dSPACE DS1103 Control Workstation Tutorial and DC Motor Speed Control

Tutorial

By

Annemarie Thomas

Advisor: Dr. Winfred Anakwa

May 11, 2009

Table of Contents

Introduction

The purpose of this tutorial is to introduce Bradley University's dSPACE DS1103 Workstation to new users. Use of this tutorial will minimize the time required for students (of senior undergraduate level or higher) to become proficient in using the workstation. This should allow them to spend less time learning about the workstation, so they can spend more time designing and implementing more complex control systems.

The example system used in this tutorial is a DC motor speed control system. The controller has been designed and simulated using both the Simulink and the dSPACE blocksets, the MATLAB-to-DSP interface libraries, Real-Time Interface to Simulink, and Real-Time Workshop, all located on the workstation PC. The controller will be downloaded onto the Texas Instruments' TM320F240 DSP [1] located on the DS1103 board.

A general block diagram for the system is shown in figure 1 below.

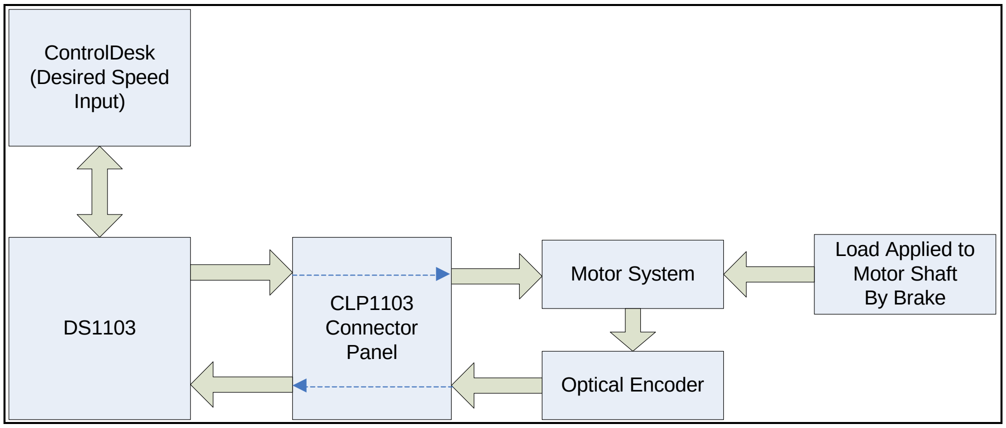

ControlDesk (Desired Speed Input)
DS1103
CLP1103 Connector Panel
Motor System
Load Applied to Motor Shaft By Brake
Optical Encoder

Figure 1: Motor Speed Control System Block Diagram.

Controller

The controller was designed using hand and MATLAB calculations and Simulink Simulation. It was then added to the Simulink Model that was downloaded to the DS1103 DSP. For more information see the final project report for this project [2].

ControlDesk

ControlDesk serves multiple purposes. It provides the interface for downloading controller models onto the DSP and provides the ability to interface with the entire system so inputs, such as the desired motor speed, can be altered and output data can be monitored on the PC display in real-time.

CLP1103 Connector Panel

The CLP1103 Connector Panel serves as an interface between the DS1103 and all external hardware. The CLP1103 Connector Panel contains connectors for twenty (20) Analog-to-Digital inputs, eight (8) Digital-to-Analog outputs, and several other connectors that can be used for Digital I/O, Slave/DSP I/O, Incremental Encoder Interfacing, CAN interfacing, and Serial Interfacing. [3]. Only the Slave I/O (for PWM output) and Incremental Encoder interfaces are used in this tutorial.

Motor System

The Motor System includes a motor and additional analog components, most of which are listed in the "Equipment" section that follows.

Optical Encoder System

The Optical Encoder System is an optical encoder connected to the motor with necessary pull-up resistors and power/ground connections added.

Equipment

Required Equipment for this tutorial is:

- The dSPACE DS1103 Workstation. (See figure 2 on the next page.)
- Pittman GM9236C534-R2 DC Motor.
- HEDS 9100 Two Channel Optical Incremental Encoder Module and 512 CPR code wheel (attached to the internal motor shaft).
- Magtrol HB-420 Brake.
- TIP120 Transistor.
- IN4004 Diode.
- SN7407 Hex Inverters.
- Connectors with wires for the pins used as inputs and outputs to the CLP1103 Connector Panel. (See more details in the "Motor Speed Control Application" section.)
- Other required/desired electronic components (resistors and wires), power supplies, and measurement devices.

All of this equipment was readily available at Bradley University at the time this tutorial was written.

Note1: The next several sections are general processes that should be followed. Skip to the section titled "Motor Speed Control Application" to follow these steps in an actual application. Even if the example application is not being used, reading through the instructions in that section may provide some additional hints and tips that might make using the system easier.

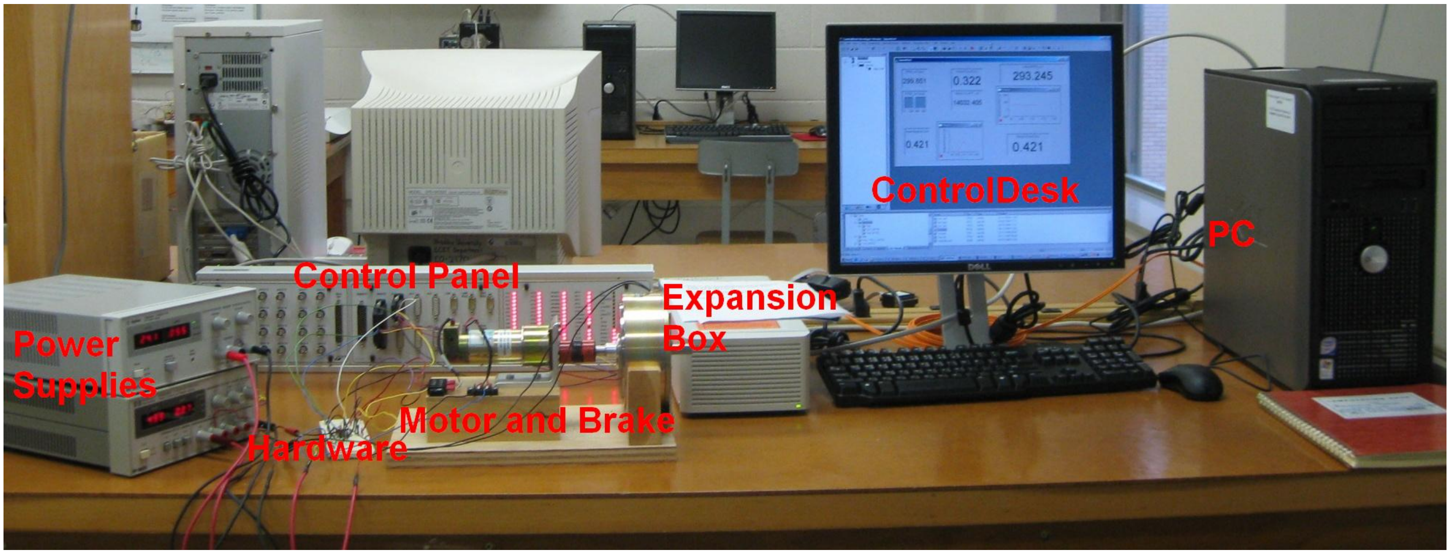


Figure 2: Workstation

Note2: If additional information is required, please refer to the final project report for this project [2] or contact Dr. Anakwa at Bradley University.

Setup/Startup

1. Turn on the DS1103 Board using the power switch on the back of the expansion box.

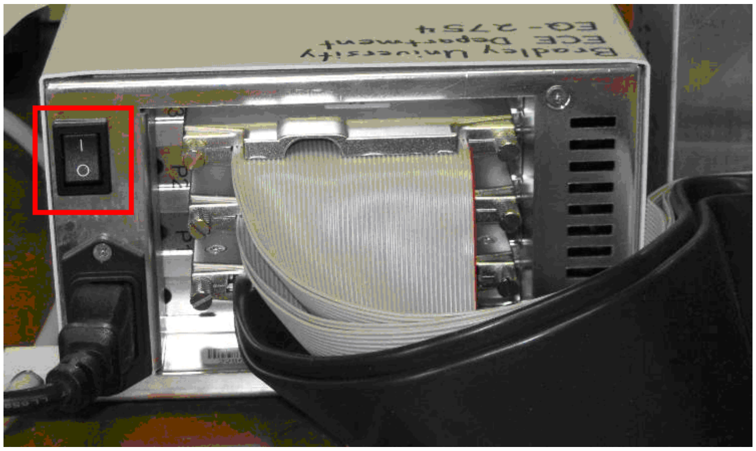


Figure 3: Back View of Expansion Box with switch highlighted.

2. Open ControlDesk from the Start Menu.
   To open ControlDesk go to: Start ► EE Applications ► dSPACE Tools ► ControlDesk

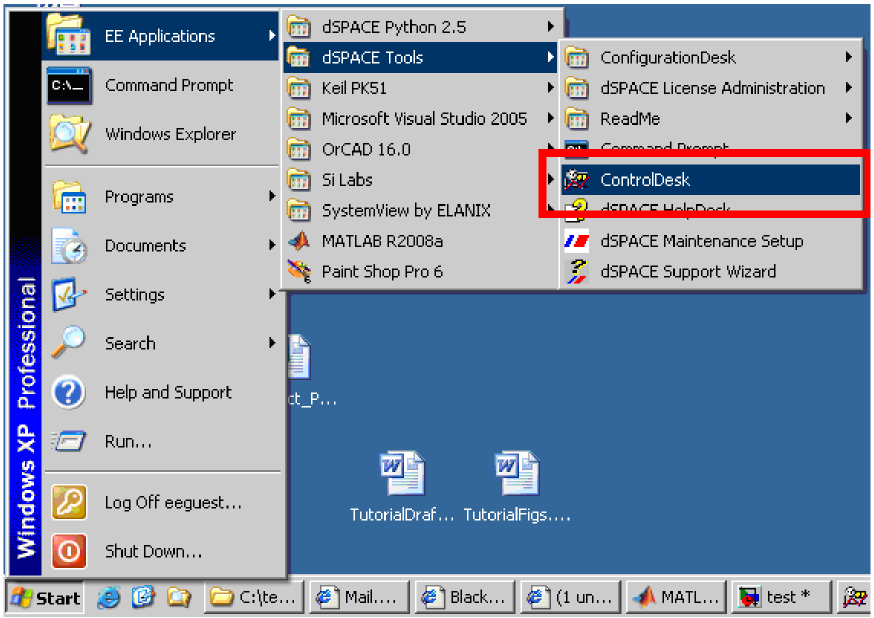
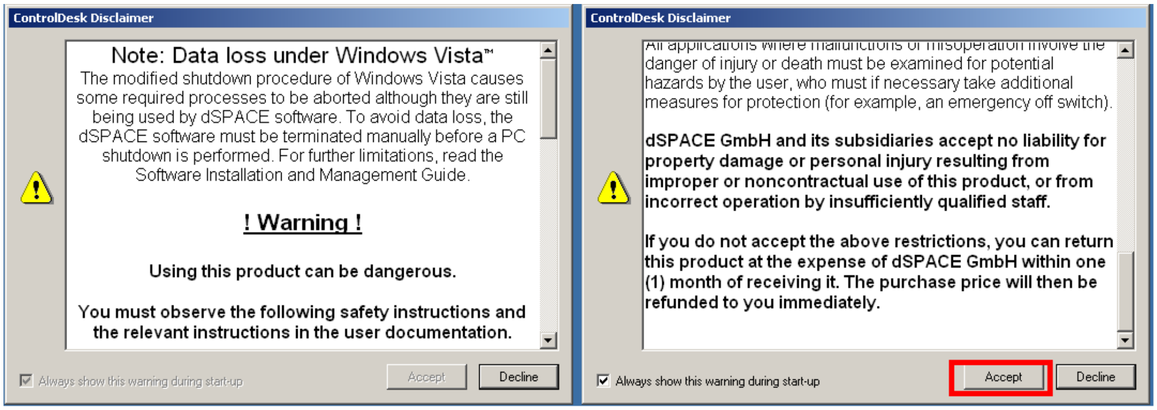


Figure 4: Location of ControlDesk Icon in the Start Menu

3. Scroll to the bottom of the ControlDesk Disclaimer dialog box when it appears and select “Accept”.



Note: Data loss under Windows Vista™

The modified shutdown procedure of Windows Vista causes some required processes to be aborted although they are still being used by dSPACE software. To avoid data loss, the dSPACE software must be terminated manually before a PC shutdown is performed. For further limitations, read the Software Installation and Management Guide.

! Warning !

Using this product can be dangerous.

You must observe the following safety instructions and the relevant instructions in the user documentation.

☑ Always show this warning during start-up

All applications where malfunctions or misoperation involve the danger of injury or death must be examined for potential hazards by the user, who must if necessary take additional measures for protection (for example, an emergency off switch).

dSPACE GmbH and its subsidiaries accept no liability for property damage or personal injury resulting from improper or noncontractual use of this product, or from incorrect operation by insufficiently qualified staff.

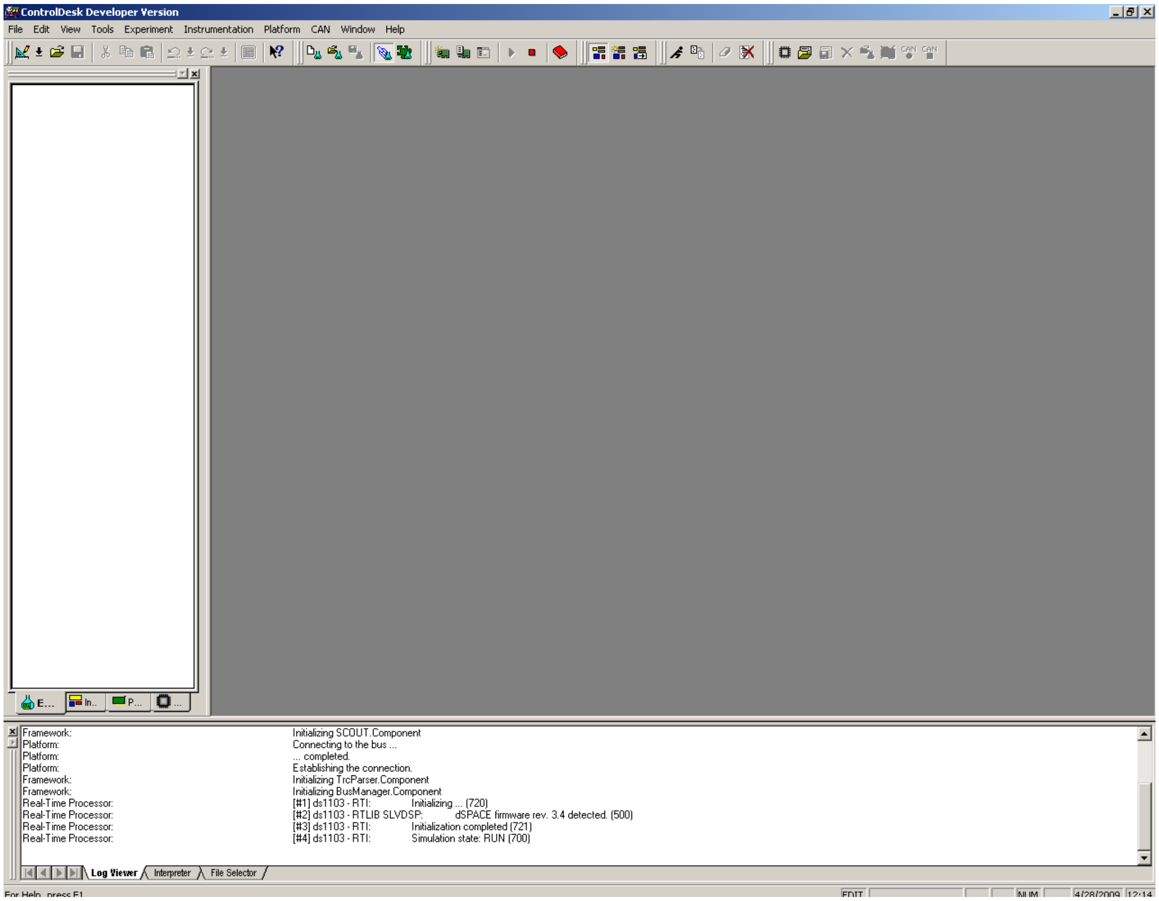
If you do not accept the above restrictions, you can return this product at the expense of dSPACE GmbH within one (1) month of receiving it. The purchase price will then be refunded to you immediately.

☑ Always show this warning during start-up

Figure 5: ControlDesk Disclaimer dialog box.

Note: This disclaimer does not affect this workstation since the PC is currently running the Windows XP operating system, not Windows Vista.

4. ControlDesk should now be open.



```
Framework:                  Initializing SCOUT.Component
Platform:                   Connecting to the bus ...
Platform:                   ... completed.
Platform:                   Establishing the connection.
Framework:                  Initializing TrcParser.Component
Framework:                  Initializing BusManager.Component
Real-Time Processor:        [#1] ds1103 - RTI:          Initializing ... (720)
Real-Time Processor:        [#2] ds1103 - RTLIB SLVDSP:        dSPACE firmware rev. 3.4 detected. (500)
Real-Time Processor:        [#3] ds1103 - RTI:          Initialization completed (721)
Real-Time Processor:        [#4] ds1103 - RTI:          Simulation state: RUN (700)
```

Figure 6: ControlDesk Window

Note: ControlDesk will complain if the DS1103 board is not turned on first.

Detecting the DS1103

Note: These steps should not need to be followed every time the system is used. These steps are generally needed the first time the workstation is setup and any time after when an error occurs in MATLAB when code is being built announces that the board has not been detected or the "ds1103" and "Slave DSP" do not appear in the "Platform" tab in ControlDesk.

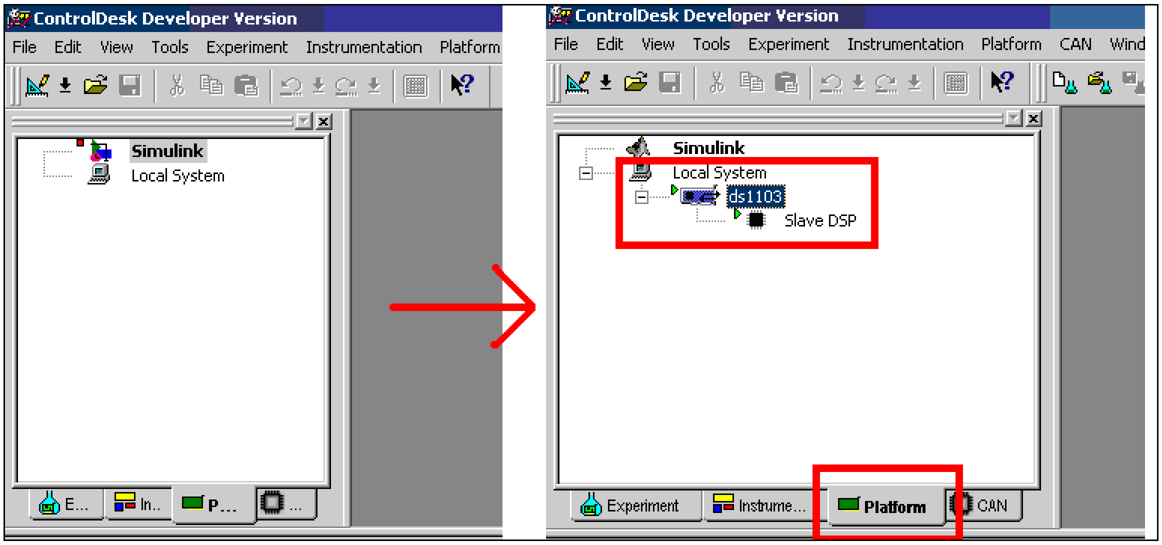

ControlDesk Developer Version
File Edit View Tools Experiment Instrumentation Platform
Simulink
Local System
E... In.. P... ...
ControlDesk Developer Version
File Edit View Tools Experiment Instrumentation Platform CAN Wind
Simulink
Local System
ds1103
Slave DSP
Experiment Instrume... Platform CAN

Figure 7: View of Platform tab in ControlDesk before and after DS1103 has been detected.

1. Select "Register" from the menu in ControlDesk.
   To select "Register", in the main menu go to: Platform ► Initialization ► Register

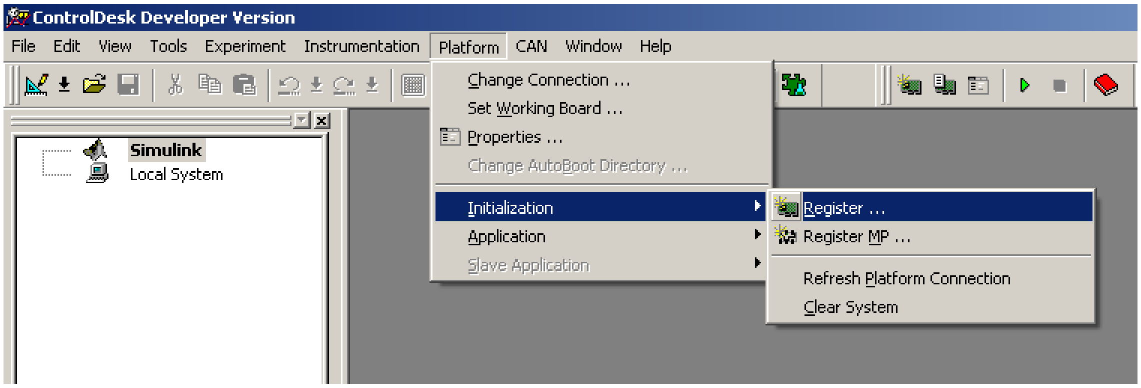

ControlDesk Developer Version
File Edit View Tools Experiment Instrumentation Platform CAN Window Help
Simulink
Local System
Change Connection ...
Set Working Board ...
Properties ...
Change AutoBoot Directory ...
Initialization
Application
Slave Application
Register ...
Register MP ...
Refresh Platform Connection
Clear System

Figure 8: Location of Register selection in ControlDesk main menu.

2. From the "Register Board" dialog box that appears, select "DS1103 PPC Controller Board" from the "Type" dropdown menu and set the "Port address" to 300. Then select the "Register" button.

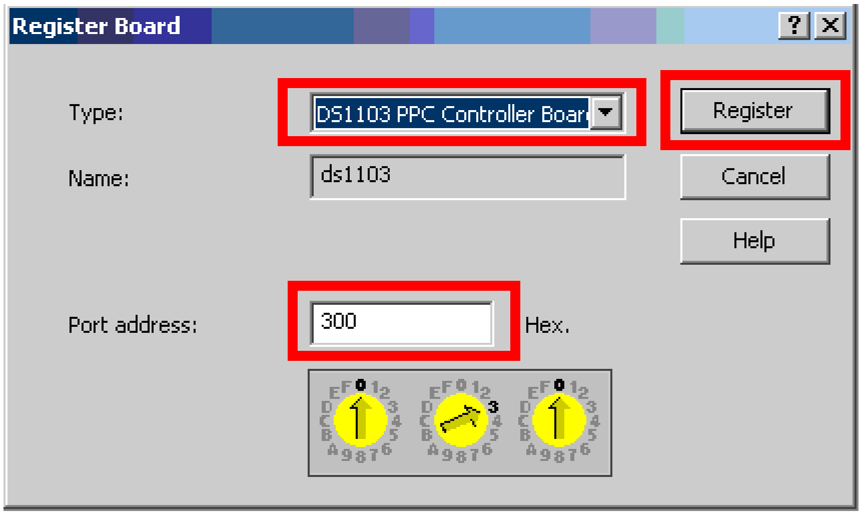


Figure 9: Correct selections in Register Board dialog box.

3. The DS1103 Board and DSP should now be reported as detected in the ControlDesk "Platform" tab.

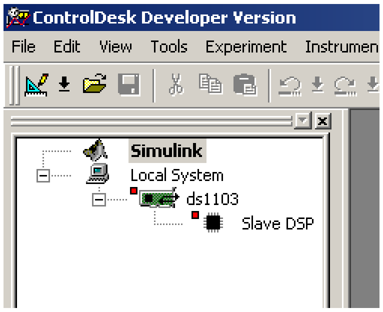
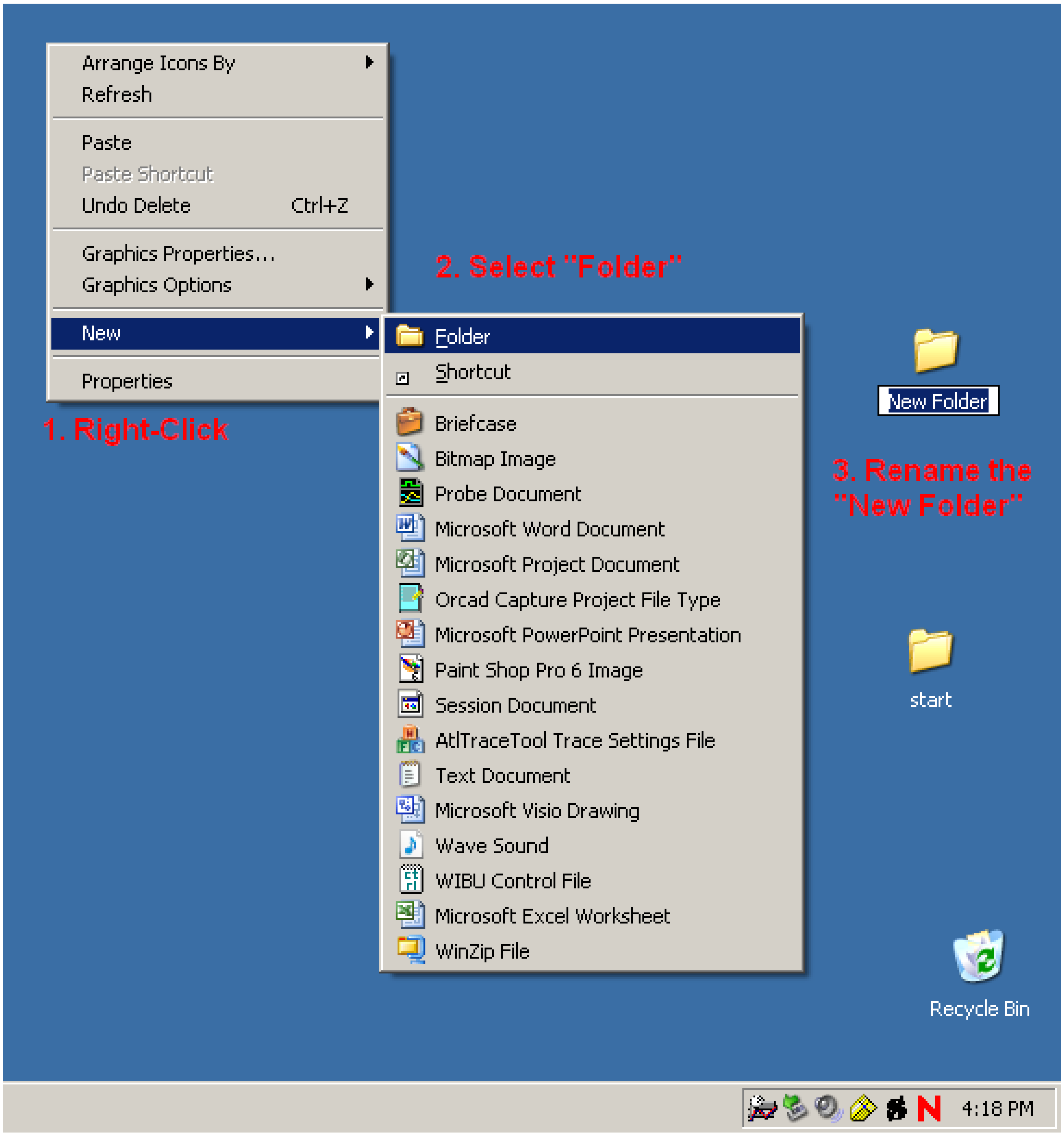

ControlDesk Developer Version
File Edit View Tools Experiment Instrumen
Simulink
Local System
ds1103
Slave DSP

Figure 10: DS1103 and Slave DSP shown in the Platform tab with the system stopped.

Creating a New Simulink Model

For best results, a new Simulink Model should be opened from ControlDesk itself. If other processes are used to create a new Simulink Model, there is a chance that there will be issues when the steps in the "Building/Downloading Model" section later in this document are followed. Once the model is created, the instructions in the "Opening MATLAB/Simulink" section can be used to re-open, edit, and build the model in the future.

1. Create a folder on the Desktop or another location on the computer for your project.



1. Right-Click

2. Select "Folder"

3. Rename the "New Folder"

2. Open ControlDesk and Select the “Platform” tab of the left menu window.

   Note: Do not close ControlDesk until all the steps in this section of the tutorial are completed.

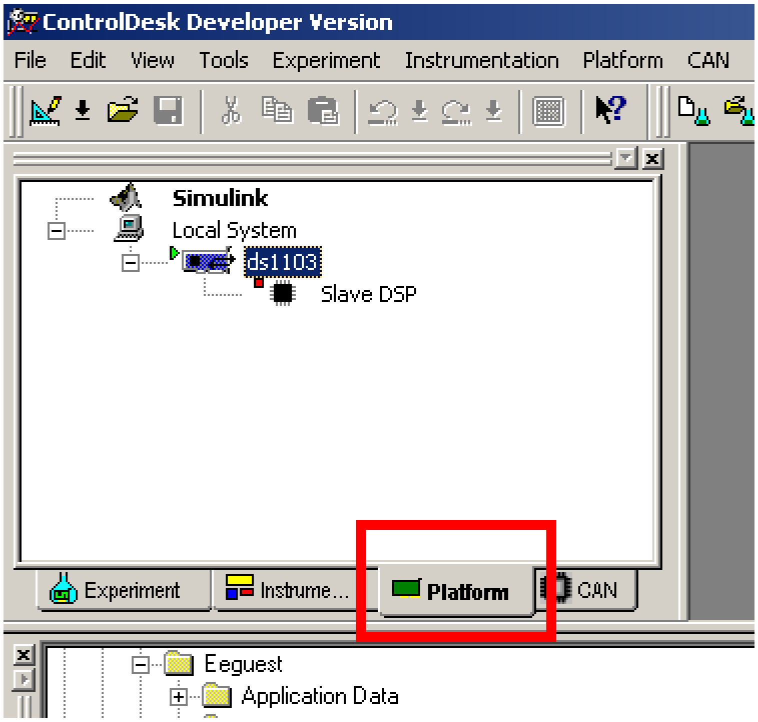


Figure 11: Location of Platform tab in ControlDesk.

3. In the “Platform” window, Right-click “Simulink” and Select “New Model…” from the menu that appears.

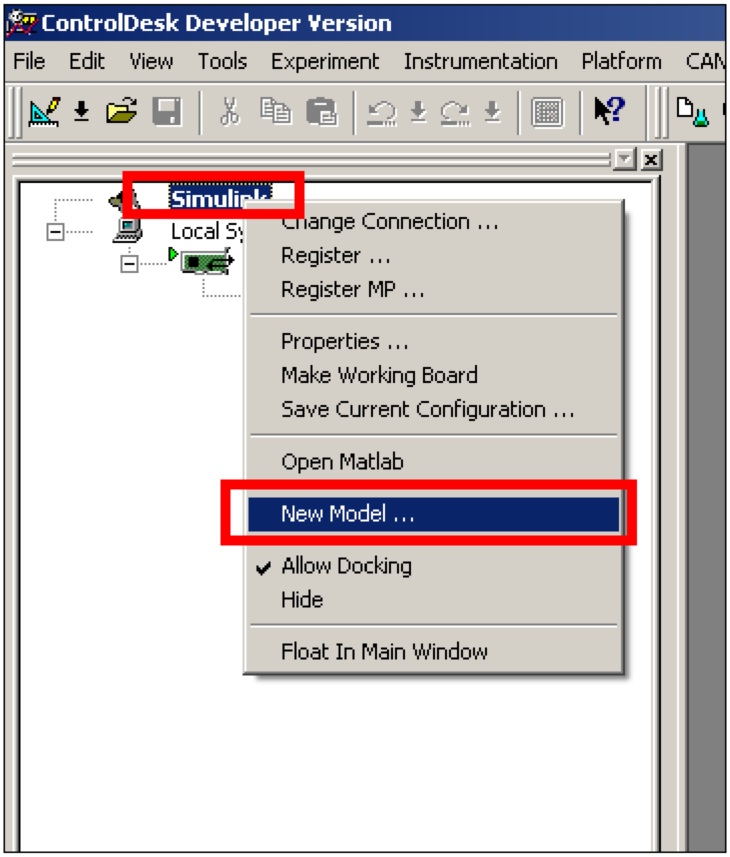
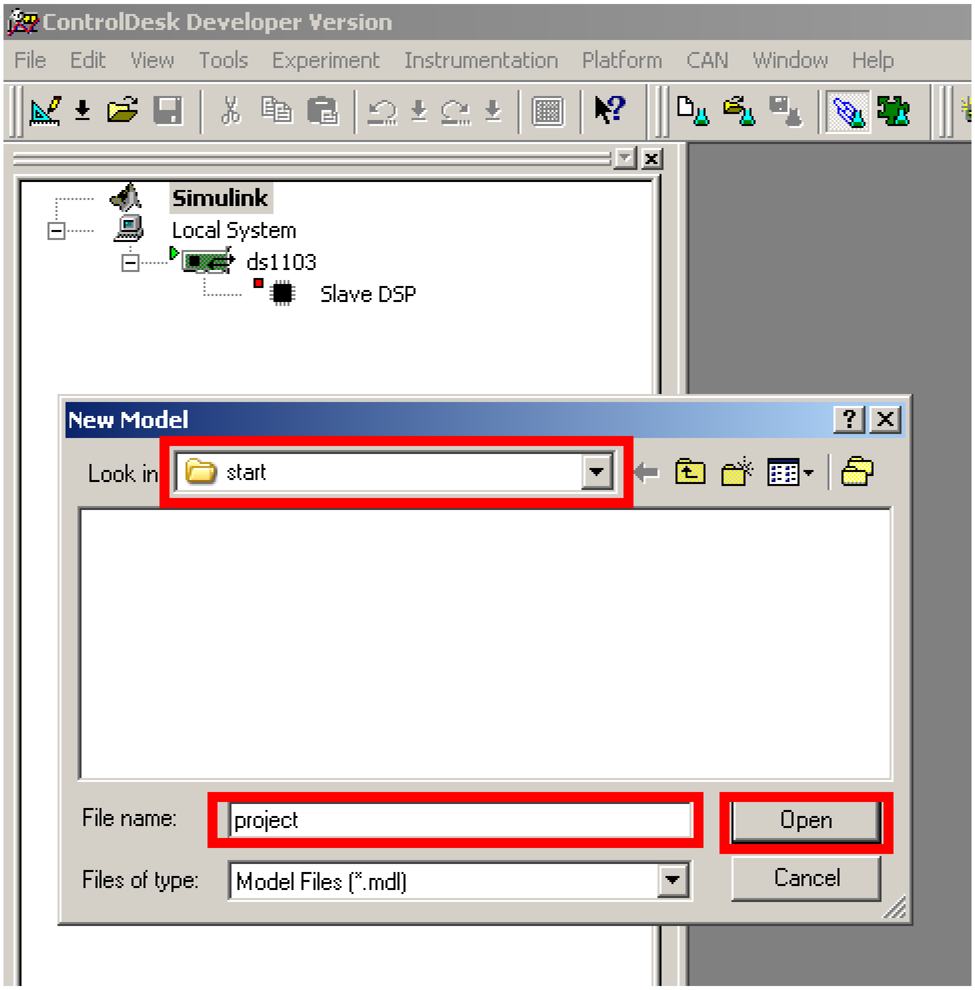


Figure 12: Creating a New Model from ControlDesk.

4. Select the folder that was created for the project from the “Look in:” drop down menu at the top of the “New Model” window. Type in a name for the new Simulink Model file in the “File Name:” box, and then, Select the “Open” button.



Look in: start

File name: project

Files of type: Model Files (*.mdl)

Figure 13: Naming the new Simulink Model File.

5. Locate the Simulink Model that opened with the name entered in the previous step and Build it.
   To Build the Model from the model's main menu go to:
   Tools ► Real-Time Workshop ► Build Model

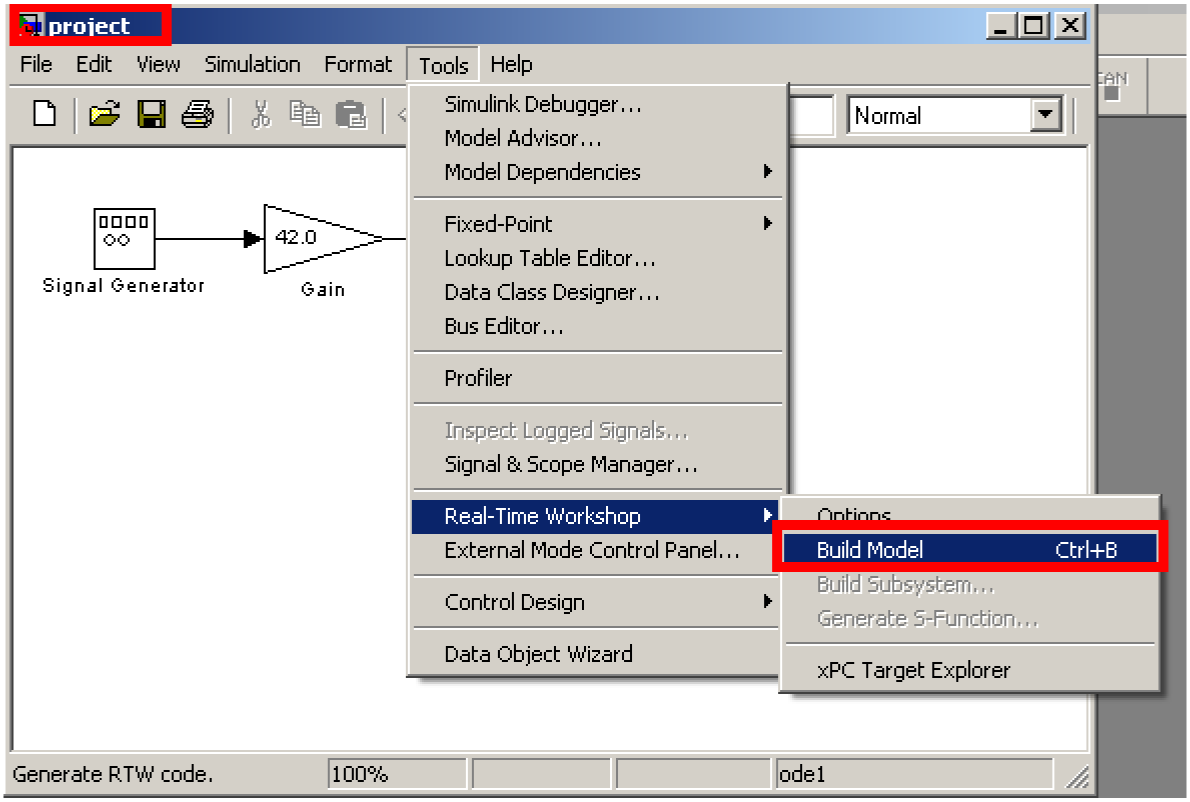


Figure 14: Selecting Build Model.

Note: To see the Build process running, view the MATLAB "Command Window".

6. Delete the Simulink blocks from the model but do not delete the grey “RTI Data” icon.

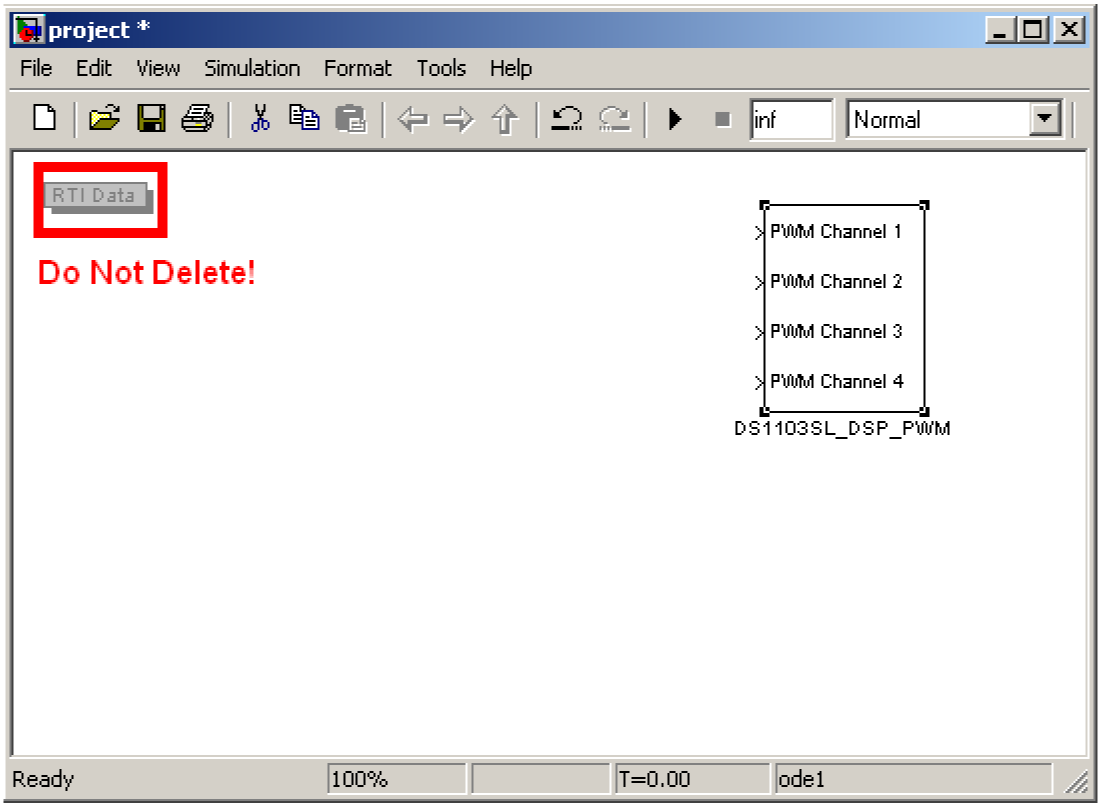


Figure 15: New Simulink model showing RTI Data icon and DS1103 PWM block added.

7. Start adding Simulink blocks to the model for the project. Save the model before closing it.

Opening MATLAB/Simulink

There are Several Ways to open MATLAB and Simulink when a Simulink model has already been created. Two possible options are listed below.

Option A:

Open the existing ".mdl" file for the project from its folder on the drive.
Note: This only works if you have an existing model.

1. Find the .mdl file on the drive and double-click the file to open it. (This will cause both MATLAB and Simulink to open.)

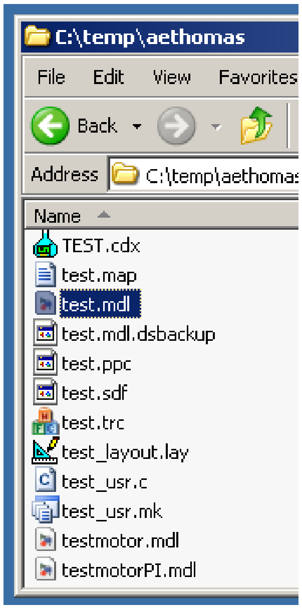
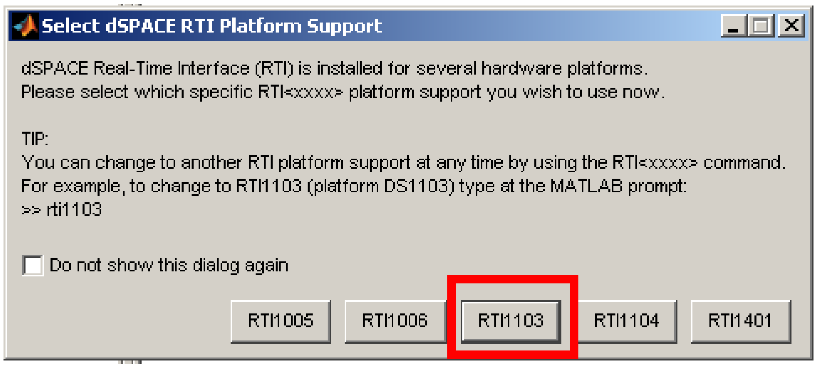


Figure 16: test.mdl file selected to be opened.

2. Select the "RTI1103" button in the "Select dSPACE RTI Platform Support" dialog box that appears in MATLAB.



dSPACE Real-Time Interface (RTI) is installed for several hardware platforms.
Please select which specific RTI<xxxx> platform support you wish to use now.

TIP:
You can change to another RTI platform support at any time by using the RTI<xxxx> command.
For example, to change to RTI1103 (platform DS1103) type at the MATLAB prompt:
>> rti1103

□ Do not show this dialog again

Figure 17: Select dSPACE RTI Platform Support dialog box.

Option B:
Open MATLAB from ControlDesk. This process does require more steps.

1. From ControlDesk, right-click "Simulink" in the "Platform" tab, and select "Open MATLAB". (This will open just a MATLAB Command Window.)

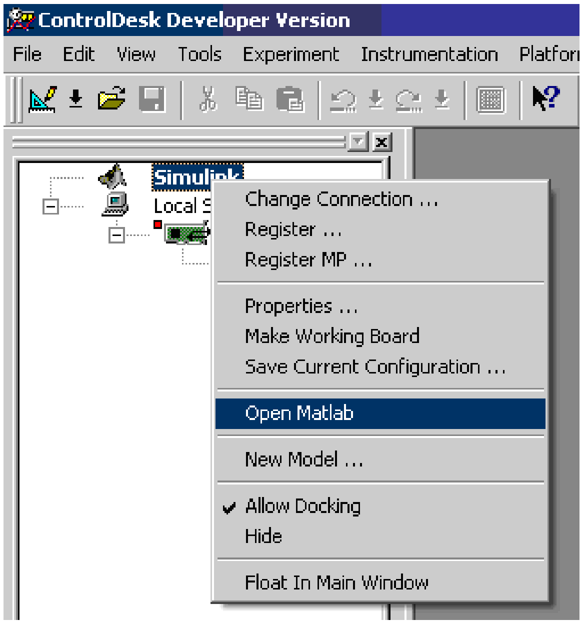
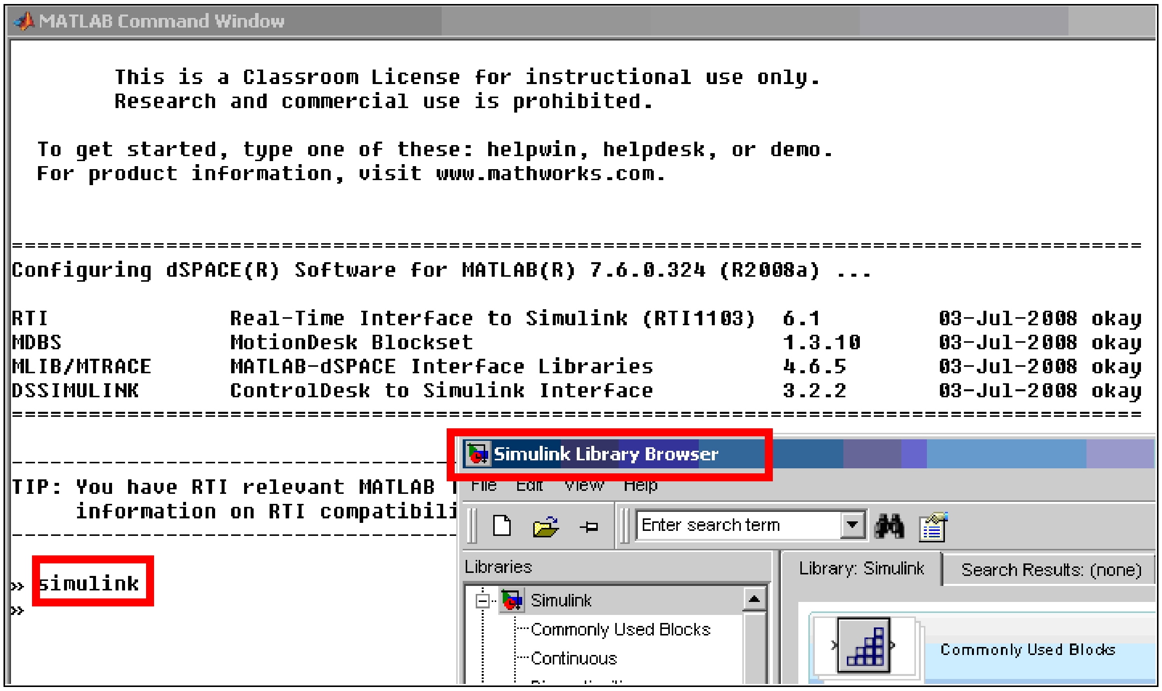


Figure 18: Link to Open MATLAB from ControlDesk.

2. To open the Simulink Library Browser, enter "simulink" into the MATLAB Command Window.



```
        This is a Classroom License for instructional use only.
        Research and commercial use is prohibited.

  To get started, type one of these: helpwin, helpdesk, or demo.
  For product information, visit www.mathworks.com.


=========================================================================================
Configuring dSPACE(R) Software for MATLAB(R) 7.6.0.324 (R2008a) ...

RTI              Real-Time Interface to Simulink (RTI1103)  6.1         03-Jul-2008 okay
MDBS             MotionDesk Blockset                        1.3.10      03-Jul-2008 okay
MLIB/MTRACE      MATLAB-dSPACE Interface Libraries          4.6.5       03-Jul-2008 okay
DSSIMULINK       ControlDesk to Simulink Interface          3.2.2       03-Jul-2008 okay
=========================================================================================

-------------------------------------
TIP: You have RTI relevant MATLAB I
     information on RTI compatibili
-------------------------------------

>> simulink
>>
```

Figure 19: Enter "simulink" into the MATLAB Command Window.

3. Next, Open the existing ".mdl" file using the "Simulink Library Browser" main menu.
   To Open an existing file go to: File ► Open.
   Then find and select your file on the drive and select the "Open" button in the "Open" dialog box.

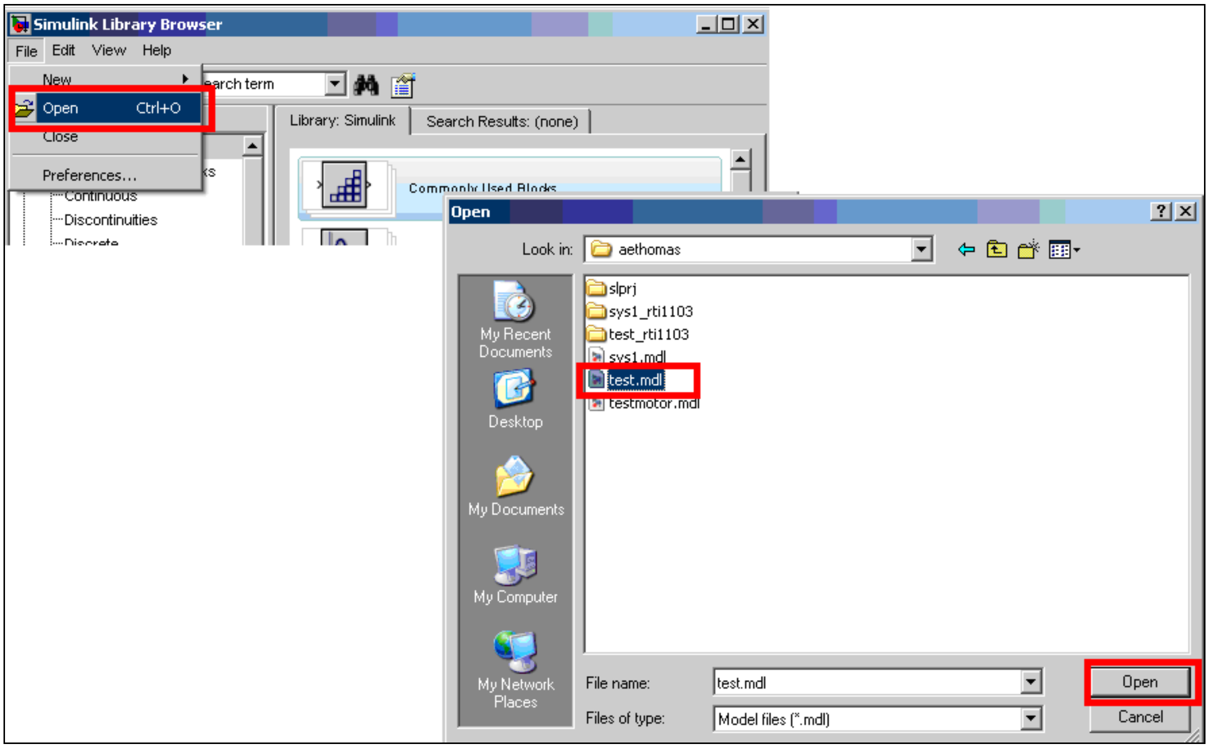


Figure 20: Opening an existing file from the Simulink Library Browser.

Building/Downloading a Model

1. Open the “Configuration Parameters” window from the main menu of your open model window.
   To open the “Configuration Parameters” window go to:
   Simulation ► Configuration Parameters…

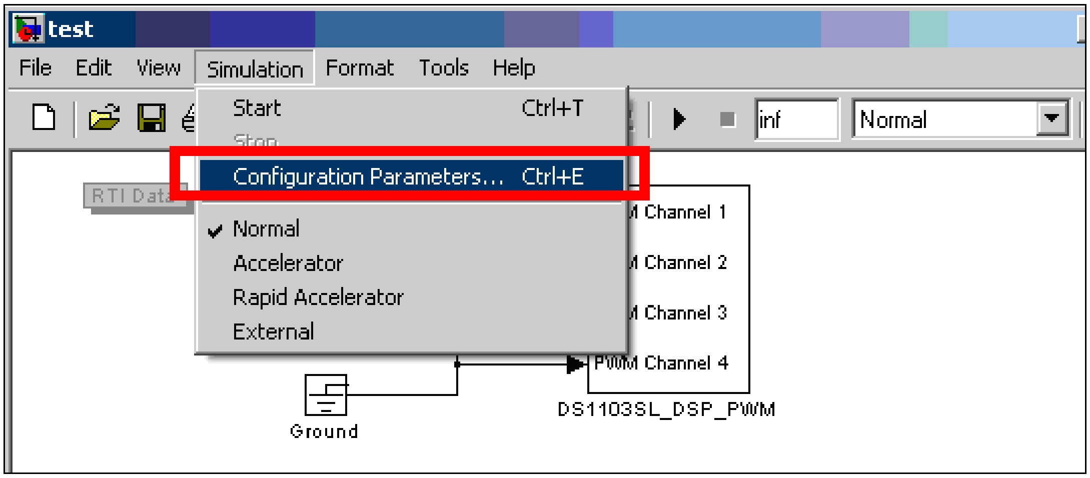
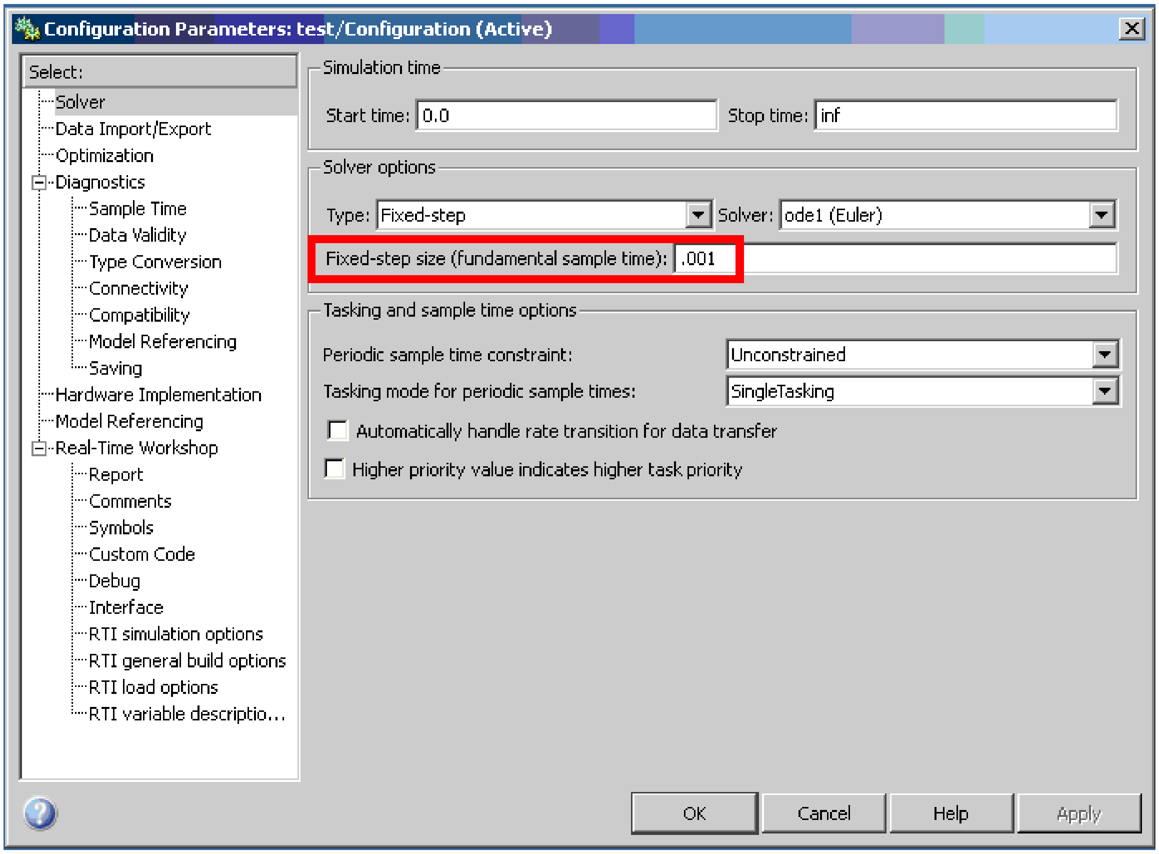


Figure 21: Opening the Configuration Parameters window.

2. Set the required/desired parameters in the “Configuration Parameters” window. Make sure to set the sampling time (“Fixed-step size”) to the correct value for the project.
   To change the sampling time go to: Solver ► Fixed-step size.
   Then, Set to desired sampling time.



Configuration Parameters: test/Configuration (Active)

Simulation time

Start time: 0.0

Stop time: inf

Solver options

Type: Fixed-step

Solver: ode1 (Euler)

Fixed-step size (fundamental sample time): .001

Tasking and sample time options

Periodic sample time constraint: Unconstrained

Tasking mode for periodic sample times: SingleTasking

□ Automatically handle rate transition for data transfer

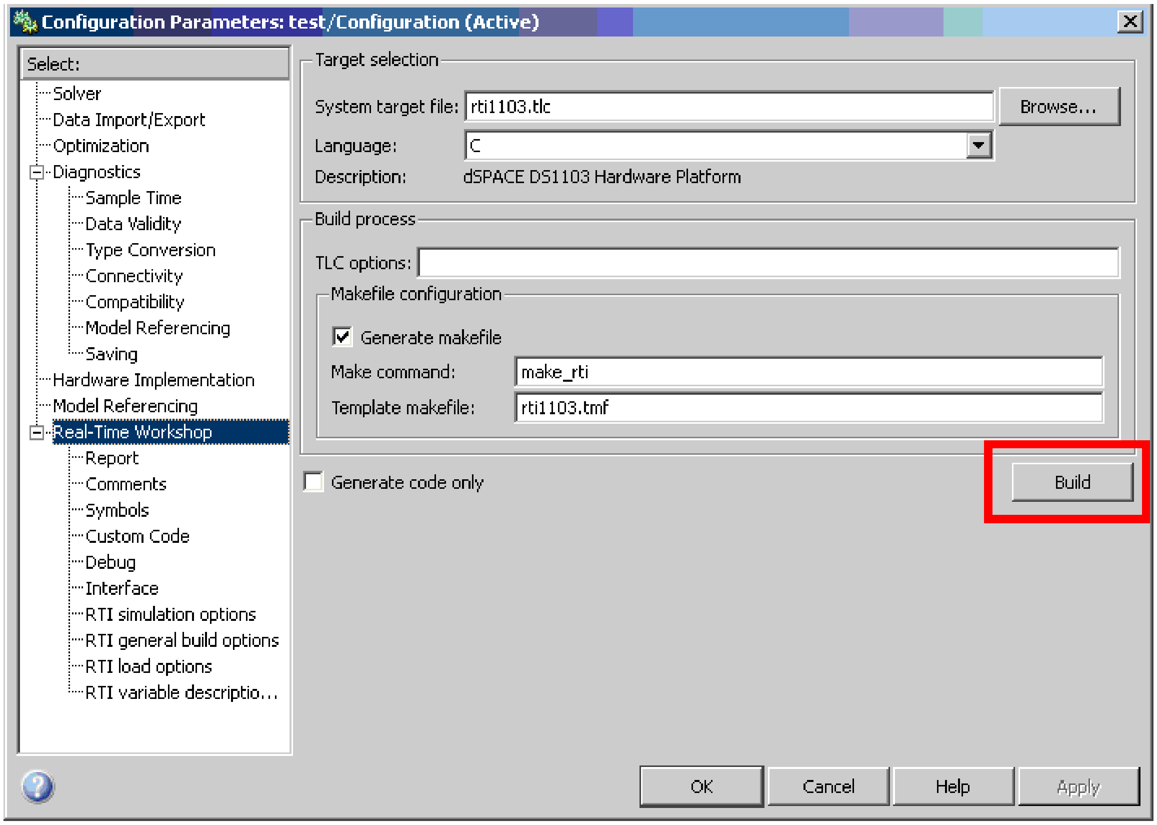
□ Higher priority value indicates higher task priority

Figure 22: Configuration Parameters window with sampling time set to 0.001s or 1ms.

3. Set the “Real-Time Workshop” settings and select the “Build” button to build and download code from the Simulink model. (Code building progress is output to the MATLAB Command Window.)

   Real-Time Workshop Settings:
   a. System target file: rti1103.tlc
   b. Language: C
   c. Make command: make_rti
   d. Template makefile: rti1103.tmf
   e. Select checkbox for “Generate makefile”

Note1: ControlDesk needs to be open for this step to complete all processes well. There will be no error message if ControlDesk is not open, but certain tabs in ControlDesk used for monitoring and controlling variables will not be automatically opened.



Configuration Parameters: test/Configuration (Active)

Target selection

System target file: rti1103.tlc

Language: C

Description: dSPACE DS1103 Hardware Platform

Build process

TLC options:

Makefile configuration

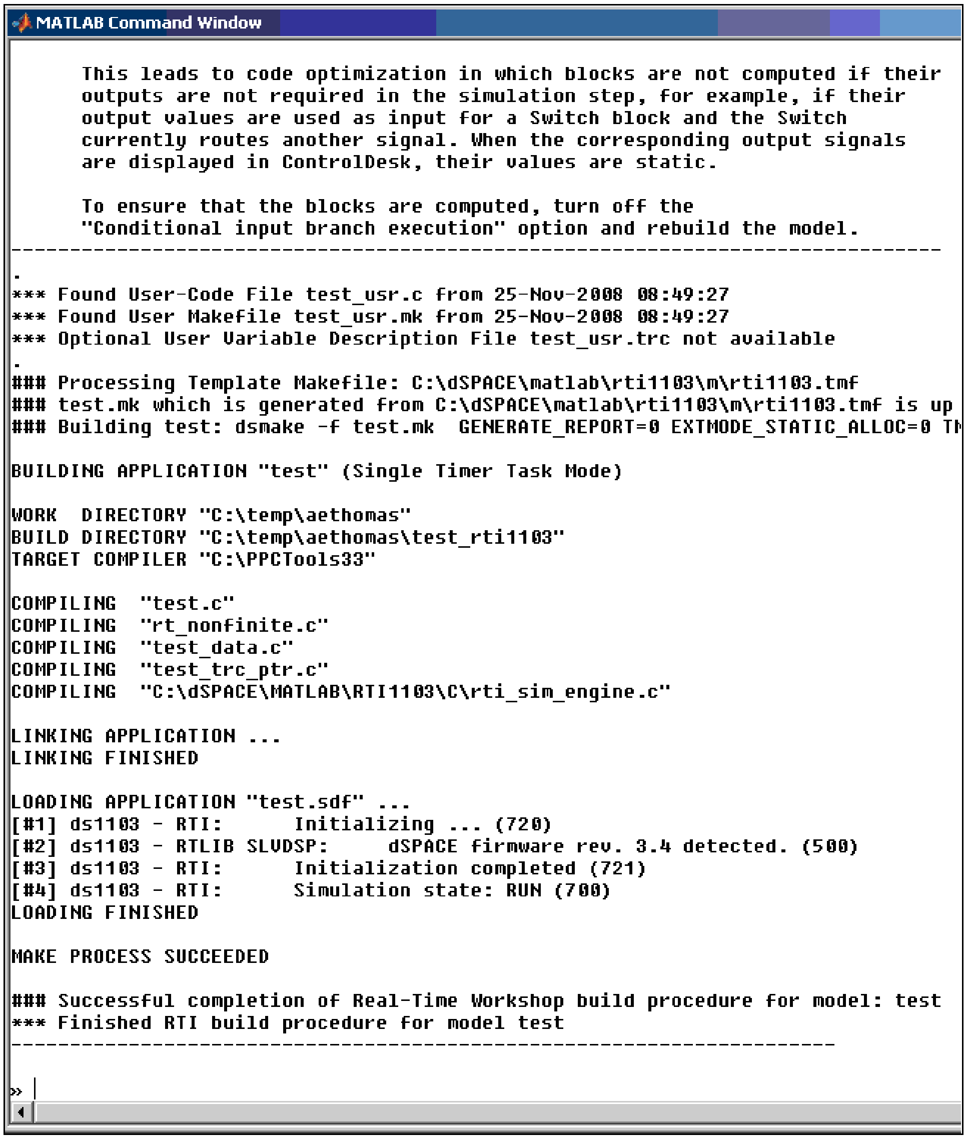
☑ Generate makefile

Make command: make_rti

Template makefile: rti1103.tmf

☐ Generate code only

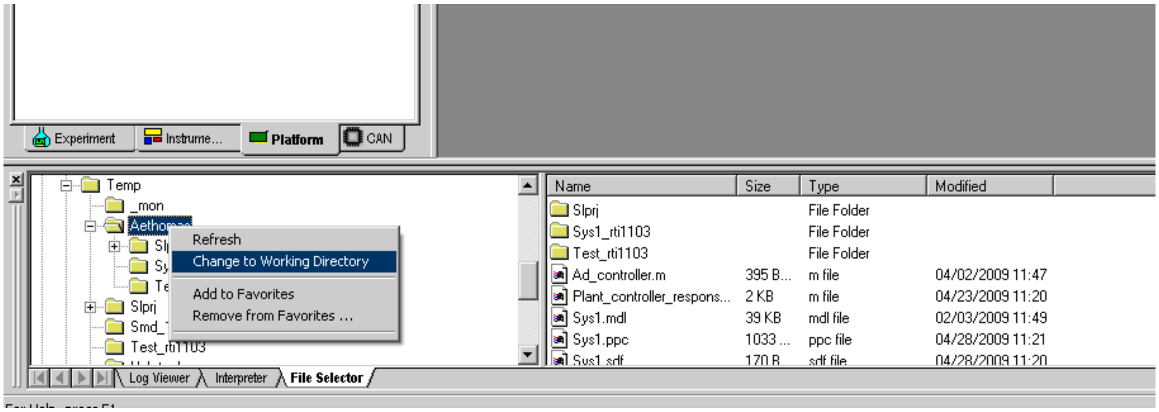
Figure 23: Real-Time Workshop Settings.



```
      This leads to code optimization in which blocks are not computed if their
      outputs are not required in the simulation step, for example, if their
      output values are used as input for a Switch block and the Switch
      currently routes another signal. When the corresponding output signals
      are displayed in ControlDesk, their values are static.

      To ensure that the blocks are computed, turn off the
      "Conditional input branch execution" option and rebuild the model.
--------------------------------------------------------------------------------
.
*** Found User-Code File test_usr.c from 25-Nov-2008 08:49:27
*** Found User Makefile test_usr.mk from 25-Nov-2008 08:49:27
*** Optional User Variable Description File test_usr.trc not available
.
### Processing Template Makefile: C:\dSPACE\matlab\rti1103\m\rti1103.tmf
### test.mk which is generated from C:\dSPACE\matlab\rti1103\m\rti1103.tmf is up
### Building test: dsmake -f test.mk  GENERATE_REPORT=0 EXTMODE_STATIC_ALLOC=0 TI

BUILDING APPLICATION "test" (Single Timer Task Mode)

WORK  DIRECTORY "C:\temp\aethomas"
BUILD DIRECTORY "C:\temp\aethomas\test_rti1103"
TARGET COMPILER "C:\PPCTools33"

COMPILING  "test.c"
COMPILING  "rt_nonfinite.c"
COMPILING  "test_data.c"
COMPILING  "test_trc_ptr.c"
COMPILING  "C:\dSPACE\MATLAB\RTI1103\C\rti_sim_engine.c"

LINKING APPLICATION ...
LINKING FINISHED

LOADING APPLICATION "test.sdf" ...
[#1] ds1103 - RTI:      Initializing ... (720)
[#2] ds1103 - RTLIB SLVDSP:     dSPACE firmware rev. 3.4 detected. (500)
[#3] ds1103 - RTI:      Initialization completed (721)
[#4] ds1103 - RTI:      Simulation state: RUN (700)
LOADING FINISHED

MAKE PROCESS SUCCEEDED

### Successful completion of Real-Time Workshop build procedure for model: test
*** Finished RTI build procedure for model test
-----------------------------------------------------------------------

>> |
```

Figure 24: End of output in MATLAB Command Window for a successful build.

Note2: To make sure the files generated are placed in the folder desired, it is advisable to select the “File Selector” tab at the bottom of the ControlDesk window, select and right-click the folder where the files should be placed, and select “Change to Working Directory” before building any code each time.



Name	Size	Type	Modified
Slprj		File Folder	
Sys1_rti1103		File Folder	
Test_rti1103		File Folder	
Ad_controller.m	395 B...	m file	04/02/2009 11:47
Plant_controller_respons...	2 KB	m file	04/23/2009 11:20
Sys1.mdl	39 KB	mdl file	02/03/2009 11:49
Sys1.ppc	1033 ...	ppc file	04/28/2009 11:21
Sys1.sdf	170 B	sdf file	04/28/2009 11:20

Figure 25: Setting the Working Directory/Build folder.

4. The code should be built and downloaded and running on the DS1103 and Slave DSP now.

Monitoring/Controlling/Recording Values

Layouts are one of the more interesting functions in ControlDesk. They allow the value of constant/input blocks, gain blocks, and perhaps some other blocks to from the Simulink model downloaded to the DS1103 to be changed while the system is running. They also allow real-time reporting of output values and saving/recording of values over a time period for analysis/viewing in MATLAB later.

Normal Process

1. Create a new layout or Open an existing layout using the ControlDesk main menu.

   A. To Open an existing file go to: File ► Open. Then find and select your file on the drive and select the "Open" button in the "Open" dialog box.

   B. To Create a new file go to: File ► New ► Layout.
   Note: Make sure to save and rename your file before closing it later.

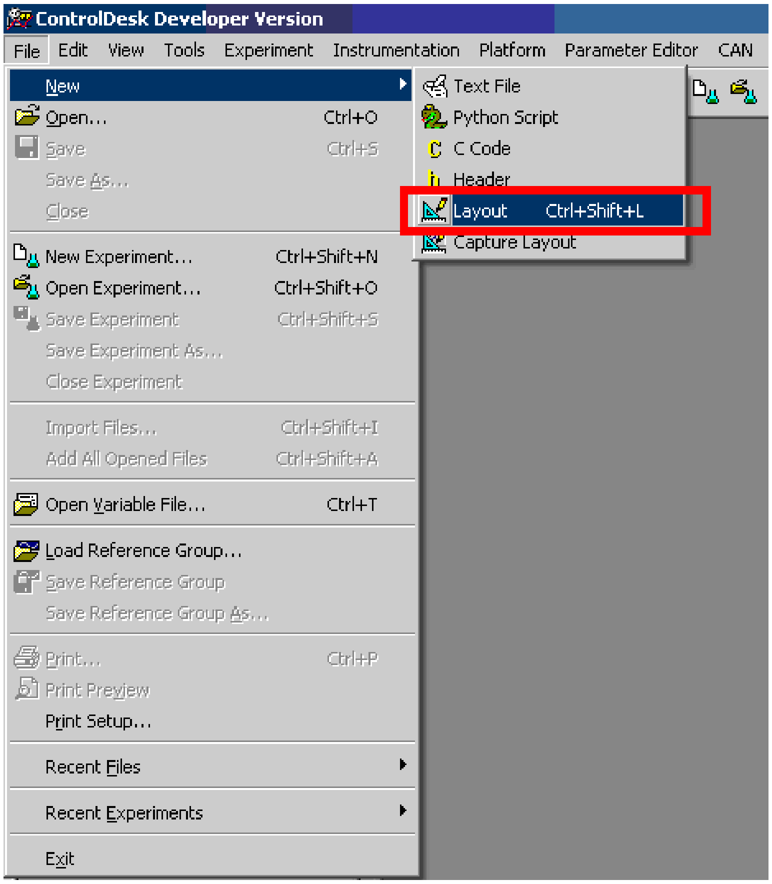


Figure 26: Create a new layout.

2. Expand the layout window that appears in the ControlDesk workspace to fit most of the workspace by dragging the double-headed arrows that appear when the mouse passes over the edges of the window.

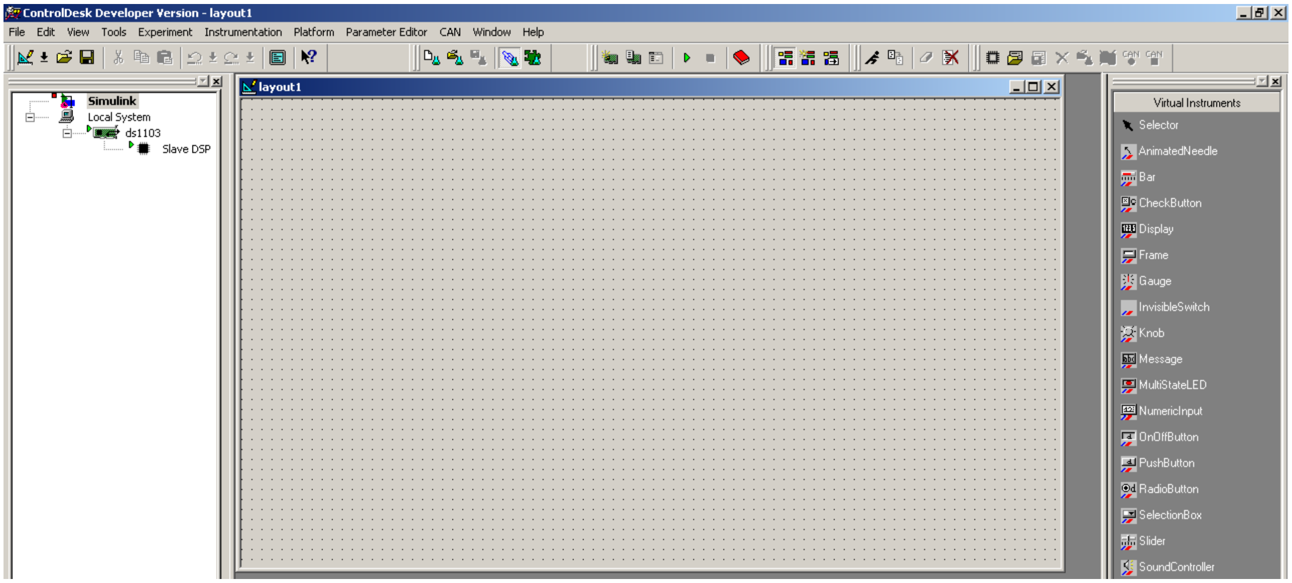


Figure 27: Expanded layout window in ControlDesk.

3. Set ControlDesk to "Edit Mode" by either selecting it from the toolbar or by selecting it from the "Instrumentation" dropdown menu of the main menu. (The "Virtual Instruments" side panel should appear.)

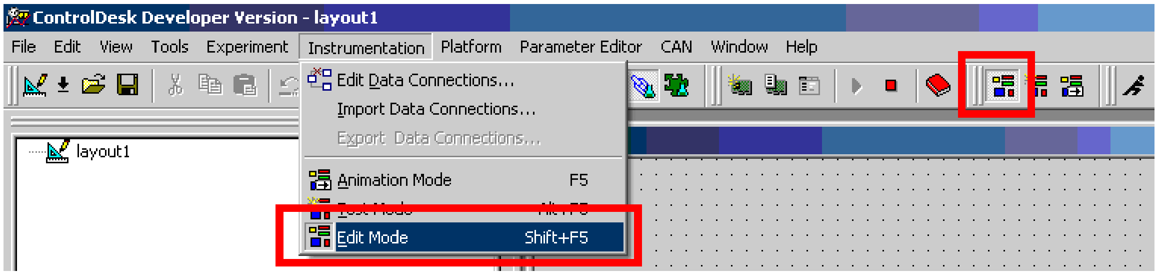

ControlDesk Developer Version - layout1
File Edit View Tools Experiment Instrumentation Platform Parameter Editor CAN Window Help
layout1
Edit Data Connections...
Import Data Connections...
Export Data Connections...
Animation Mode F5
Test Mode Alt+F5
Edit Mode Shift+F5

Figure 28: Selecting Edit Mode in ControlDesk.

4. Now “Virtual Instruments” and “Data Acquisition” Tools can be added to the layout. This is done by single-clicking instruments in the side panel (without holding the mouse button down), then clicking and then holding the mouse button down in the layout window while dragging the corner of the box that begins to appear for that particular instrument.

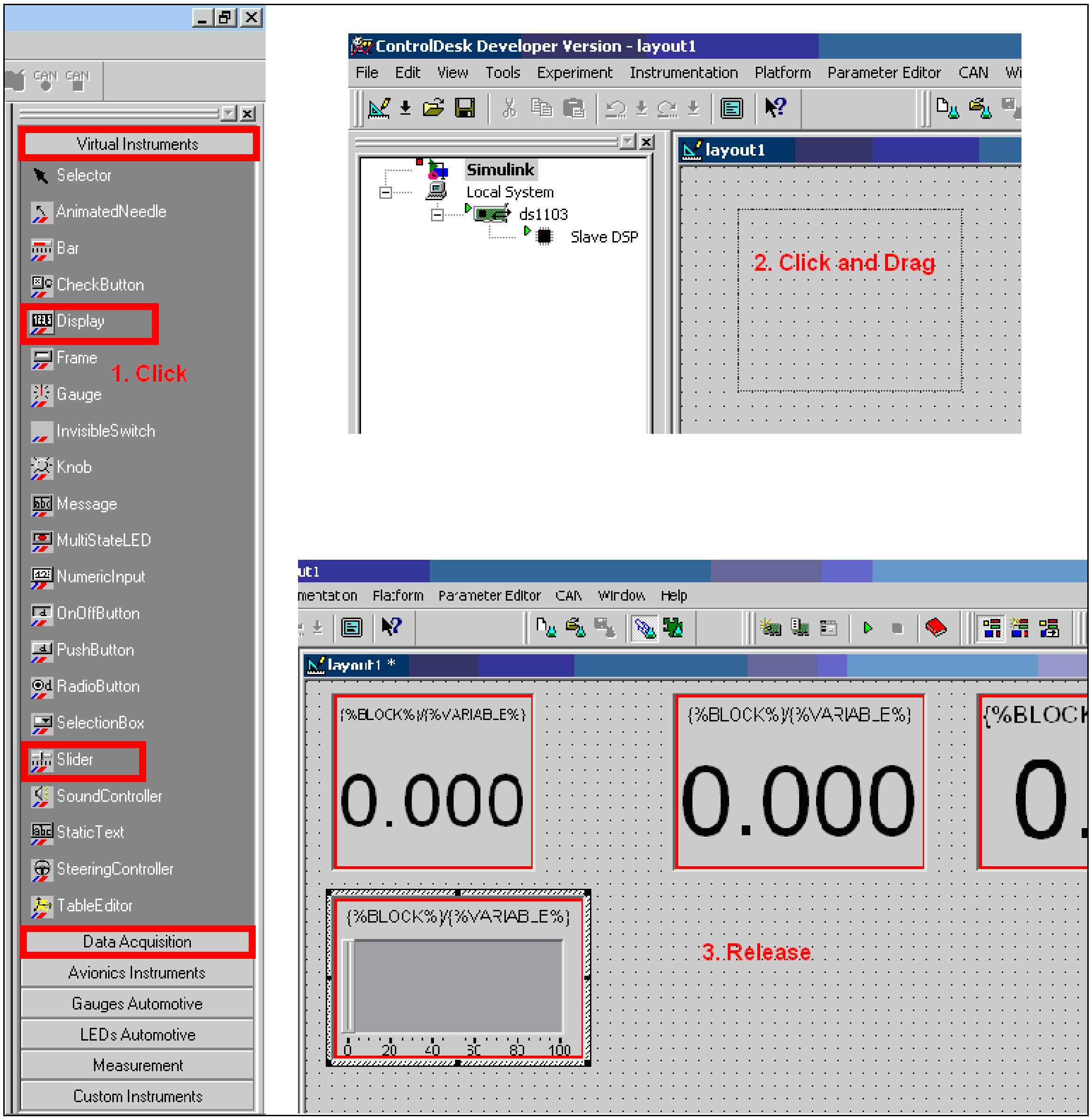


Figure 29: Adding a Virtual Instrument.

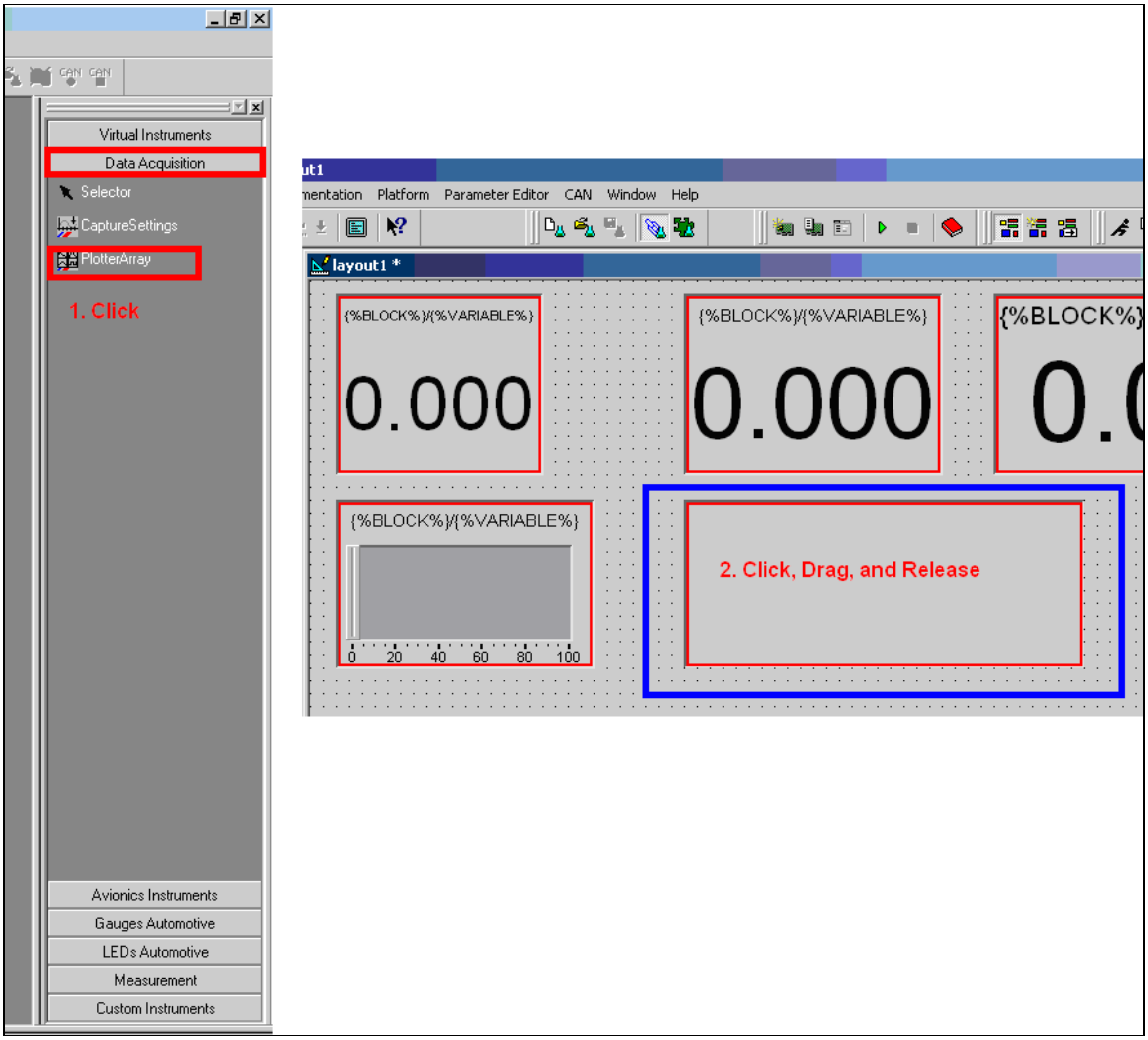
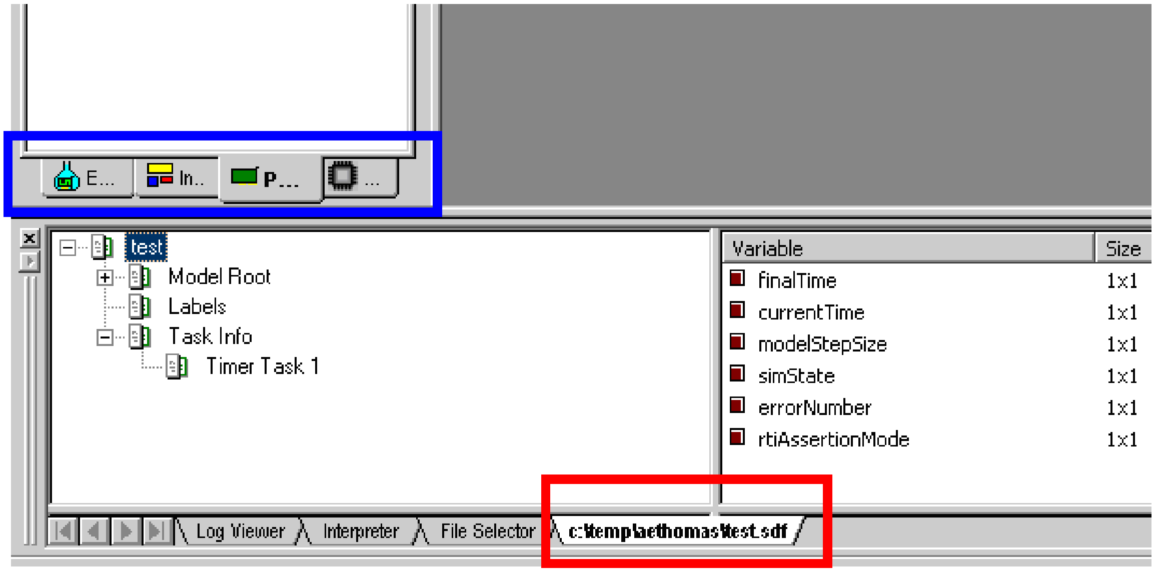


Figure 30: Adding Plotter Array from the Data Acquisition Menu.

5. Verify a tab for the ".sdf" file generated for the project is available near the bottom of the ControlDesk window. If it is not, rebuild the project using the instructions in the "Building/Downloading a Model" section of this tutorial, beginning on page 13.



Variable	Size
finalTime	1x1
currentTime	1x1
modelStepSize	1x1
simState	1x1
errorNumber	1x1
rtiAssertionMode	1x1

Figure 31: Location of .sdf tab in the ControlDesk window.

6. To connect/assign variables from the Simulink model downloaded to the DS1103, locate the variable in the ".sdf" menu(s) and then click-and-drag it to its instrument in the layout. (Generally, the values of constants and gain blocks in the model can be monitored and modified/controlled in real-time and outputs/connections between blocks can be monitored only.)

Note: To connect to the outputs of dSPACE blocks in the downloaded Simulink model, a gain (of 1) block needs to be connected to the output of the block if nothing else is connected to it other than a terminator block.

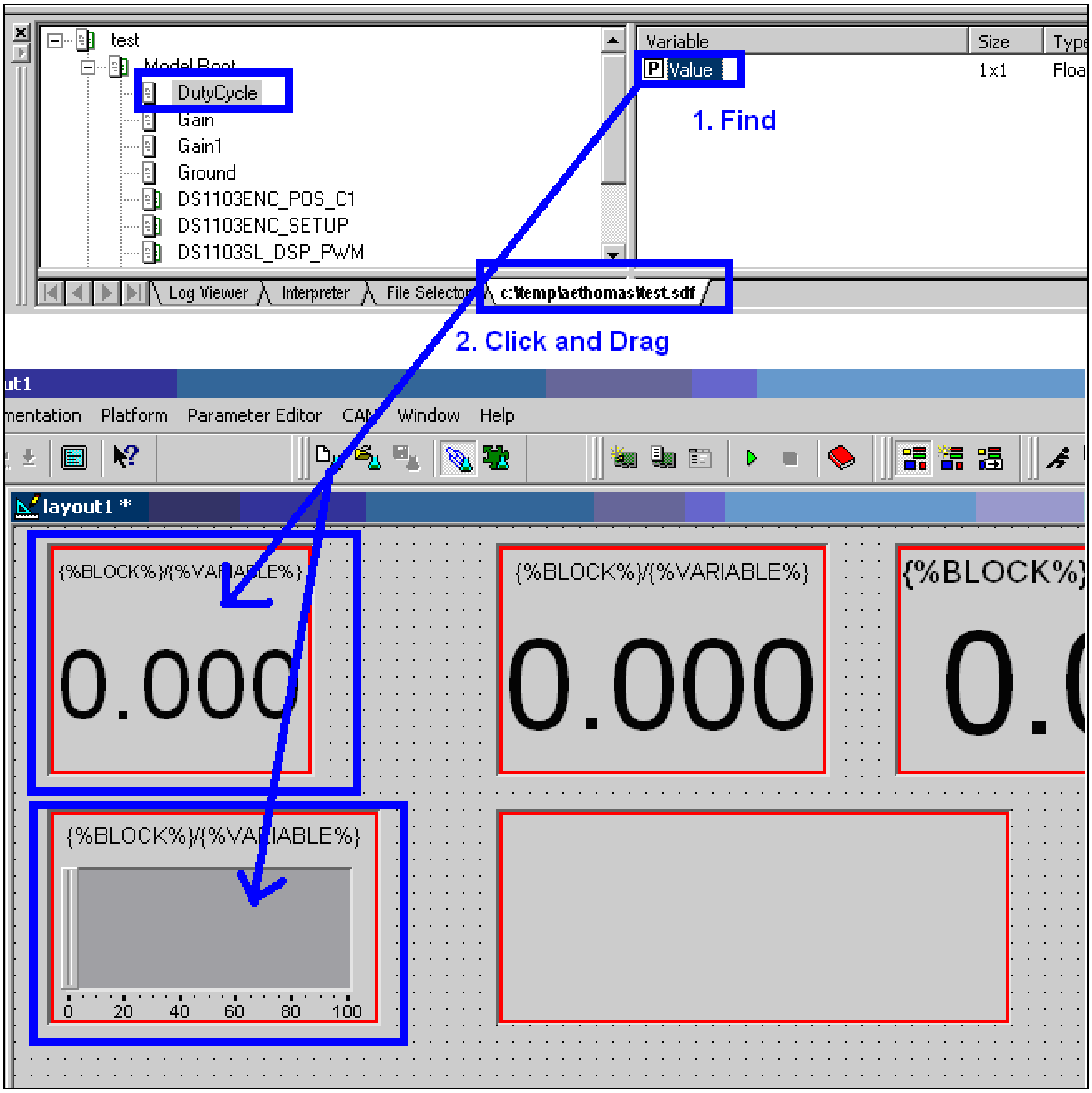
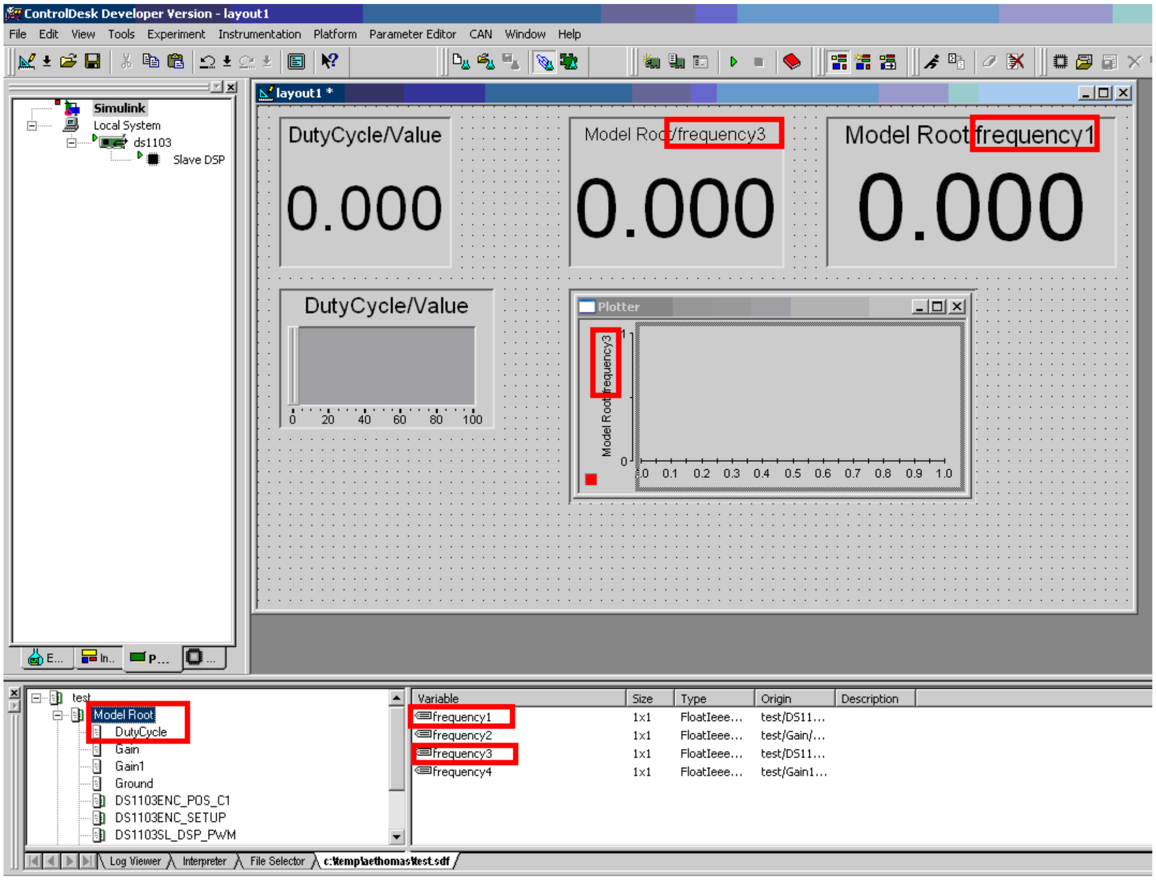


Figure 32: Connecting Variables to Instruments.



Variable	Size	Type	Origin	Description
frequency1	1x1	FloatIeee...	test/DS11...	
frequency2	1x1	FloatIeee...	test/Gain/...	
frequency3	1x1	FloatIeee...	test/DS11...	
frequency4	1x1	FloatIeee...	test/Gain1...	

Figure 33: Completed layout with location of additional connections not shown in Figure 28 highlighted.

7. Change the properties of the instruments as desired.
   Accessing the "Properties" menu(s) for a particular instrument done by either:
   A. Left-clicking the instrument to select it and then selecting "Properties" from the ControlDesk main menu "View" menu.
   B. Right-clicking the instrument and selecting "Properties".

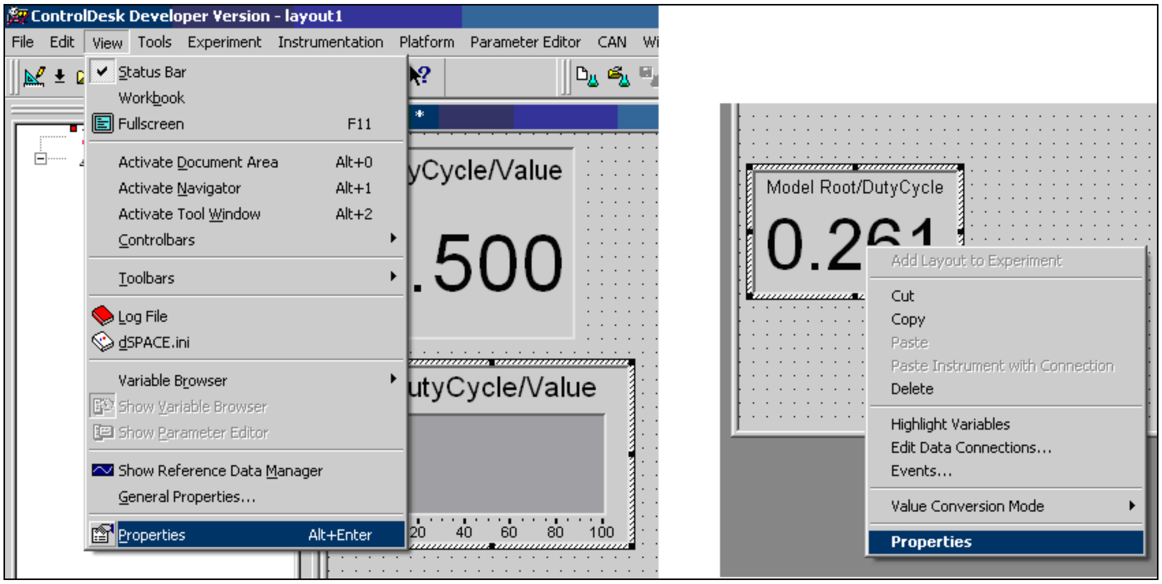
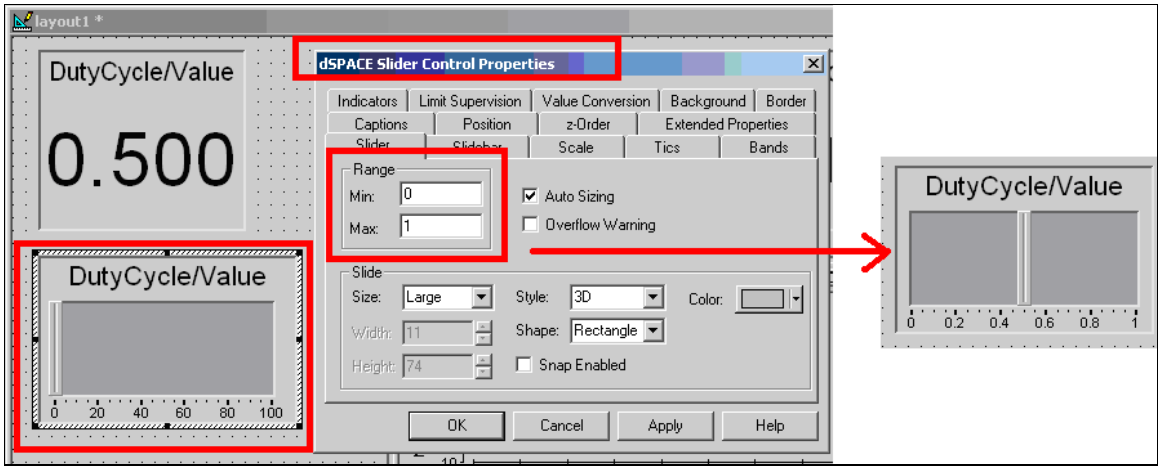


Figure 34: Two methods of opening the Properties dialog for an instrument.

Find the property that should change under one of the tabs and change it. (Generally, only Max and Min "Range" values need to be changed and/or Range "Check Mode" needs to be Enabled, but other properties such as the color of the instrument and text can be changed as well.)



Range
Min: 0
Max: 1
☑ Auto Sizing
☐ Overflow Warning
Slide
Size: Large
Width: 11
Height: 74
Style: 3D
Shape: Rectangle
☐ Snap Enabled
Color:

Figure 35: Changing the Range of the DutyCycle Slider input.

8. Verify the instruments now functions as desired without affecting the system (or the other instruments) using "Test Mode", which is found in a similar manner to "Edit Mode" in Step 3 of this section. (This is useful to verify that the "Range" enabled for "Check Mode" for an instrument such as the "NumericInput" instrument - not used in this section - rejects values input that are outside that range.)

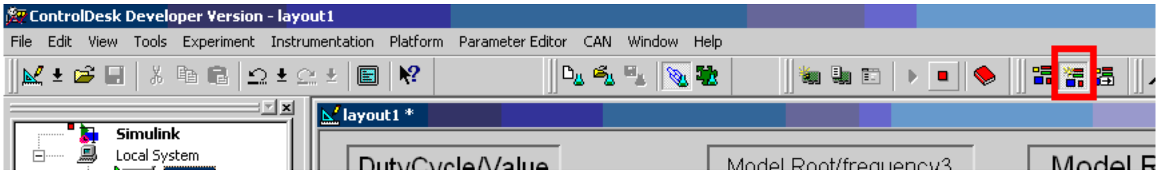

ControlDesk Developer Version - layout1
File Edit View Tools Experiment Instrumentation Platform Parameter Editor CAN Window Help
layout1 *
Simulink
Local System
DutyCycle/Value
Model Root/frequency3
Model R

Figure 36: Location of Test Mode button on main toolbar.

9. To use the instruments to control and monitor that actual, running, real-time system, select "Animation Mode" in a similar manner to selecting "Edit Mode" in Step 3 of this section. (It appears "Properties" can also be modified in this mode.)

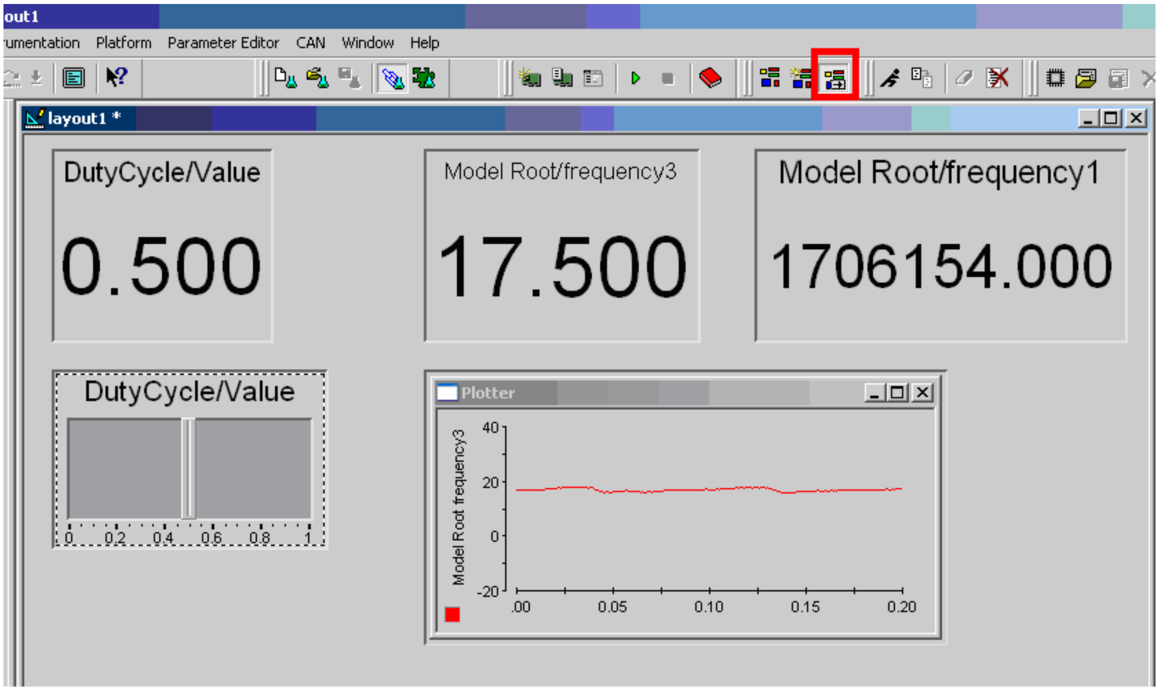


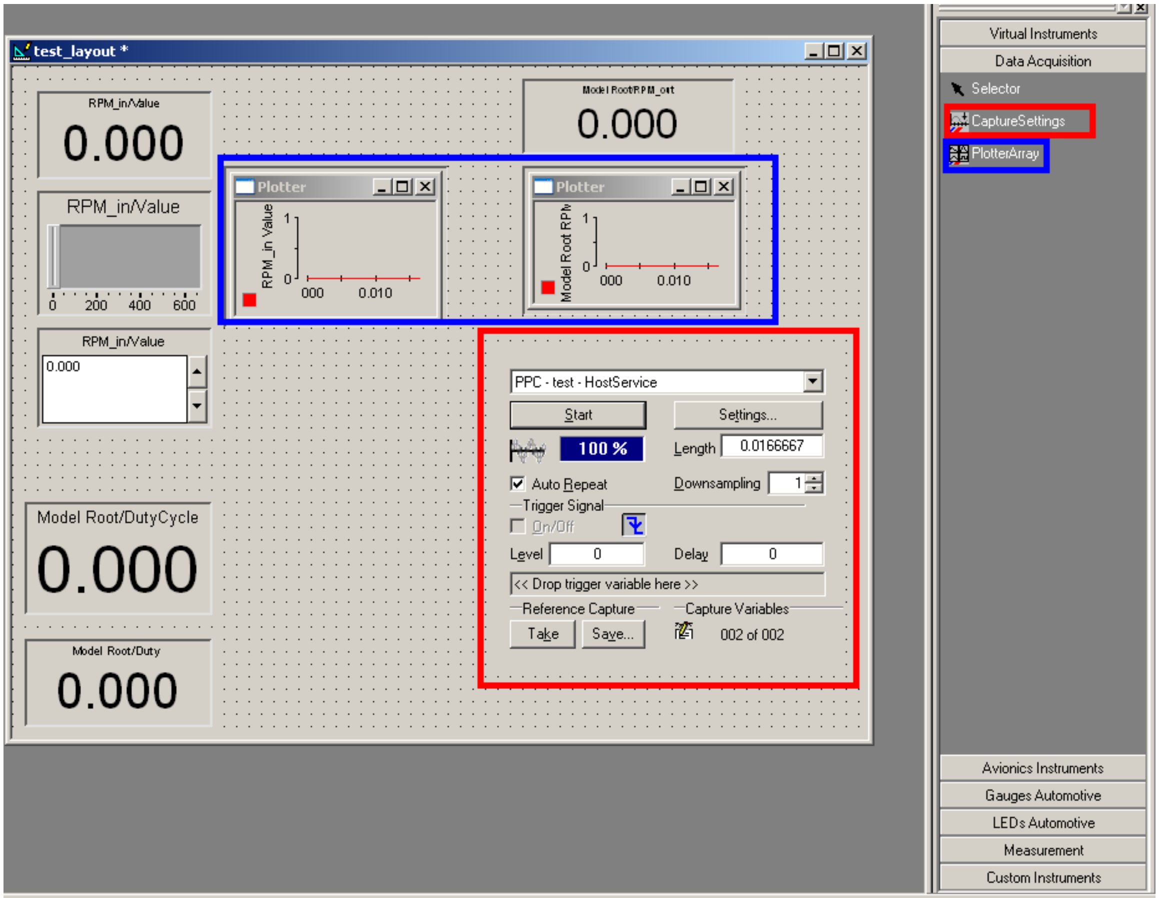
Figure 37: System running with Animation Mode active.

10. Additional instruments such as the "CaptureSettings" instrument, used for saving/recording data, can be added later by repeating Steps 3-9 of this section for the new instrument. It should also be possible to create and use multiple layouts at the same time.

Note: For instructions/details about the "CaptureSettings" instrument, see the "CaptureSettings Instrument" subsection that follows.

CaptureSettings Instrument

The “CaptureSettings” instrument found in the “Data Acquisition” instrument menu allows data output on “PlotterArray” instruments (and only “PlotterArray” instruments) to be saved/recorded for later analysis/use in MATLAB or perhaps some other program.



Length 0.0166667

☑ Auto Repeat

Downsampling 1

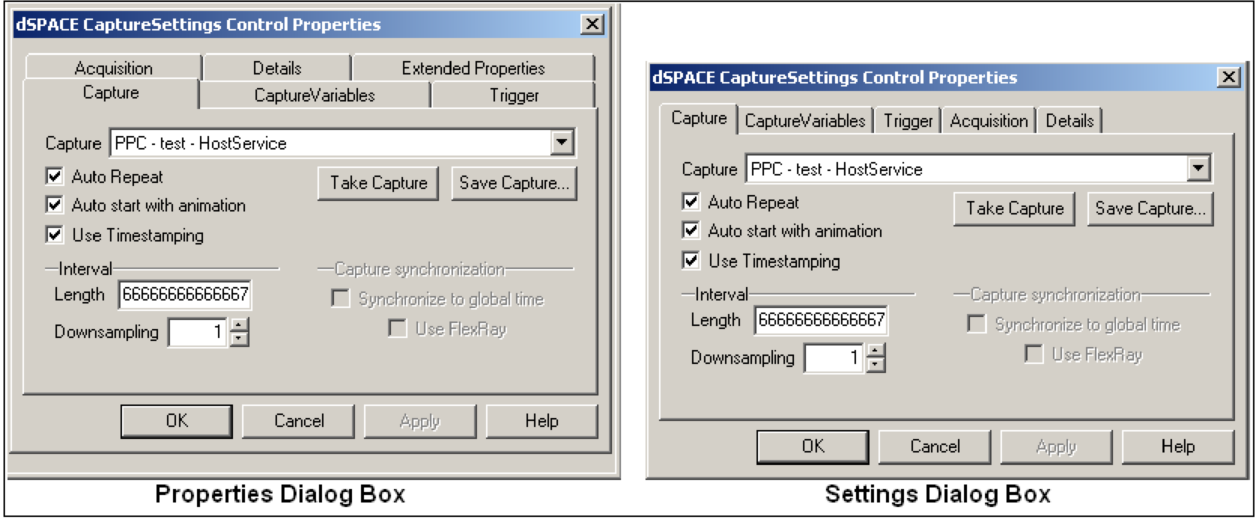
□ On/Off

Level 0

Delay 0

Figure 38: CaptureSettings instrument (red boxes) added to a layout with associated PlotterArrays (blue boxes). A NumericInput instrument is also included in this layout below the slider.

The “CaptureSettings” instrument is added in the same way as any other instrument (See Steps 3-9 in the “Normal Process” subsection previous to this one.), but the space provided for it must be large enough to access all its functions and most of the “Properties” can also be accessed using the ”Settings…” button on the instrument.



Capture: PPC - test - HostService
☑ Auto Repeat
☑ Auto start with animation
☑ Use Timestamping
Length: 6666666666667
Downsampling: 1
☐ Synchronize to global time
☐ Use FlexRay

Properties Dialog Box

Capture: PPC - test - HostService
☑ Auto Repeat
☑ Auto start with animation
☑ Use Timestamping
Length: 6666666666667
Downsampling: 1
☐ Synchronize to global time
☐ Use FlexRay

Settings Dialog Box

Figure 39: Most tabs in the Properties and Settings dialog boxes for the CaptureSettings instrument are exactly the same.

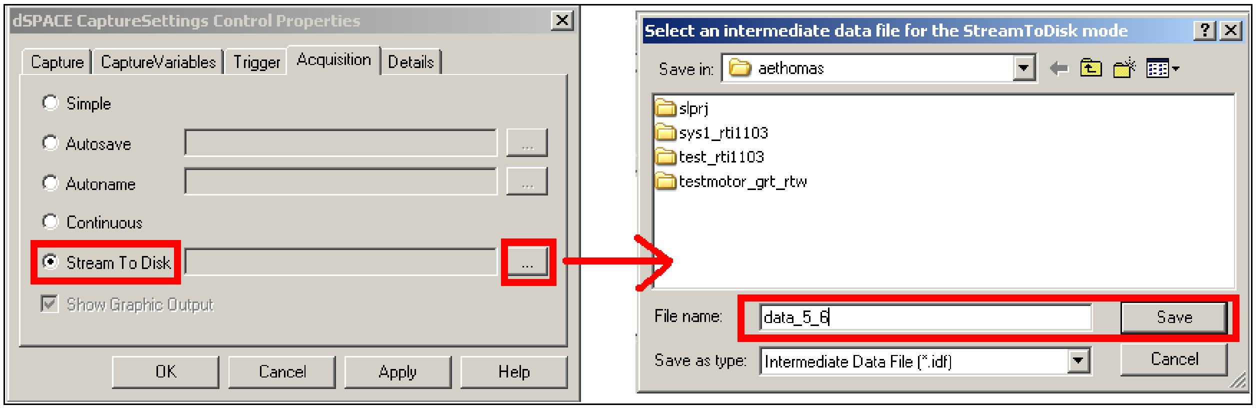
Note: For this tutorial only the “Stream to Disk” “Acquisition” method was used in its simplest form. To see more functions of this instrument, search for “How to Generate and Save Reference Data” or “Working with Data Captures” in the ControlDesk “Help” menu found on the ControlDesk main menu bar.

After the CaptureSettings instrument has been added and PlotterArrays have been added for the variables that should be recorded:

1. Set the top dropdown menu for the instrument is to “PPC - HostService”. (The name for the project will be included as well.)

2. Open the Settings or Properties dialog for the CaptureSettings Instrument and select the “Acquisition” tab. Select the bullet box for “Stream to Disk”.

3. Select the button with "…" at the end of the line for "Stream to Disk" and create or select a file for the data to be saved to. The file will be saved as an ".idf" file.



Save in: aethomas

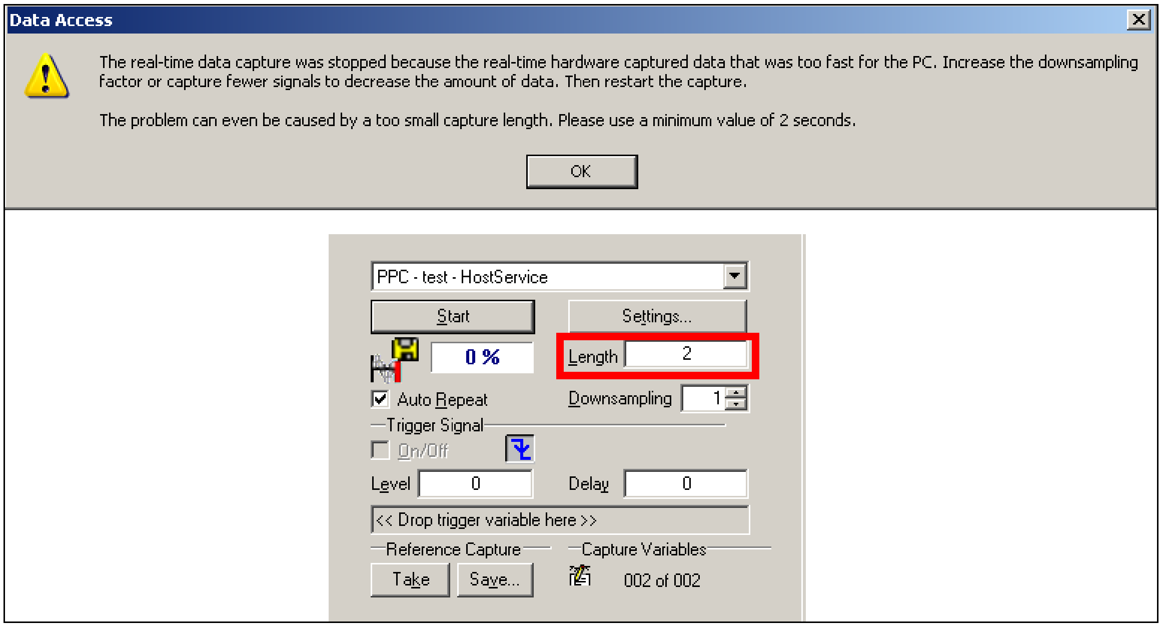
File name: data_5_6

Save as type: Intermediate Data File (*.idf)

Figure 40: Set Stream to Disk mode and create/select location for captured data.

4. Select "OK" at the bottom of the dialog box to exit the menu.

5. If the "Start" button on the instrument is selected now, ControlDesk may report an error with the "Length" or "Downsampling" Settings. Arbitrarily setting the "Length" to "2" will fix this for the example application.



The real-time data capture was stopped because the real-time hardware captured data that was too fast for the PC. Increase the downsampling factor or capture fewer signals to decrease the amount of data. Then restart the capture.

The problem can even be caused by a too small capture length. Please use a minimum value of 2 seconds.

PPC - test - HostService

0 %

Length 2

☑ Auto Repeat

Downsampling 1

☐ On/Off

Level 0

Delay 0

002 of 002

Figure 41: Setting the Length for the capture to 2 seconds.

6. Select “Start” on the instrument to start saving data, and then Select “Stop” when the desired data capture is completed.

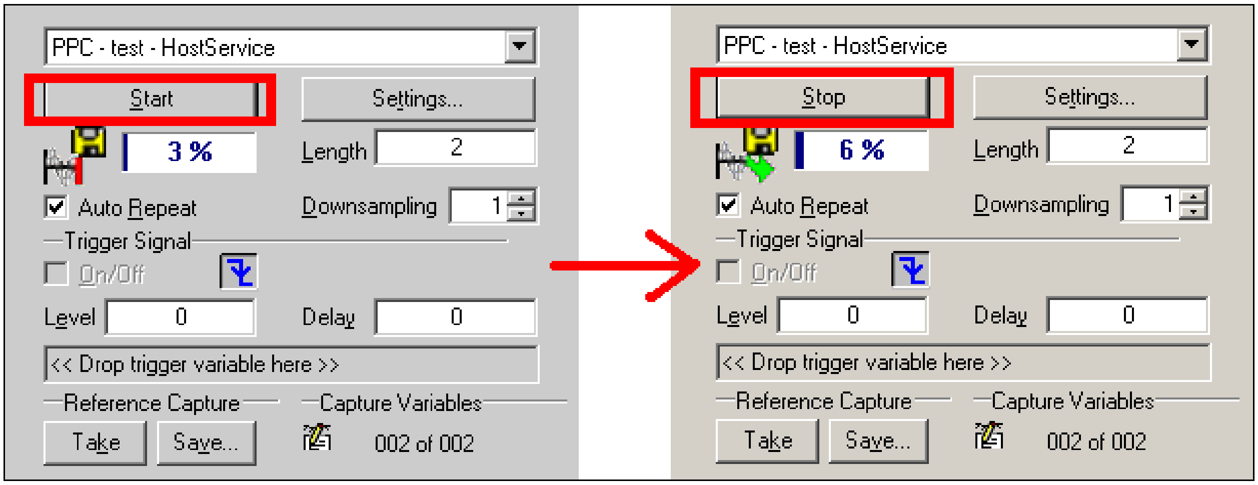
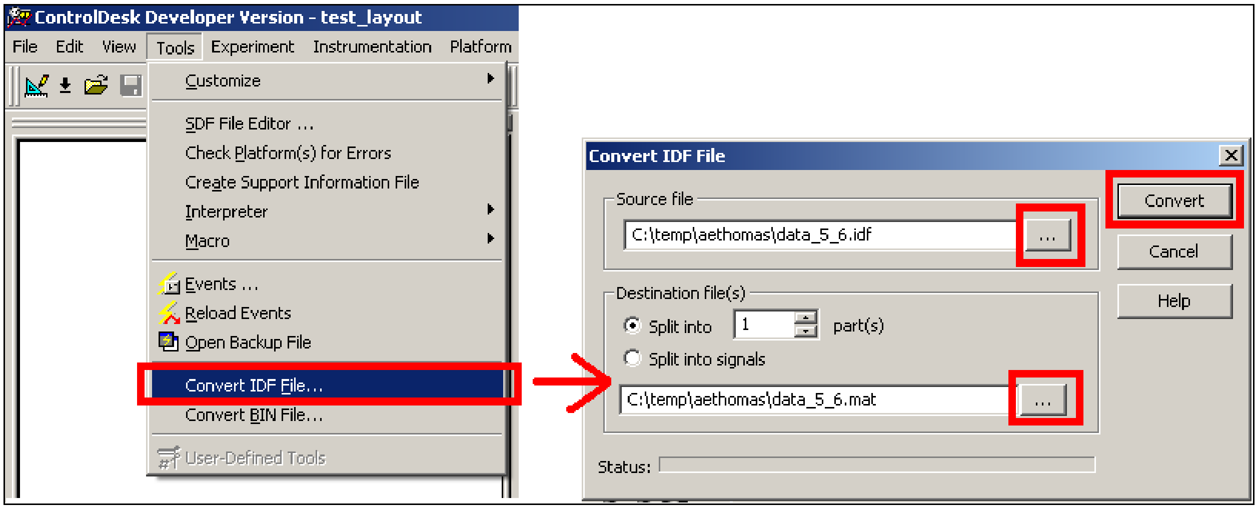


Figure 42: Saving data.

Note: This will overwrite any data stored previously in the “.idf” file created earlier. If the data should not be rewritten follow Steps 1-2 again between captures an Create/Select a “.idf” file with a different name.

7. Convert the “.idf” file to a MATLAB “.mat” file by Selecting “Convert IDF File…” from the “Tools” menu on the ControlDesk main menu, and then Selecting the “.idf” file containing the saved data and Selecting/Creating the “.mat” file the converted data should be saved to using the “Source File” and “Destination File(s)” “…” buttons and Selecting the “Convert” button.


ControlDesk Developer Version - test_layout

Convert IDF File

Source file: C:\temp\aethomas\data_5_6.idf

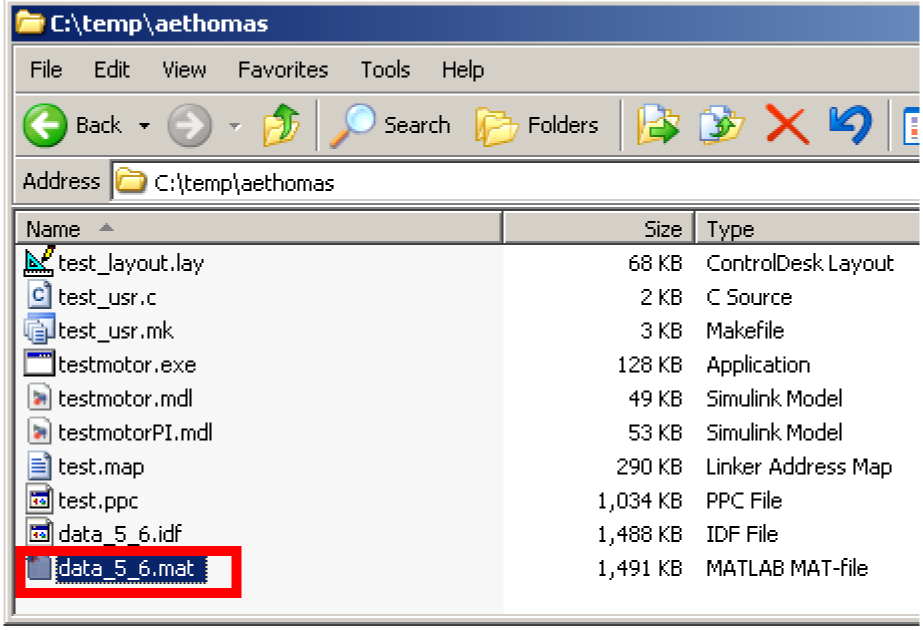
Destination file(s): ● Split into 1 part(s)

C:\temp\aethomas\data_5_6.mat

Status:

Figure 43: Converting an .idf file to a .mat file.

8. Open the ".mat" file from the location it was saved to on the drive.

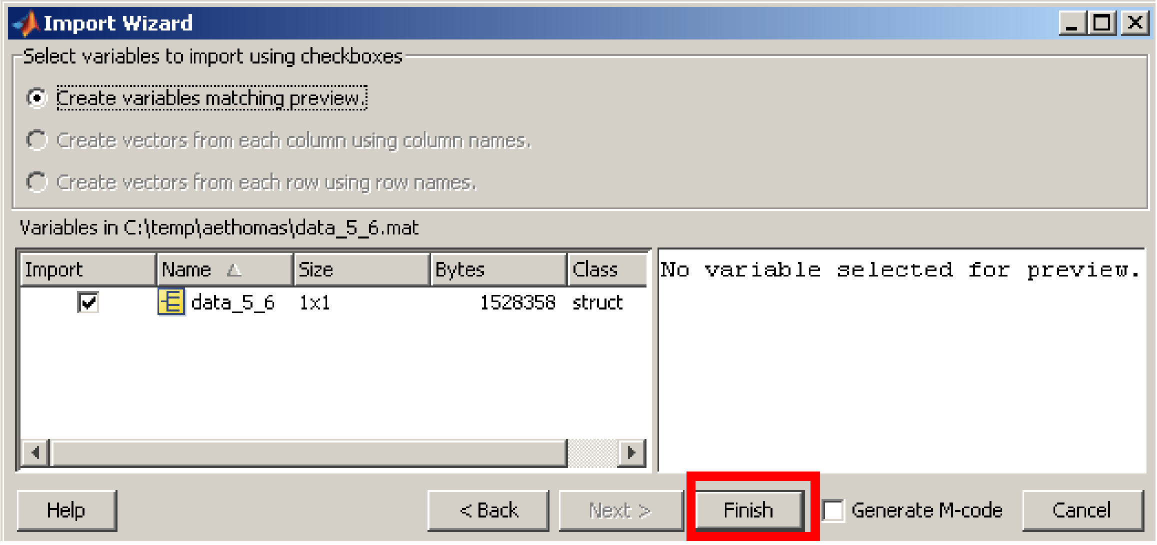


Address C:\temp\aethomas

Name	Size	Type
test_layout.lay	68 KB	ControlDesk Layout
test_usr.c	2 KB	C Source
test_usr.mk	3 KB	Makefile
testmotor.exe	128 KB	Application
testmotor.mdl	49 KB	Simulink Model
testmotorPI.mdl	53 KB	Simulink Model
test.map	290 KB	Linker Address Map
test.ppc	1,034 KB	PPC File
data_5_6.idf	1,488 KB	IDF File
data_5_6.mat	1,491 KB	MATLAB MAT-file

Figure 44: Open the .mat file.

9. Select the "Finish' button from the MATLAB "Import Wizard" dialog box that opens.



Select variables to import using checkboxes

● Create variables matching preview.

○ Create vectors from each column using column names.

○ Create vectors from each row using row names.

Variables in C:\temp\aethomas\data_5_6.mat

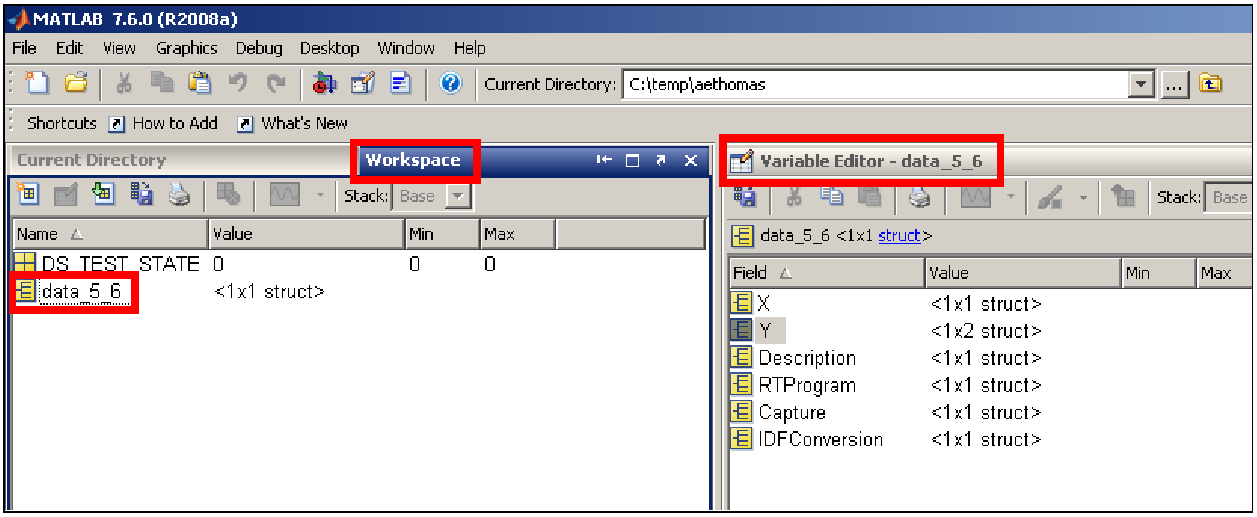
Import	Name	Size	Bytes	Class
☑	data_5_6	1x1	1528358	struct

```
No variable selected for preview.
```

□ Generate M-code

Figure 45: MATLAB Import Wizard dialog box.

10. From the MATLAB “Workspace”, Double-click (or Right-click and Select “Open Selection”) the structure with the same name as the “.mat” file to open the “Variable Editor” for the file.


MATLAB 7.6.0 (R2008a)
File Edit View Graphics Debug Desktop Window Help
Shortcuts How to Add What's New
Current Directory
Workspace
Stack: Base
Variable Editor - data_5_6
Stack: Base

Current Directory: C:\temp\aethomas

Name	Value	Min	Max
DS_TEST_STATE	0	0	0
data_5_6	<1x1 struct>		

data_5_6 <1x1 struct>

Field	Value	Min	Max
X	<1x1 struct>		
Y	<1x2 struct>		
Description	<1x1 struct>		
RTProgram	<1x1 struct>		
Capture	<1x1 struct>		
IDFConversion	<1x1 struct>		

11. Plot the data (or manipulate it in some other manner).

- The “X” structure contains the time values in its “Data” array.
- The values for the variables that were recorded are found in the “Data” arrays for each “<1x1 struct>” of the “Y” structure. To determine which variable is found in each “<1x1 struct>” of “Y”, Double-click the structure to open it and look at the “Value” of its “Name” label. (See figure 47 on next page.)
- The full identifying label for each variable can be found at the top of the “Variable Editor” window when it is open.
- Example MATLAB code to plot both variables from the figures in this section on the same plot:

```
>> plot(data_5_6.X.Data, data_5_6.Y(1,1).Data) %in or RPM_in
>> hold on
>> plot(data_5_6.X.Data, data_5_6.Y(1,2).Data) %out or RPM_out
>> hold off
```

(Plot shown in figure 46 on the next page.)

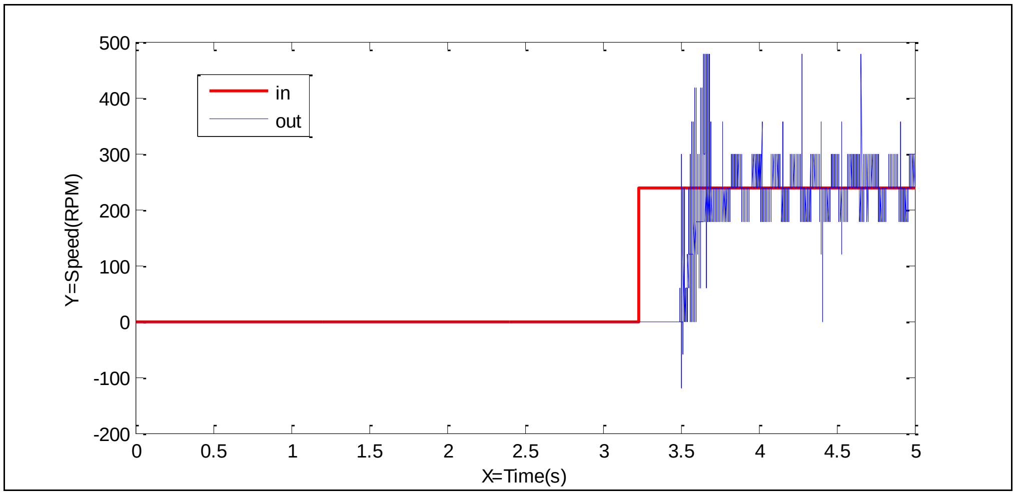
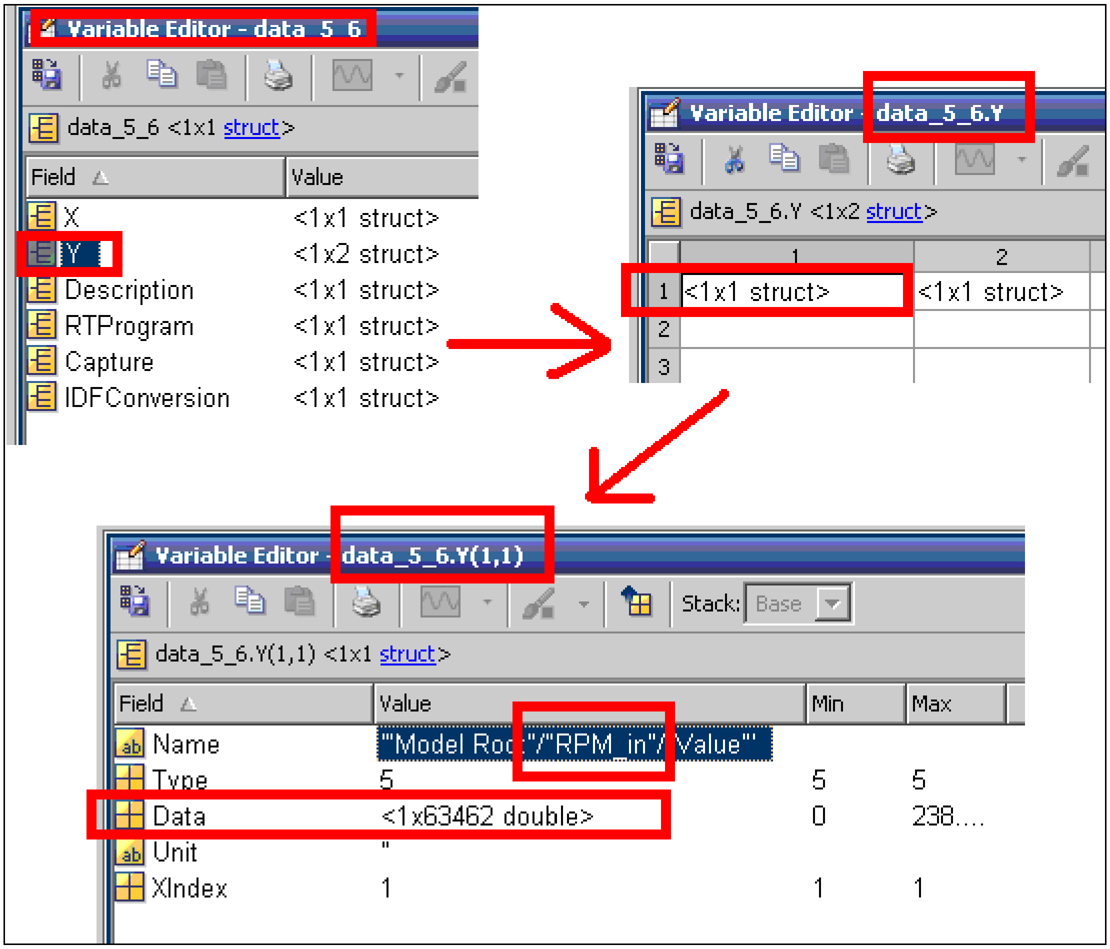


Figure 46: Example data plotted in MATLAB with labels/color added manually.


Variable Editor - data_5_6

data_5_6 <1x1 struct>

Field	Value
X	<1x1 struct>
Y	<1x2 struct>
Description	<1x1 struct>
RTProgram	<1x1 struct>
Capture	<1x1 struct>
IDFConversion	<1x1 struct>

Variable Editor - data_5_6.Y

data_5_6.Y <1x2 struct>

	1	2
1	<1x1 struct>	<1x1 struct>
2		
3		

Variable Editor - data_5_6.Y(1,1)
Stack: Base

data_5_6.Y(1,1) <1x1 struct>

Field	Value	Min	Max
Name	'"Model Roo"/"RPM_in"/ Value"'		
Type	5	5	5
Data	<1x63462 double>	0	238....
Unit	''		
XIndex	1	1	1

Figure 47: Locating the Data for the RPM_in variable. (RPM_out was in data_5_6.Y(1,2).)

Motor Speed Control Application

1. Construct the motor subsystem shown in the figure below, also connecting the optical encoder and code wheel to the internal motor shaft and adding the brake. Also add the 5 Volt power connection and the 2.7 kΩ pull-up resistors between 5 Volts and the encoder channels, A and B. Verify 5 Volt and 25 Volt DC power supplies are set correctly before connecting to the system.

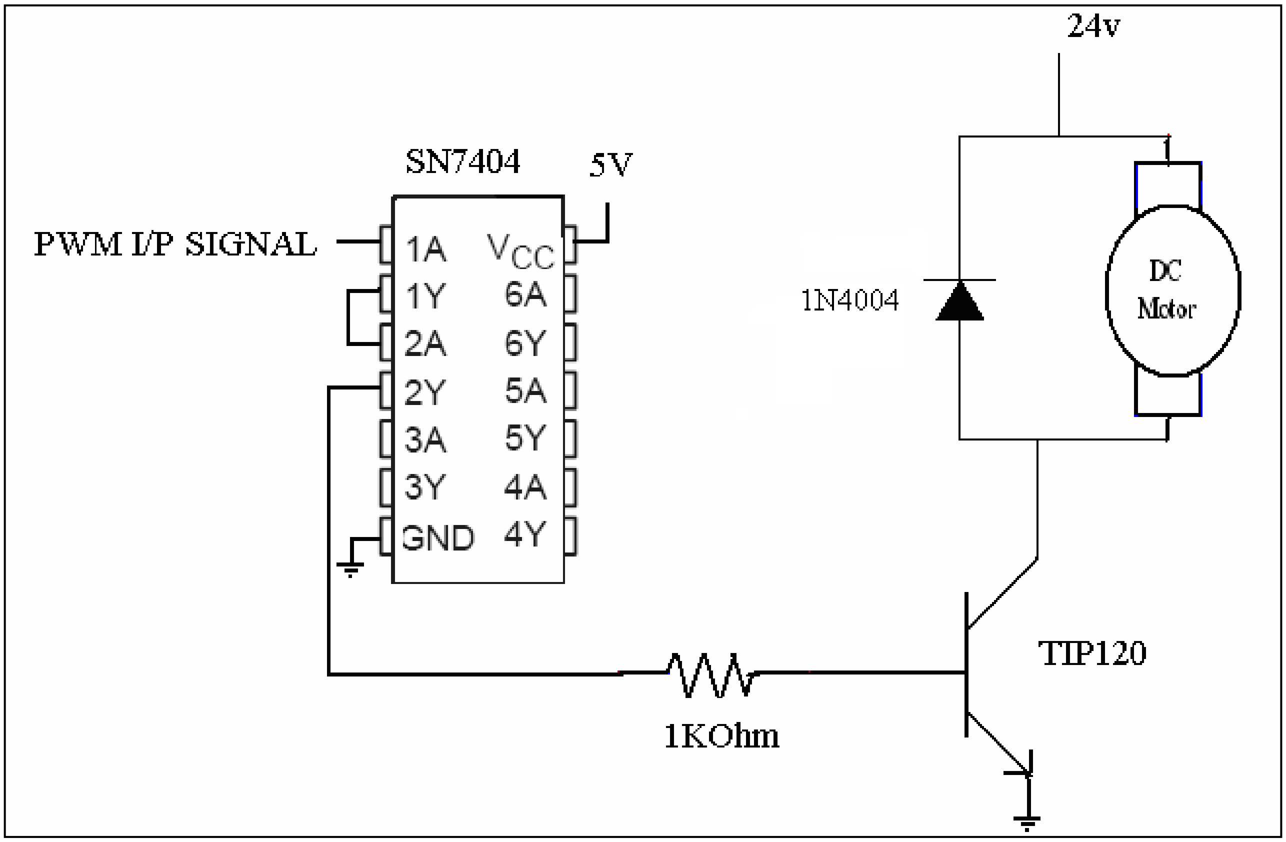


Figure 48: Motor Hardware Schematic.

For the motor-optical encoder system provided at Bradley University, the motor and optical encoder connections are shown in the figure on the next page. The pin assignment for the TIP120 NPN transistor is shown in the figure below.

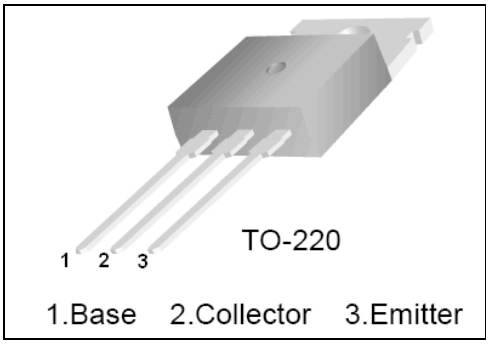
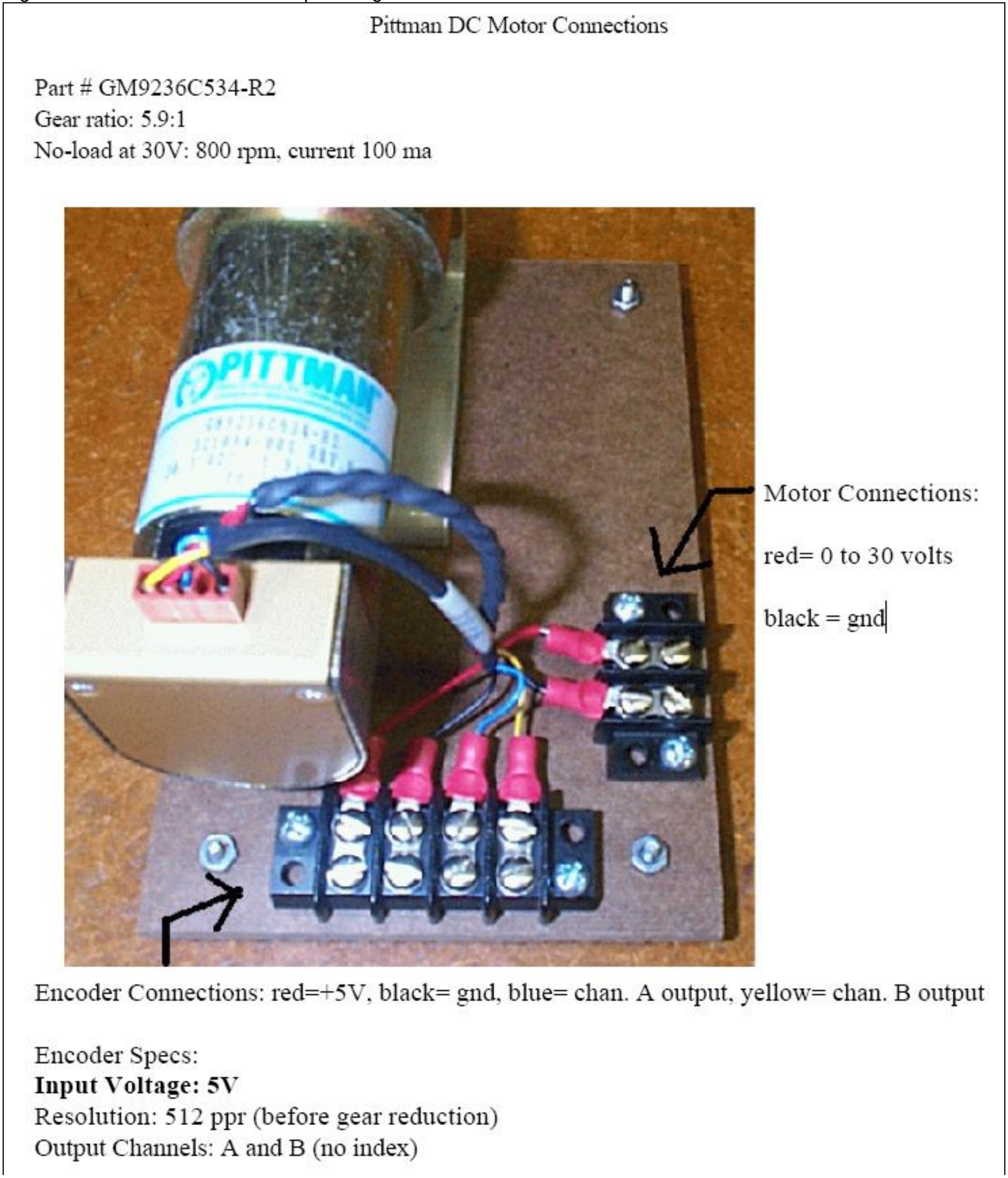


Figure 49: TIP120 NPN transistor pin assignment.



Pittman DC Motor Connections

Part # GM9236C534-R2
Gear ratio: 5.9:1
No-load at 30V: 800 rpm, current 100 ma

Motor Connections:

red= 0 to 30 volts

black = gnd

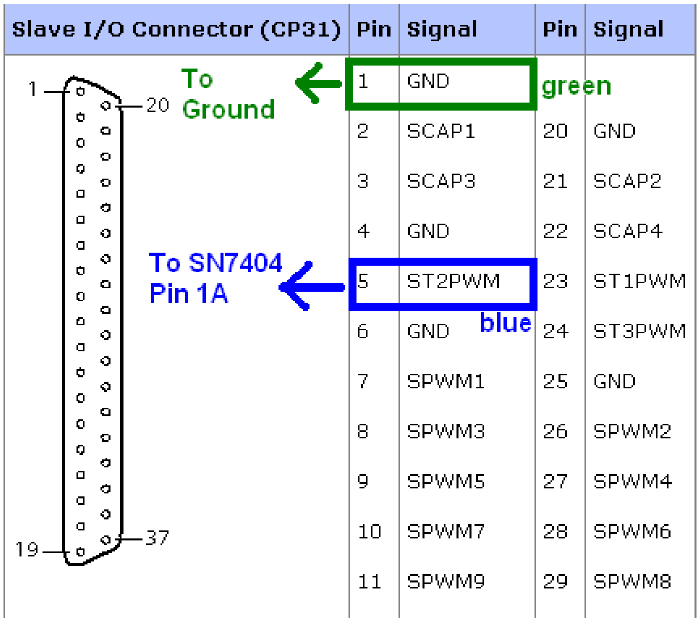
Encoder Connections: red=+5V, black= gnd, blue= chan. A output, yellow= chan. B output

Encoder Specs:
Input Voltage: 5V
Resolution: 512 ppr (before gear reduction)
Output Channels: A and B (no index)

Figure 50: Motor and encoder connections for system at Bradley University.

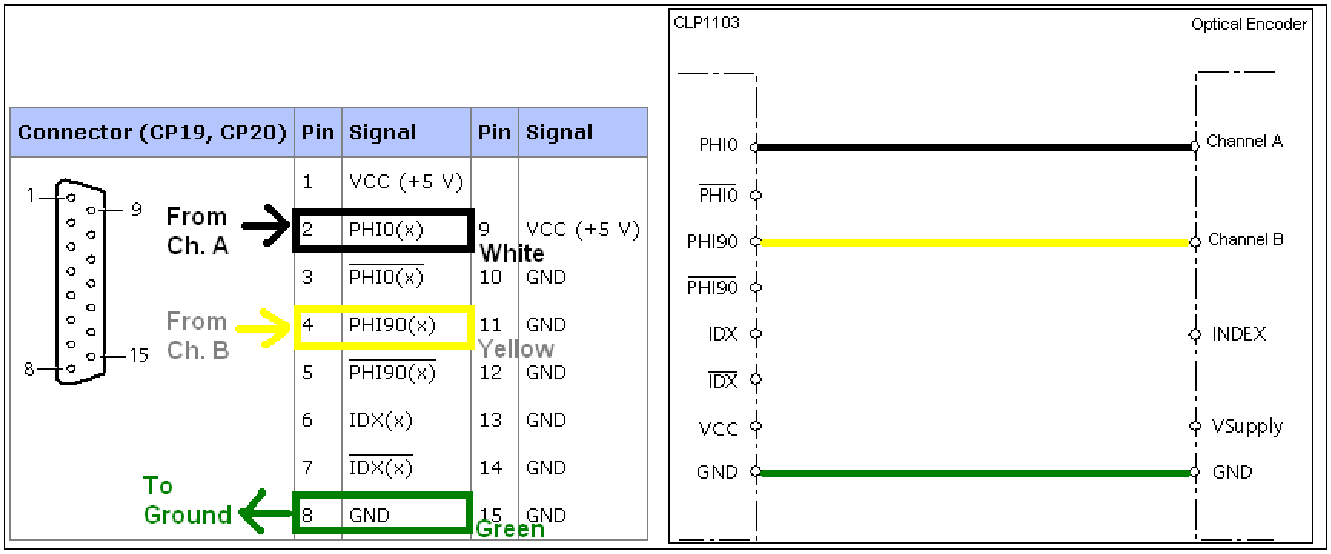
2. Make all connections between the Connector Panel, the encoder and the Motor System. The required connections to the Slave I/O and Inc1 connectors are shown in the figures that follow.

   Note: Colors listed are for the wires used on the current version of the connectors at Bradley University.



Slave I/O Connector (CP31)	Pin	Signal	Pin	Signal
	1	GND		
	2	SCAP1	20	GND
	3	SCAP3	21	SCAP2
	4	GND	22	SCAP4
	5	ST2PWM	23	ST1PWM
	6	GND	24	ST3PWM
	7	SPWM1	25	GND
	8	SPWM3	26	SPWM2
	9	SPWM5	27	SPWM4
	10	SPWM7	28	SPWM6
	11	SPWM9	29	SPWM8

Figure 51: PWM connections on Slave I/O connector.



Connector (CP19, CP20)	Pin	Signal	Pin	Signal
	1	VCC (+5 V)		
	2	PHI0(x)	9	VCC (+5 V)
	3	$\overline{\text{PHI0(x)}}$	10	GND
	4	PHI90(x)	11	GND
	5	$\overline{\text{PHI90(x)}}$	12	GND
	6	IDX(x)	13	GND
	7	$\overline{\text{IDX(x)}}$	14	GND
	8	GND	15	GND

Figure 52: Encoder Connections to/from Inc1 connector.

Note: Once code has been downloaded at later steps of these instructions, it is generally best to turn the power supplies on after downloading the code and turn them off before stopping the code from running in ControlDesk.

3. Now that all the hardware connections are complete, the work with ControlDesk and Simulink can begin. Start by following the instructions in the sections of this tutorial labeled:
   a. Setup/Startup
   b. Detecting the DS1103
   c. Creating a New Simulink Model

   Leave the model open after saving it or re-open it using the instructions in the "Opening MATLAB/Simulink" section.

4. Now, Add the DS1103 dSPACE blocks needed for PWM generation and Encoder input capture can be added. These blocks are found in the "Simulink Library Browser" under the "dSPACE RTI1103" heading on the left menu window.

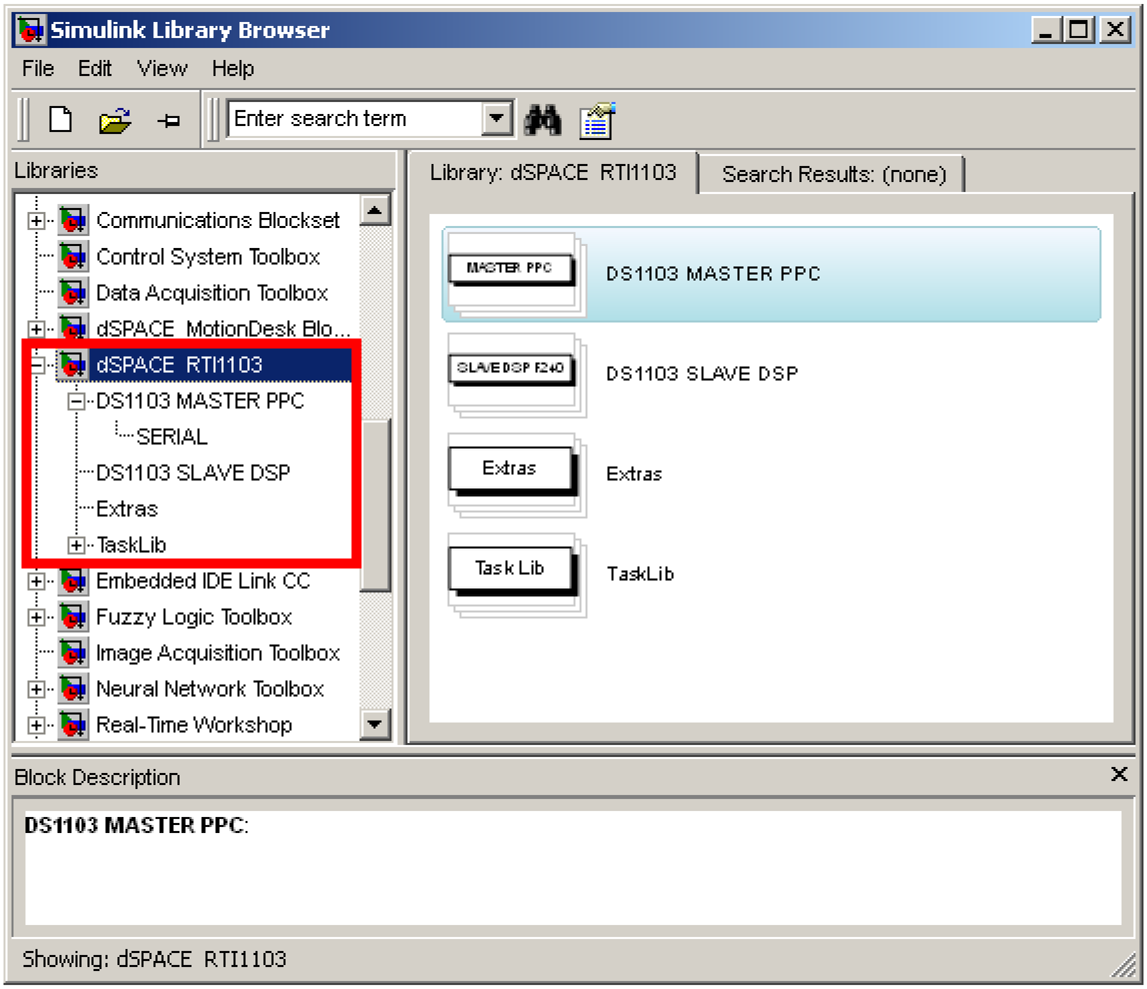
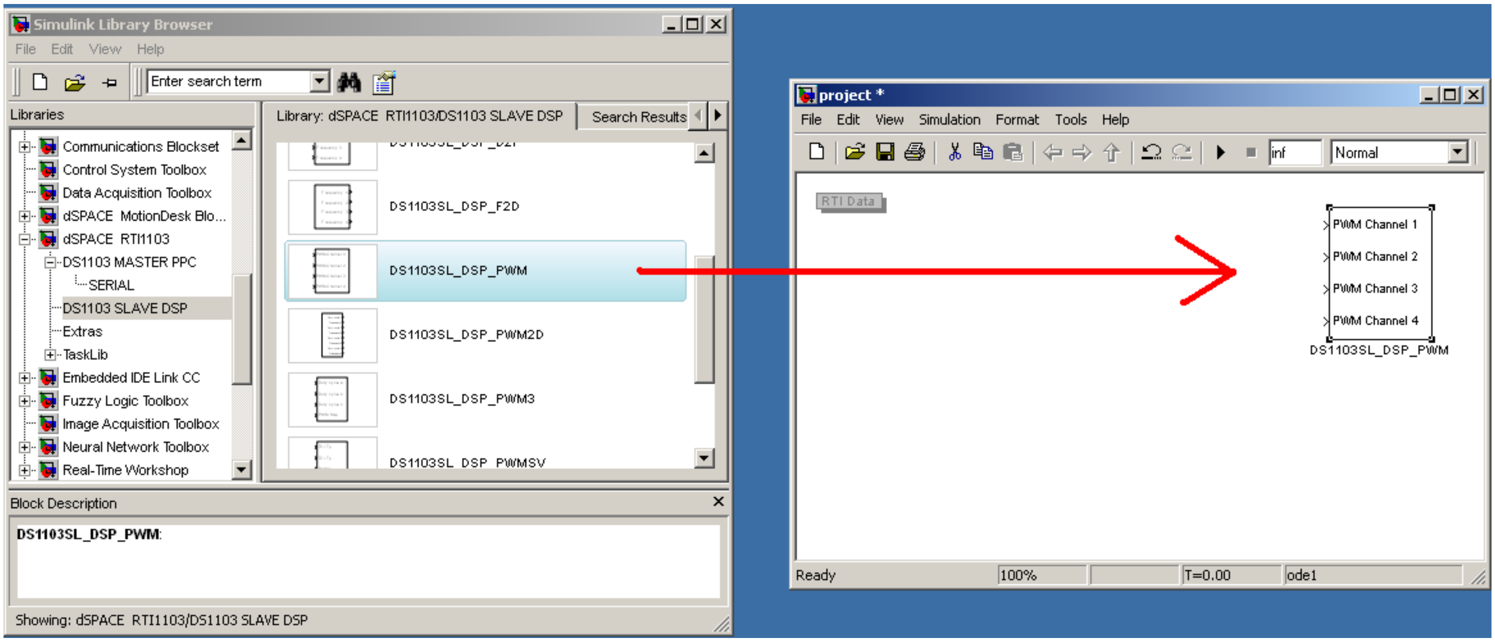


Figure 53: Location of dSPACE DS1103 blocks in the Simulink Library Browser.

4a. Locate the PWM block labeled “DS1103SL_DSP_PWM” at:
dSPACE RTI1103 ► DS1103 SLAVE DSP ► DS1103SL_DSP_PWM.
Click and Drag the block into the model created for the project.

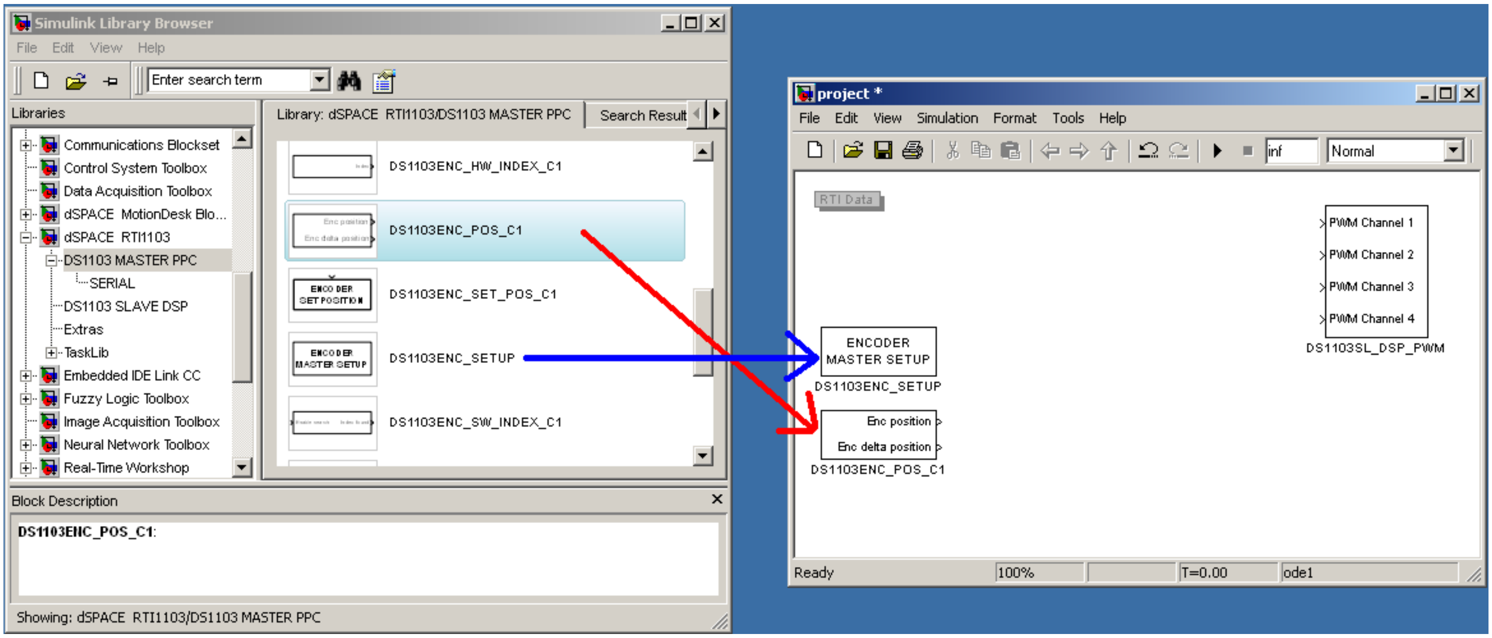


DS1103SL_DSP_PWM:

Showing: dSPACE RTI1103/DS1103 SLAVE DSP

Figure 54: Locating and Adding the PWM block.

4b. Some features such as the encoder blocks require more than one block to function. One block has outputs. The other block is for more general settings that would also apply to other related (encoder) blocks if they were used.
Locate the Encoder blocks labeled “DS1103ENC_SETUP” and “DS1103ENC_POS_C1” at:
dSPACE RTI1103 ► DS1103 MASTER PPC ► DS1103ENC_SETUP.
And
dSPACE RTI1103 ► DS1103 MASTER PPC ► DS1103ENC_POS_C1.
Click and Drag both blocks into the model created for the project.



DS1103ENC_POS_C1:

Showing: dSPACE RTI1103/DS1103 MASTER PPC

Figure 55: Locating and Adding the encoder blocks.

5. Add and Connect constant, gain, terminator, and ground blocks to the Simulink project then save the project, but do not close it.
    a. Add a constant block and connect it to “PWM Channel 1” of the PWM block. Set its value to a number between 0 and 1. Change the label of the block to “DutyCycle”. (This is the range for the input to the block as a percentage (0.5 = 50%) of the duty cycle.)
    b. Connect gain blocks to “Enc position” and “Enc delta position”. Set the gain of these blocks to “1”. Label the connections going into or out of these blocks “Enc1” and “Enc2”. (If these gain blocks are not added at this point, the outputs will probably not be monitorable in ControlDesk later.)
    c. Connect the “PWM Channel 2”, “PWM Channel 3”, and “PWM Channel 4” inputs to the PWM block to ground since they are not being used in this project. (Any connections that are open need to be attached to either a terminator or ground.)
    d. Connect the output of the gain blocks to terminator blocks.

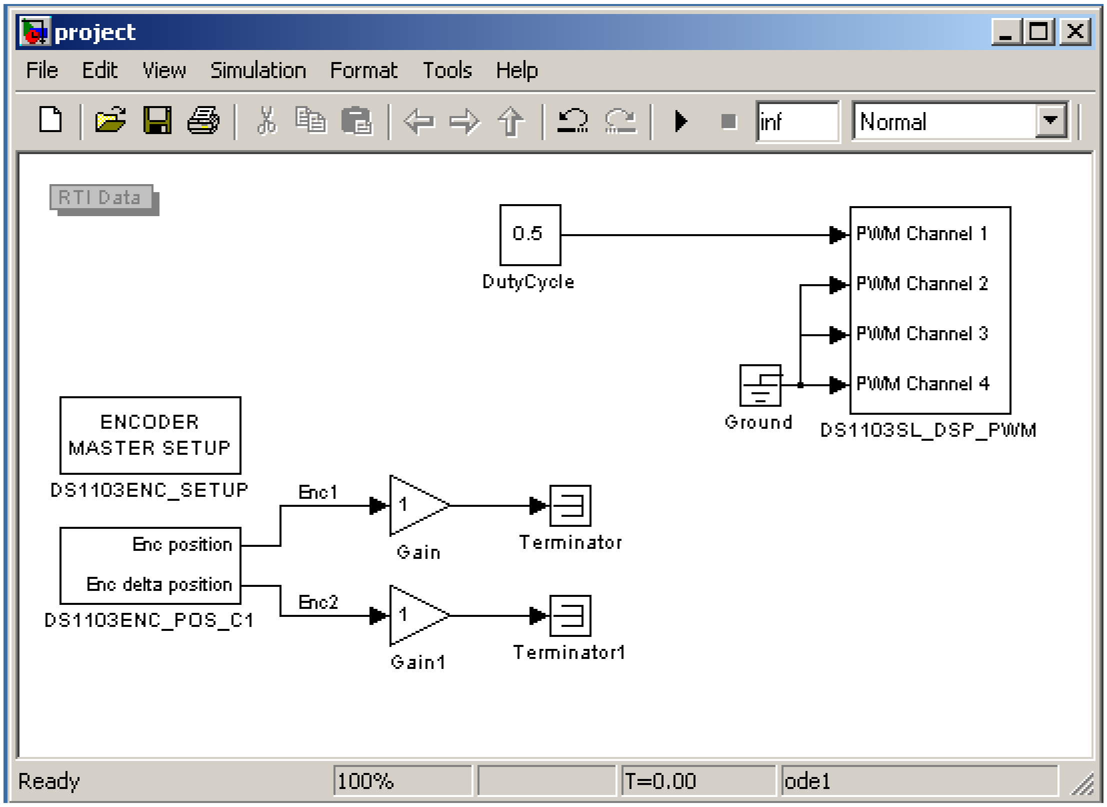
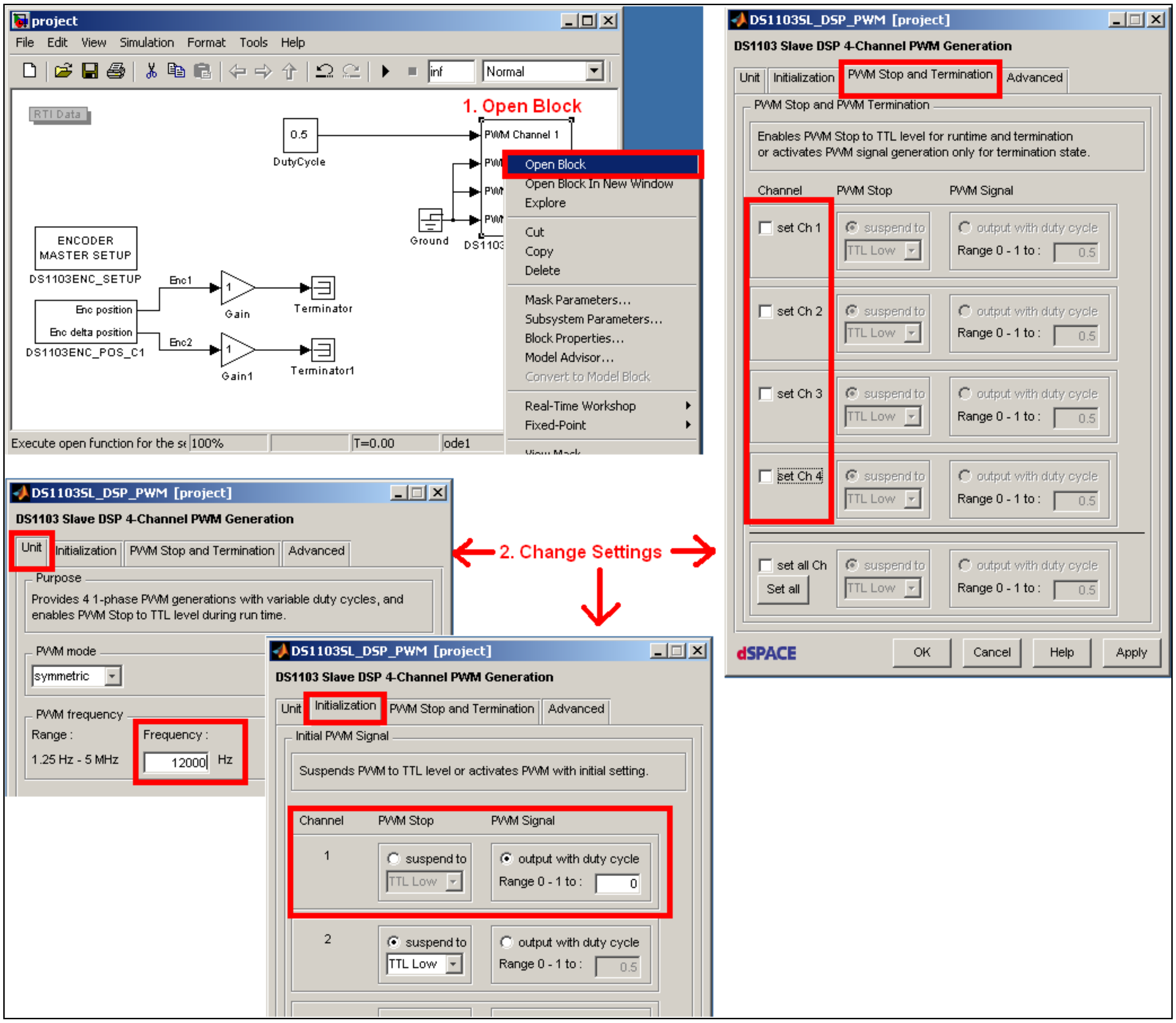


Figure 56: Simulink model after adding blocks and connections and saving. The DutyCycle is set to 50%.

6. Change the Properties/Settings/Parameters of the PWM block. (No encoder settings need to be changed since the "Inc1" connector input is being used.) To do this, Right-click the block and Select "Open Block" from the drop down menu. Then, change the appropriate settings.

   Change the following DS1103SL_DSP_PWM Settings in the dialog box that appears:

   a. "Unit" tab: Frequency = 12000 Hz
   b. "Initialization" tab: For Channel 1: PWM Stop = "output with duty cycle", Range = 0 (The ST2PWM output being used is Channel 1 of this block.)
   c. "PWM Stop and Termination" tab: Remove check from box for "set Ch" X for all channels.



DS1103SL_DSP_PWM [project] — DS1103 Slave DSP 4-Channel PWM Generation

Unit | Initialization | PWM Stop and Termination | Advanced

Purpose: Provides 4 1-phase PWM generations with variable duty cycles, and enables PWM Stop to TTL level during run time.

PWM mode: symmetric

PWM frequency: Range : 1.25 Hz - 5 MHz; Frequency : 12000 Hz

Initial PWM Signal: Suspends PWM to TTL level or activates PWM with initial setting.

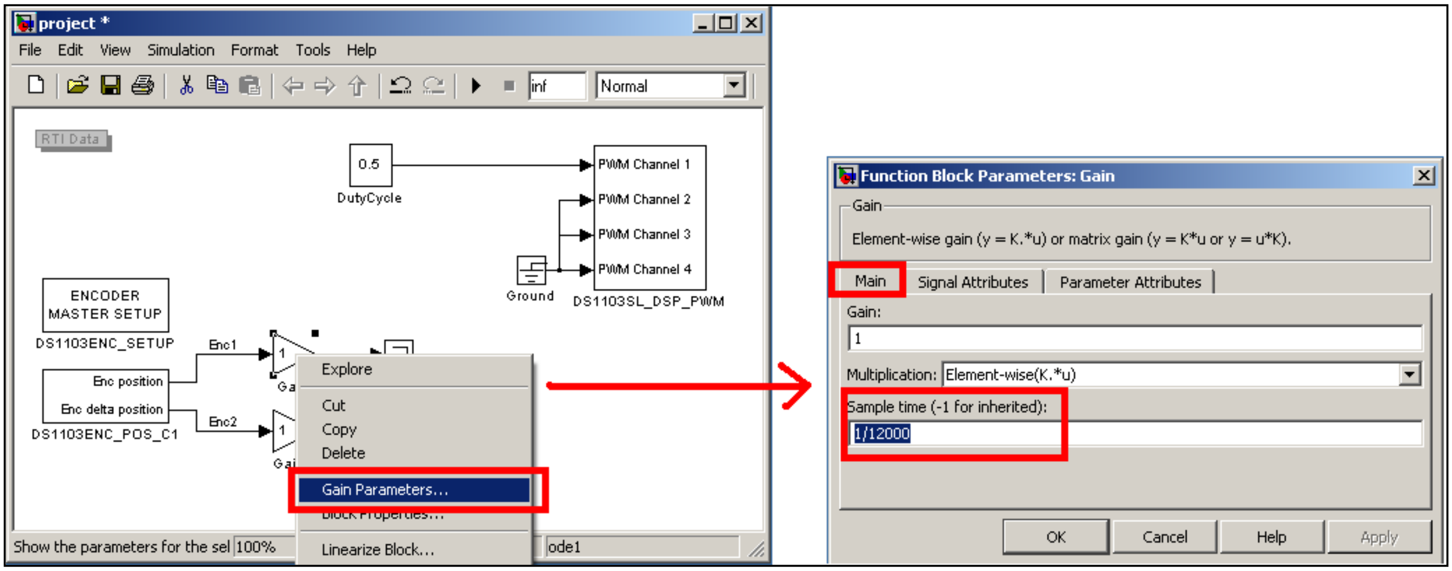
Channel	PWM Stop	PWM Signal
1	○ suspend to TTL Low	● output with duty cycle Range 0 - 1 to : 0
2	● suspend to TTL Low	○ output with duty cycle Range 0 - 1 to : 0.5

PWM Stop and PWM Termination: Enables PWM Stop to TTL level for runtime and termination or activates PWM signal generation only for termination state.

Channel	PWM Stop	PWM Signal
□ set Ch 1	● suspend to TTL Low	○ output with duty cycle Range 0 - 1 to : 0.5
□ set Ch 2	● suspend to TTL Low	○ output with duty cycle Range 0 - 1 to : 0.5
□ set Ch 3	● suspend to TTL Low	○ output with duty cycle Range 0 - 1 to : 0.5
□ set Ch 4	● suspend to TTL Low	○ output with duty cycle Range 0 - 1 to : 0.5
□ set all Ch Set all	● suspend to TTL Low	○ output with duty cycle Range 0 - 1 to : 0.5

Figure 57: PWM block settings.

7. Change the “Sample time” of the Gain blocks to “1/12000 seconds” by Right-clicking the blocks, Selecting “Gain Parameters…”, and Changing the “Sample time” in each block to “1/12000” on the “Main” tab of the dialog window that appears.



Gain

Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).

Gain: 1

Multiplication: Element-wise(K.*u)

Sample time (-1 for inherited): 1/12000

Figure 58: Changing the sampling time of gain blocks.

8. Connect the channels of an oscilloscope to the PWM output and one of the channels of the optical encoder in the hardware circuit.

9. Build and Download the model to the DS1103 board using the instructions found in the section of this tutorial labeled “Building/Downloading a Model”, setting the “Fixed-step size” for the model to “1/12000”. (When this step is complete, the system should be running and the DutyCycle should be initialized to the value set in the model.)

10. Now, it is time to have some fun with ControlDesk. Create a layout in ControlDesk following the directions in the “Monitoring/Controlling/Recording Values” section of this tutorial. Connect the following variables/values to the type of instrument(s) indicated.
    a. DutyCycle: Slider (Change Max Range to “1”) and Display and PlotterArray
    b. Enc1: Display
    c. Enc2: Display

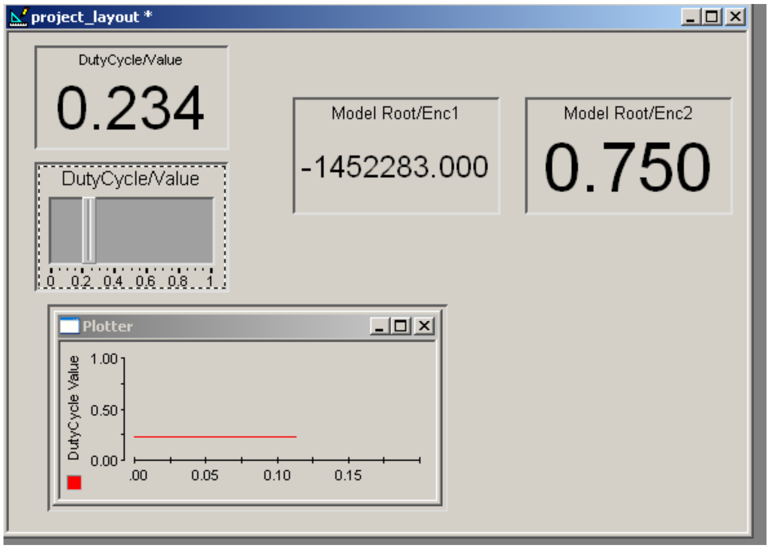


Figure 59: Layout in ControlDesk with system running and slider used to change the DutyCycle.

11. Observe that changing the DutyCycle value using the slider changes the duty cycle of the PWM measurement on the oscilloscope to a similar percentage and changes the encoder frequency observed. Also observe that the DutyCycle has approximately a 12 kHz frequency using the oscilloscope.

12. Observe that in the case of this system, the absolute value of the Enc1 output on the ControlDesk Layout generally increases since the motor shaft is only rotating in one direction and it is probably measuring distance. Multiplying the Enc2 output on the ControlDesk Layout by the inverse of the sampling time (the frequency), 12000 in this case should approximately correspond to the Encoder frequency reported on the oscilloscope.

13. Now, Delete the instruments from the ControlDesk layout and save it. Also, Delete the "DutyCycle" and "Gain" blocks from the Simulink model, Delete the "Enc1" label, and reconnect the "Enc position" output to its terminator block. Save the model.

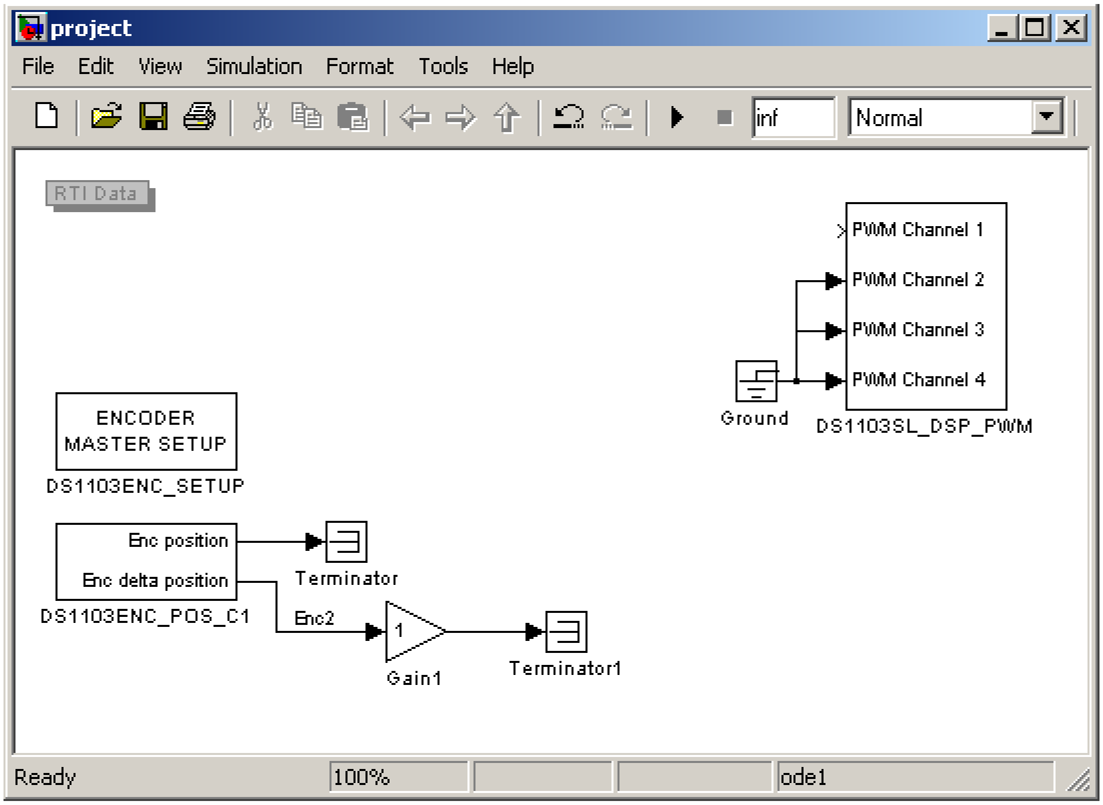


Figure 60: Simulink model after Step 13 is completed.

14. Setup/Add the blocks to the Simulink model for the RPM output calculation/conversion.
    a. Re-label the “Enc delta position” output connection “count_out”.
    b. Add another gain block between “Gain1” and the terminator block.
    c. Label these two gain blocks “count_to_Hz” and “Hz_to_RPM” and set the sample time for each block to “1/12000” (See Step 7 in this section).
    d. Change the gain of the “count_to_Hz” block to “12000” and of the “Hz_to_RPM” block to “0.019881”.
    e. Label the connection between the two gain blocks “Hz” and between the second gain block and the terminator “RPM_out”. (RPM_out is the actual output for the system.)

    Note: Leave enough space between the encoder block output and the first gain block for an additional connection in the upward direction later.)

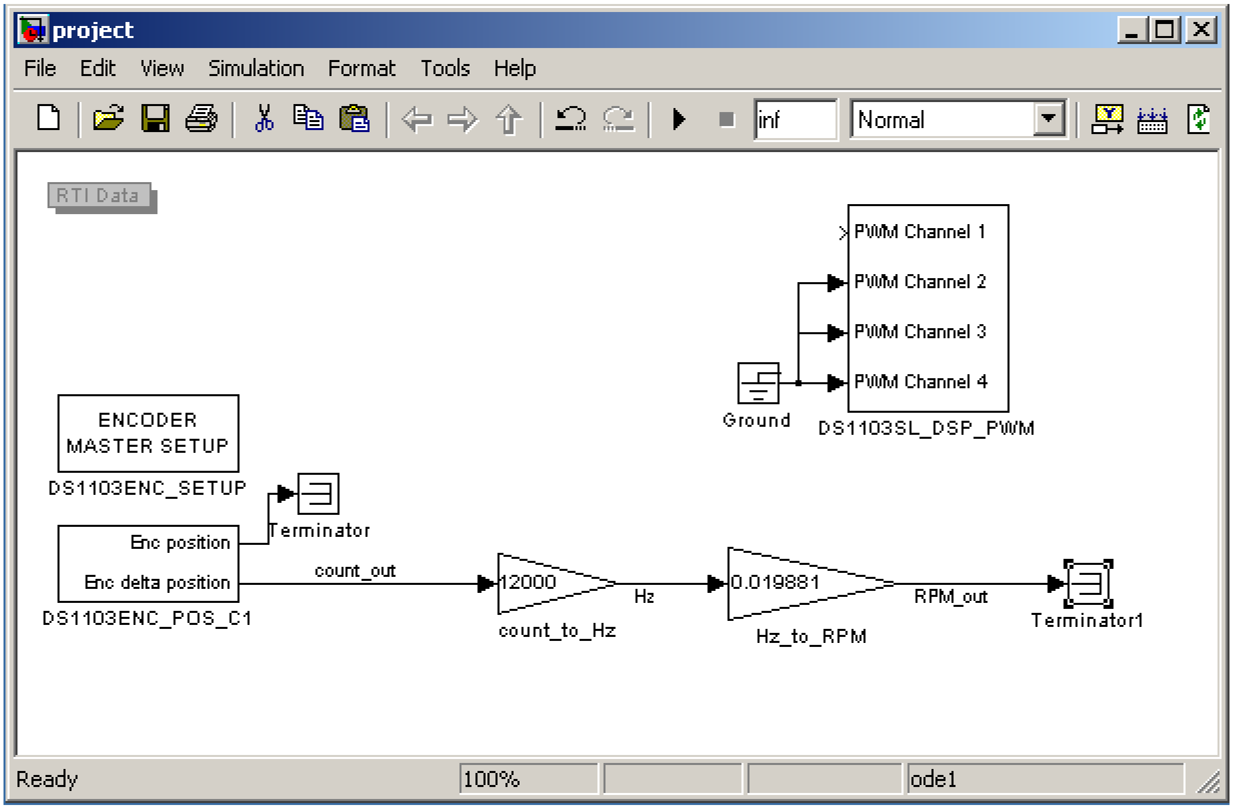


Figure 61: Simulink model after Step 14 is completed.

15. Add the desired RPM input block and conversion blocks.
    a. Add a Constant block and two (2) Gain blocks toward the top on the Simulink model.
    b. Label the constant block "RPM_in" and change its gain to "0".
    c. Connect the two gain blocks to the output of the "RPM_in" block in series with one another.
    d. Label the first Gain block "RPM_to_Hz" and the second Gain block "Hz_to_count" and set the sample time for both blocks to "1/12000".
    e. Change the gain of the "RPM_to_Hz" block to "50.2996" and of the "Hz_to_count" block to "", and label the connection between the two gain blocks "Hz".

Note: It will be necessary to increase the size of the window for the model and rearrange some blocks to fit all the blocks in the model at this step or at some point later in the process.

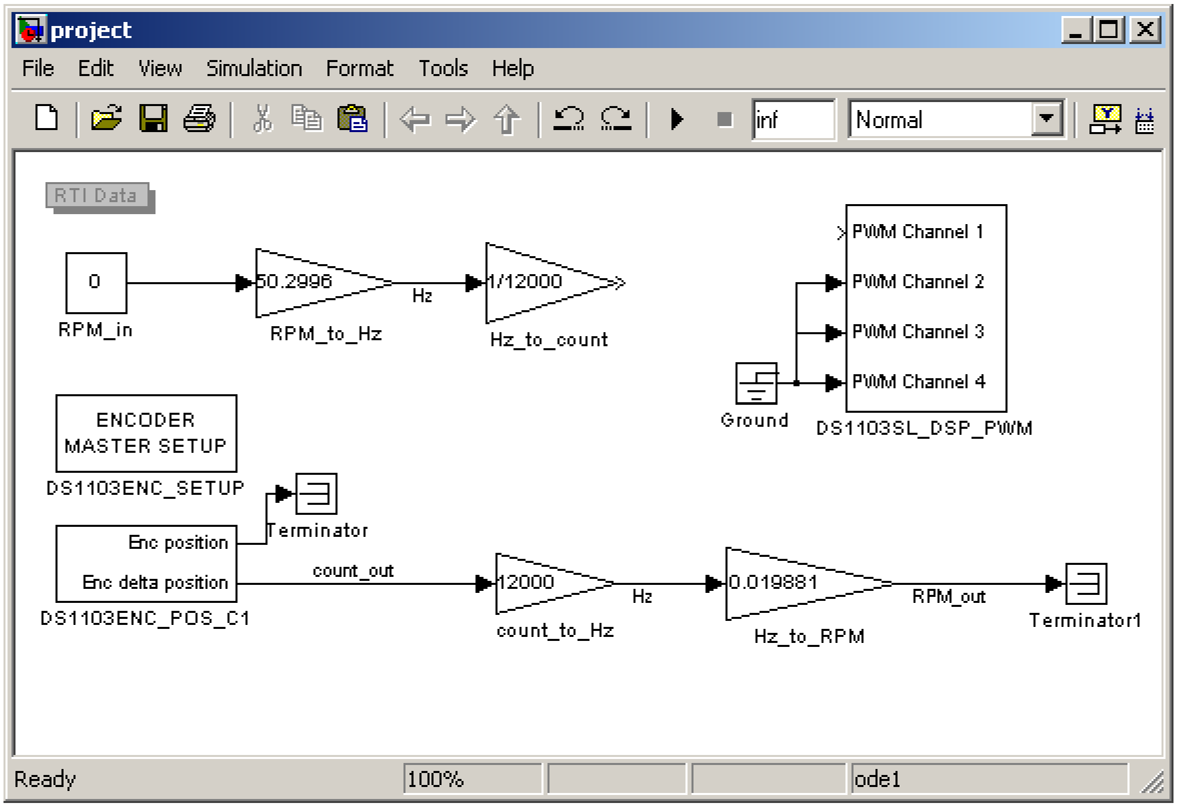
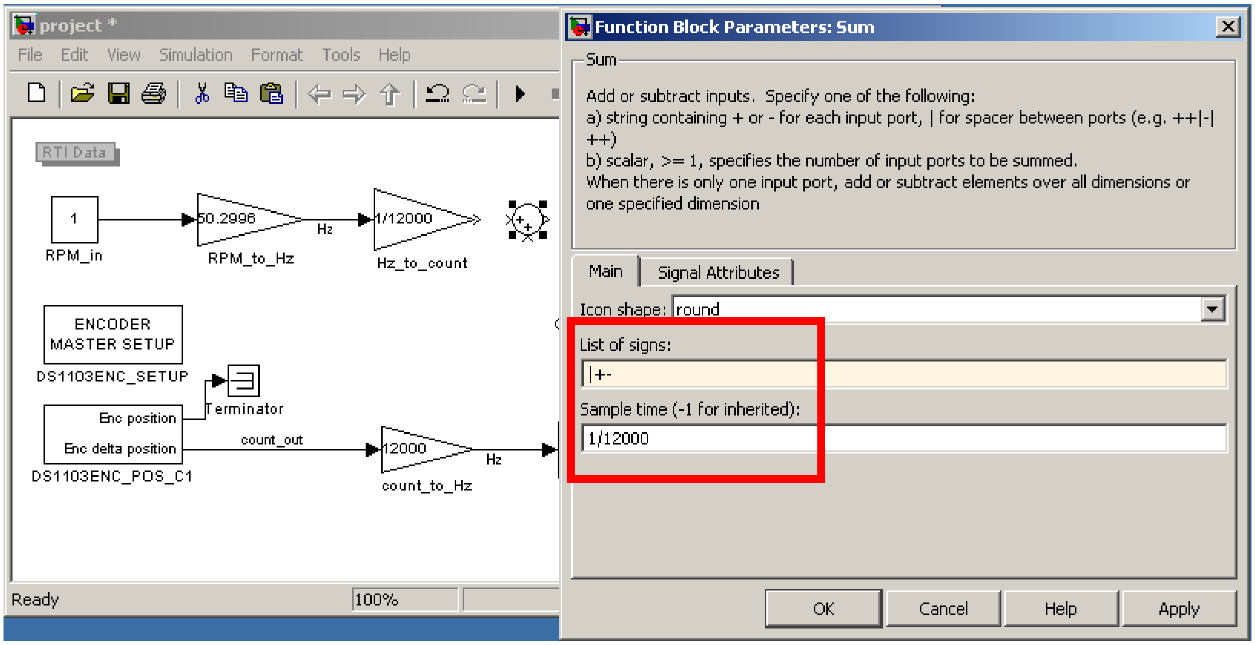


Figure 62: Simulink model after Step 15 is completed.

16. Add a Sum block with one positive and one negative input with a sample time of “1/12000”. Connect “count_out” to the negative terminal and the output of the “Hz_to_count” block to the positive terminal, and label this connection “count_in”.



Function Block Parameters: Sum

Sum

Add or subtract inputs. Specify one of the following:
a) string containing + or - for each input port, | for spacer between ports (e.g. ++|-|++)
b) scalar, >= 1, specifies the number of input ports to be summed.
When there is only one input port, add or subtract elements over all dimensions or one specified dimension

Main | Signal Attributes

Icon shape: round

List of signs: |+-

Sample time (-1 for inherited): 1/12000

OK | Cancel | Help | Apply

Figure 63: New Sum block parameters.

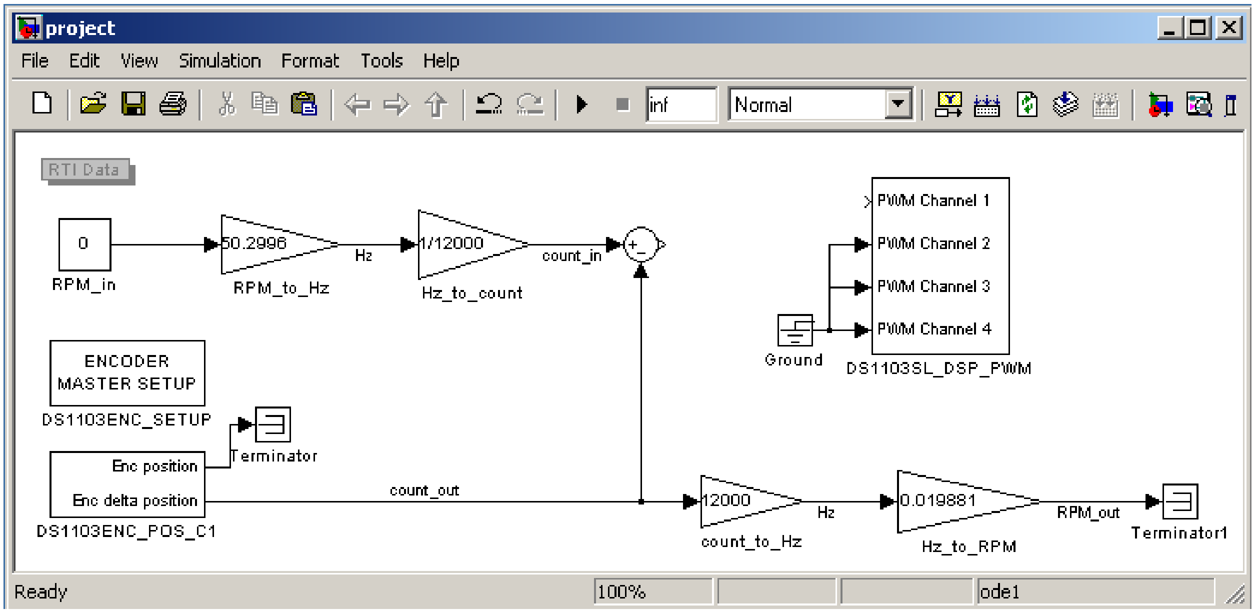
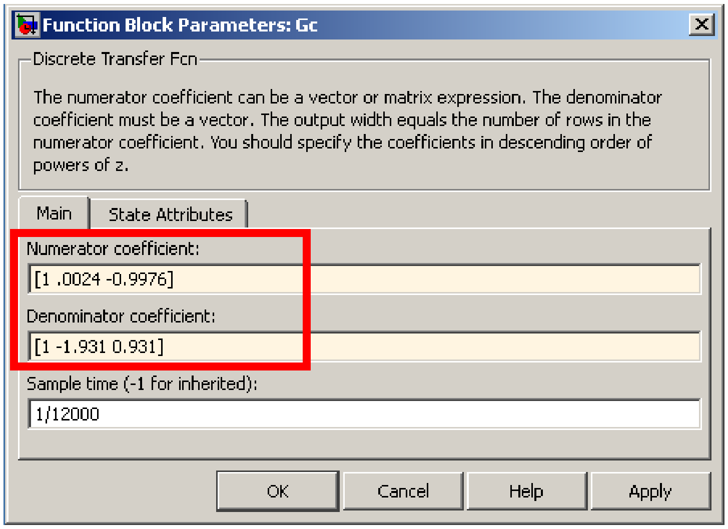


Figure 64: Simulink model after Step 16 is completed.

17. Add the Controller.
    a. Add three (3) Gain blocks and a "Discrete Transfer Fcn" block to the Simulink model and set the sample time for each block to "1/12000".
    b. Connect the Gain blocks in series with one another. Then connect the first block to the output of the Sum block and the third block to the input of the "Discrete Transfer Fcn" block.
    c. Label the Gain blocks (from first to last) "K", "Kz", and "Kz3" and the "Discrete Transfer Fcn" block "Gc".
    d. Change the gain of "K" to "72.0631", of "Kz" to "0.482282", and of "Kz3" to "7".
    e. In the "DiscreteTransferFcn Parameters" dialog box for "Gc", change the "Numerator coefficient" to "[1 .0024 -0.9976]" and the "Denominator coefficient" to "[1 -1.931 0.931]".



Function Block Parameters: Gc

Discrete Transfer Fcn

The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of z.

Main | State Attributes

Numerator coefficient:
[1 .0024 -0.9976]

Denominator coefficient:
[1 -1.931 0.931]

Sample time (-1 for inherited):
1/12000

Figure 65: New Discrete Transfer Function block parameters.

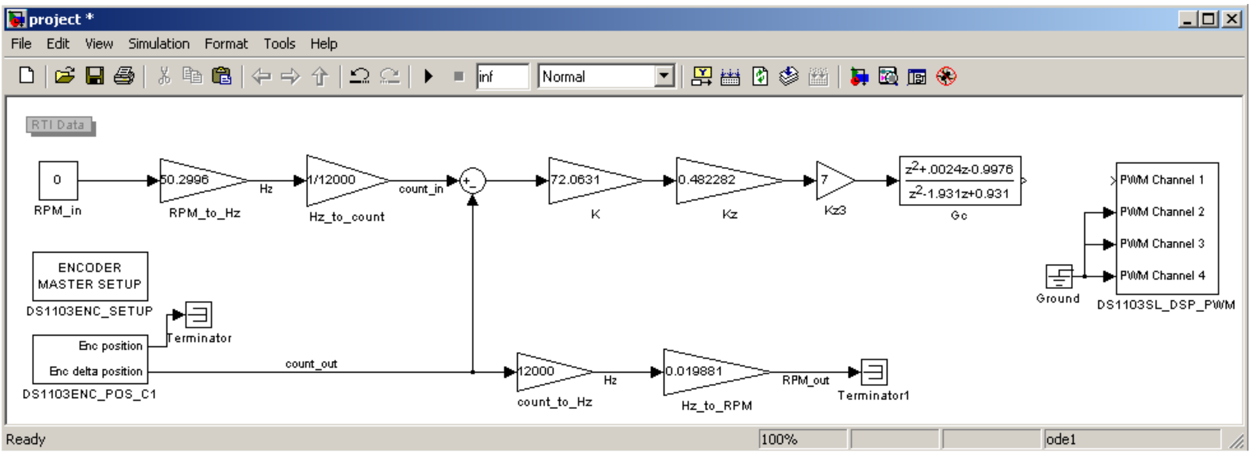
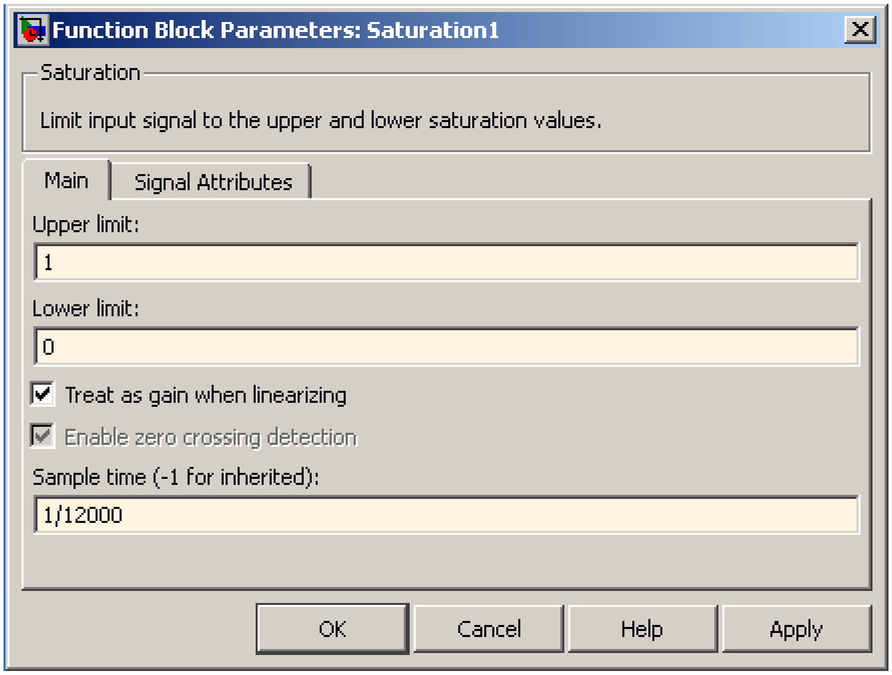


Figure 66: Simulink model after Step 17 is completed.

18. Add a Gain block and connect it to the output of "Gc". Label it "Hz_DutyCycle". Also label the connection between the two blocks "TF_out". Set the gain of "Hz_DutyCycle" to "0.00003" and its sample time to "1/12000". (This block is required because the output of "Gc" is a frequency and the "PWM Channel 1" input is a percentage. (This block coverts the frequency in Hz to a percentage of the maximum input the system is designed for, approximately 660 RPM.)

19. Add a Saturation block and connect it between the "Hz_DutyCycle" output and the "PWM Channel 1" input. Set its sample time to "1/12000", and Set its "Upper limit" to "1" and its "Lower limit" to "0". Label the block "0_to_1" and its connection to "PWM Channel 1" "DutyCycle".



Function Block Parameters: Saturation1

Saturation

Limit input signal to the upper and lower saturation values.

Upper limit:

1

Lower limit:

0

☑ Treat as gain when linearizing

☑ Enable zero crossing detection

Sample time (-1 for inherited):

1/12000

Figure 67: New Saturation block parameters.

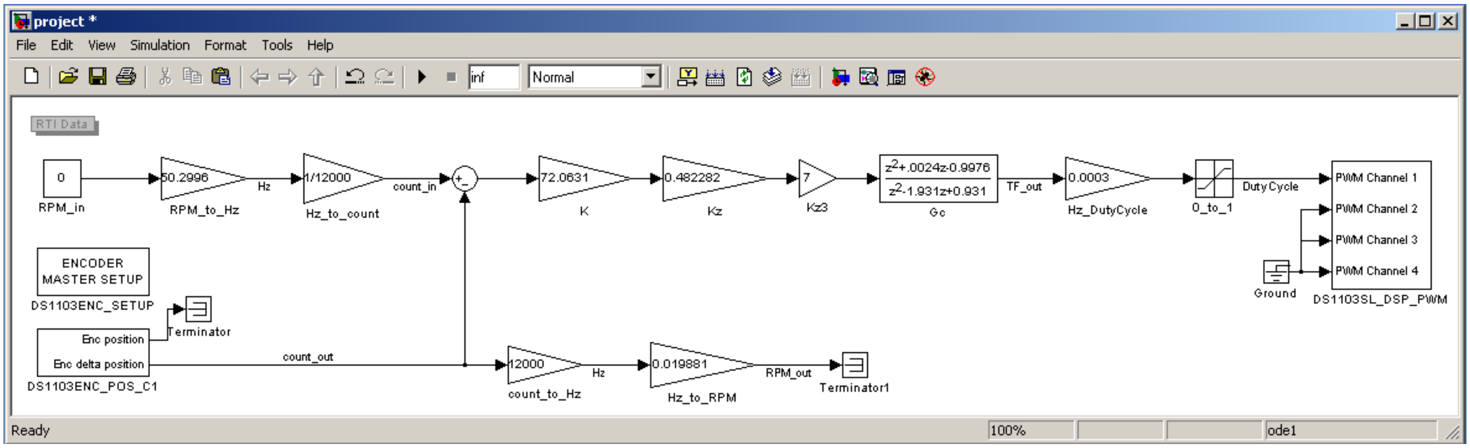


Figure 68: Simulink model after Step 19 is completed.

20. Make the Controller subsystem by selecting the “K”, “Kz”, “Kz3”, and “Gc” blocks (Ctrl+Shift and Click OR Drag box over the blocks), and then Selecting “Create Subsystem” from the “Edit” selection on the main menu of the Simulink model. Label the Subsystem “Controller”.

21. Re-label the connection between the “Controller” and “Hz_DutyCycle” “TF_out”.

22. Double-click the “Controller” to open it and change the label of “In1” to “Sum” and of “Out1” to “TF_out”. Also delete the label of the connection labeled “TF_out”. (The Simulink model is now complete.)

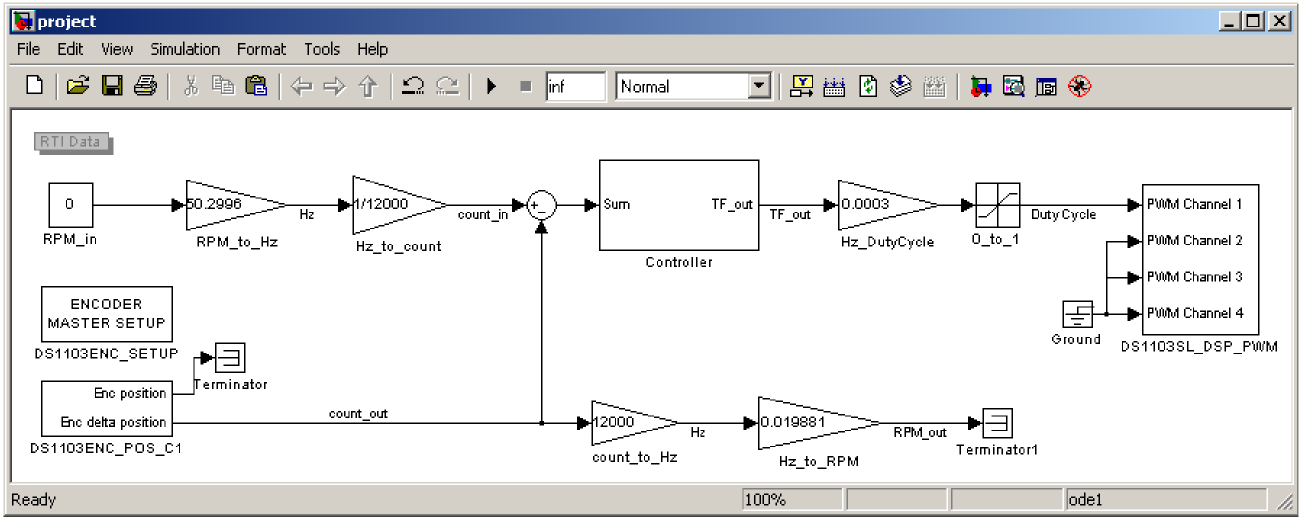


Figure 69: Simulink model after Step 22 is completed.

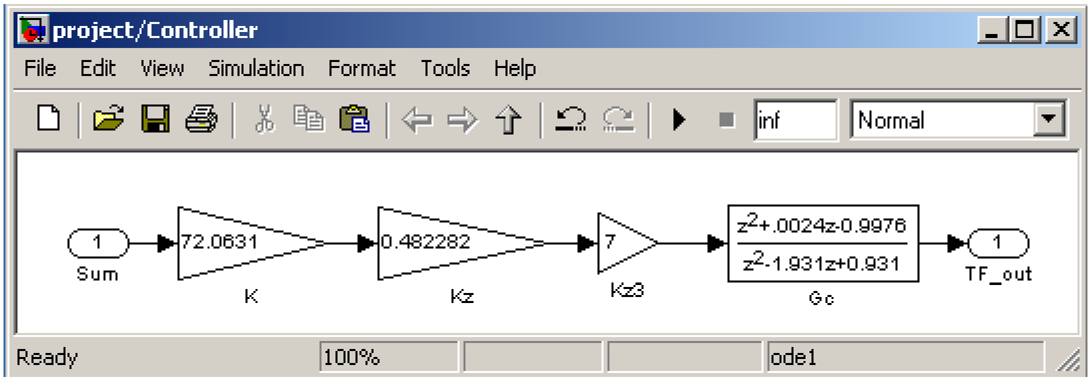


Figure 70: Controller subsystem shown in the figure above.

23. Make sure to Save the completed Simulink model.

24. Build and Download the code to the DS1103 board by following the instructions in the “Building/Downloading a Model” section of this tutorial. (The Fixed-step size should still be “1/12000”)

25. Now, it is time to edit the ControlDesk layout. Open the existing layout and follow the instructions in the "Monitoring/Controlling/Recording Values" section of this tutorial to Connect the following variables/values to the type of instrument(s) indicated.
    a. RPM_in: Display, Slider, NumericInput, and PlotterArray. (Change the Max Range of the Slider and NumericInput to "660".)
    b. RPM_out: Display and PlotterArray
    c. DutyCycle: Display

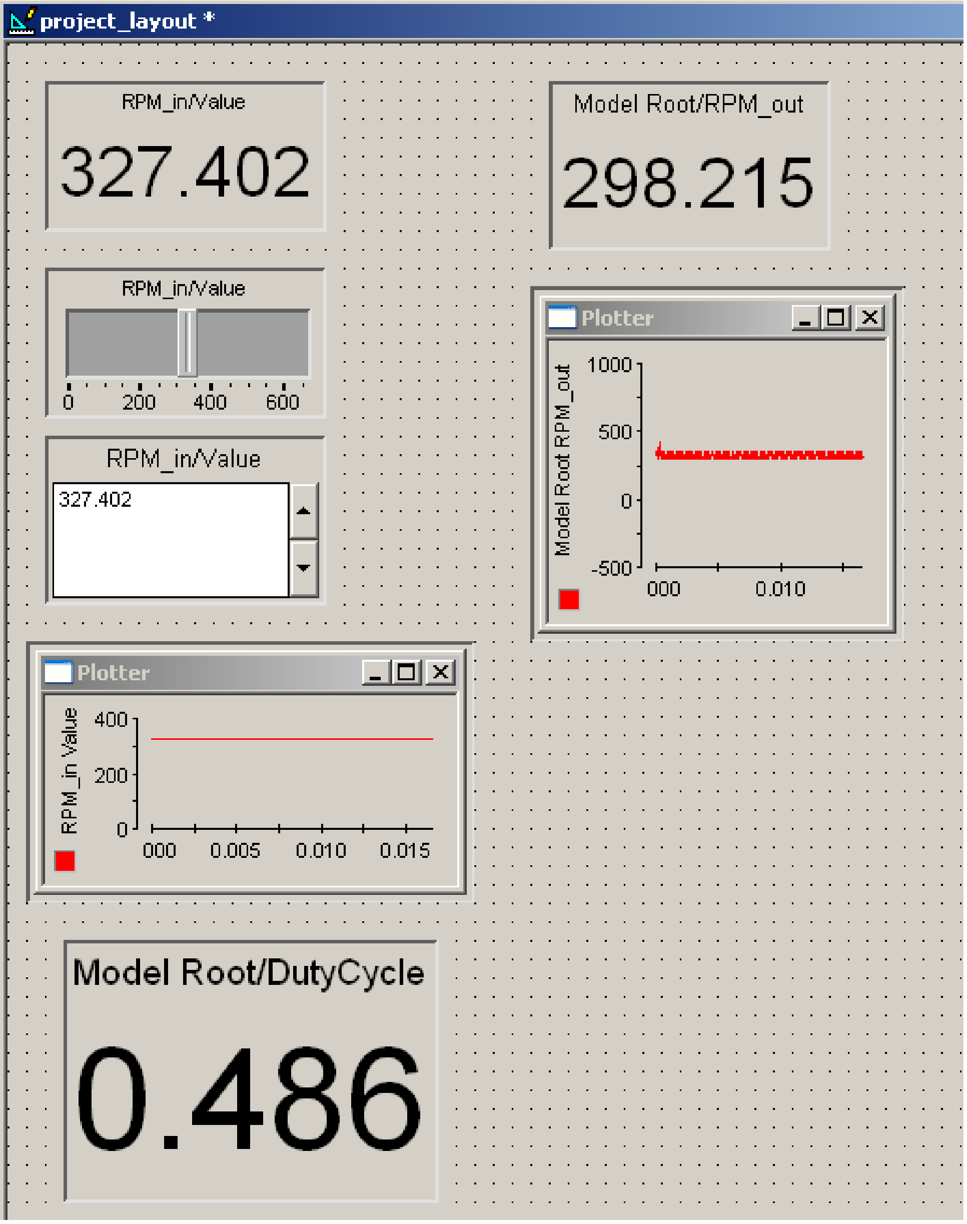
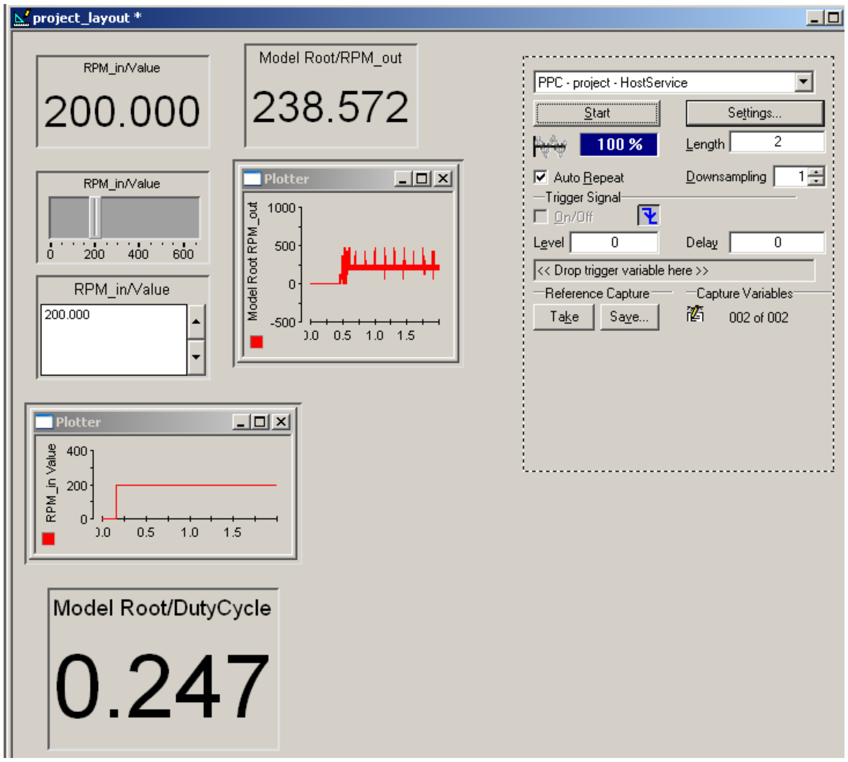


Figure 71: Layout in ControlDesk with system running and slider used to change RPM_in.

26. Observe that using the NumericInput for RPM_in provides for better observation of the actual Step Response of the System. Also observe that if voltage is applied to the brake, especially at mid-range input values, the DutyCycle will increase to compensate and maintain RPM_out near the desired value. (Generally RPM_in of 200 RPM was used in testing. For high speed inputs such as 660 RPM, RPM_out will show decreases in its value as more brake voltage is applied but will e stable.)

    Note: The maximum value before the brake stops is around 10+ Volts if the voltage is increased slowly. Dropping the voltage to somewhere between 5 and 8 Volts should cause the motor to start running again, but it may take some time for the system to start the motor and stabilize its speed.

27. In order to record data, Add a CaptureSettings instrument by following the instructions in the "CaptureSettings Instrument" subsection of the "Monitoring/Controlling/Recording Values" section of this tutorial.



project_layout *

RPM_in/Value 200.000

Model Root/RPM_out 238.572

RPM_in/Value 200.000

Model Root/DutyCycle 0.247

PPC - project - HostService

100 %

Length 2

☑ Auto Repeat

Downsampling 1

Trigger Signal

□ On/Off

Level 0

Delay 0

<< Drop trigger variable here >>

Reference Capture

Capture Variables 002 of 002

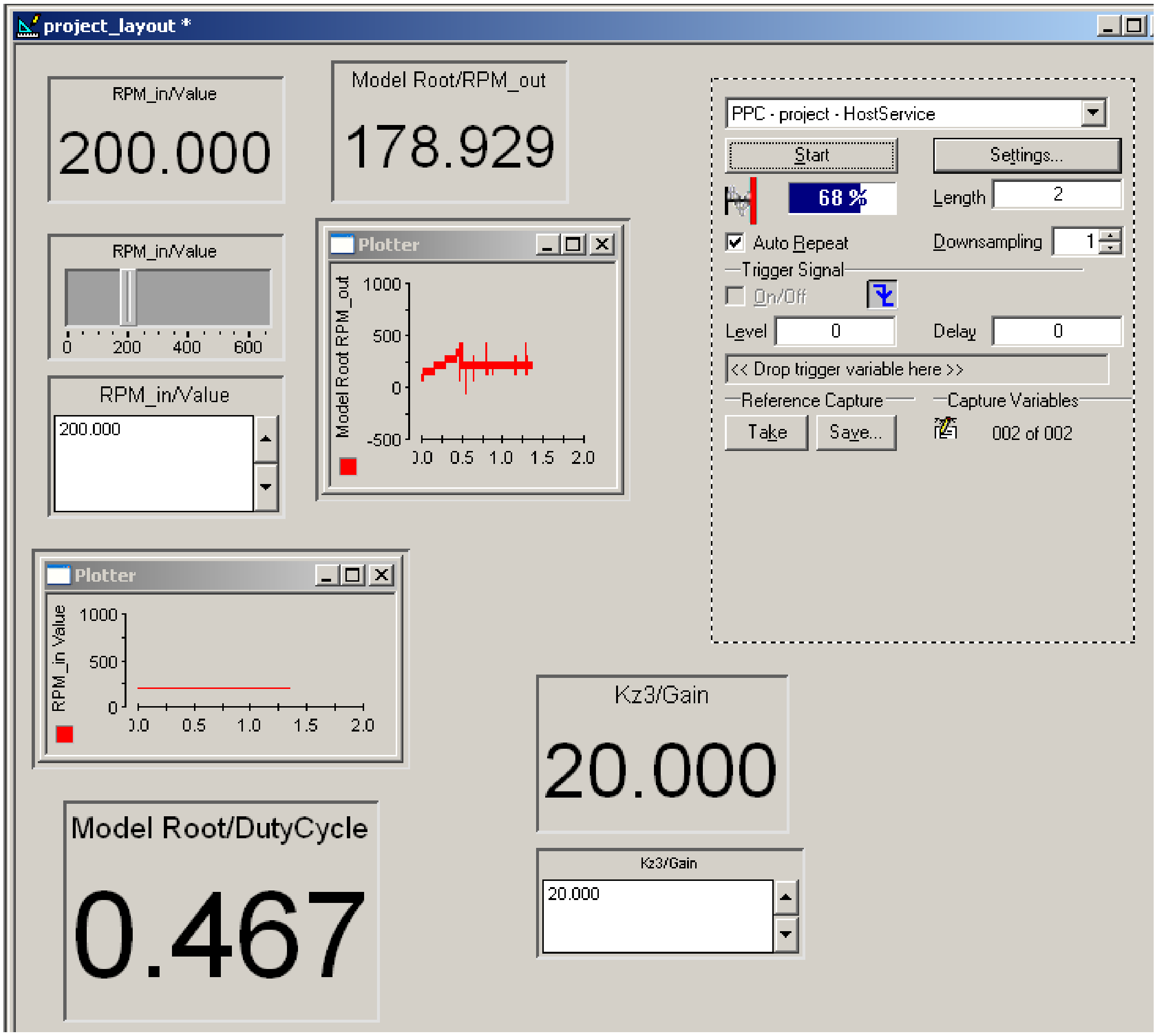
Figure 72: ControlDesk layout with CaptureSettings instrument added and the system running. (Step response for 200 RPM observed using NumericInput to change RPM_in from 0 to 200 while system was running.)

28. Capture data and Plot it in MATLAB. (This is also explained in the "CaptureSettings Instrument" subsection mentioned in Step 27 above.)

29. Now, for another cool feature of ControlDesk. The value of any gain or constant block in the Simulink model can be changed without rebuilding and downloading code and while the system is running. (There might be other blocks that work as well.) Use the instructions in the "Monitoring/Controlling/Recording Values" section of this tutorial to connect "Kz3" (Gain) to a Display and a NumericInput instrument in the ControlDesk Layout.

30. Apply 10 Volts to the brake. Set RPM_in to "0". Start the CaptureSettings instrument. Quickly change RPM_in to "200". Observe RPM_out on its PlotterArray, and quickly Stop the CaptureSettings instrument before the 0 to 200 transition disappears. The data can be plotted in MATLAB if desired.

31. Set "Kz3" to "20" and repeat Step 30 above. Compare the results. (There should be a visible spike/overshoot in RPM_out for Kz3 = 20 that is not in the Kz3 = 7 RPM_out output.)



project_layout *

RPM_in/Value: 200.000

Model Root/RPM_out: 178.929

RPM_in/Value: 200.000

PPC - project - HostService

68 %

Length: 2

☑ Auto Repeat

Downsampling: 1

□ On/Off

Level: 0

Delay: 0

002 of 002

Kz3/Gain: 20.000

Kz3/Gain: 20.000

Model Root/DutyCycle: 0.467

Figure 73: ControlDesk layout with Kz3 instruments added and system running with RPM_out response to RPM_in=200, Kz3=20, and brake voltage=10V shown.

32. Set the brake voltage to 0 Volts and "Kz3" back to "7" The tutorial is completed.

References

[1] “DS1103 PPC Controller Board”, Germany: dSPACE, July 2008.
[2] Annemarie Thomas. "dSPACE DS1103 Control Workstation Tutorial and DC Motor Speed Control: Project Report", Senior Project Report, Bradley University ECE Department, May 2009.
[3] “Connector and LED Panels,” *Catalog 2008*, Germany: dSPACE GmbH, 2008, p. 302.
[4] *ControlDesk Experiment Guide For ControlDesk 3.2*, Germany: dSPACE GmbH, 2008, Release 6.1.
[5] *dSPACE System First Work Steps For DS1103, DS1104, DS1005, DS1006, and Micro Auto Box*, Germany: dSPACE GmbH, 2007, Release 6.0.
[6] *Real-Time Interface (RTI and RTI-MP) Implementation Guide*, Germany: dSPACE GmbH, 2008, Release 6.1.
[7] *DS1103 PPC Controller Board Hardware Installation and Configuration*, Germany: dSPACE GmbH, 2007, Release 6.0.