOS) boot messages (Detecting RAM, finding media devices (floppy, flash), and displaying found PCI devices). This takes about 10 seconds. After this, DOS is loaded from the internal flash disk, and the stand-alone xPC Target self-test application, which has been loaded on the flash disk, begins.

   xPC TargetBox 107/207/108/208 — Because these boxes are equipped with a QuickBoot BIOS that boots in a very short time, any output to the monitor (VGA) is suppressed. The first message that you can see on the monitor is when FreeDOS is loaded. After FreeDOS loads from the internal flash disk, the stand-alone xPC Target self-test application, which has been loaded on the flash disk, begins.

3 Observe the monitor screen.

   The monitor should show a simple graphical interface for the self-test. Confirm that the self-test is running by observing in the upper left window that the running time is increasing. A running self-test is the first major sign that the xPC TargetBox is operating properly.



```
Loaded App: xpctbtst
Memory:     124MB
Mode:       RT, single
Logging:    t tet
StopTime:   Inf d
SampleTime: 0.001
AverageTET: 0.0001489
Execution:  stopped
```

```
maximal TET: 0.000168 at time 0.500000
System: execution started (sample time: 0.001000)
System: execution stopped at 4.000000
minimal TET: 0.000145 at time 0.142000
maximal TET: 0.000168 at time 1.008000
System: execution started (sample time: 0.001000)
System: execution stopped at 4.000000
minimal TET: 0.000145 at time 0.780000
maximal TET: 0.000168 at time 1.015000
```

```
F1 SC1 14 12
      301.100000
        1.000000
```

```
F2 SC2 15 13
      301.200000
        1.000000
```

4 Observe the lower section of the monitor screen.

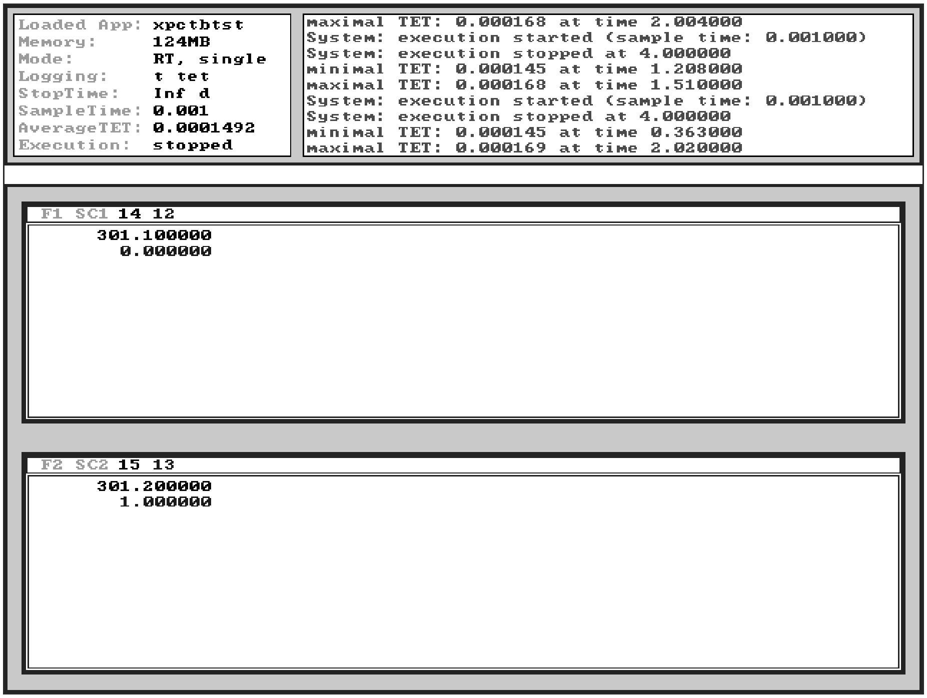
The monitor screen displays a window for each connector attached to an I/O option. The first value is the I/O option number and connector. For example, 301.1 is the analog connector for the IO 301 option, and 301.2 is the digital connector for the same I/O option. Below the I/O option number is a numerical value.

- A numerical value of 1 indicates that the loop-back self-test was successful for that particular I/O option. If all numerical values are 1, the self-test is running successfully and your xPC TargetBox is fully functional. Unplug the power cord and proceed to the next section, “Removing Test Dongles” on page 2-15.
- If one of the numerical values is 0, a failure was detected when exercising a particular I/O option in the loop-back mode. Unplug the power cord and proceed to “Troubleshooting the Loop-Back Self-Test” following.

Troubleshooting the Loop-Back Self-Test

The self-test exercises the I/O options in your xPC TargetBox and identifies problems. With information from the self-test you can than locate where the problems occurred.

1 On the target screen, find the window or windows where a value of `0` is shown. For example, the screen below shows that IO option 301 analog (`301.10000`) failed (`0.000000`) the test.



```
Loaded App:  xpctbtst
Memory:      124MB
Mode:        RT, single
Logging:     t tet
StopTime:    Inf d
SampleTime:  0.001
AverageTET:  0.0001492
Execution:   stopped
```

```
maximal TET: 0.000168 at time 2.004000
System: execution started (sample time: 0.001000)
System: execution stopped at 4.000000
minimal TET: 0.000145 at time 1.208000
maximal TET: 0.000168 at time 1.510000
System: execution started (sample time: 0.001000)
System: execution stopped at 4.000000
minimal TET: 0.000145 at time 0.363000
maximal TET: 0.000169 at time 2.020000
```

```
F1 SC1 14 12
      301.100000
        0.000000
```

```
F2 SC2 15 13
      301.200000
        1.000000
```

2 Write down the board names that show a numerical value of `0`.

3 Unplug the AC power cord.

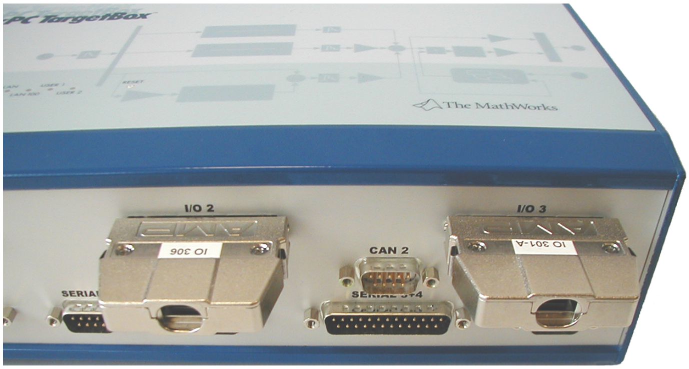
4 Turn the xPC TargetBox over and find the configuration label on the bottom. Look for a listing of I/O options for this box.

The I/O options are attached to the connectors labeled I/O 1 through I/O 6, CAN1, and CAN2. Using the following table and the board name you wrote down, determine the corresponding I/O option and connector for this board.

Board Name	**IO Option**
Diamond MM-32-AT	IO 301
Diamond Ruby-MM-1612	IO 302
Diamond Ruby-MM-416	IO 303
Diamond Onyx-MM	IO 304
Diamond Quartz-MM-10	IO 305
RealTime Devices DM6814	IO 306
Softing CAN-AC2-104	IO 308

5 Using the I/O option number, locate the connector number on the configuration label, and then locate the corresponding I/O connector on the front or rear panel of the xPC TargetBox. The connectors are marked, on the front and rear panels of an xPC TargetBox, I/O 1 through I/O 6 and CAN1 through CAN3.

6 Press the two latches on the test dongle connected to the corresponding I/O connector, remove it from the I/O connector, and then connect it again.



Removing and reattaching a test dongle ensures that the dongle is not loose.

7 Repeat the above step for each test dongle with a corresponding self-test that failed.

8 Power up the system again and let the system self-test run another time. If some numerical values in the target screen still show values of 0, unplug the system and contact Technical Support at The MathWorks for help.

xPC TargetBox Installation for Regular Use

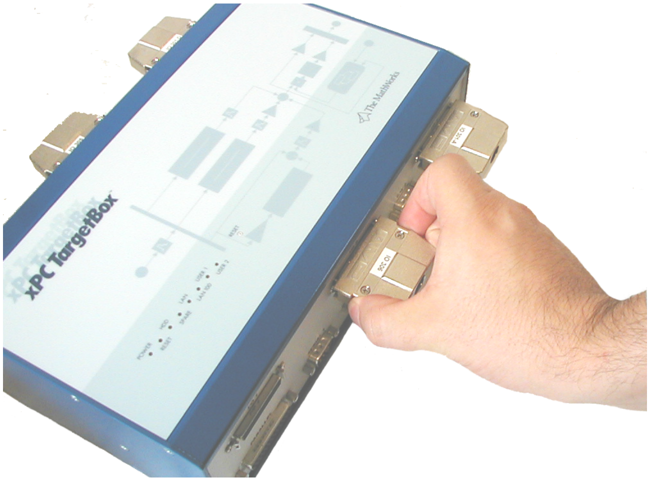
If you completed the self-test successfully, you will have already connected a monitor to the xPC TargetBox. To further use your xPC TargetBox, you need to attach an external floppy disk drive, keyboard, and optionally a mouse. This section includes the following topics:

- **Removing Test Dongles** — Remove and save test dongles for later troubleshooting of I/O boards
- **Connecting Additional Peripherals** —Connect a mouse and keyboard to directly interact with the target application running on the xPC TargetBox
- **Connecting the External Floppy Disk Drive** — Connect the external floppy drive and copy files from the host PC to the xPC TargetBox

Removing Test Dongles

After successfully running the self-test you can remove the test dongles from the xPC TargetBox.

1. Unplug the AC power cord.
2. Press the two latches on the sides of each dongle at the same time and remove the dongle from the connector.



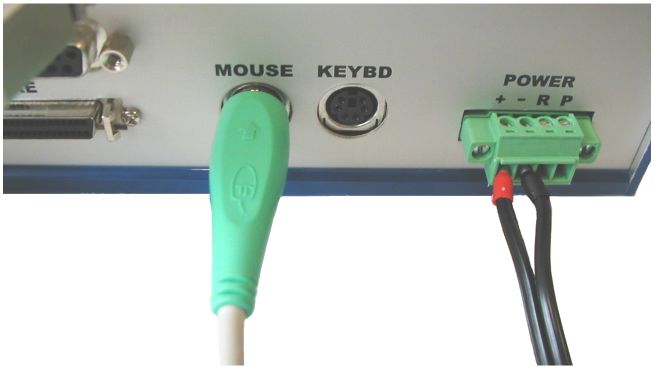
3 Store the test dongles you removed in the lower compartment of the shipping case for later use.

If you encounter a problem with an I/O option in the future you will need these test dongles to narrow down the problem.

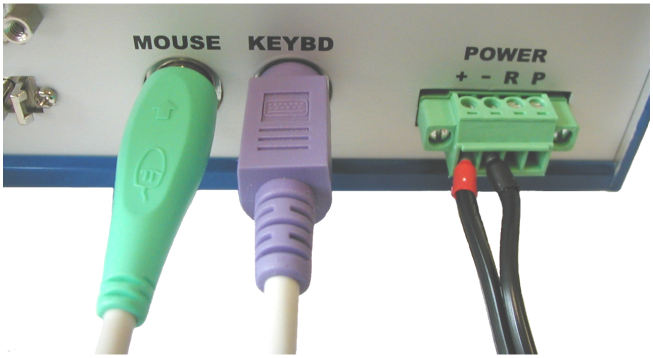
Connecting Additional Peripherals

When using Boot-Floppy and DOSLoader mode, the xPC TargetBox is connected to a host PC. From this host PC you can control and interact with the target application running on the xPC TargetBox. However, when using StandAlone mode, there is no connection to the host PC. In this case, interaction is only by using a mouse and keyboard connected to the xPC TargetBox.

1 Use a mouse with a PS/2 connection and attach to the back panel of your xPC TargetBox.



2 USe a keyboard with a PS/2 connection and attach to the back panel of your xPC TargetBox.



With a mouse and keyboard attached to your xPC TargetBox, you can use the target command-line interface to interact with target applications. See “Target PC Command-Line Interface” in Chapter 1 of the xPC Target Getting Started documentation.

Connecting the External Floppy Disk Drive

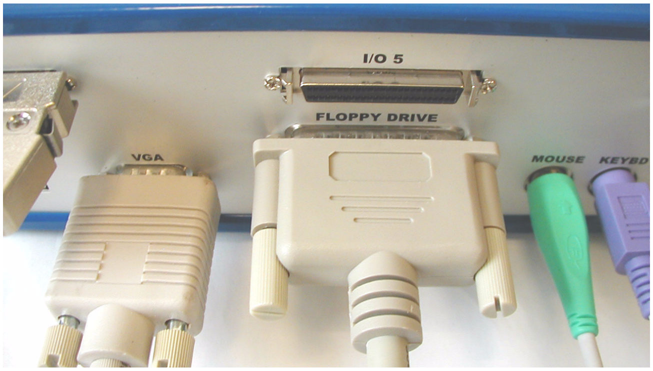
You use the external floppy disk drive to boot the xPC Target kernel, and with DOSLoader and StandAlone modes, copy the xPC Target kernel and target application to the flash disk. With DOSLoader and StandAlone mode, you can copy the files, and then remove the disk drive.

1 In the shipping case and from the upper layer, remove the external floppy disk drive. Check it visually for physical damage.

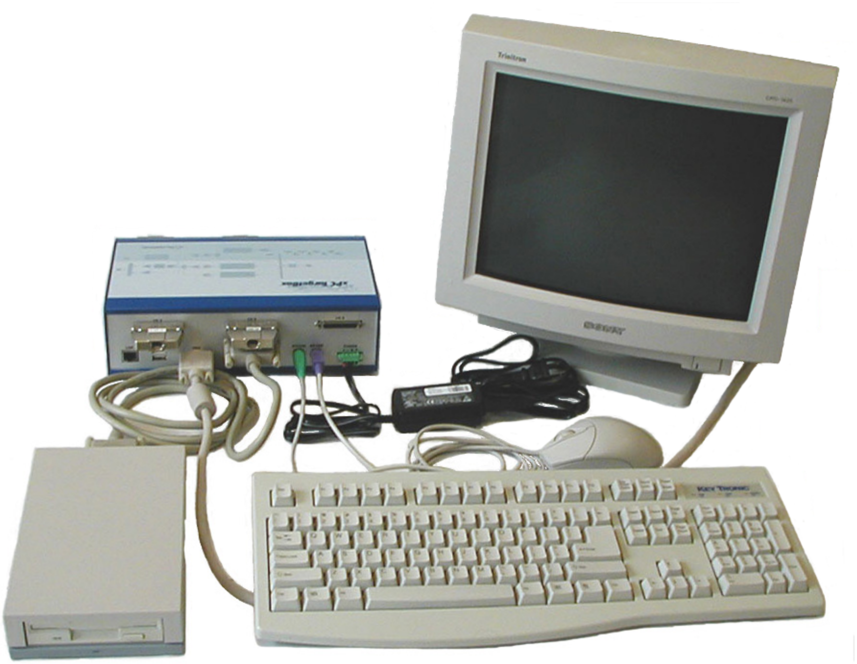
2 From the compartment below the disk drive, remove the external disk drive cable.

3 Connect the correct end of the cable to the connector on the back end of the external floppy disk drive. Tighten the security screws.

4 On the back plate of the xPC TargetBox, locate the external floppy disk drive connector. This special 25-pin port connector is clearly marked. Connect the other end of the external floppy disk drive cable to this connector and tighten the security screws.



5 Leave the xPC TargetBox unpowered.



The next step is to connect the xPC TargetBox to a host computer using either a serial or network connection. See “Serial Communication” on page 3-3 or “Network Communication” on page 3-6.

Connecting Hardware to the xPC TargetBox

Every practical xPC Target and xPC TargetBox application consists of several physical connections to your hardware under test (plant, for rapid control prototyping, or controller, for hardware-in-the-loop simulation). These connections are between the xPC TargetBox I/O options (I/O boards) and the hardware under test. This section describes how to make these connections.

- **Introduction** — Determine the electrical and mechanical limitations of your hardware and xPC TargetBox I/O options before making any connections
- **Configuration Label for I/O Options** — Identify all I/O options and associated I/O connectors
- **Planning I/O Hardware Connections** — Read hardware manuals and draw a diagram of the connections
- **Connecting Hardware to Screw Terminal Boards** — Attach cables between the xPC TargetBox and screw terminal boards

Introduction

Your xPC TargetBox is shipped with external cables and screw terminal boards. You use these boards to connect your equipment to the I/O boards in your system. The label on the bottom of the xPC TargetBox indicates the I/O options in the system and the connector ports these options are connected to.

Caution You have to be extremely careful that you do not damage an I/O option or even the entire xPC TargetBox. You can damage an I/O option by making a wrong connection or by not connecting hardware according to the specifications of the I/O option. Because the physical connections depend on your application, you must make sure on your own that the electrical and mechanical connections are within the boundaries of the I/O options.

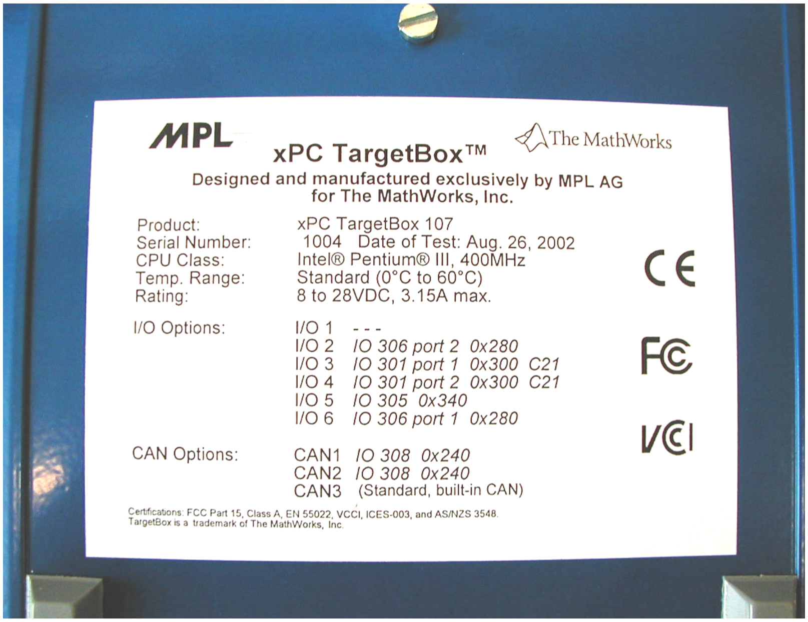
External connectors for I/O options — Each I/O option installed in your xPC TargetBox is accessible through one or two external I/O connectors on the front and rear panels of the xPC TargetBox. These connectors are clearly marked I/O 1 through I/O 6 and CAN1 through CAN3. The first six connectors are of type SCSI-II with 50 pins while the three CAN connectors have 9 pins. The layout of the 50 pins depends on which I/O option is internally connected to which I/O connector on the outside of the xPC TargetBox.

Configuration Label for I/O Options

The layout of the I/O connectors on the xPC TargetBox is the same for all models and is independent of the I/O options installed in your xPC TargetBox. The mapping of which I/O option is connected to which I/O connector is listed on the xPC TargetBox configuration label on the bottom of the xPC TargetBox.

Warning Always unplug the xPC TargetBox when locating the label.

Three additional copies of these labels are in the Read First envelope found in the upper compartment of the shipping case. These additional labels are for your reference. You can either use them for your documentation materials or attach them to your application setup at a convenient location.



MPL

The MathWorks

xPC TargetBox™

Designed and manufactured exclusively by MPL AG
for The MathWorks, Inc.

Product:	xPC TargetBox 107
Serial Number:	1004 Date of Test: Aug. 26, 2002
CPU Class:	Intel® Pentium® III, 400MHz
Temp. Range:	Standard (0°C to 60°C)
Rating:	8 to 28VDC, 3.15A max.
I/O Options:	I/O 1 - - -
	I/O 2 *IO 306 port 2 0x280*
	I/O 3 *IO 301 port 1 0x300 C21*
	I/O 4 *IO 301 port 2 0x300 C21*
	I/O 5 *IO 305 0x340*
	I/O 6 *IO 306 port 1 0x280*
CAN Options:	CAN1 *IO 308 0x240*
	CAN2 *IO 308 0x240*
	CAN3 (Standard, built-in CAN)

CE

FC

VCI

Certifications: FCC Part 15, Class A, EN 55022, VCCI, ICES-003, and AS/NZS 3548.
TargetBox is a trademark of The MathWorks, Inc.

Product — Product name (xPC TargetBox) and model number (106, 107, 108, 206, 207, 208)

Serial Number —Serial number with date of system test

CPU class — CPU name (Pentium II, Pentium III) and speed (266 MHz, 400 MHz, 700 MHz)

Temp. Range — Standard (0°C to 60°C), Extended models 206 and 207 (-40°C to 75°C), or Extended model 208 ((-40°C to 65°C)

Rating — Always 8 to 28 VDC, 3.15 A max

I/O Options — Each line on the label corresponds to a connector on the xPC TargetBox (I/O 1 through I/O 6) with the following fields:

- The I/O option associated with the connector (IO 301, IO 302, IO 303, IO 304, IO 305, IO 306).
- For options with more than one connector, a distinction between connectors (port 1, port 2).
- The base address of the I/O option (0x220, 0x240, 0x280, 0x300).
- If an I/O option (I/O board) includes jumper settings, the label includes a unique code to identify in which position the hardware jumpers are inserted. See table below.

CAN Options — Each line on the label corresponds to a CAN connector on the xPC TargetBox (CAN1 through CAN3).

- The I/O option associated with the connector (IO 308)
- The base address of the I/O option (0x220, 0x240, 0x280, 0x300)

The following table lists all I/O options with the number of associated I/O connectors and possible jumper codes.

.

Table 2-1: Jumper Codes for I/O Options

I/O Option	Number of Connectors	Jumper Codes
IO 301 Diamond MM-32-AT	2	**Analog input (A/D) jumper codes** (first number after the C) • C1# — 32 single-ended • C2# — 16 differential • C3# — 16 single-ended/8 differential **Analog output (D/A) jumper codes** (second number after the C) • C#1— ±10 V • C#2— ±5 V • C#3 — 0-10 V • C#4 — 0-5 V Default configuration is C21.
IO 302 Ruby-MM-1612	1	**Analog output (D/A) jumper codes** (first number after the C for the first 8 channels) • C1#— ±10 V • C2#— ±5 V • C3#— ±2.5 V • C4#— 0-10 V • C5#— 0-5 V • C6#— 0-2.5 V The third character (second number) in the code indicates the configuration of the second 8 channels. The default configuration is C11.

Table 2-1: Jumper Codes for I/O Options

I/O Option	Number of Connectors	Jumper Codes (Continued)
IO 303 Ruby-MM-416	1	**Analog output (D/A) jumper codes for first channel** • C1#— ±10 V • C2#— ±5 V • C3# — 0-10 V The third, fourth, and fifth characters in the code indicate the configuration for the second, third, and fourth D/A channels. The default configuration is C1111
IO 304 Onyx-MM	2	None
IO 305 Quartz-MM-10	1	None
IO 306 DM6814	2	None
IO 308 CAN	Special connectors	None

If you have identified all I/O options and associated I/O connectors on your xPC TargetBox, you are ready to connect the xPC TargetBox through its I/O connectors to your hardware under test. See “Planning I/O Hardware Connections” on page 2-25.

Planning I/O Hardware Connections

To ensure correct operation, the shipping case includes an I/O board user's manual from the original manufacturer for each particular I/O option, in the lower compartment of the xPC TargetBox shipping case. Make sure to consult these manuals to verify that your I/O connections comply with the manufacturers' specifications.

1. Make a drawing of the I/O connections you want to establish.
2. Verify that the voltage levels, current loads, and so forth are in accordance with the specification of a particular I/O option.

Note Neither The MathWorks nor MPL will be liable for any damage resulting from connecting hardware to the xPC TargetBox in any way that is not in compliance with the I/O board specification from the original manufacturer.

---

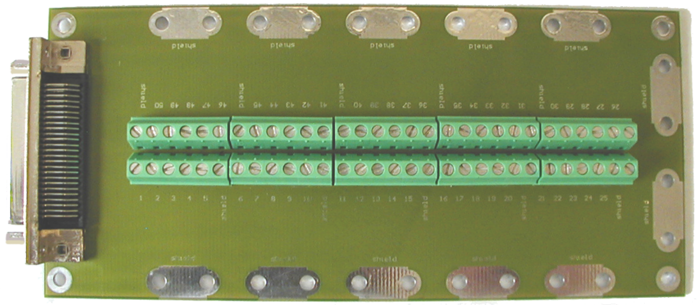
3. From the following list determine the hardware manuals for your particular I/O options.
   - IO 301 — Diamond Systems DIAMOND-MM-32 AT User Manual V2.61
   - IO 302 — Diamond Systems RUBY-MM-1612 User Manual V1.1
   - IO 303 — Diamond Systems RUBY-MM-416 User Manual V1.1
   - IO 304 — Diamond Systems ONYX-MM-XT User Manual V1.4
   - IO 305 — Diamond Systems QUARTZ-MM User Manual V1.5
   - IO 306 — DM5814/DM6814 User's Manual from Real Time Devices
   - IO 308 — Softing CAN-AC2-104 User Manual Version 4.0

Connecting Hardware to Screw Terminal Boards

Be extremely careful when making connections to the terminal boards. For your convenience the shipping case includes one or two I/O cables and one or two screw terminal boards (depending on the I/O option) for each I/O option installed in your xPC TargetBox. You can use these parts to quickly and conveniently make the connections between your xPC TargetBox and your hardware under test.

Note Unplug the xPC TargetBox when connecting any I/O cables and screw terminal boards to the an xPC TargetBox.

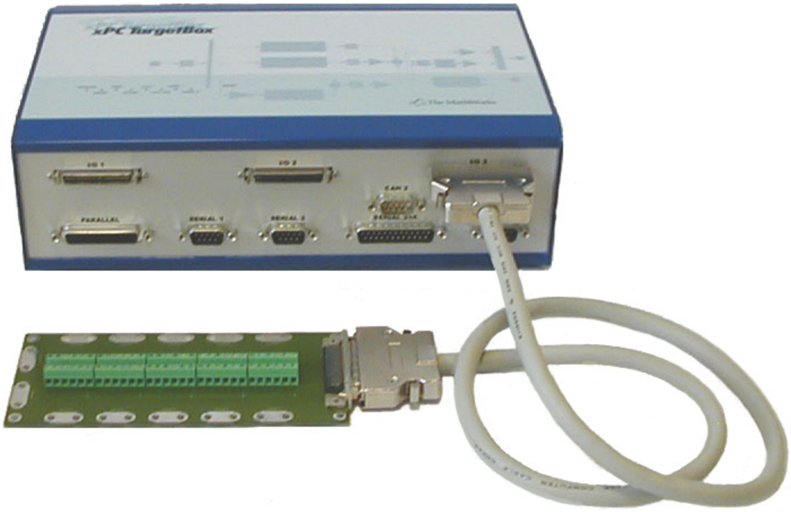
1 You can find the screw terminal boards in the shipping case at the front of the top compartment.



2 You can find the I/O cables in the lower compartment of the shipping case.

3 Plug one end of the I/O cable into an I/O connector. Press both latches at the plug at the same time and insert the plug. Check that the plug is properly connected to the I/O connector.

4 Connect the screw terminal board to the other side of the I/O cable in the same way. Make sure that the screw terminal does not have any wires connected to it and especially make sure that no physical connections exist between the screw terminal board and the hardware under test, which could possibly be powered up and therefore immediately damage I/O options or the xPC TargetBox.



It is good practice to do some loop-back testing of an I/O option before connecting it to your real hardware under test. See Chapter 4, “xPC TargetBox I/O Options.”

The pin layout for each I/O option is provided in tables found in Chapter 4, “xPC TargetBox I/O Options.”

3

Installation and Configuration

The software environment for xPC Target uses two separate computers. Because of this added complexity, installation and configuration are more involved. This chapter includes the following sections:

xPC Target Installation

Before you can build and download a target application from the host PC to the xPC TargetBox, you need to properly install any MathWorks software. This section includes the following topics:

- **Installing xPC Target** — Install MATLAB, Simulink, Stateflow and Stateflow Coder (optional), Real-Time Workshop, xPC Target, and a third-party C/C++ compiler on the host PC.
- **Installing xPC Target Embedded Option** — Install xPC Target Embedded Option on the host PC.

Installing xPC Target

You need to have an xPC Target license for each xPC TargetBox you purchase. The xPC Target software is installed entirely on the host PC. Installing software on the xPC TargetBox is not necessary.

For information on installing software on the host computer, see the section "Installation on the Host PC" in the xPC Target Getting Started documentation.

Installing xPC Target Embedded Option

If you have an xPC Target Embedded Option license, you can use the features of the embedded option to

- Boot the kernel from the FlashRAM module.
- Create stand-alone applications and deploy then on an xPC TargetBox.

For information on installing software on the host computer, see the section "Installation on the Host PC" in the xPC Target Getting Started documentation.

Serial Communication

Before you can create and run a target application, you need to set up the connection between your host PC and xPC TargetBox. You can use either serial or network communication. This section includes the following topics:

- **Hardware for Serial Communication** — Connect a null modem cable
- **Environment Properties for Serial Communication** — Select the host PC COM port and baud rate

Hardware for Serial Communication

Before you install the xPC Target software and configure it for serial communication, you must install the following hardware:

Null modem cable — Connect the host and target computers with the null modem cable supplied by The MathWorks with the xPC Target software. You can use either the COM1 or COM2 port.

Environment Properties for Serial Communication

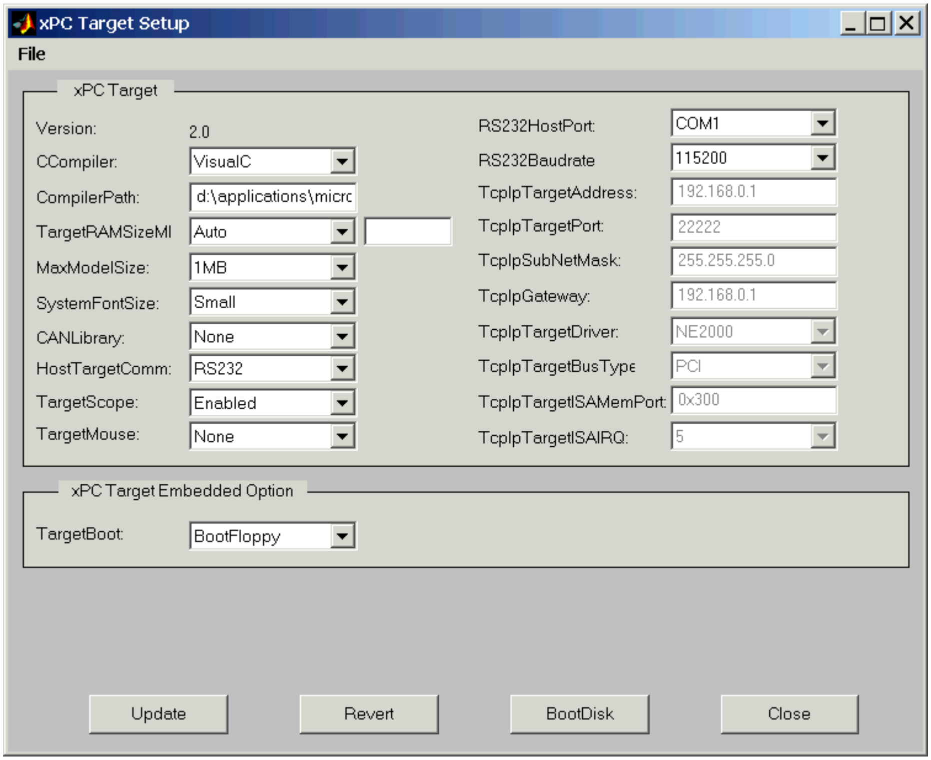
The xPC Target environment is defined by a group of properties. These properties give xPC Target information about the software and hardware products that it works with.

After you install xPC Target, you can set the environment properties for the host and target computers. You need to change these properties before you can build and download a target application.

1 In the MATLAB Command Window, type

```
xpcsetup
```

The **xPC Target Setup** window opens.



xPC Target

Version:	2.0	RS232HostPort:	COM1
CCompiler:	VisualC	RS232Baudrate	115200
CompilerPath:	d:\applications\micrc	TcpIpTargetAddress:	192.168.0.1
TargetRAMSizeMl	Auto	TcpIpTargetPort:	22222
MaxModelSize:	1MB	TcpIpSubNetMask:	255.255.255.0
SystemFontSize:	Small	TcpIpGateway:	192.168.0.1
CANLibrary:	None	TcpIpTargetDriver:	NE2000
HostTargetComm:	RS232	TcpIpTargetBusType	PCI
TargetScope:	Enabled	TcpIpTargetISAMemPort:	0x300
TargetMouse:	None	TcpIpTargetISAIRQ:	5

xPC Target Embedded Option

TargetBoot: BootFloppy

The **xPC Target Setup** window has two sections:

- xPC Target
- xPC Target Embedded Option

The xPC Target Embedded Option gives you the additional **TargetBoot** choices of **DOSLoader** and **StandAlone** mode.

2 From the **CCompiler** list, select either `VisualC` or `Watcom`.

3 In the **CompilerPath** box, enter the root path where you installed your C/C++ compiler.

4 From the **HostTargetComm** list, select `RS232`.

5 From the **RS232HostPort** list, select either `COM1` or `COM2` for the connection on the host PC. xPC Target automatically determines the COM port you use on the xPC TargetBox.

6 When you finish changing the properties, click the **Update** button.

   xPC Target updates the environment with the new properties.

You do not have to exit and restart MATLAB after making changes to the xPC Target environment, even if you change the communication between the host and target from RS232 to TCP/IP. However, you have to recreate the target boot disk and rebuild the target application from the Simulink model.

For more information on the xPC Target environment, see Chapter 8, “Software Environment,” in the xPC Target User’s documentation.

Your next task is to create a target boot disk. See “Target Boot Disk” on page 3-10.

Network Communication

Before you can create and run a target application, you need to set up the connection between the host and target computers. You can use either serial or network communication. This section includes the following topics:

- **Hardware for Network Communication** — Connect to a LAN or use a crossover cable
- **Environment Properties for Network Communication** — Enter the IP address and network information for the xPC TargetBox

This manual does not include information for installing network cards or the TCP/IP protocol on your host computer. For correct installation and setup of your network cards and the TCP/IP protocol, contact your system administrator.

Hardware for Network Communication

You must install the following hardware before you install the xPC Target software and configure it for network communication:

1. When using the xPC TargetBox with TCP/IP, you must have a network adapter card correctly installed on the host PC. On the other hand, the xPC TargetBox has an Intel 82559ER Ethernet controller onboard.

2. Connect the host and target computers with an unshielded twisted pair (UTP) cable to your local area network (LAN). On the front panel of the xPC TargetBox, locate the 10/100 MB/s Ethernet connector, clearly marked. Plug the target side of the Ethernet cable into the Ethernet plug of the xPC TargetBox.

   You can also directly connect your computers. Use a crossover UTP cable with RJ45 connectors.

Environment Properties for Network Communication

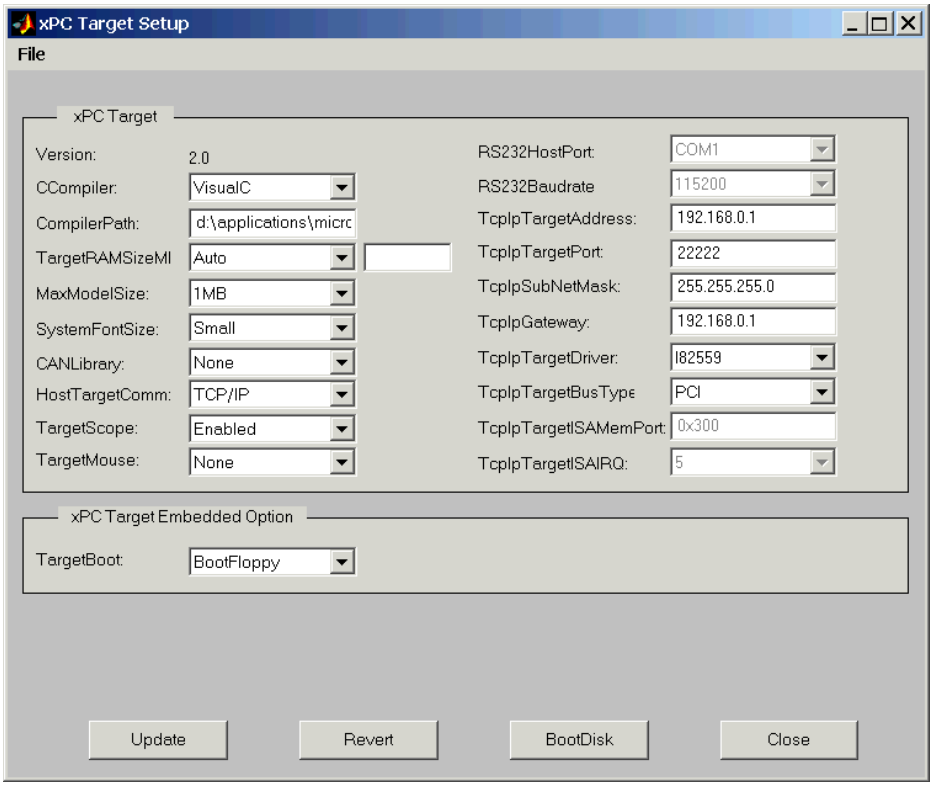
The xPC Target environment is defined by a group of properties. These properties give xPC Target information about the software and hardware that it works with.

After you have installed xPC Target, you can set the specific environment properties for your host and target computers. You must change these environment properties before you can build and download a target application.

1 In the MATLAB Command Window, type

```
xpcsetup
```

The **xPC Target Setup** window opens.



xPC Target

Version:	2.0	RS232HostPort:	COM1
CCompiler:	VisualC	RS232Baudrate	115200
CompilerPath:	d:\applications\micrc	TcpIpTargetAddress:	192.168.0.1
TargetRAMSizeMI	Auto	TcpIpTargetPort:	22222
MaxModelSize:	1MB	TcpIpSubNetMask:	255.255.255.0
SystemFontSize:	Small	TcpIpGateway:	192.168.0.1
CANLibrary:	None	TcpIpTargetDriver:	I82559
HostTargetComm:	TCP/IP	TcpIpTargetBusType	PCI
TargetScope:	Enabled	TcpIpTargetISAMemPort:	0x300
TargetMouse:	None	TcpIpTargetISAIRQ:	5

xPC Target Embedded Option

TargetBoot: BootFloppy

The **xPC Target Setup** window has two sections:

- xPC Target
- xPC Target Embedded Option

The xPC Target Embedded Option gives you the additional **TargetBoot** choices of **DOSLoader** and **StandAlone** mode.

2 From the **CCompiler** list, select either `VisualC` or `Watcom`.

3 In the **CompilerPath** box, enter the path to where you installed your C/C++ compiler.

4 From the **HostTargetComm** list, select `TCP/IP`.

The TCP/IP text boxes become active.

You must enter the following properties with the correct values according to your LAN environment. Ask your LAN system administrator for values for the following settings:

- **TcpIpTargetAddress** — This is the IP address for your xPC TargetBox. An example of an IP address is `192.168.0.1`.
- **TcpIpSubNetMask** — This is the Subnet Mask address of your LAN. An example of a Subnet Mask address is `255.255.0.0`.

You enter the following properties depending on your specific circumstances:

- **TcpIpTargetPort** — This property is set by default to `22222`. This value should not cause any problems, because this number is higher than the reserved area (`telnet`, `ftp`, ...) and it is only of relevance on the xPC TargetBox. If necessary this property value can be changed to any value higher than `20000` and less than `65536`.
- **TcpIpGateway** — This property is set by default to `255.255.255.255`. This means that you do not use a gateway to connect to your xPC TargetBox.

  If you communicate with the xPC TargetBox from within your LAN, you might not need to define a gateway and change this setting.

  If you communicate from a host PC located in a LAN different from your xPC TargetBox, you need to define a gateway and enter its IP address.

This is especially true if you want to work over the Internet. Ask your system administrator for the IP address of the appropriate gateway.

The following properties are specific for the Ethernet card on your xPC TargetBox:

- **TcpIpTargetDriver** — Because the xPC TargetBox includes an Intel 82559ER Ethernet controller, from the list, select `I82559`.
- **TcpIpTargetBusType** — Select `PCI`.
- **TcpIpISAMemPort and TcpIpISAIRQ** — Because the xPC TargetBox uses a PCI Ethernet card, you do not need to enter values for these properties.

5 When you finish changing the properties, click the **Update** button.

xPC Target updates the environment with the new properties.

You do not have to exit and restart MATLAB after making changes to the xPC Target environment, even if you change the communication between the host and target from RS-232 to TCP/IP. However, you have to recreate the target boot disk and rebuild the target application from the Simulink model.

For more information on the xPC Target Environment, see “Software Environment” on page 8-1 in the xPC Target User’s documentation.

Your next task is to create a target boot disk. See “Target Boot Disk” following.

Target Boot Disk

The target boot disk includes the xPC Target kernel specific for either serial or network communication. If you select StandAlone mode, the target boot disk also includes the target application. This section includes the following topics:

- **Creating a Target Boot Disk with a Graphical User Interface**— Use the `xpcsetup` command to open a dialog box and create a boot disk
- **Creating a Target Boot Disk with Command-Line Interface** — Use the `xpcbootdisk` command to create a boot disk from the MATLAB Command Window

Creating a Target Boot Disk with a Graphical User Interface

You use the target boot disk to load and run the xPC Target kernel. After you make changes to the xPC Target environment properties, you need to create or update a target boot disk.

To create a target boot disk for the current xPC Target environment, use the following procedure. Alternatively, see “Creating a Target Boot Disk with Command-Line Interface” on page 3-12.

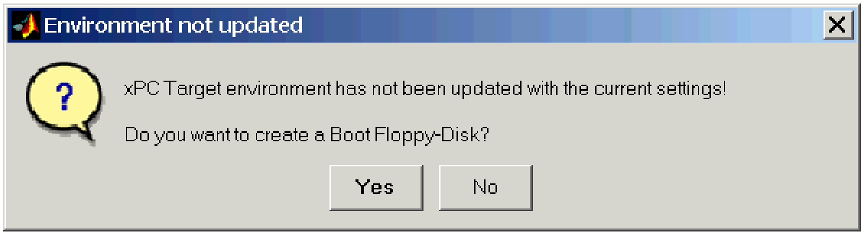
1 In the MATLAB Command Window, type

```
xpcsetup
```

The xPC Target Setup window opens.

2 Click the **BootDisk** button.

If you didn’t update the current settings, the following message box opens.



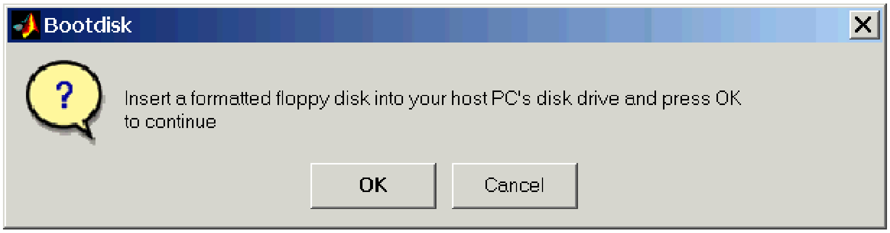
Environment not updated

xPC Target environment has not been updated with the current settings!

Do you want to create a Boot Floppy-Disk?

Click **No**. Click the **Update** button, and then click the **BootDisk** button again.

After you update the current properties and click the **BootDisk** button, the following message box opens.

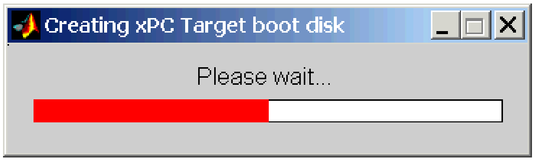


Insert a formatted floppy disk into your host PC's disk drive and press OK to continue

3 Insert a formatted 3.5 inch floppy disk into the host PC disk drive, and then click **OK**.

Note All data on the disk is erased.

xPC Target displays the following dialog box while creating the boot disk. The process takes about 1 to 2 minutes.



Please wait...

4 Close the **xPC Target Setup** window.

5 Remove the target boot disk from the host PC disk drive, and insert it into the external disk drive connected to your xPC TargetBox.

Your next task is to test your installation. See "Testing xPC Target Installation" on page 3-13.

Creating a Target Boot Disk with Command-Line Interface

You use the target boot disk to load and run the xPC Target kernel. After you make changes to the xPC Target environment properties, you need to create or update a boot disk.

To create a target boot disk for the current xPC Target environment, use the following procedure:

1. In the MATLAB Command Window, type

   ```
   xpcbootdisk
   ```

   xPC Target displays the following message:

   ```
   Insert a formatted floppy disk into your host PC’s
   disk drive and press any key to continue.
   ```

2. Insert a formatted floppy disk into the host PC disk drive, and then press any key.

   The write procedure starts and, while creating the boot disk, the MATLAB Command Window displays the following status information. On Windows NT systems, the status information is displayed only at the end of the write process.

   ```
   Creating xPC Target boot disk ... Please wait

   xPC Target boot disk successfully created.
   ```

Your next task is to test your installation. See “Testing xPC Target Installation” on page 3-13.

Testing xPC Target Installation

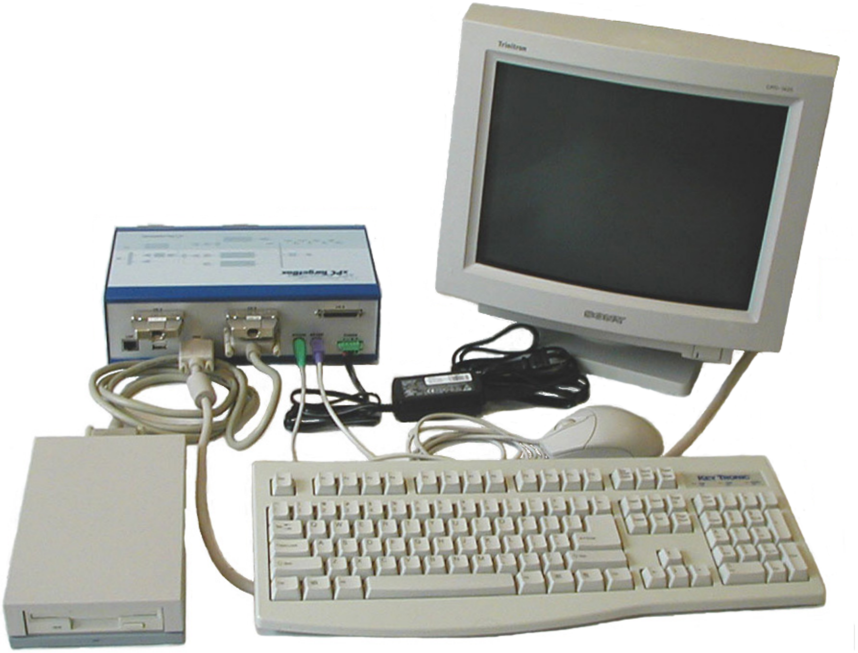
Use this section to troubleshoot connection and communication problems between your host PC and the xPC TargetBox. This section includes the following topics:

- **Powering Up the xPC TargetBox** — Check xPC TargetBox connections to peripherals and power supply
- **Testing the Installation**— Run the `xpctest` command
- **Test 1, Ping Target System Standard Ping**— Test network communication with a standard ping command
- **Test 2, Ping Target System xPC Target Ping**— Test either network or serial communication with the host PC
- **Test 3, Reboot Target Using Direct Call** — Test rebooting the xPC TargetBox from the host PC
- **Test 4, Build and Download Application**— Test downloading and running a target application on the xPC TargetBox

Powering Up the xPC TargetBox

After you create a boot disk in the **xPC Target Setup** dialog box, you are ready to power up the xPC TargetBox again, but this is a good time to check the xPC TargetBox setup:

1. Check that the external AC power supply adapter is properly connected to the xPC TargetBox, the external floppy drive is properly connected, the monitor is connected and powered up, an Ethernet or RS-232 cable is properly connected, and the xPC TargetBox is still unpowered.



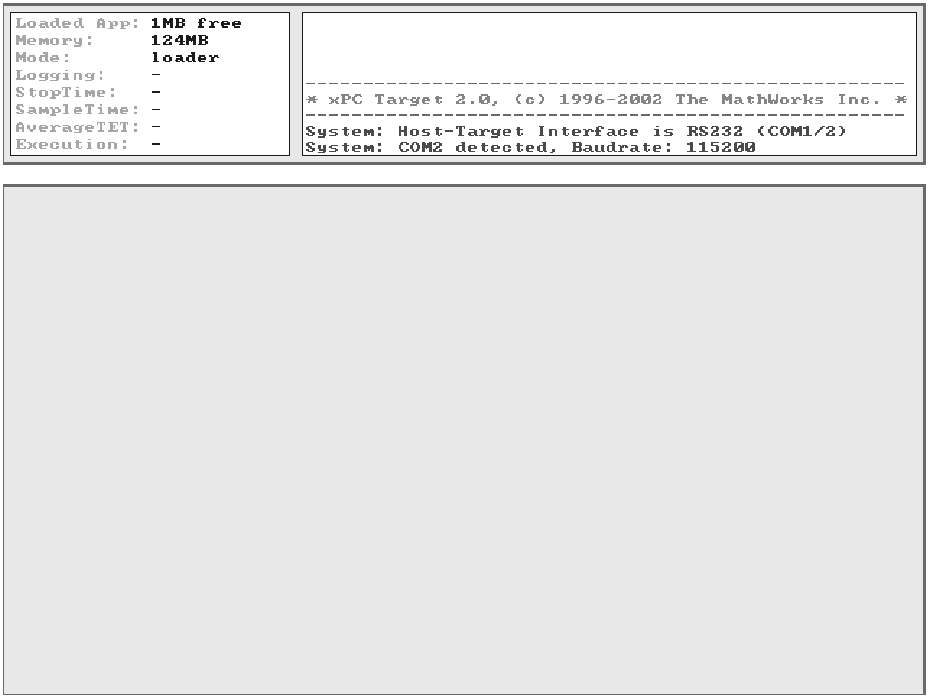
2 Insert the xPC Target boot disk into the external floppy drive.

3 Plug in the AC power cord.

On the top of the xPC TargetBox, check that the Reset LED and then the Power LED turn on.

4 After the BIOS loads (time varies with xPC TargetBox model), the floppy disk drive is accessed (verify by observing the green LED on the front of the external floppy disk drive and listening for an audible sound from the drive).

The xPC TargetBox loads the xPC Target kernel on the disk and as soon as the entire image has been loaded, the xPC Target kernel starts executing.



```
Loaded App: 1MB free
Memory:     124MB
Mode:       loader
Logging:    -
StopTime:   -
SampleTime: -
AverageTET: -
Execution:  -
```

```
-------------------------------------------------------
* xPC Target 2.0, (c) 1996-2002 The MathWorks Inc. *
-------------------------------------------------------
System: Host-Target Interface is RS232 (COM1/2)
System: COM2 detected, Baudrate: 115200
```

The xPC TargetBox and the loaded xPC Target kernel are now ready to accept commands from the host computer through either the RS-232 or Ethernet connection.

Your next step is to test the installation. See “Testing the Installation” following.

Testing the Installation

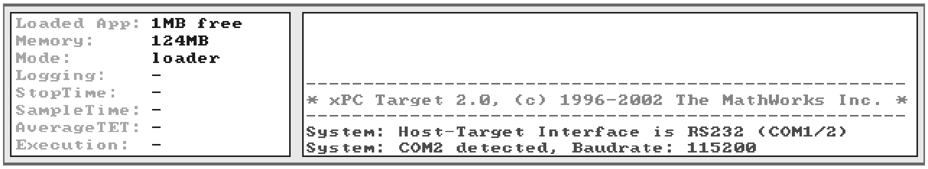
xPC Target uses a test script to test the entire installation. This test checks both the host computer xPC Target setup and the xPC TargetBox by building, downloading, and running a simple test Simulink model.

After you install the xPC Target software, set the environment settings, and create a target boot disk, you can test your installation.

1 Insert your target boot disk into the external disk drive.

2 Press the **Reset** button on the xPC TargetBox.

After loading the BIOS, xPC Target boots the kernel and displays the following screen on the target PC monitor.



```
Loaded App: 1MB free
Memory:     124MB
Mode:       loader
Logging:    -
StopTime:   -
SampleTime: -
AverageTET: -
Execution:  -
```

```
------------------------------------------------------
* xPC Target 2.0, (c) 1996-2002 The MathWorks Inc. *
------------------------------------------------------
System: Host-Target Interface is RS232 (COM1/2)
System: COM2 detected, Baudrate: 115200
```

3 In the **MATLAB Current Directory** window, select a current directory outside the MATLAB root directory.

Note During the build process, Real-Time Workshop does not allow files to be saved within the MATLAB tree root. If you select a current directory within the MATLAB tree, the xPC Target test procedure will fail when trying to build a target application.

4 In the MATLAB Command Window, type

```
xpctest
```

MATLAB runs the test script and displays messages indicating the success or failure of a test. If you use RS-232 communication, the first test is skipped.

```
### xPC Target Test Suite 2.0
### Host-Target interface is: TCP/IP (Ethernet)
### Test 1, Ping target system using standard ping: ... OK
### Test 2, Ping target system using xpctargetping: ... OK
### Test 3, Reboot target using direct call: ....... OK
### Test 4, Build and download xPC Target application: ... OK
### Test 5, Check host-target communication for commands: .. OK
### Test 6, Download xPC Target application using OOP: ... OK
### Test 7, Execute xPC Target application for 0.2s: ... OK
### Test 8, Upload logged data and compare with simulation:. OK
### Test Suite successfully finished
```

If any of the tests fails, see the appropriate test section:

- "Test 1, Ping Target System Standard Ping" on page 3-17
- "Test 2, Ping Target System xPC Target Ping" on page 3-19
- "Test 3, Reboot Target Using Direct Call" on page 3-19
- "Test 4, Build and Download Application" on page 3-20

If all of the subtests are successful, the host computer and xPC TargetBox are properly set up for regular use. Your next steps are to

- Connect the xPC TargetBox to the hardware you are testing through the I/O connector ports. See "Connecting Hardware to the xPC TargetBox" on page 2-20.
- Build and download a target application to the xPC TargetBox. See Chapter 3, "Basic Tutorial," in the xPC Target Getting Started documentation.

Test 1, Ping Target System Standard Ping

If you are using a network connection, this is a standard system ping to your target computer. If this test fails, try troubleshooting with the following procedure:

1. Open a DOS shell, and type the IP address of the xPC TargetBox:

   ```
   ping xxx.xxx.xxx.xxx
   ```

   DOS should display a message similar to the following:

   ```
   Pinging xxx.xxx.xxx.xxx with 32 bytes of data:
   Replay from xxx.xxx.xxx.xxx: bytes-32 time<10 ms TTL=59
   ```

2. Check the messages on your screen.

   Ping command fails — If the DOS shell displays the following message,

```
Pinging xxx.xxx.xxx.xxx with 32 byte of data:
Request timed out.
```

The ping command failed, and the problem might be with your network cables.

To solve this problem, check your network cables. You might have a faulty network cable, or if you are using a coaxial cable, the terminators might be missing.

Ping command fails, but cables are okay — If the cables are okay, the problem might be that you entered an incorrect property in the **Setup** window.

To solve this problem, in the MATLAB Command Window, type

```
xpcsetup
```

Check that **TcpIpTargetAddress**, **TcpIpSubNetMask**, and **TcpIpGateway** have the correct values. On the host PC, open the **xPC Target Setup** dialog box, change the TCP/IP options, click the **Update** button, and update/create a new boot floppy disk. On the xPC TargetBox, reboot with the corrected boot floppy disk.

The xPC TargetBox uses a PCI-bus:

- Check that **TcpIpTargetBusType** is set to PCI instead of ISA.

Ping succeeds, but test 1 with the command xpctest fails — The problem might be that you have incorrect IP and gateway addresses entered in the **Setup** window.

To solve this problem, in the MATLAB Command Window, type

```
xpcsetup
```

Enter the correct addresses. Click the **Update** button. Recreate the target boot disk by inserting a floppy disk into the host disk drive and then clicking the **BootDisk** button.

If you still cannot solve your problem, see “If You Still Need More Help” on page 3-20.

Test 2, Ping Target System xPC Target Ping

This test is an xPC Target ping to your xPC TargetBox. If this test fails, try troubleshooting with the following procedure.

1 In the MATLAB Command Window, type

```
tg=xpc
```

2 Check the messages in the MATLAB window.

   MATLAB should respond with the following messages.

```
xPC Object
   Connected           = Yes
   Application         = loader
```

   Target object does not connect — If you do not get these messages, the problem might be that you have a bad target boot disk.

   To solve this problem, create another target boot disk with a new floppy disk. See "Target Boot Disk" on page 3-10.

If you still cannot solve your problem, see "If You Still Need More Help" on page 3-20.

Test 3, Reboot Target Using Direct Call

This test tries to boot your xPC TargetBox using an xPC Target command. If this test fails, try troubleshooting with the following procedure.

1 In the MATLAB Command Window, type

```
xpctest noreboot
```

   This command reruns the test without using the `reboot` command and displays the message

```
### Test 3, Reboot target using direct call: ... SKIPPED
```

2 Observe the messages in the MATLAB Command Window during the build process.

If you still cannot solve your problem, see "If You Still Need More Help".

Test 4, Build and Download Application

This test tries to build and download the model `xpcosc.mdl`. If this test fails, try troubleshooting with the following procedure:

1. In the MATLAB Command Window, check the error messages.

   These messages help you locate where there is a problem.

2. If you get the error message

   ```
   xPC Target loader not ready
   ```

   reboot your xPC TargetBox. This error message is sometimes displayed even if the target screen shows the loader is ready.

3. Open the xPC Target setup dialog box. Type

   ```
   xpcsetup
   ```

4. Check the path to the C compiler. A common error when creating a target application is setting the path to the C compiler incorrectly.

If you still cannot solve your problem, see “If You Still Need More Help”.

If You Still Need More Help

If you cannot solve your problem, contact The MathWorks directly for help.

Internet	`http://www.mathworks.com/support/`
E-mail	`mailto:support@mathworks.com`
Telephone	508-647-7000 Ask for Technical Support.

4

xPC TargetBox I/O Options

xPC TargetBox is available with seven I/O board options. Each option includes one or two test dongles that you attach to the external connectors in order to run a loop-back test. This chapter includes the following sections:

Introduction to I/O Options

xPC TargetBox includes screw terminal boards and cables to connect the I/O options to your hardware. This section includes the following topics:

- **Pin Layout and Screw Terminal Boards** — Pin numbers on a screw terminal board match the pin numbers on an xPC TargetBox connector
- **Making Your Own Terminal Boards** —Add application-specific signal conditioning circuits

Pin Layout and Screw Terminal Boards

The pin layout for each I/O option is provided in a table. The numbers printed on each screw terminal board correspond directly to the pin numbers provided in the table.

Each screw terminal board is provided with rubber feet and distance spacers for easy placement on a workbench. A nice feature of the screw terminal boards is that you can stack two or more together to reduce space requirements on your workbench.

Making Your Own Terminal Boards

You might want to make your own connector boards with signal conditioning circuitry that is specific for your application. The 50 pin connectors on the PC TargetBox and the screw terminal boards are from the AMPLIMIT .050 series III from Tyco Electronics. The part number is

```
Tyco Electronics P/N: 787082-5
```

If you want more information, see `http://www.amp.com` and the TERM50 hardware manual included in the xPC TargetBox shipping case.

Loop-Back Testing of I/O Options

Use a loop-back test to initially determine if your xPC TargetBox is working correctly, determine if your xPC TargetBox is continuing to work correctly, and to learn more about an I/O option. This section includes the following topics:

- **Loop-Back Testing Process** — Connect a test dongle to an xPC TargetBox connector and run a target application specific for that connector and I/O option
- **Determining the Success of a Loop-Back Test** — Observe a scope on the xPC TargetBox monitor that displays the results from a loop-back test
- **Uses for Loop-Back Testing** — Test your xPC TargetBox when you first unpack the box, during use, and to learn about an I/O option
- **Creating a Simulink Model for Loop-Back Testing** — Recreate the Simulink model to test all or some of the I/O options in your xPC TargetBox
- **Running the Target Application for Specific I/O Option Testing** — Create and run a target application to test I/O options

Loop-Back Testing Process

Loop-back testing is a technique to make sure that the xPC TargetBox is working properly with a certain I/O option or all I/O options working together. The technique consists of

1. A simple Simulink model with appropriate I/O driver blocks and analysis blocks.

2. A target application built from a Simulink model using Real-Time Workshop, xPC Target, and a C/C++ compiler.

3. Running the target application on the xPC TargetBox where the corresponding I/O option is exercised. Signals generated at the outputs are looped back through physical connectors to the inputs of the same I/O option.

4. The analysis part of the target application compares the generated signal with the looped-back signal. Depending on the result of the comparison, a scope displays success = 1 or failure = 0.

Determining the Success of a Loop-Back Test

A successful loop-back test for an I/O option ensures that the following are functioning properly:

- xPC TargetBox is running properly. It can execute a target application.
- The I/O option (I/O board) is accessed through its base address. This address was set by the manufacturer when you purchased your xPC TargetBox. This is the address you enter in the `xpctgboxtest` command to create a Simulink model for loop-back testing.
- The I/O option, consisting of the I/O board, I/O connector, test dongle, I/O cable, and screw terminal board, is fully functional both mechanically and electrically.

A failed loop-back test of an I/O option signifies that one of the parts has a failure that makes additional testing necessary.

Uses for Loop-Back Testing

You can use loop-back testing of an I/O option to accomplish various testing tasks at different stages of using the xPC TargetBox:

- Initial full system test of the xPC TargetBox and all I/O options when unpacking the system for the first time after delivery

  At the time you unpack the xPC Target box, you can attach test dongles and a monitor, power up the system, and immediately observe the proper functioning of the xPC TargetBox and all installed I/O options.
- Testing a particular I/O option during xPC TargetBox operation

  This task is useful if you are working with xPC Target and the xPC TargetBox and you believe that something might be wrong with an I/O option. For example, you could have a damaged I/O board. This may also be useful when you are working with a more complex real-world model (application) and it is not working to your satisfaction. You can then use loop-back testing of a particular I/O option as a unit test for a certain feature of the xPC TargetBox and narrow down the possible causes of the overall failure.
- Getting familiar with a certain I/O option and its features

Creating a Simulink Model for Loop-Back Testing

During the final system test at the manufacturer, a Simulink model was created that does a loop-back test of all I/O options installed on your system. This Simulink model was built into a stand-alone target application and put onto the flash disk of your xPC TargetBox. The stand-alone application is called `xpctgtst.rtb` and is located in the directory `c:\work`.

It is good practice to back up this file for later use in case the flash disk is corrupted. Nevertheless, you can create the same Simulink model on which `xpctgtst.rtb` is based at any time. You can then build and download that target application and redo the entire system test if the original test stand-alone application no longer exists.

1 Check the configuration label for the I/O options installed in your xPC TargetBox and the base addresses.

2 Enter the command to create the I/O option Simulink model. For example, if you want a model to test I/O option 301 and I/O option 305 with base addresses 0x300 and 0x280, in the MATLAB Command Window, type

```
xpctgboxtest({'IO301','0x300','C21'},{'IO305','0x280'})
```

For I/O options with jumpers on the boards, enter the jumper codes. See “Configuration Label for I/O Options” on page 2-21.

3 On the bottom of the xPC TargetBox, find the jumper codes for your I/O options. Use the table in “Configuration Label for I/O Options” on page 2-21 to determine the jumper settings from the jumper codes.

4 From the list of jumper options, enter the number corresponding to the jumper settings for your box.

A Simulink test model is created. This model is the system test model that was used to create the target application initially delivered on the flash disk as a stand-alone application. You can use this model to build and run a target application on the xPC TargetBox.

5 Save the Simulink test model in a directory outside the MATLAB root directory.

Caution Make sure that you have connected the correct dongles to the correct I/O connectors; otherwise, you can damage an I/O option or even the entire xPC TargetBox.

Running the Target Application for Specific I/O Option Testing

If you want to exercise a loop-back test of a certain I/O option, take the following steps:

1. Power down the xPC TargetBox. Never plug or unplug any connectors while the xPC TargetBox is powered up.

2. Select the correct test dongle or dongles. Some I/O options come with two dongles. Each test dongle is labeled with the I/O option number (for example, IO 301) and its port number. For example, IO 301, has two ports labeled IO 301-1 (analog) and IO 301-2 (digital).

   On the bottom panel of the xPC TargetBox, check the xPC TargetBox configuration label. This label lists the I/O connector to connect the test dongle to. Also write down the base address for the I/O option (I/O board) you want to exercise.

   Warning Make sure that you connect the selected test dongles to the corresponding I/O connectors; otherwise, you can damage an I/O option or the entire xPC TargetBox when powering it up again.

3. Boot the xPC TargetBox so that the xPC Target kernel starts running using either

   - Boot Floppy mode from a floppy disk.
   - DOSLoader mode from the flash disk.

4 From the MATLAB Command Window, type

```
xpctargetping
```

This command checks for a proper connection between your xPC TargetBox and host PC, and if successful, responds with

```
ans =
success
```

5 Enter the command to create the I/O option Simulink model. For example, if you want a model to exercise I/O option 301 with the I/O board base address 0x300, with the analog input jumper set for differential mode, and the analog output jumper set to ± 10 volts, type

```
xpctgboxtest({'IO301','0x300','C21'})
```

The corresponding Simulink test model is created.

6 Save the Simulink model.

7 In the Simulink window, and from the **Tools** menu, point to **Real-Time Workshop**, and then click **Build**.

Real-Time Workshop, xPC Target, and a C/C++ compiler build and download a target application to the xPC TargetBox.

8 If the build and download process was successful, type

```
+tg
```

The target application starts running.

9 Check the numerical values on the scopes.

- If the scopes show numerical values of 1, then the loop-back test was successful and the problem with your application might lie somewhere else.
- If one of the displayed values is 0, then either the board or the test dongle might be damaged. In this case, contact your MathWorks representative for help.

xPC TargetBox IO 301

The Diamond MM-32-AT is an analog and digital board with 32 analog input (A/D) channels, 4 analog output (D/A) channels, and 24 digital I/O lines. This section includes the following topics:

- Wiring for IO 301 Test Dongles
- Pin Layout IO 301 (1 of 2 Connectors)
- Pin Layout IO 301 (2 of 2 connectors)
- Testing Model IO 301

Note xPC TargetBox does not support the counters on this board.

Wiring for IO 301 Test Dongles

xPC Target communicates with this board using two connectors. Some of the pins in the test dongles are connected together to allow for loop-back testing of this board.

- Analog Header (J3) Dongle — connect pins 3—38, 7—37, 11—35, 15—36, and 2—4—8—12—16
- Digital Header (J4) Dongle — connect pins 12 —20, and 16—24

Pin Numbering for Connectors and Screw Terminal Boards

The screw terminals on the screw terminal boards are numbered sequentially, but the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.

Pin Layout IO 301 (1 of 2 Connectors)

The following figure is the pin layout for the IO 301 option connector. This is the J3 connector in the manufacturer's data sheet at
`http://www.diamondsystems.com/files/binaries/DMM32v2.61.pdf.`

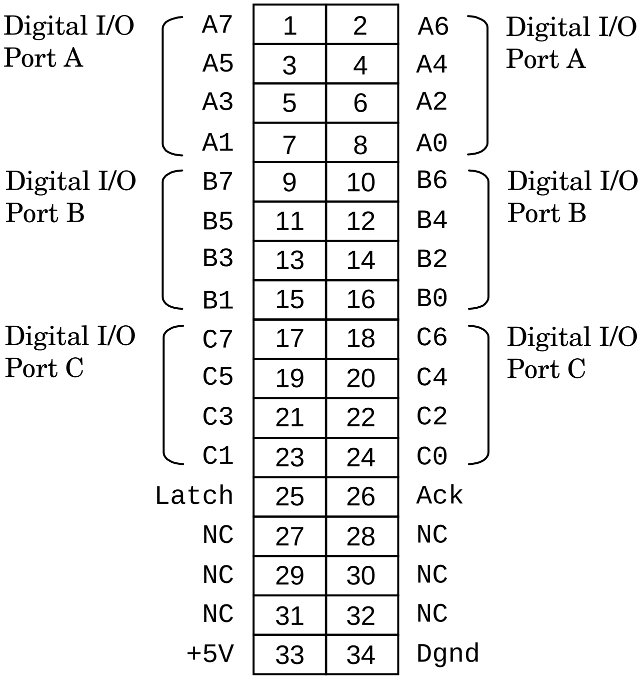
The terminals on the screw terminal boards are numbered sequentially while the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.

	Agnd	1	2	Agnd	
Analog input	Vin 0 / 0+	3	4	Vin 16 / 0-	Analog input
	Vin 1 / 1+	5	6	Vin 17 / 1-	
	Vin 2 / 2+	7	8	Vin 18 / 2-	
	Vin 3 / 3+	9	10	Vin 19 / 3-	
	Vin 4 / 4+	11	12	Vin 20 / 4-	
	Vin 5 / 5+	13	14	Vin 21 / 5-	
	Vin 6 / 6+	15	16	Vin 22 / 6-	
	Vin 7 / 7+	17	18	Vin 23 / 7-	
	Vin 8 / 8+	19	20	Vin 24 / 8-	
	Vin 9 / 9+	21	22	Vin 25 / 9-	
	Vin 10 / 10+	23	24	Vin 26 / 10-	
	Vin 11 / 11+	25	26	Vin 27 / 11-	
	Vin 12 / 12+	27	28	Vin 28 / 12-	
	Vin 13 / 13+	29	30	Vin 29 / 13-	
	Vin 14 / 14+	31	32	Vin 30 / 14-	
	Vin 15 / 15+	33	34	Vin 31 / 15-	
Analog output	Vout 3	35	36	Vout 2	Analog output
	Vout 1	37	38	Vout 0	
	Vref Out	39	40	Clr2 Out / Dout 2	
	A/D Convert	41	42	Clr0 Out / Dout 0	
	Dout 1	43	44	Extgate / Din 2	
	Extclk /Din 3	45	46	Clk 0 / Din 2	
	Gate 0 / Din 1	47	48	Clk 0 / Din 0	
	+5V	49	50	Dgnd	

Pin Layout IO 301 (2 of 2 connectors)

The following figure is the pin layout for the IO 301 option connector. This is the J4 connector in the manufacturer's data sheet at
`http://www.diamondsystems.com/files/binaries/DMM32v2.61.pdf`

The terminals on the screw terminal boards are numbered sequentially while the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.



Digital I/O Port A	A7	1	2	A6	Digital I/O Port A
	A5	3	4	A4	
	A3	5	6	A2	
	A1	7	8	A0	
Digital I/O Port B	B7	9	10	B6	Digital I/O Port B
	B5	11	12	B4	
	B3	13	14	B2	
	B1	15	16	B0	
Digital I/O Port C	C7	17	18	C6	Digital I/O Port C
	C5	19	20	C4	
	C3	21	22	C2	
	C1	23	24	C0	
	Latch	25	26	Ack	
	NC	27	28	NC	
	NC	29	30	NC	
	NC	31	32	NC	
	+5V	33	34	Dgnd	

Note This connector does not use pins 35 to 50.

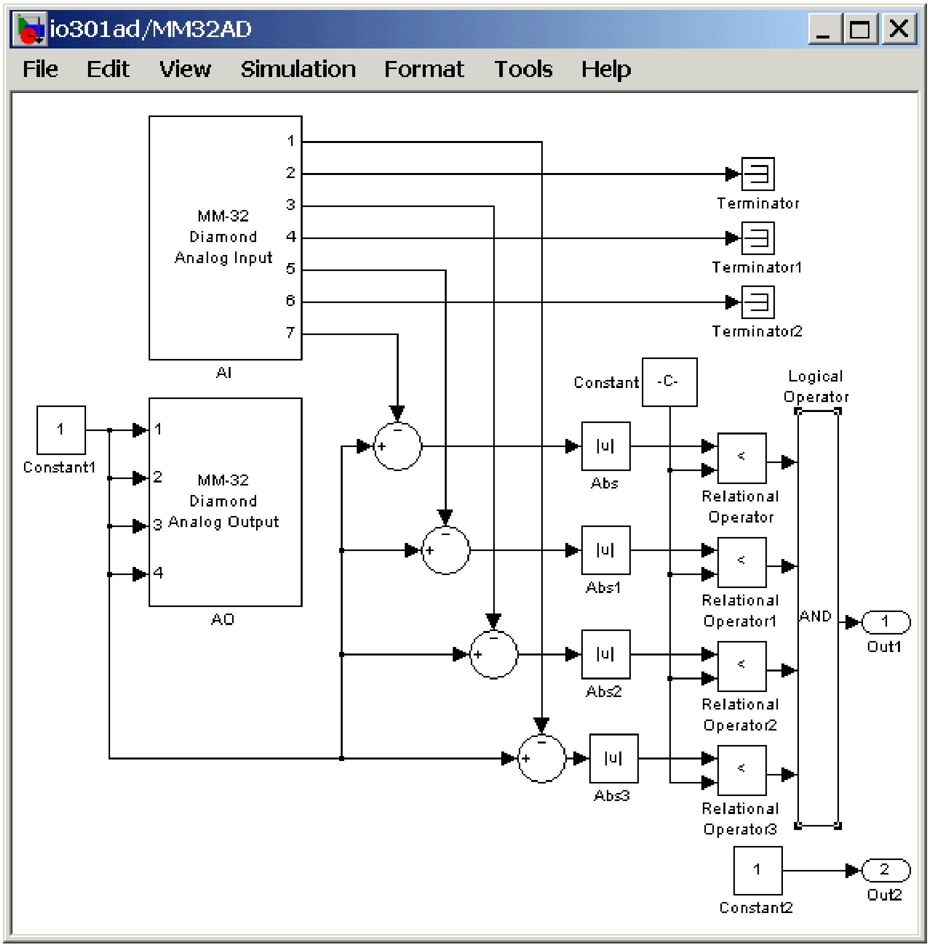
Testing Model IO 301

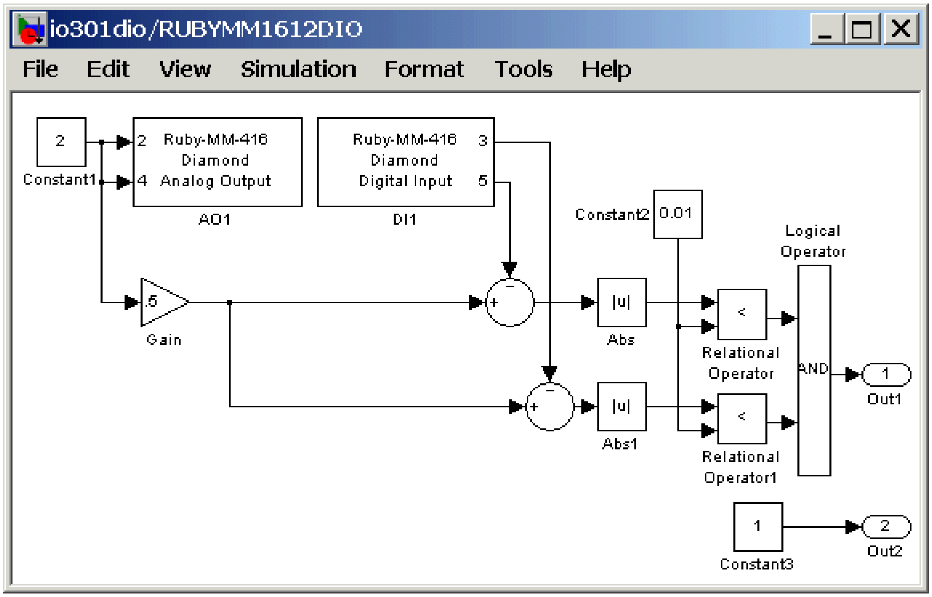
This model tests the analog channels. A constant value is written to an A/D driver block and then read from the D/A driver block. The values are compared with an error defined in a second constant block.

1 In the MATLAB Command Window, type

```
io301
```

The Simulink model for testing the IO 301 option opens.




io301dio/RUBYMM1612DIO
File
Edit
View
Simulation
Format
Tools
Help
2
Constant1
2
4
Ruby-MM-416
Diamond
Analog Output
AO1
Ruby-MM-416
Diamond
Digital Input
3
5
DI1
Constant2
0.01
Logical
Operator
.5
Gain
+
−
+
−
|u|
Abs
|u|
Abs1
<
Relational
Operator
<
Relational
Operator1
AND
1
Out1
1
Constant3
2
Out2

xPC TargetBox IO 302

The Diamond Ruby-MM-1612 is an analog and digital board with 16 analog output (D/A) channels and 24 digital I/O lines. This section includes the following topics:

- Wiring for IO 302 Test Dongle
- Pin Layout IO 302
- Testing Model IO 302

Note xPC TargetBox does not support the external trigger on this board.

Wiring for IO 302 Test Dongle

xPC TargetBox communicates with this board using one connector. Some of the pins in the test dongle are connected together to allow for loop-back testing of this board.

Header (J3) Dongle — connect pins 4—46, and 8 —44

Pin Layout IO 302

The following figure is the pin layout for the IO 302 option connector. This is the J3 connector in the manufacturer's data sheet at `http://www.diamondsystems.com/files/binaries/RMM1612v11.pdf.`

The terminals on the screw terminal boards are numbered sequentially while the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.

	Agnd	1	2	Vout 0	Analog output
	Agnd	3	4	Vout 1	
	Agnd	5	6	Vout 2	
	Agnd	7	8	Vout 3	
	Agnd	9	10	Vout 4	
	Agnd	11	12	Vout 5	
	Agnd	13	14	Vout 6	
	Agnd	15	16	Vout 7	
Analog output	Vout 8	17	18	Vout 9	
	Vout 10	19	20	Vout 11	
	Vout 12	21	22	Vout 13	
	Vout 14	23	24	Vout 15	
Digital I/O Port A	DIO A7	25	26	DIO A6	Digital I/O Port A
	DIO A5	27	28	DIO A4	
	DIO A3	29	30	DIO A2	
	DIO A1	31	32	DIO A0	
Digital I/O Port B	DIO B7	33	34	DIO B6	Digital I/O Port B
	DIO B5	35	36	DIO B4	
	DIO B3	37	38	DIO B2	
	DIO B1	39	40	DIO B0	
Digital I/O Port C	DIO C7	41	42	DIO C6	Digital I/O Port C
	DIO C5	43	44	DIO C4	
	DIO C3	45	46	DIO C2	
	DIO C1	47	48	DIO C0	Ext Trig
	+5V	49	50	Dgnd	

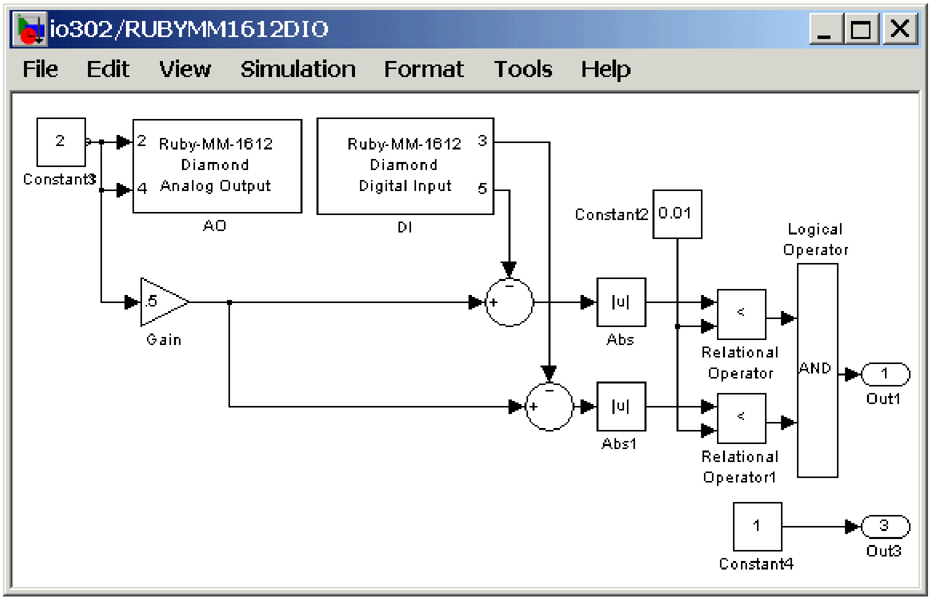
Testing Model IO 302

A constant value is written to the A/D driver block and then read from the D/A driver block. The values are compared with an error defined in a second constant block, and the results sent to an Outport block.

1 In the MATLAB Command Window, type

```
io302
```

The Simulink model for testing the IO 302 option opens.



xPC TargetBox IO 303

The Diamond Ruby-MM-416 is an analog and digital board with 4 analog output (D/A) channels and 24 digital I/O lines. This section includes the following topics:

- Wiring for IO 303 Test Dongle
- Pin Layout IO 303
- Testing Model IO 303

Note xPC TargetBox does not support the external trigger on this board.

Wiring for IO 303 Test Dongle

xPC TargetBox communicates with this board using one connector. Some of the pins in the test dongle are connected together to allow for loop-back testing of this board.

Header (J3) Dongle — connect pins 4—46, and 8—44

Pin Layout IO 303

The following figure is the pin layout for the IO 303 option connector. This is the J3 connector in the manufacturer's data sheet at `http://www.diamondsystems.com/files/binaries/RMM416v11.pdf.`

The terminals on the screw terminal boards are numbered sequentially while the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.

	Agnd	1	2	Vout 0	Analog output
	Agnd	3	4	Vout 1	
	Agnd	5	6	Vout 2	
	Agnd	7	8	Vout 3	
	NC	9	10	NC	
	NC	11	12	NC	
	NC	13	14	NC	
	NC	15	16	NC	
	NC	17	18	NC	
	Agnd	19	20	+15V	
	-15V	21	22	Agnd	
	Dgnd	23	24	Ext Trig	
Digital I/O Port A	DIO A7	25	26	DIO A6	Digital I/O Port A
	DIO A5	27	28	DIO A4	
	DIO A3	29	30	DIO A2	
	DIO A1	31	32	DIO A0	
Digital I/O Port C	DIO C7	33	34	DIO C6	Digital I/O Port C
	DIO C5	35	36	DIO C4	
	DIO C3	37	38	DIO C2	
	DIO C1	39	40	DIO C0	
Digital I/O Port B	DIO B7	41	42	DIO B6	Digital I/O Port B
	DIO B5	43	44	DIO B4	
	DIO B3	45	46	DIO B2	
	DIO B1	47	48	DIO B0	
	+5V	49	50	Dgnd	

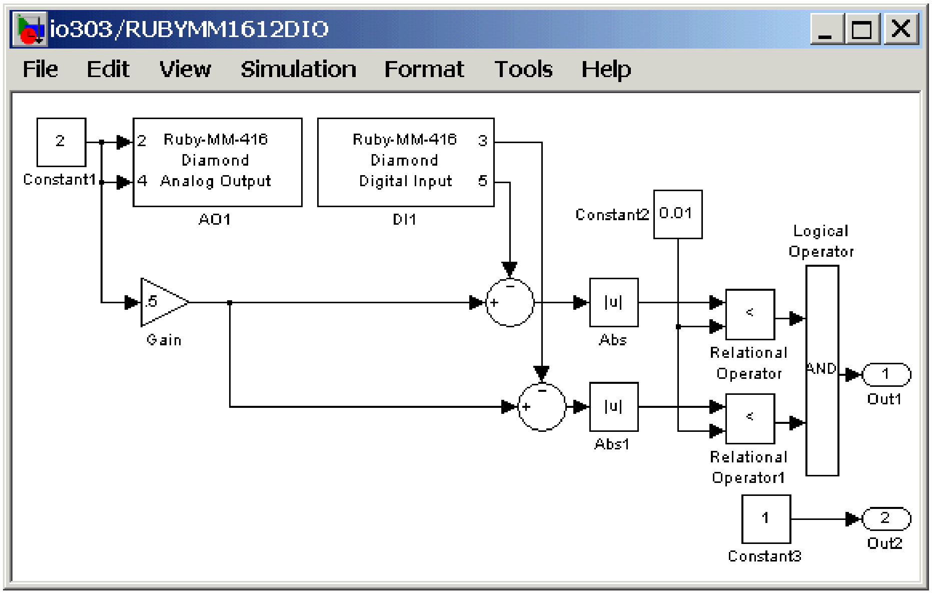
Testing Model IO 303

A constant value is written to the A/D driver block and then read from the digital input driver block. The values are compared with an error defined in a second constant block, and the results sent to an Outport block.

1 In the MATLAB Command Window, type

```
io303
```

The Simulink model for testing the IO 303 option opens.



xPC TargetBox IO 304

The Diamond Onyx-MM board has 48 digital I/O lines. They can be programmed for input or output in groups of 8 lines. This section includes the following topics:

- Wiring for IO 304 Test Dongle
- Pin Layout IO 304 (Both Connectors)
- Testing Model IO 304

Note xPC TargetBox does not support the counter/timers and external interrupts on this board.

Wiring for IO 304 Test Dongle

xPC TargetBox communicates with this board using two connectors. Some of the pins in the test dongles are connected together to allow for loop-back testing of this board.

- Header (J3) Dongle — connect pins 23—39, and 31—47
- Header (J4) Dongle — connect pins 23—39, and 31—47

Pin Layout IO 304 (Both Connectors)

The following figure is the pin layout for both of the IO 304 option connectors. These are the J3 and J4 connectors in the manufacturer's data sheet at `http://www.diamondsystems.com/files/binaries/OMMv14.pdf.`

The terminals on the screw terminal boards are numbered sequentially while the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.

Digital I/O Port A	A7	1	2	Gnd
	A6	3	4	Gnd
	A5	5	6	Gnd
	A4	7	8	Gnd
	A3	9	10	Gnd
	A2	11	12	Gnd
	A1	13	14	Gnd
	A0	15	16	Gnd
Digital I/O Port C	C7	17	18	Gnd
	C6	19	20	Gnd
	C5	21	22	Gnd
	C4	23	24	Gnd
	C3	25	26	Gnd
	C2	27	28	Gnd
	C1	29	30	Gnd
	C0	31	32	Gnd
Digital I/O Port B	B7	33	34	Gnd
	B6	35	36	Gnd
	B5	37	38	Gnd
	B4	39	40	Gnd
	B3	41	42	Gnd
	B2	43	44	Gnd
	B1	45	46	Gnd
	B0	47	48	Gnd
	+5V	49	50	Gnd

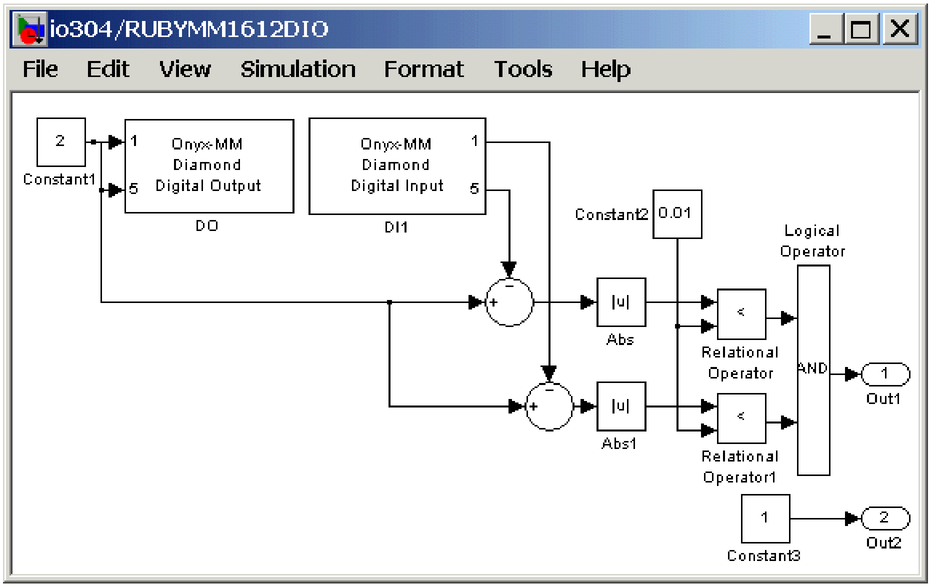
Testing Model IO 304

A constant value is written to a digital output driver block and then read from a digital input block. The values are compared with an error defined in a second constant block, and the results sent to an Outport block.

1 In the MATLAB Command Window, type

```
io304
```

The Simulink model for testing the IO 304 option opens.



xPC TargetBox IO 305

The Diamond Quartz-MM-10 board has 10 counters, 8 digital I/O lines, and one interrupt line. This section includes the following topics:

- Wiring for IO 305 Test Dongle
- Pin Layout IO 305
- Testing Model IO 305

Wiring for IO 305 Test Dongle

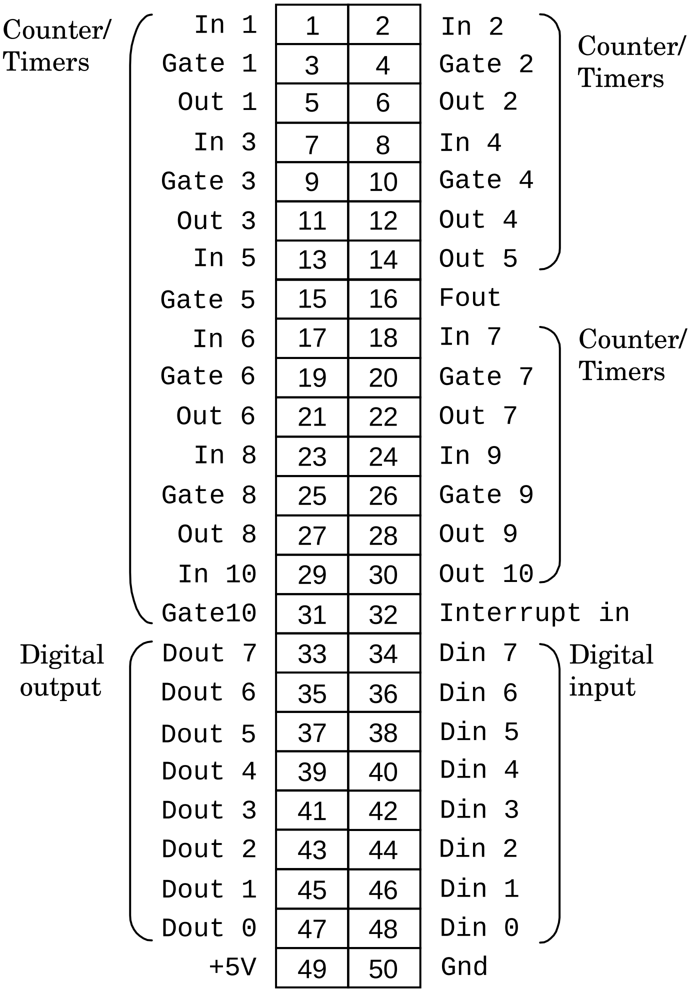
xPC TargetBox communicates with this board using one connector. Some of the pins in the test dongles are connected together to allow for loop-back testing of this board.

Header (J3) Dongle — connect pins 6—17, 12—18, and 43—48

Pin Layout IO 305

The following figure is the pin layout for the IO 305 option connector. These are the J3 and J4 connectors in the manufacturer's data sheet at `http://www.diamondsystems.com/files/binaries/QMMv15.pdf.`

The terminals on the screw terminal boards are numbered sequentially while the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.



Counter/ Timers	In 1	1	2	In 2	Counter/ Timers
	Gate 1	3	4	Gate 2	
	Out 1	5	6	Out 2	
	In 3	7	8	In 4	
	Gate 3	9	10	Gate 4	
	Out 3	11	12	Out 4	
	In 5	13	14	Out 5	
	Gate 5	15	16	Fout	
	In 6	17	18	In 7	Counter/ Timers
	Gate 6	19	20	Gate 7	
	Out 6	21	22	Out 7	
	In 8	23	24	In 9	
	Gate 8	25	26	Gate 9	
	Out 8	27	28	Out 9	
	In 10	29	30	Out 10	
	Gate10	31	32	Interrupt in	
Digital output	Dout 7	33	34	Din 7	Digital input
	Dout 6	35	36	Din 6	
	Dout 5	37	38	Din 5	
	Dout 4	39	40	Din 4	
	Dout 3	41	42	Din 3	
	Dout 2	43	44	Din 2	
	Dout 1	45	46	Din 1	
	Dout 0	47	48	Din 0	
	+5V	49	50	Gnd	

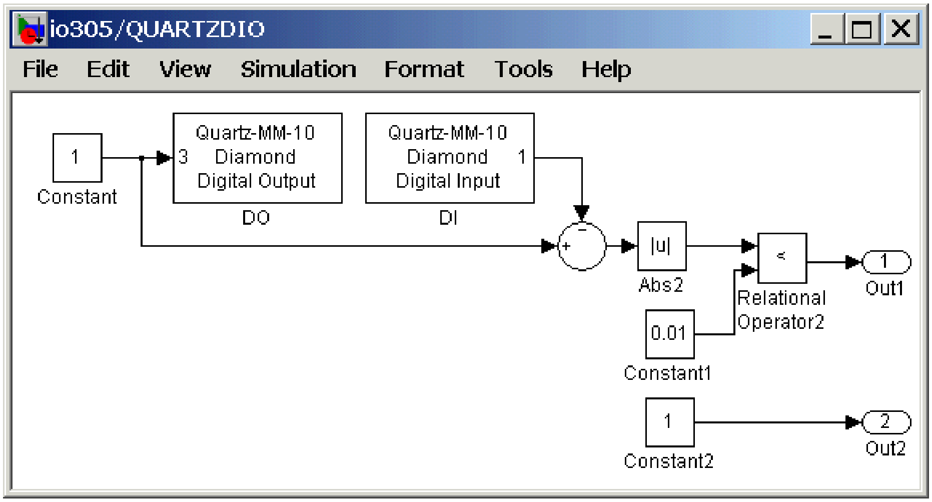
Testing Model IO 305

A constant value is written to a digital output driver block and then read from a digital input block. The values are compared with an error defined in a second constant block, and the results sent to an Outport block.

1 In the MATLAB Command Window, type

```
io305
```

The Simulink model for testing the IO 305 option opens.



xPC TargetBox IO 306

The RealTime Devices DM6814 board has 3 encoders, 12 digital I/O lines, and 2 interrupts. This section includes the following topics:

- Wiring for IO 306 Test Dongle
- Pin Layout IO 306 (1 of 2 Connectors)
- Pin Layout IO 306 (2 of 2 Connectors)
- Testing Model IO 306

Note xPC TargetBox does not support the counter/timers on this board.

Wiring for IO 306 Test Dongle

xPC Target communicates with this board using two connectors. Some of the pins in the test dongle are connected together to allow for loop-back testing of this board.

Header (P2) Dongle — connect pins 5 —13, 7 —15, 21—29, 23—31, 37—45, 39—47

Note Header (P3) — While this header is connected to an I/O connector and shipped with a terminal board and cable, the xPC TargetBox is not shipped with a dongle attached to this I/O connector.

Pin Layout IO 306 (1 of 2 Connectors)

The following figure is the pin layout for the IO 306 option connector. This is the P2 connector on the manufacturer's data sheet `http://www.rtdusa.com/manuals/DM/dm5810.pdf`.

The terminals on the screw terminal boards are numbered sequentially while the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.

Encoder	Inc Enc 3 IRQin	1	2	Ext int1
	Overflow 3 Out	3	4	Dgnd
	Inc Enc 3 ChB	5	6	Dgnd
	Inc Enc 3 ChA	7	8	Dgnd
D In	P4.3	9	10	Dgnd
	P4.2	11	12	Dgnd
	P4.1	13	14	Dgnd
	P4.0	15	16	Dgnd
Encoder	Inc Enc 2 IRQin	17	18	Dgnd
	Overflow 3 Out	19	20	Dgnd
	Inc Enc 3 ChB	21	22	Dgnd
	Inc Enc 3 ChA	23	24	Dgnd
D In	P2.3	25	26	Dgnd
	P2.2	27	28	Dgnd
	P2.1	29	30	Dgnd
	P2.0	31	32	Dgnd
Encoder	Inc Enc 1 IRQin	33	34	Dgnd
	Overflow 3 Out	35	36	Dgnd
	Inc Enc 3 ChB	37	38	Dgnd
	Inc Enc 3 ChA	39	40	Dgnd
D In	P0.3	41	42	Dgnd
	P0.2	43	44	Dgnd
	P0.1	45	46	Dgnd
	P0.0	47	48	Dgnd
	+5V	49	50	Dgnd

Pin Layout IO 306 (2 of 2 Connectors)

The following figure is the pin layout for the IO 306 option connector. This is the P3 connector on the manufacturer's data sheet `http://www.rtdusa.com/manuals/DM/dm5810.pdf`.

The terminals on the screw terminal boards are numbered sequentially while the pin numbers on the connectors are numbered alternately left and right. However, the numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below.

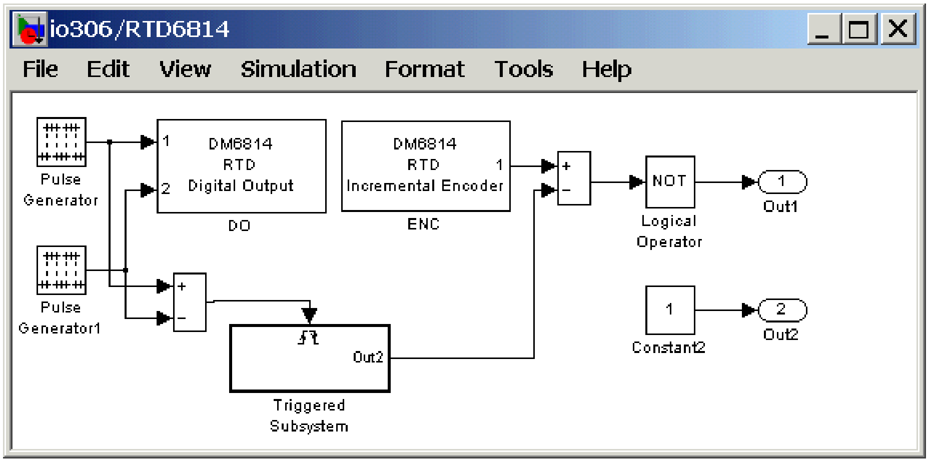
	NC	1	2	Ext int2
	NC	3	4	Dgnd
	NC	5	6	Dgnd
	NC	7	8	Dgnd
D In	P5.3	9	10	Dgnd
	P5.2	11	12	Dgnd
D In/Out	P5.1	13	14	Dgnd
	P5.0	15	16	Dgnd
	NC	17	18	Dgnd
	NC	19	20	Dgnd
	NC	21	22	Dgnd
	NC	23	24	Dgnd
D In	P3.3	25	26	Dgnd
	P3.2	27	28	Dgnd
D In/Out	P3.1	29	30	Dgnd
	P3.0	31	32	Dgnd
	NC	33	34	Dgnd
	NC	35	36	Dgnd
	NC	37	38	Dgnd
	NC	39	40	Dgnd
D In	P1.3	41	42	Dgnd
	P1.2	43	44	Dgnd
D In/Out	P1.1	45	46	Dgnd
	P0.0	47	48	Dgnd
	+5V	49	50	Dgnd

Testing Model IO 306

1 In the MATLAB Command Window, type

```
io306
```

The Simulink model for testing the IO 306 option opens.



xPC TargetBox IO 308

The Softing CAN-AC2-104 board has two CAN channels. This section includes the following topics:

- Wiring for IO 308 Test Dongle
- Pin Layout IO 308
- Testing Model IO 308

Wiring for IO 308 Test Dongle

xPC Target communicates with this board using two CAN serial connectors.

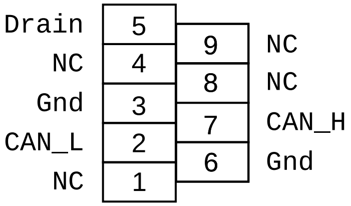
The test dongle is a CAN serial cable, with termination resistors, connected between the two CAN 9-pin connectors.

Note The IO 308 option is shipped with terminal resistors enabled. When you connect this option to a CAN field bus, you need to add terminal resistors to your cable.

Pin Layout IO 308

The following figure is the pin layout for both of the IO 308 option connectors. The pin layout for the D-Sub 9 connector is the same for both CAN channels.

The numbers printed on each screw terminal board correspond directly to the pin numbers provided in the figure below. However, the pin numbers on the terminal board are numbered sequentially.



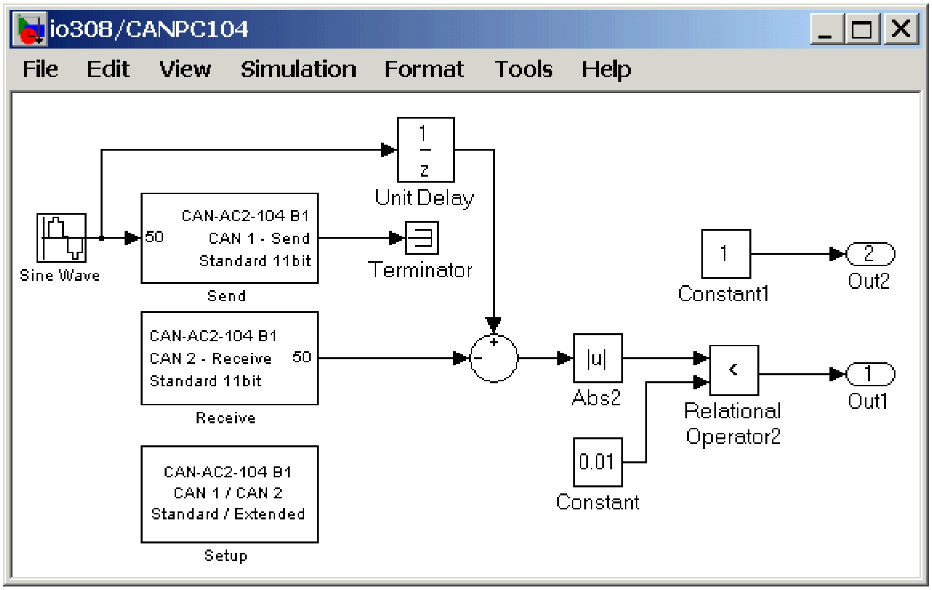
Testing Model IO 308

A sine wave signal is written to a CAN output driver block and then read from a CAN input block. After delaying the input signal, the values are compared with an error defined in a second constant block, and the results sent to an Outport block.

1 In the MATLAB Command Window, type

```
io308
```

The Simulink model for testing the IO 308 option opens.



Using xPC TargetBox

This chapter includes the following sections:

xPC TargetBox Library

The xPC TargetBox library is a group of Simulink blocks that represent the drivers for the xPC TargetBox I/O options, LEDs, and watchdog circuitry. This section includes the following topics:

- **Drivers in the xPC TargetBox Library** — Drivers from the xPC Target library are conveniently grouped into an xPC TargetBox library
- **Using the Panel LEDs** — Control the two LEDs on the xPC TargetBox using driver blocks from the xPC TargetBox library
- **Using the Watchdog Timer** — Perform a system reset when a programmable timeout occurs

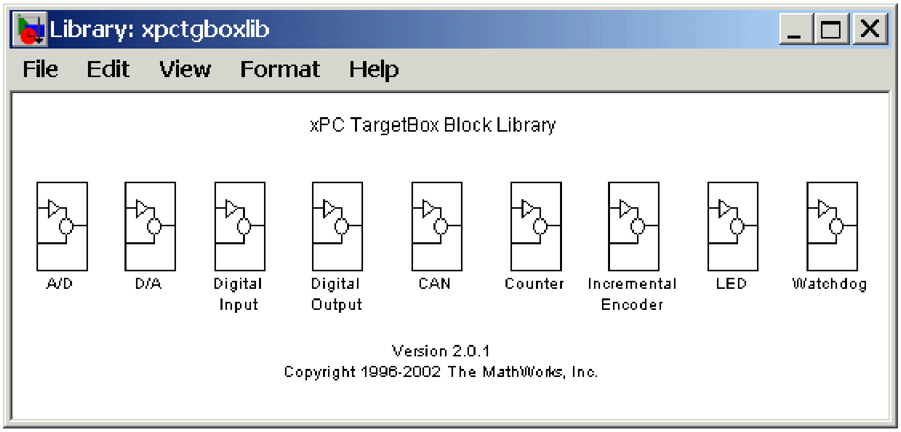
Drivers in the xPC TargetBox Library

The drivers in the xPC TargetBox library are links to drivers from the various function groups in the xPC Target driver library. Use these blocks in the same way as you would use the original driver blocks.

1 In the MATLAB Command Window, type

```
xpctgboxlib
```

The **xPC TargetBox Library** window opens.



2 Drag and drop a driver block into your Simulink model.

Except for the LED and Watchdog blocks, the driver blocks in the xPC TargetBox library are aliases to drivers in the xPC Target library. For information about a driver block,

- Click the **Help** button in the **Block Parameters** dialog box.
- For recently added drivers, use the updated xPC Target I/O Reference documentation on the xPC Target Product News page:

```
http://www.mathworks.com/support/author/xpc/index.shtml
```

Using the Panel LEDs

Each xPC TargetBox has two LEDs that you can control. These LEDs are labeled USER 1 and USER 2 on the xPC TargetBox.

From the xPC TargetBox library (`xpctgbox.lib`), drag an LED block into your Simulink model. You can have only one block of this type in any model.

Driver Block Parameters

LED vector — Enter numbers between `1` and `2`. This driver allows the selection of individual LEDs in any order. The number of elements defines the number of LEDs used. For example, to control the USER 1 and USER 2 LEDs, enter

```
[1 2]
```

Reset vector — Enter `0` or `1` for each LED. This parameter controls the behavior at model termination. A value of `1` causes the corresponding LED to be reset to the value specified in **Initial value vector**. A value of 0 causes the corresponding LED to be left at the most recent value set while the model was running. The **Reset vector** should either be a scalar value or the same length as the LED vector. For scalar values, the block performs a scalar expansion and assumes the same **Reset vector** value for each element in the vector.

Initial value vector — Enter 0 or `1` for each LED. This parameter specifies the initial value (`0` or `1`) to which the LED should be set between the time the model is downloaded and the time it is started. The **Initial vector value** should be the same length as the LED vector. If you do not have the same vector length, the block performs a scalar expansion and assumes the same **Initial vector value** value for each element in the vector.

Sample time — Enter a base sample time or a multiple of the base sample time.

Using the Watchdog Timer

The xPC TargetBox provides a watchdog that you can program to perform a system reset when a programmable timeout occurs. The timeout interval can range from 1 to 255 seconds with one-second resolution, or from 1 to 255 minutes with one-minute resolution. When the timeout expires, an IRQ15 interrupt is asserted. With the **Reboot upon expiration of watchdog** parameter, you can specify that the system reboot at this time. At most, one block of this type can be included in any model.

Driver Block Parameters

Timeout units — From the list, select either `seconds` or `minutes`. These are the units for the time.

Timeout interval — Enter a number between `1` and `255`.

Show reset port (R) — Select this check box to display an input port labeled `R`. A signal connected to this port resets the watchdog whenever its value exceeds 0.5.

Reset upon keyboard activity — Select this check box to reset the watchdog whenever there is keyboard activity.

Reset upon mouse activity — Select this check box to reset the watchdog whenever there is mouse activity.

Reboot upon expiration of watchdog — Select this check box to reboot the system when the watchdog timer expires and asserts IRQ15.

Sample time — Enter a base sample time or a multiple of the base sample time.

DOSLoader Mode

DOSLoader mode allows you to copy the xPC Target kernel to the flash disk, remove the floppy disk drive, and then boot the xPC Target kernel. The target application is still downloaded from the host PC.

Use this mode for applications where the xPC TargetBox is not easily accessible. This section includes the following topics:

- **Updating Environment Properties and Creating a Boot Disk** — Select DOSLoader mode in the **Setup** dialog box
- **Copying the Kernel to Flash Memory** — Copy the xPC Target kernel to the flash disk on an xPC TargetBox and then start the kernel running
- **Creating a Target Application** — Create, download, and run a target application from a host PC
- **Booting DOS Instead of the xPC Target Kernel** — Boot an xPC TargetBox into DOS instead of starting the xPC Target kernel

---

Note To use DOSLoader mode you need the xPC Target Embedded Option.

Updating Environment Properties and Creating a Boot Disk

xPC Target uses the environment properties to determine what files to create for the various target boot modes.

This procedure assumes you have serial or network communication working correctly between your host computer and the xPC TargetBox with **TargetBoot** in the **Setup** dialog box set to `BootFloppy`.

1 On the host computer, start MATLAB.

2 In the MATLAB Command Window, type

```
xpcsetup
```

The **xPC Target Setup** window opens.

3. From the **TargetBoot** list, choose `DOSLoader`. Click **Update**.

   xPC Target updates the environment properties and grays the **Update** button.

4. Click **BootDisk**.

   A message box opens with the following message.

   ```
   Insert a formatted floppy disk into your host PC disk drive and
   click OK to continue.
   ```

5. Insert a floppy disk, and then click **OK**.

   The files `checksum.dat`, `xpcsgo1.rtb`, `xpcboot.com`, and `autoexec.bat` are copied to the disk.

Copying the Kernel to Flash Memory

1. Boot the xPC TargetBox without a floppy disk.

   xPC TargetBox starts running FreeDOS from flash memory.

   ```
   C:\>
   ```

2. Since the process of setting up the xPC TargetBox for DOSLoader mode overwrites the current `autoexec.bat` file, you might want to save a copy with a different name. For example:

   ```
   c:\>copy autoexec.bat autoex01.bat
   ```

3. Insert the boot floppy disk with the xPC Target kernel into the external floppy disk drive.

4. Type

   ```
   copy a:\xpcsgo1.rtb c:
   copy a:\xpcboot.com c:
   copy a:\autoexec.bat c:
   ```

5 Remove the floppy disk, and then, on the xPC TargetBox, press the **Reset** button.

The file autoexec.bat includes the command xpcboot.com to start the xPC Target kernel running.

Note If you get the error message

```
Cannot boot RTTarget-32 from Virtual 8086 mode.
```

edit the config.sys file to comment out or remove the line that loads the high memory driver:

```
REM DEVICE=C:\DROS\EMM386.EXE DPMI=OFF FRAME=NO
```

Creating a Target Application

1 In the MATLAB Command Window, type the name of a Simulink model. For example, type

```
xpc_osc3
```

A Simulink window opens with the model.

2 From the **Tools** menu, point to **Real-Time Workshop**, and then click **Build Model**.

3 Real-Time Workshop and xPC Target create a target application and download it to the xPC TargetBox.

Booting DOS Instead of the xPC Target Kernel

After you copy the file autoexec.bat for DOSLoader mode to the flash disk, the xPC TargetBox will always start DOS and then immediately boot the xPC Target kernel. You might want to return your system to boot into DOS.

1 On the xPC TargetBox, press the **Reset** button.

2 When you see the message for loading DOS, press **Ctrl-C**.

The xPC Target kernel boot process is stopped and the DOS prompt is displayed.

3 Copy the backup autoexec file. For example, if you saved a previous version of the autoexec file as autoex01.bat, type

```
[DR-DOS] c:\>copy autoex01.bat autoexec.bat
```

4 Press the **Reset** button.

The xPC TargetBox boots into DOS.

StandAlone Mode

StandAlone mode combines the target application with the kernel and boots them together on the xPC TargetBox from flash memory. The host PC does not have to be connected to the xPC TargetBox. This section includes the following topics:

- **Updating Environment Properties** — Select StandAlone mode in the **Setup** dialog box
- **Creating a Kernel/Target Application** — On the host PC, create a stand-alone application
- **Copying the Kernel/Target Application to Flash Memory** — Copy the combined xPC Target kernel and target application to the flash disk on an xPC TargetBox
- **Booting DOS Instead of the xPC Target Kernel** — Boot an xPC TargetBox into DOS instead of starting the target application

---

Note To use StandAlone mode you need the xPC Target Embedded Option.

Updating Environment Properties

xPC Target uses the environment properties to determine what files to create for the various target boot modes.

This procedure assumes you have serial or network communication working correctly between your host computer and the xPC TargetBox with **TargetBoot** in the **Setup** dialog box set to `BootFloppy`.

1. On the host computer, start MATLAB.

2. In the MATLAB Command Window, type

   ```
   xpcsetup
   ```

   The **xPC Target Setup** window opens.

3. From the **TargetBoot** list, choose `StandAlone`.

4. Click **Update**.

   xPC Target updates the environment properties and the build process is ready to create a stand-alone kernel/target application.

Note For StandAlone mode, you do not create an xPC Target boot disk. Instead, you copy files created from the build process onto a formatted floppy disk.

Creating a Kernel/Target Application

Using the xPC Target Embedded Option, create a boot disk with the xPC Target kernel and target application.

1. In the MATLAB Command Window, type the name of a Simulink model. For example, type

   ```
   xpc_osc3
   ```

   A Simulink window opens with the model.

2. From the **Tools** menu, point to **Real-Time Workshop**, and then click **Build Model**.

   Real-Time Workshop and xPC Target create a directory `xpc_osc3_xpc_emb` with the following files:

   ```
   xpc_osc3.rtb
   xpcboot.com
   autoexec.bat
   ```

3. Copy the above files to a formatted floppy disk.

Copying the Kernel/Target Application to Flash Memory

You build a target application on a host PC using Real-Time Workshop, xPC Target, and a C/C++ compiler. One method for transferring the files from the host PC to an xPC TargetBox is to use an external floppy disk drive.

After you set up your xPC TargetBox to boot into DOS and build a stand-alone application on a host PC, you can copy files from a floppy disk to the flash disk. See “Booting DOS Instead of the xPC Target Kernel” on page 5-7 and “Creating a Kernel/Target Application” on page 5-10.

1. Boot the xPC TargetBox without a floppy disk into DOS.

   The system starts running FreeDOS from flash memory.

   ```
   C:\>
   ```

2. Insert the boot floppy disk with the xPC Target kernel into the external floppy disk drive.

3. Type

   ```
   copy a:\xpc_osc3.rtb c:
   copy a:\xpcboot.com c:
   copy a:\autoexec.bat c:
   ```

4. Remove the floppy disk, and then, on the xPC TargetBox, press the **Reset** button.

   The file `autoexec.bat` includes the command `xpcboot.com` to start the xPC Target kernel and the target application running.

   Note If you get the error message

   ```
   Cannot boot RTTarget-32 from Virtual 8086 mode.
   ```

   edit the `config.sys` file to comment out or remove the line that loads the high memory driver.

   ```
   REM DEVICE=C:\DROS\EMM386.EXE DPMI=OFF FRAME=NO
   ```

5. On the xPC TargetBox keyboard, press the spacebar.

   A command line opens on the xPC TargetBox screen.

For a complete list of target PC commands, see the xPC Target user documentation.

Booting DOS Instead of the xPC Target Kernel

After you copy the file `autoexec.bat` for DOSLoader or StandAlone mode to the flash disk, the xPC TargetBox will always start DOS and then immediately boot the xPC Target kernel and, with stand-alone mode, start the target application. You might want to return your system to boot into DOS. Use one of the following methods:

- Stop the xPC Target kernel from booting using **Ctrl-C**, and then copy the backup file `autoexec01.bat` with DOS on the flash disk to `autoexec.bat`.
- Create the DOS system on floppy disk, boot DOS from the floppy disk, and then copy the backup file `autoexec01.bat` with DOS on disk to `autoexec.bat`.

FTP File Transfer

You can create larger target applications (up to 16 MB) that do not fit on a floppy disk. For these applications you can boot the xPC TargetBox in DOSLoader mode and transfer the target application from the host PC (serial or network communication), or you can copy files to the xPC TargetBox using a network connection and File Transfer Protocol (FTP).

Another application for FTP transfer is when the xPC TargetBox is in a rugged environment where you do not want to use a mechanical disk drive. In this case, you can remove the floppy disk drive and attach a crossover Ethernet cable from an xPC TargetBox to a host PC.

This section includes the following topics:

- **Directories and Files on the xPC TargetBox** — Description of the directories and files on the flash disk
- **Booting xPC TargetBox with FTP Server** — Edit the file `autoexec.bat` to boot the xPC TargetBox with the FTP server running
- **Using FTP Commands** — List of common FTP commands you enter from the host PC
- **Copying Files with DOS and FTP** — Use FTP to copy stand-alone applications from a host PC to an xPC TargetBox
- **Copying Files with MATLAB and FTP** — Use an xPC Target function from MATLAB to establish an FTP connection and copy stand-alone applications from a host PC to an xPC TargetBox
- **Booting xPC TargetBox to DOS** — Edit the file `autoexec.bat` to boot the xPC TargetBox into DOS

Directories and Files on the xPC TargetBox

You need these files to run the FTP server with the network card in the xPC TargetBox.

Directory or Filename	Description
`kernel.sys`	FreeDOS
`command.com`	FreeDOS

`autoexec.bat`
`autoexec.tst`
`autoexec.dos`
`autoexec.ftp`
	Copy one of the `autoexec.*` files to `autoexec.bat` to boot into DOS (`autoexec.dos`), start the FTP server (`autoexec.ftp`), or run the xPC Target kernel and self-test (`autoexec.tst`).

`FDOS <DIR>`	FreeDOS
`FTP <DIR>`	Backup files for the FTP server
`INTEL <DIR>`	Drivers for the Ethernet connection

`TESTS <DIR>`
`doststｰexe`
`xpcboot.com`
`xpctbtst.rtb`
`autoexec.bat`
	Files to run a self-test for the xPC TargetBox and I/O boards. Do not confuse `c:\work\autoexec.bat` with `c:\autoexec.bat`. `c:\work\autoexec.bat` loads and runs the target application or self-test.

`WORK <DIR>`
`xpcboot.com`
`application.rtb`
`autoexec.bat`
	Files created from building a stand-alone application on the host PC and copied to the xPC TargetBox

`ftpbin.exe`
`telpass.exe`
`config.tel`
`password.tel`
	Files to start the FTP server (`ftpbin.exe`), set IP address (`config.tel`), and change password (`telpass.exe`)

`unldftp.bat`
`bootxpc.exe`
`loadxpc.bat`
	Unload FTP server (`unldftp.bat`), and unload FTP server with Ethernet drivers (`bootxpc.exe`, `loadxpc.bat`)

`reboot.exe`	Reboot the xPC TargetBox

Booting xPC TargetBox with FTP Server

When you first receive your xPC TargetBox, the system is configured to run a self-test. If you set up a network connection to xPC Target, you can use an FTP server to copy files from a host PC to the xPC TargetBox.

After you set up your host PC for network communication, you can connect to an xPC TargetBox using FTP. For information on connecting your host PC to a network, see your system administrator.

1 On the xPC TargetBox, press the **Reset** button.

2 You need to cancel the DOS LED test and stop the xPC Target kernel from loading. Press the **Ctrl-C** keys repeatedly until you see the message

   ```
   Terminate batch file ’C:\AUTOEXEC.BAT’ (Yes/No/All)?
   ```

3 Type `Y`. The system boots into the self-test directory

   ```
   c:\work
   ```

4 Copy the autoexec file configured for running the FTP server. Type

   ```
   copy c:\autoexec.ftp c:\autoexec.bat
   ```

5 Open the FTP configuration file. Type

   ```
   cd c:\work
   edit config.tel
   ```

6 Change the IP address to your xPC TargetBox. For example, enter

   ```
   myip=192.168.0.1
   ```

7 Save and close the file. Press **Alt-F** to open the `File` menu, use the down arrows to select `Exit`, and then press **Enter**.

8 On the xPC TargetBox, press the **Reset** button.

   The system boots the FTP server.

   ```
   ftp>
   ```

Using FTP Commands

After you start the FTP server on the host PC and passwords are checked, you are prompted for individual FTP commands. These commands are documented in the manuals for the host computer, but most FTP implementations have similar commands because they are modeled after the Berkeley UNIX version of FTP. The following table lists FTP commands that are common to most implementations.

Command	Action
`ascii`	Set transfer mode to ASCII (default).
`binary`	Set transfer mode to binary. This is the mode you need to transfer target applications to the xPC TargetBox.
`cd <path>`	Set a new default directory on the xPC TargetBox.
`dir`	Show directories and files in the xPC TargetBox default directory.
`pwd`	Show current xPC TargetBox directory name.
`get <filename>`	Get a file from the xPC TargetBox and send it to the host PC.
`put <filename>`	Send a file from the host PC to the xPC TargetBox. For stand-alone applications you need to transfer the files `xpcboot.com`, `autoexec.bat`, and `<application_name>.rtb.`
`quit`	Exit the FTP server on the host computer.

If you need more information about the File Transfer Protocol, you can find a wealth of information on the internet. One Web site is

```
http://archive.ncsa.uiuc.edu/SDG/Software/PCTelnet/
```

Copying Files with DOS and FTP

You do not need a floppy disk drive attached to an xPC TargetBox to copy stand-alone applications from the host PC. Instead, you can connect your xPC TargetBox to a network and use FTP to transfer files to the flash disk.

After you create a xPC Target stand-alone application, you can copy the files to the xPC TargetBox flash disk. To create the needed stand-alone files, see "StandAlone Mode" on page 5-9.

1. Boot the xPC TargetBox with the FTP server. See "Booting xPC TargetBox with FTP Server" on page 5-15.

2. On the host computer, open a DOS shell. Start the FTP server with the IP address for your box. For example, type

   ```
   ftp 192.168.0.1
   ```

   The host PC displays

   ```
   Connected to 192.168.0.1
   220 XPC PC FTP server v2.55 -- at 192.168.0.1
   User (192.168.0.1:(none)):
   ```

3. Log in to the server by pressing `Enter`.

   The host PC logs into the directory `c:\work` and displays the messages

   ```
   230 User logged in
   ftp>
   ```

4. Set the host FTP to binary transfer. On the host PC, type

   ```
   binary
   ```

5. Copy files from the host PC to the xPC TargetBox. For example, if you created a stand-alone application from the Simulink model `xpcosc.mdl` in your MATLAB working directory `c:\mwd`, type

   ```
   put c:\mwd\xpcosc_xpc_emb\xpcosc.rtb
   ```

The host PC displays the following:

```
200 Selected data port
150 Trasferring binary file “xpcosc.com”
226 Transfer complete
ftp: 16536 bytes sent in 0.00 Seconds
ftp>
```

6 Repeat step 2 for the boot loader and the `autoexe.bat` file that loads the kernel and application.

```
put c:\mwd\xpcosc_xpc_emb_xpcboot.com c:
put c:\mwd\xpcosc_xpc_emb\autoexec.bat c:
```

7 Quit the FTP server. On the xPC TargetBox, type

```
quit
```

The directory changes to `c:\xpc`.

8 Unload the FTP server. Type

```
unldftp
```

9 Run the target application. For example, type

```
autoexec
```

This file was created by the build process for a stand-alone application and includes the command `xpcboot xpcosc.rtb`. First the kernel is loaded, then the target application, and finally the target application is started.

If you want your stand-alone application to run when you press the **Reset** button on your xPC TargetBox, you need to edit the file `c:\autoexec.bat` to include the lines

```
cd c:\work
autoexec
```

Copying Files with MATLAB and FTP

As an alternative to transferring files to an xPC TargetBox using a DOS shell and FTP, you can use MATLAB.

After you build a stand-along application, you can transfer the application and utility files (`xpcboot.com`, `autoexec.bat`) to an xPC TargetBox.

1. Boot the xPC TargetBox with the FTP server. See "Booting xPC TargetBox with FTP Server" on page 5-15.

   The system boots the FTP server.

2. In the MATLAB Command Window, type

   ```
   xpcsetup
   ```

   The **xPC Target Setup** dialog box opens.

3. Enter information for a network connection. See"Network Communication" on page 3-6.

4. For example, if your application is `xpcosc.rtb`, type

   ```
   xpctgboxdl('xpcosc')
   ```

   An FTP connection is established between the host PC and the xPC TargetBox. The application and the required `bat` file (`autoexec.bat`) are downloaded to the directory `c:\work`.

   If the download was successful, MATLAB displays a confirming message. Otherwise, MATLAB displays an error message.

Booting xPC TargetBox to DOS

When you first receive your xPC TargetBox, the system boots the xPC Target kernel and runs a self-test. If you want your system to boot into DOS, you need to edit the file `autoexec.bat`.

1 On the xPC TargetBox, press the **Reset** button.

2 You need to cancel the DOS LED test and loading the xPC Target kernel. Press the **Ctrl-C** keys repeatedly until you see the message

```
Terminate batch file 'C:\AUTOEXEC.BAT' (Yes/No/All)?
```

3 Type `Y`. The system boots into the self-test directory.

```
c:\tests
```

4 Copy the `autoexec` file configured for booting into DOC. Type

```
copy c:\autoexec.dos c:\autoexec.bat
```

5 On the xPC TargetBox, press the **Reset** button. The screen displays

```
Loading FreeDOS
. . .
C:\>
```

Index