A

Quick Start

xPC Target Quick Start A-2
Installation and set up. A-2
Running a Simple xPC Target Application A-4
Other useful commands: A-6

xPC Target Quick Start

In this appendix, we provide a very brief summary for those who are anxious to start creating and executing xPC Target real-time applications on the target PC without further delay.

This quick start provides the minimal number of steps to get a basic real-time application running without the inclusion of any I/O hardware. For more detailed instructions, or if you have any difficulties, please refer to "Setting Up Serial Communication" on page 2-11, "Creating a Target Application" on page 3-6, and "Running the Target Application on the Target PC" on page 3-12.

Installation and Setup

1. Install MATLAB, Simulink, Real-Time Workshop, and xPC Target from your CD-ROM or Web downloadable.

2. Install your C compiler. You can use either Microsoft Visual C/C++ version 5.0 or 6.0, or Watcom C/C++ compiler version 10.6 or above. If you are unsure of the compiler installation, you may prefer running the compiler CD from your CD drive. However, as you continue to use the product, you will probably prefer to have a permanent installation of the compiler on your hard disk.

3. Connect the serial cable from your host PC to your target PC on any of the available serial ports. Record whether you have selected to use COM1 or COM2 on the host computer. This information will be needed shortly.

   A serial cable and an ethernet card are provided with xPC target. This enables you to choose either a serial or a network connection between your host PC and target PC. For this quick start chapter, we recommend using the serial cable connection, which simplifies connectivity. Although communication is at a slower rate, it is easy to set up. You can always use the network connection later.

   Note On some older PCs you may need to enable a particular serial port by pressing delete on start-up and selecting this setting manually. Be sure that the serial ports on both the host PC and target PC are enabled in the BIOS set up.

4 Run MATLAB, and then bring up the set-up GUI by typing

```
xpcsetup
```

5 In the xPC Target Setup Window, enter the letter of the driver and root path where you have installed your Microsoft Visual C/C++ compiler, or your Watcom C/C++ compiler.

6 In the **RS232HostPort** text box, select either **COM1** or **COM2** for the connection on the host PC that you have recorded in step 3.

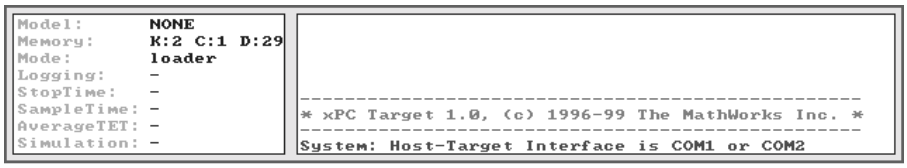
Note The port on the target is automatically determined.)

7 Click the **Update** button.

8 Place a formatted, writable 3.5-inch floppy disk in the host PC. Click the **BootDisk** button in the xPC Target Setup Window to create a target boot disk.

9 Once xPC Target has completed writing to the target boot disk, remove the target boot disk from the host PC drive and insert it into the target PC.

10 Reboot the target PC. During reboot, xPC Target boots from the target boot disk. If a monitor is connected to your target PC, the monitor should appear as follows:



11 At the MATLAB prompt, test your setup by typing

```
xpctest
```

If any error messages appear, see “Troubleshooting Serial and Network Connections” on page 2-26.

Running a Simple xPC Target Application

Assuming you have not encountered any error messages, the following section illustrates how you can run an xPC Target application on your target PC.

1 Open the Simulink oscillator model.

```
xpcosc
```

2 In the Simulink window, and from the **Tools** menu, click **RTW Build** to initiate the build process and automatically download the real-time application.

3 Once the download process is completed, run the application by typing

```
+tg
```

4 View the target object by typing

```
tg
```

In this case, you will now notice that `TimeLog`, `StateLog`, `OutputLog`, and `TETLog` all have matrices with nonzero dimensions since the model has already run.

5 Plot the logged signal data versus time.

```
plot(tg.TimeLog, tg.OutputLog)
```

In this case, the signals that are plotted are those which are connected to the Simulink Scope block.

6 Plot the task execution time (TET), which shows the actual computation time required at each time step.

```
plot(tg.TimeLog, tg.TETLog)
```

7 Add a scope object, which provides greater flexibility for viewing data. First, start by increasing the time at which the real-time application will stop.

```
tg.StopTime = 1000
```

If you are using particularly slow hardware, (for example, a 386 or 486 target PC), we recommend reducing the sample time. If your target PC has a Pentium processor, it should be able to run at the faster default rate. To reduce the sample time to 1 millisecond, type

```
tg.SampleTime=.001
```

8 Obtain signal IDs for Intergrator1 and Signal Generator blocks.

```
sigid1 = getsignalid(tg, 'Integrator1')
sigid2 = getsignalid(tg, 'Signal Generator')
```

9 Create a scope object that runs on the target and has identifier 1.

```
sc1 = addscope(tg, 'target')
```

10 Add signals to the scope object.

```
addsignal(sc1, sigid1)
addsignal(sc1, sigid2)
```

11 Start the scope object which runs on the target.

```
+sc1
```

12 With the scope object already running, now start the target application, by typing

```
+tg
```

Signals should be visible on the target PC monitor. Congratulations! You now have your first xPC Target application example up and running in real time.

Other Useful Commands

Command	Summary
`xpclib`	For adding hardware device driver blocks to your model.
`xpcscope`	For tracing signals and displaying them on the host PC.
`xpctgscope`	For tracing signals on the target PC.
`help xpc`	Full list of xPC Target commands.