

第3章 xPC Targetのハロー・ワールド

ホストとターゲットに関してパソコンの準備できたので、システムが正常に動作するかどうかテストします。

xPC Targetにはテスト・プログラムが用意されています。最初に、このプログラムを実行してシステムの動作を検証します。

テストの結果が合格ならば、パソコンの準備作業は正しく行われたということになります。

もし、不合格ならば、第2章に戻りパソコンの準備作業を見直す必要があります。

テストに合格した場合は、次にテストと同じ過程を手作業で実行し、細部の動作を確認します。簡単な問題を設定してSimulinkモデルを作り、ブロック線図をビルドしてプログラムを作成します。そして、そのプログラムをターゲットにダウンロードして、実行した結果をチェックします。

これが、ここで行うxPC Targetのハロー・ワールドの作業手順です。

■ 3.1 ターゲットの起動

ホストとターゲットをケーブルで結び、電源をONにして、システムが正常に起動するかどうかをチェックします。

3台のターゲットを用意しましたが、処理の進行はほとんど同じなので、紙面を省くためにディスク分離モデル、すなわちターゲット1の場合について述べます。

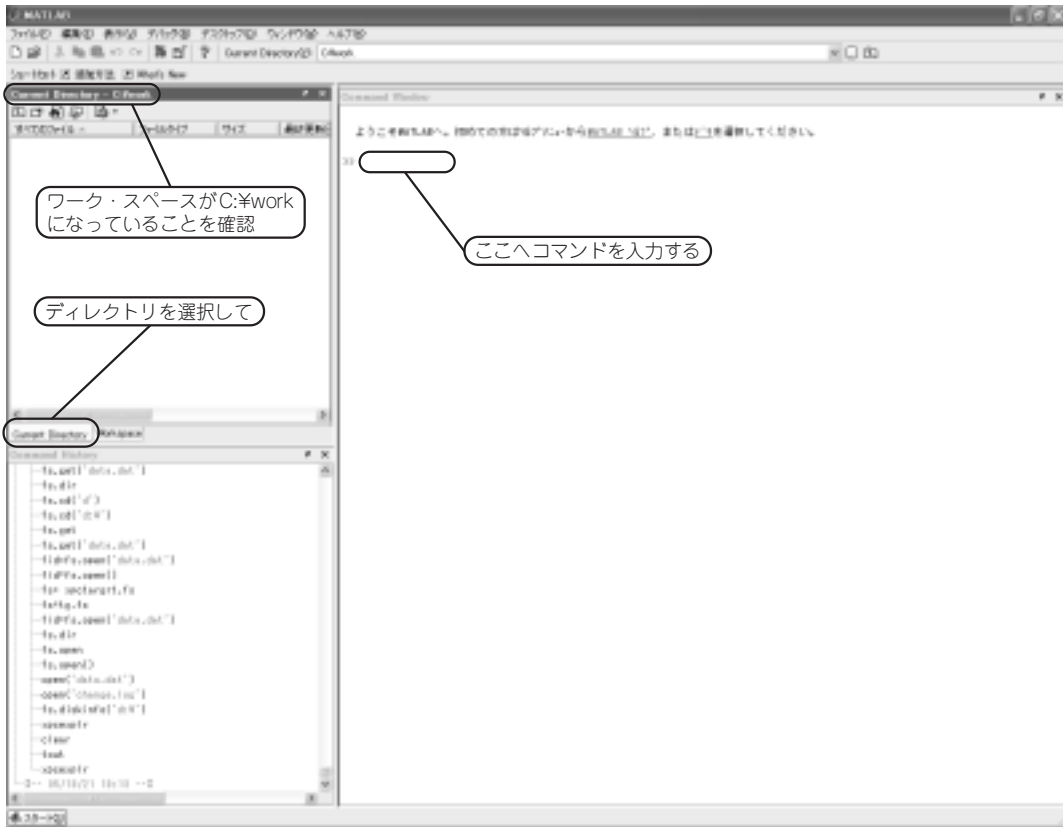
ターゲットのモデルの違いによって、画面表示や処理結果などが異なる場合のみ補完的にターゲット2、ターゲット3に関する説明を追加します。

以下において、「ターゲット」は「ターゲット1」と読み替えてください。

まず、ターゲットの電源をOFFにしてホストを立ち上げ、MATLABを起動します。画面3.1となります。

ここで、MATLABのワーク・スペースがMATLABのシステム・ディレクトリの外にあることをチェックします。

画面に示したように、[Current Directory]タブをクリックして、表示のワーク・スペースがシステム・ディレクトリの外(すなわち、私の場合は、c:\work)になっていることを確認します。そうなる



画面3.1 MATLABの起動画面

いない場合は、ワーク・スペースの場所を移動します。

MATLABのコマンド・ラインから、

```
>> cd c:\work
```

と入力すると、[Current Directory]はc:\workディレクトリへ移動します。

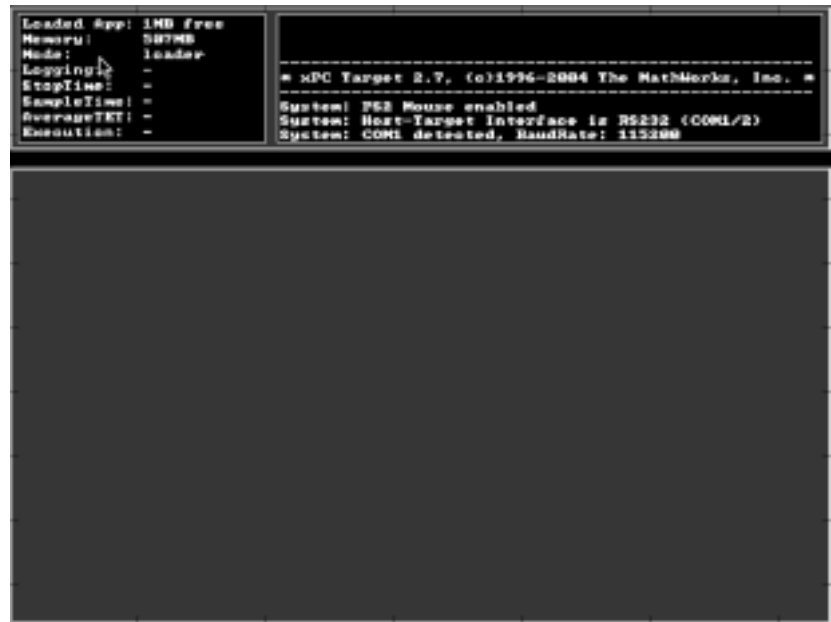
今回は、実際にワーク・スペースを使用してプログラムをビルドするので、厳密にチェックします。

ホストのパソコンからターゲットのパソコンへ場所を移動し、前章で作成したブート用のフロピ・ディスクをターゲットへ挿入して、電源スイッチをONにします。

ターゲットは、ブート・ディスクを読み込み、ホストからプログラムを受け付ける状態に入ります。ターゲットの初期画面を画面3.2に示します。

もし、この時点でターゲットがこの画面を表示しない場合、あるいは表示したとしてもここで示した画面と異なる場合は、前章へ戻ってパソコンの設定が正しいかどうか再点検してください。

細かいことを言うと、ターゲットの起動画面は画面3.2と同じではありません。右上のペインにおける表示は、



画面3.2 ターゲットの起動画面

PS2 Mouse enabled

Host-Target Interface is RS232 (COM1/2)

2行のみであり、第3行目の、

COM1 is detected, BaudRate: is 115200

は、この時点においては表示されていません。

第3行目は、ホストとターゲットが通信を行ったあとに表示されます。

ターゲットの画面を取り込むために、ホストとターゲットが通信を行ったので、画面3.2となりました。

ターゲット起動直後の画面は取り込むことができません。皆さんの実機において確認してください。ただし、これは表示の問題であって、プログラムの本質にかかわる問題ではありません。

ここでは、ターゲットにおいて正常な画面が表示されたとします。ターゲットの画面の左上方に、マウスのアイコンがあります。ターゲットのマウスをつかんで、動かしてください。画面上のマウスのアイコンは動きません。マウスは、現在非アクティブの(表示されているが動作しない)状態です。

画面に表示される内容について説明します。画面3.2の左上方に、

Loaded App	1MB free
Memory	507MB
Mode	loader

と表示されています。

ブート・ディスクを作成する際に、アプリケーション・プログラムの容量を1MBに設定しました。このプログラム領域は、現在1MBフリーなので、アプリケーション・プログラムの領域は空の状態、すなわち、プログラムはダウンロードされていないことを示しています。

私のターゲット1は、512MBのメイン・メモリを搭載しています。その中の507MBが使用可能状態です。この数値は、使用したパソコンに搭載されているメイン・メモリの容量によって変わります。

ターゲットのモードはローダの状態、すなわちホストがプログラムをダウンロードするのを待っている状態です。

画面3.2の右上方に、

PS2 Mouse enabled

Host-Target Interface is RS232 (COM1/2)

COM1 is detected, BaudRate: is 115200

と表示されています。

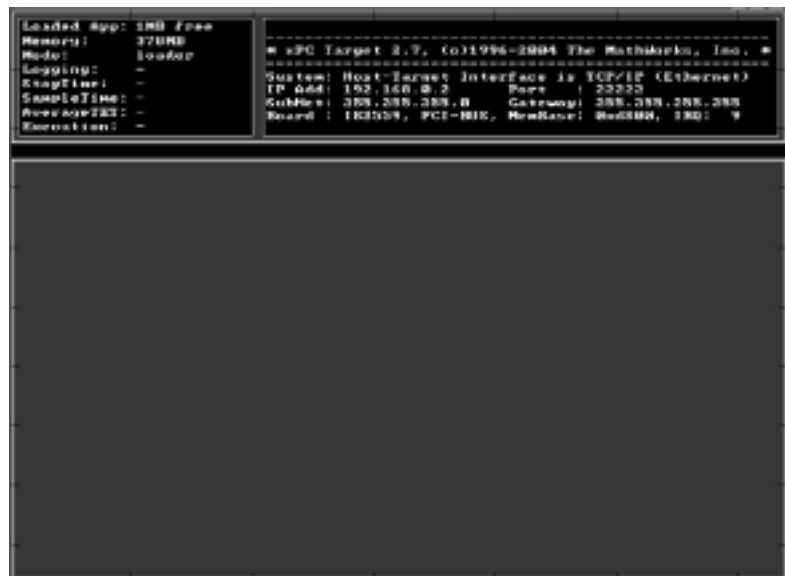
第1行目は、マウスがPS2であることを示します。

第2行目は、ホスト-ターゲット間通信がシリアル通信であることを示します。

第3行目は、最初の通信が行われたあとに表示される行です。

COM1が使用され、ボー・レートは115200となります。これらの記述は、ブート・ディスクを作成したときの条件と一致することを確認します。

ターゲットは正常に起動しました。参考のために、TCP/IPの場合、すなわちターゲット2の場合の起動画面を画面3.3に示します。



画面3.3 LANモデルの起動画面

右上方の表示にTCP/IPの設定値が表示されています。

第1行目は、ホスト-ターゲット間通信がTCP/IP、イーサネットであることを示します。

第2行目は、IPアドレスとポート番号を示します。

第3行目は、サブネット・マスクとゲートウェイのアドレスを示します。

第4行目は、使用したネットワーク・ボードに対して割り付けられたIOアドレス、割り込みポート番号などを示します。

これらの設定値は、ブート・ディスクを作成したときの条件と一致します。ターゲット2も正常に起動しています。ターゲット3も同様に正常に起動しました。

私が制作した3台のターゲットは、正常に起動しました。

■ 3.2 テストの実施

xPC Targetは、新規にシステムを構築した際の診断用としてテスト・プログラムを用意しています。

まず、このテスト・プログラムを実行し、システムの構築に誤りがないかどうかをチェックします。

テストはホストとターゲット間において実施します。このため、説明はホストとターゲットの間を往復します。区切りを明確に示すために、本節では、ホストからターゲットへ、あるいはターゲットからホストへ切り替わる場所において、**ホスト**、**ターゲット**の表示によって注意を喚起しています。

ホストにおいて[xPC Target Explore]を立ち上げます。

MATLABのコマンドランから、

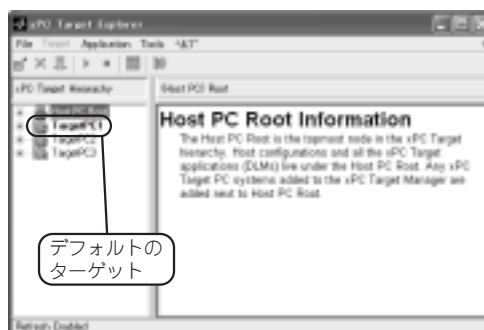
```
>> xpcexplr
```

と入力します。画面3.4の[xPC Target Explorer]が開きます。

[xPC Target Explorer]の[xPC Target Hierarchy]ペインにおいて、ターゲット1がデフォルトのターゲットになっています。

画面3.4の左のペインにおいて、

TargetPC1



画面3.4 xPCのエクスプローラ

の文字が、ここに示したように太字の表示になっています。これがデフォルト・ターゲットの表示です。
もし、ターゲット1がデフォルトになっていない場合は、[xPC Target Hierarchy]のペインにおいて、

TargetPC1

を右クリックしてコンテキスト・メニューを開き、

[Set As Default]

をクリックします。TargetPC1の文字が太字表示のTargetPC1に変わります。

テストは、デフォルトのターゲットに対して実行されるので、ここをかならずチェックします。

ターゲットが1台の場合は、そのターゲットが常にデフォルトになるので、ここで述べた確認作業は不要です。直ちにテスト・プログラムを実行する手順に入ります。

それでは、テストを実行します。

ターゲットは、ブート直後の状態(すなわち、ローダの状態)とします。

ターゲットは、ホストから送られてくるプログラム待ちの状態です。

このターゲットに対して、ブート用の3.5インチのフロッピー・ディスクをセットします。ここでセットというのは、ターゲットがいつでもフロッピー・ディスクを読むことができる状態を意味します。

ホストのMATLABのコマンド・ラインから、

```
>> xpctest
```

と入力します。MATLABはテスト・スクリプトを読み込み、ホスト-ターゲット間における自動テストを開始します。テストは#1から#8まで8個のテストを行います。

まず、ホストは、ホスト-ターゲット間の通信が正常に動作するかどうかをチェックします。ホスト-ターゲット間の通信はシリアル通信です。

TCP/IPのPingを発行する#1テストは、ターゲット1の場合実施する意味がないので、スキップします。

#2テストは、シリアル通信のチェックを行うために用意されているxpctargetpingを実行します。

このテストがNGならば通信不能、つまりホストからターゲットへデータを送ることができません。すなわちテストの続行は不可です。この場合、テストは中断です。

OKならばホストとターゲット間の通信は確立されているので、#3のテストへ入ります。

まず、ホストはターゲットに対して[リセット]コマンドを発行します。このコマンドを以下において、ソフトウェア・リセットと呼びます。

これに対して、ターゲット・パソコンのスイッチを押してターゲット・パソコンを物理的にリセットする行為を、ハードウェア・リセットと呼びます。本書では、両者を厳密に区別します。

ターゲットに、ソフトウェア・リセットがかかって、ターゲットの画面はクリアされます。

続いて、パソコンのBIOSの処理が行われ、そのあとにフロッピー・ディスクが読み込まれて、画面3.2(または画面3.3)の初期画面が再度表示されます。

このテストから、ホストからのリセット・コマンドによってターゲットが自動的に立ち上がることが検証されました。

画面3.5 xpcostの結果

```

>> apcost

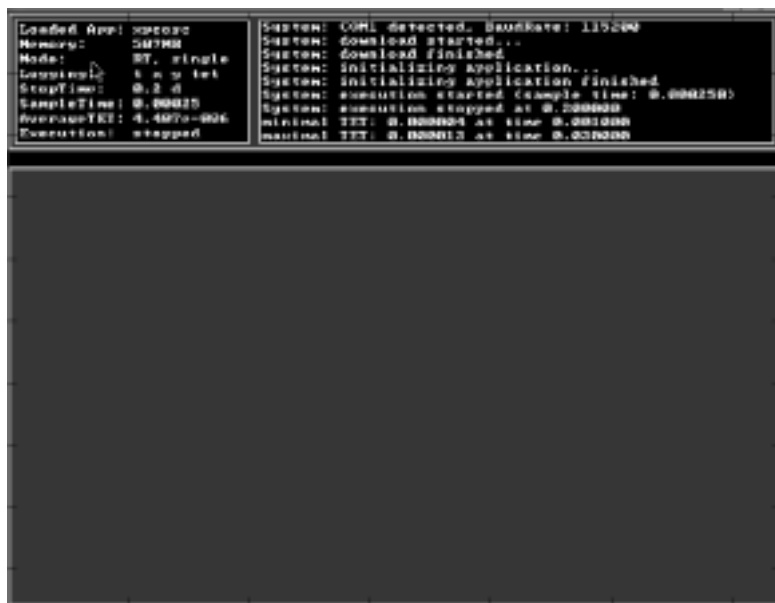
### xPC Target Test Suite 3.7.2
### Host-Target interface is: R2012c R2012c
### Test 1, Ping target system using standard ping ... OK
### Test 2, Ping target system using ncat/targeted ... OK
### Test 3, Reboot target using direct call ... OK
### Test 4, Build and download xPC Target application using model aspect ... OK
### Test 5, Check host-target communication for commands ... OK
### Test 6, Download xPC Target application ... OK
### Test 7, Execute xPC Target application for R2012c ... OK
### Test 8, Upload logged data and compare it with simulation ... OK
### Test Suite successfully finished
>>

```

ターゲット1はCOM1を使うのでTCP/IPはスキップ

テストは成功した

画面3.6 テスト後のターゲットの画面



#4テスト～#8テストは、ホストにおいてSimulinkモデルを構築し、そのモデルをビルドして、ターゲットへダウンロードして実行し、結果のデータが正しいかどうかをチェックするテストです。

このテストに関しては、次節で手作業で実行過程をたどることにします。

以上の手続きが正常に終了すると、MATLABのコマンド・ウィンドウは画面3.5となります。

最後の行に、テストは成功したと表示されています。ここが肝心なところです。ターゲットの画面は、画面3.6となりました。

この画面から、テストにおいて、Simulinkモデル、

xpcosc.mdl

がビルドされて、ターゲットへダウンロードされ、実行されたことがわかります。これでコンパイラの設定も含めて、一連のテストは成功しました。

ほっと一息、コーヒーなどで一服しましょう。