Matlab LinCon-8000 User Manual
Version 1.0

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assume no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

Table of Contents

1 Introduction

The LinCon-8000 embedded controller has been integrated into one of the new generation MATLAB Embedded Controller solution of ICPDAS. Therefore **Matlab-LinCon** is a combination of **LinCon-8000 embedded controller** with **MATLAB/Simulink/**. Its purpose is to achieve "**Development of Rapid Controller Platform**" and it will help the engineers grealy to develop their own controller. In the LinCon-8000, it provides VGA、PS/2、USB、Ethernet、Series ports 、CF Card and I/O expansion slots for Matlab development environment. For this application, there have been over 20 I/O bridges and system-level Simulink Blocks developed and built in Matlab/Simulink Blockset. By using Simulink development environment and these Matlab Driver's I/O blocks, control algorithm can be easily constructed and verified. Once the algorithm has been verified, users just click a "build" button, and the simulink/model will be converted to executable file. **The process from model to executable file is without writing any code**. Then download the executable file to the LInCon-8000 embedded controller for test or practical application via Ethernet TCP/IP communication. It also provides users to upload the experimental data for further analysis to improve the result. All in all, engineers can put more focus on advanced control algorithm design and development.

1.1. Hardware Specifications for LinCon-8000

[General environment]

Operating temperature: -25°C to +75°C
Storage temperature: -30°C to +85°C
Humidity: 5 ~ 95%
Built-in power protection & network protection circuit

[System Specification]

CPU：Intel StrongARM 206MHz
SDRAM：64M bytes
FLASH ROM：32M bytes
EEPROM：16byte
COM2=RS-232, COM3=RS-485
Ethernet：10 BaseT
Real-Time Clock
Built-in 64-bit hardware unique serial number
Built-in Watchdog Timer

Built-in VGA, PS/2, USB port

Built-in CF Card (128M bytes)

I/O Expansion Slot：0-slot for L-8038 ； 3-slot for L-8338 ； 7-slot for L-8738

1.2. Features

The MATLAB driver for LinCon-8000 embedded controller combines with MATLAB/Simulink/Stateflow perfectly and easily. With the Matlab/Simulink library support, many extended powerful blocks of the I-8000 series modules including AO, AI, DO, DIO, DI and DataToFile function etc. have been built in. Therefore the sophisticated tasks of creating, analyzing, and simulating block diagram models can all be solved with MATLAB/Simulink/Stateflow conveniently.

The features of Matlab-LinCon are described as follows：

1. **Ethernet Communication Interfaces:** The Matlab-LinCon solution provides the Ethernet Communication Interface to let users download their programs and upload the experimental data quickly and stably..

2. **Easy-to-use platform:** The Matlab-LinCon solution provides a friendly communication GUI interface which enables users to communicate with LinCon-8000 target hardware conveniently.

3. **Reduce the design cycle of your product:** This Matlab-LinCon solution combines LinCon-8000 with MATLAB/Simulink/Stateflow which provides users the “model-based” control design approach and this approach is a timesaving and cost-effective approach. Therefore engineers can work with a single model of a function or a complete system in an integrated software environment.

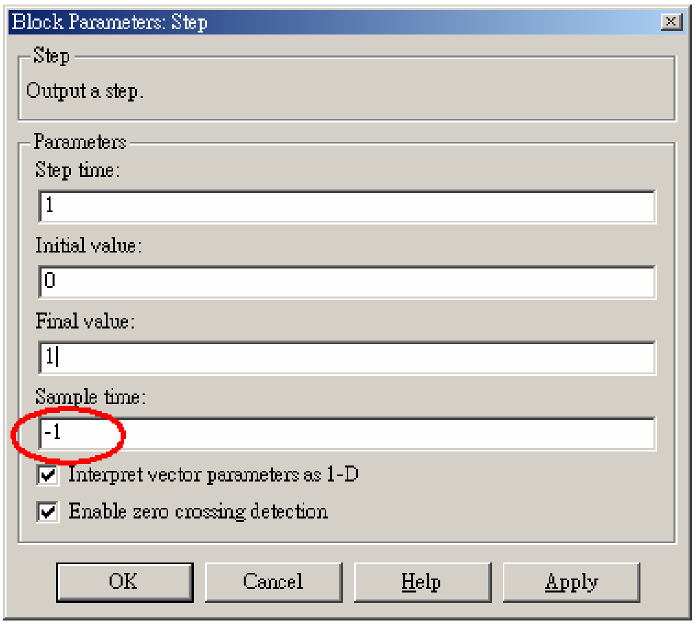
4. **The extensibility of I/O driver blocks:** This Matlab-LinCon solution is based on the LinCon-8000 controller which provides 3 or 7 expansion slots. Currently, Matlab-LinCon driver provides over 20 I/O blocks to cooperate with Matlab development environment, which including DI, DO, DIO, AI, AO, and Relay blocks. More I/O blocks will be supported in the future.

5. **The Open Source Code：**
   The source code produced by Matlab RTW is open for users.

1.3. Limitations

The MATLAB-LinCon Driver only supports **single tasking** and **Fixed-step** modes, due to the limitations of the RTW Embedded Coder.

1. **Single tasking:** In Simulink, *single tasking* means that only one sample rate can be used in the whole control system. That is, every block must have the same sampling rate. It is suggested that users set the *Sample time* to be " **–1** " when the option is available in the block.



Block Parameters: Step

Step

Output a step.

Parameters

Step time: 1

Initial value: 0

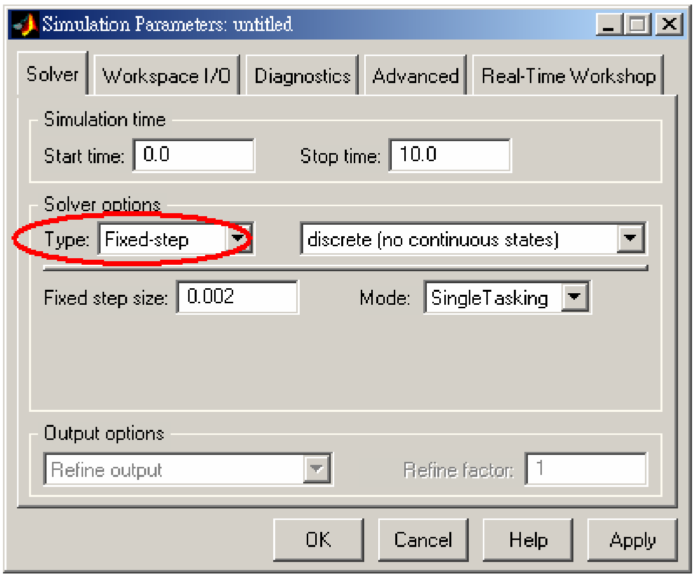
Final value: 1

Sample time: -1

☑ Interpret vector parameters as 1-D

☑ Enable zero crossing detection

2. **Fixed-step:** Because the RTW 4.x or 5.0 have not supported variable step time, the *Solver options* on the *Simulation Parameters* dialog box can only be set to *Fixed-step*.



Simulation Parameters: untitled

Simulation time

Start time: 0.0 Stop time: 10.0

Solver options

Type: Fixed-step discrete (no continuous states)

Fixed step size: 0.002 Mode: SingleTasking

Output options

Refine output Refine factor: 1

Furthermore, the **RTW Embedded Coder** does not support the following built-in Simulink blocks yet:

1. **Simulink\Continuous**
    - No blocks in this library are supported

2. **Simulink\Discrete**
    - First-Order Hold

3. **Simulink\Function and Tables**
    - MATLAB Fcn

4. **Simulink\Math**
    - Algebraic Constraint
    - Matrix Gain

5. **Simulink\Nonlinear**
    - Rate Limiter

6. **Simulink\Signals & System**
    - Bus Selector
    - IC

7. **Simulink\Sinks**
    - XY Graph
    - Display
    - To File

8. **Simulink\Sources**
    - Clock
    - Chirp Signal
    - Pulse Generator
    - Ramp
    - Repeating Sequence
    - Signal Generator

1.4. Module list supported for Matlab-LinCon Driver

The following table is a list of currently supported I-8000 I/O modules by Matlab-LinCon Driver. They include：DI, DO, DIO, AI, AO, Relay and Encoder modules. If users want to get more information about the specification or function of these modules, please visit the website (http://www.icpdas.com/products/PAC/lincon-8000/matlab_lincon.htm) to obtain the current driver situation and download the related documents and latest driver.

Type	Description	Module Model
DI	Digital input module	I-8040, I-8051, I-8052,I-8053, I-8058
DO	Digital output module	I-8041, I-8056, I-8057
DIO	Digital input & output module	I-8042, I-8054, I-8055, I-8063
AI	Analog input module	I-8017H
AO	Analog output module	I-8024
Relay	Relay output module	I-8060, I-8064
Encoder	Encoder counter module	I-8090

2. Software Installation

Before installing the MATLAB-LinCon Driver, users need to check：

(1) If Matlab v6.5 software development system installed in the operation system.

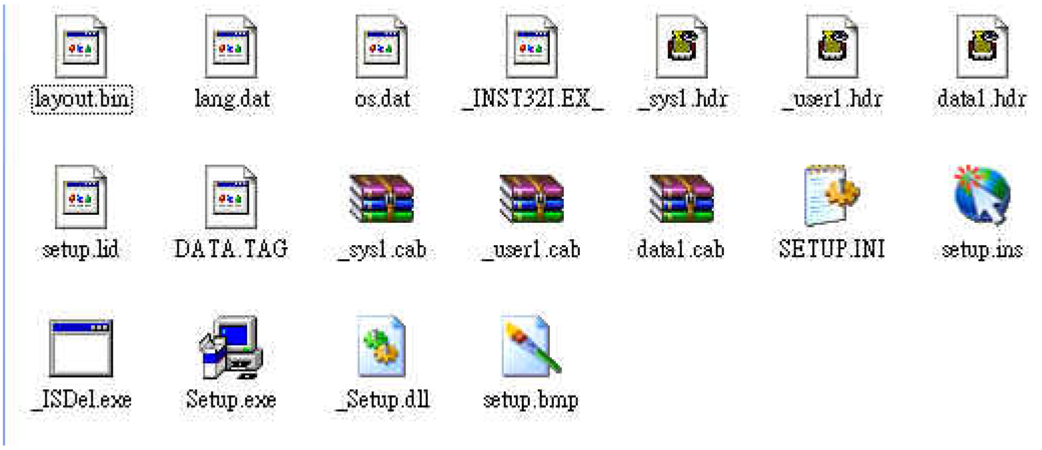
(2) The requirement toolbox of the MATLAB software for Matlab-LinCon solution is listed as follows:

- **MATLAB 6.5 is installed**.
- **Simulink 5.0 installed**.
- **Real-Time Workshop 5.0 installed**.
- **Real-Time Workshop Embedded Coder 3.0 installed**.
- **Stateflow and Stateflow Coder 5.0** (it is optional if users needs the function.)

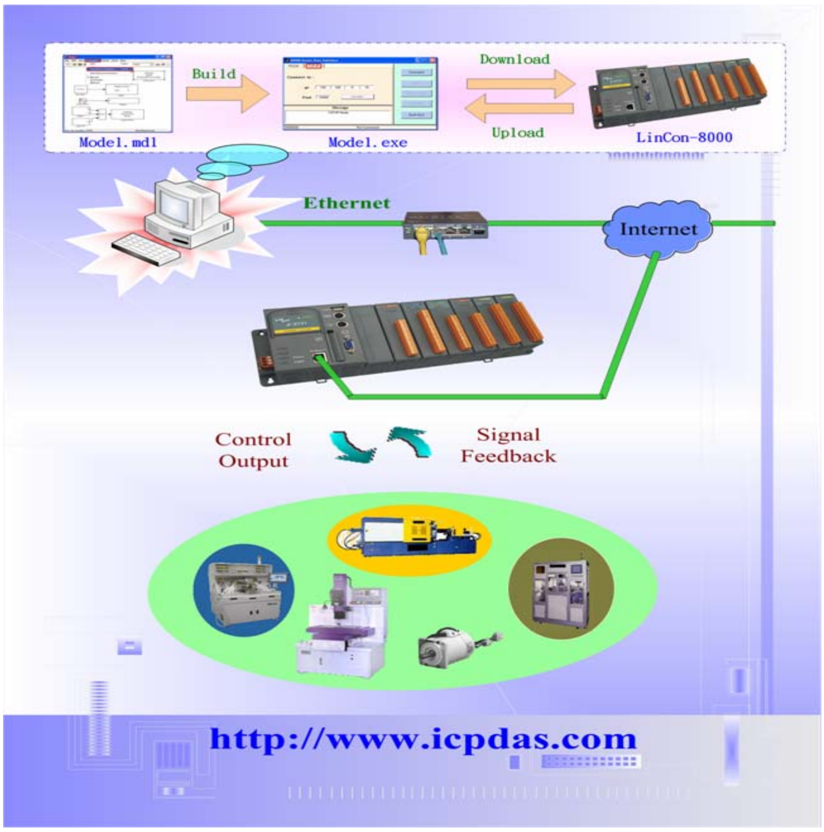
Note that current solution is only tested in the Window system, which is Win98/2000/XP. In the following sections, we will demonstrate the installation procedures step by step to make everything registered correctly.

2.1. Driver Installation

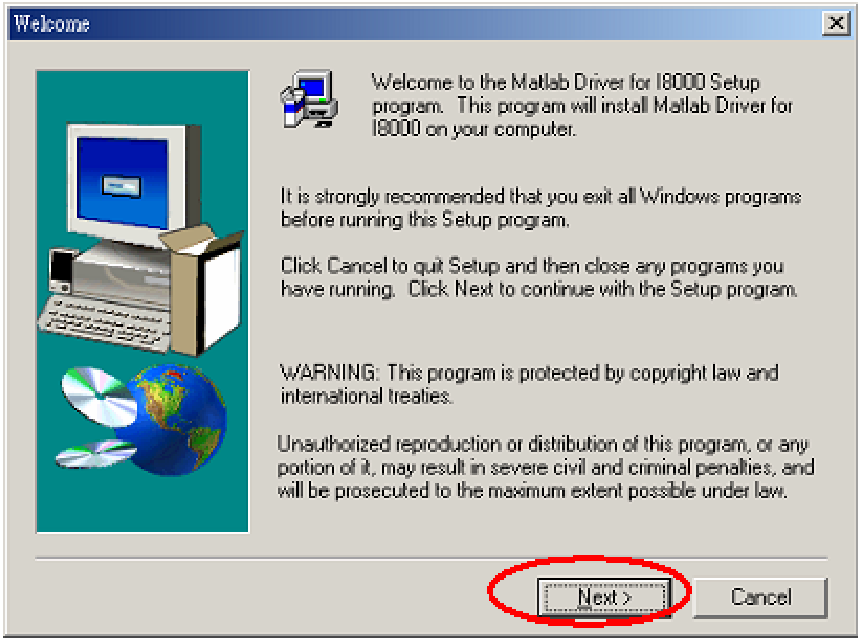
Step 1: Insert the LinCon-8000 CD and enter the Matlab-LinCon Driver directory ： **\ Matlab_LinCon \ Driver \ Ver1.0 ** or in the web path: http://www.icpdas.com/products/PAC/lincon-8000/download.htm can be downloaded. The files listed should look like the figure below. Double click "*Setup.exe*" to start the installation procedure.



Step 2: An ICPDAS MATLAB-LinCon solution logo will appear and then click "*Next*" button to continue.



ICP DAS MATLAB solution logo



Welcome to the Matlab Driver for I8000 Setup program. This program will install Matlab Driver for I8000 on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

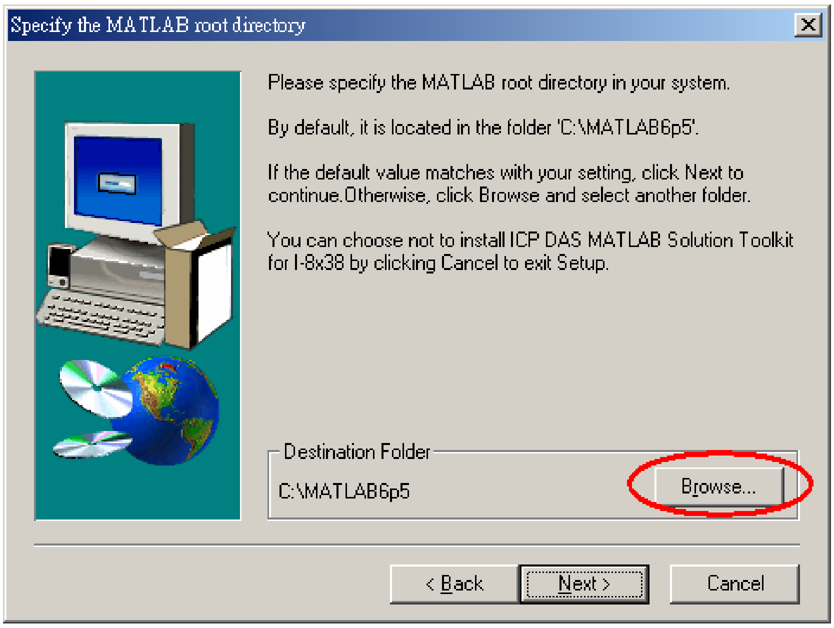
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

The “Welcome” dialog box

Step 3: The Installation directory for MATLAB dialog box shows the current directory information that MATLAB software has been installed, as shown in the figure below. The default MATALB folder is "**C:\Matlab6p5**". **If MATALB 6.5 was not installed on the default path, users must click the Browse button to choose the correct path. Otherwise, the MATLAB Solution Toolkit will not work properly.** Press "*Next*" to the next step.



Please specify the MATLAB root directory in your system.

By default, it is located in the folder 'C:\MATLAB6p5'.

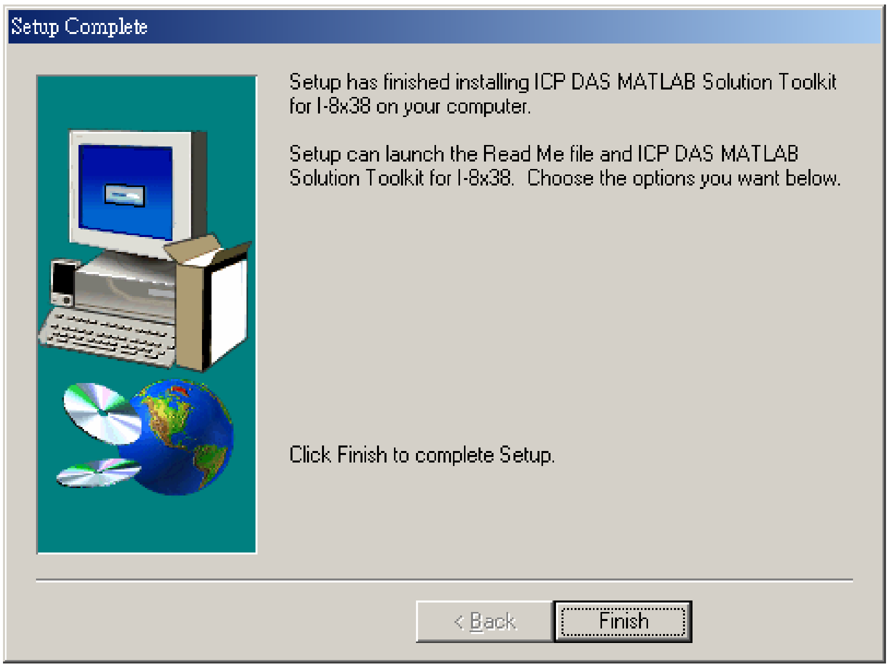
If the default value matches with your setting, click Next to continue.Otherwise, click Browse and select another folder.

You can choose not to install ICP DAS MATLAB Solution Toolkit for I-8x38 by clicking Cancel to exit Setup.

Destination Folder: C:\MATLAB6p5

Specify the MATLAB root directory

Step 4: After copying files has been completed successfully, the Setup Complete dialog box appears. Click "*Finish*" to exit the installation program.

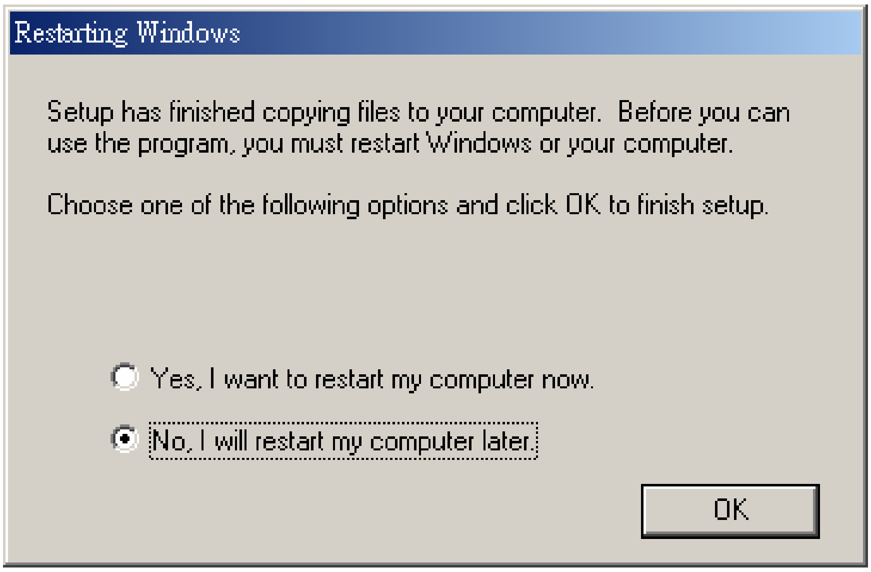


Setup has finished installing ICP DAS MATLAB Solution Toolkit for I-8x38 on your computer.

Setup can launch the Read Me file and ICP DAS MATLAB Solution Toolkit for I-8x38. Choose the options you want below.

Click Finish to complete Setup.

Step 5: After the installation has been completed, **Please reboot your system to let the settings take effect.**


Restarting Windows

Setup has finished copying files to your computer. Before you can use the program, you must restart Windows or your computer.

Choose one of the following options and click OK to finish setup.

○ Yes, I want to restart my computer now.

● No, I will restart my computer later.

Reboot your system

Step 6: After reboot, **"C:/7188/MATLAB"** (default) directory will be established automatically and the necessary files are placed in it.

[Note]

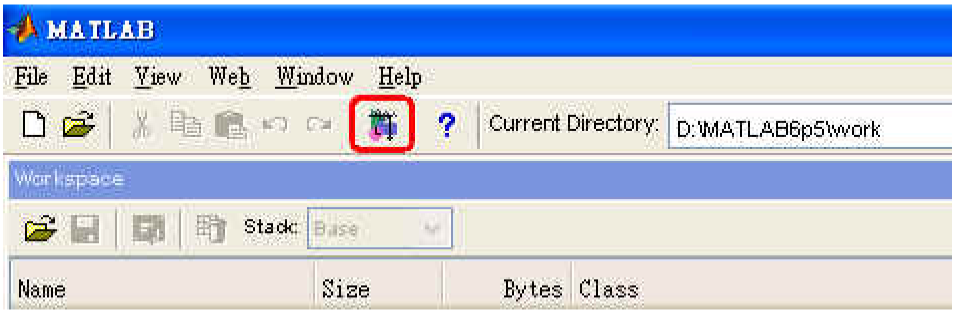
1. Users can also download the latest Matlab_LinCon_Driver for Matlab v6.5 from the web path : http://www.icpdas.com/PAC/lincon-8000/download.htm.

3. How to work with MATLAB/Simulink

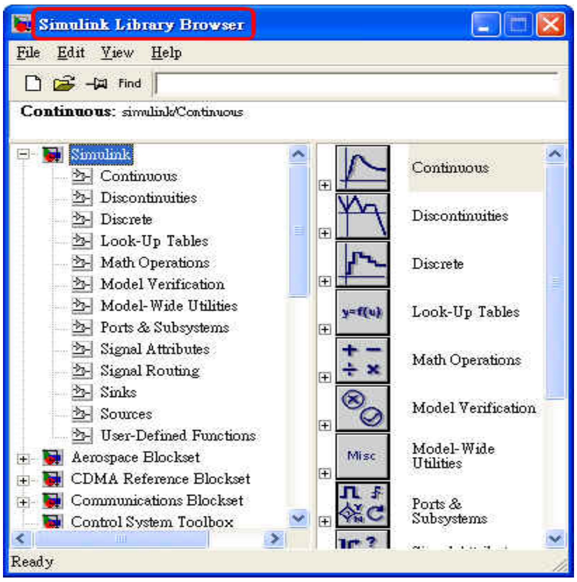
This chapter gives an example to demonstrate how to construct your control model with Matlab-LinCon driver blocks by using MATLAB/Simulink. In the meanwhile, users will learn how to build the model into an executable file by employing RTW and RTW Embedded Coder. In the last two sections of this chapter, we will also show you how to cooperate the Stateflow Coder with Matlab-LinCon driver blockset. After you completely follow this chapter instruction, it is expected that you can construct your own model, and build it into an executable file and download it to the LinCon-8000 target system for application. Finally you can easily start your own project and upload the data for further analysis.

3.1. Create a control model using Simulink

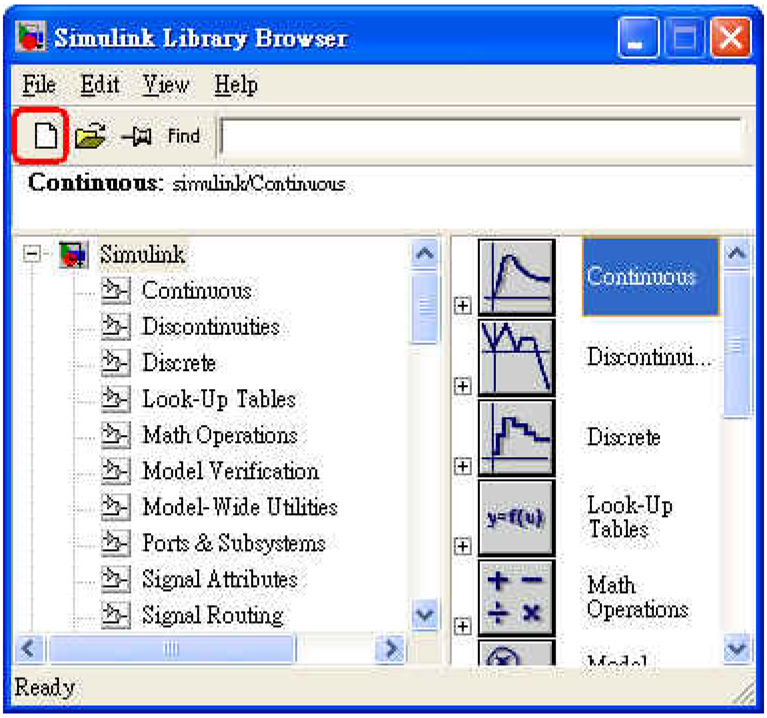
To start Simulink, you must first start MATLAB. And then you can start Simulink by clicking the Simulink icon on the MATLAB toolbar.

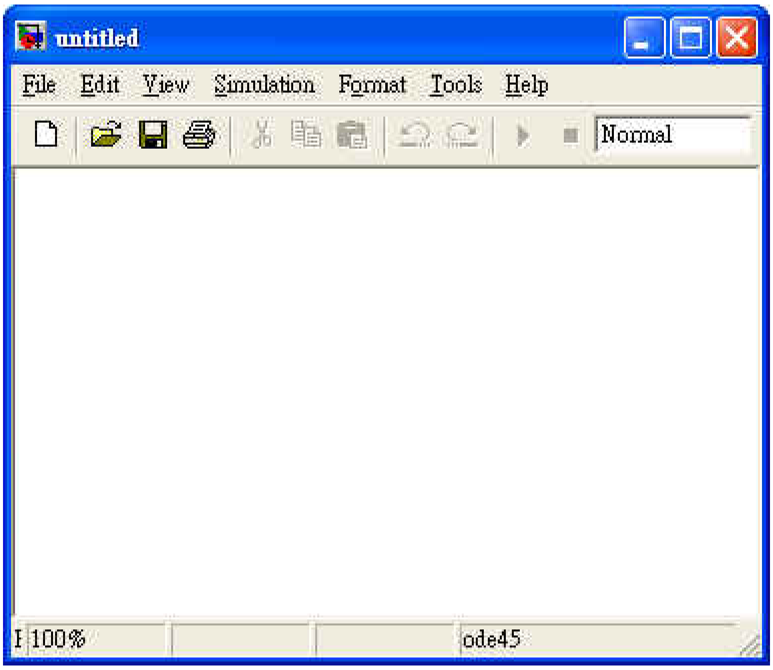

MATLAB
File Edit View Web Window Help
Current Directory: D:\MATLAB6p5\work
Workspace
Stack: Base
Name
Size
Bytes
Class

On Microsoft Windows platform, starting Simulink displays the Simulink Library Browser window as shown below.

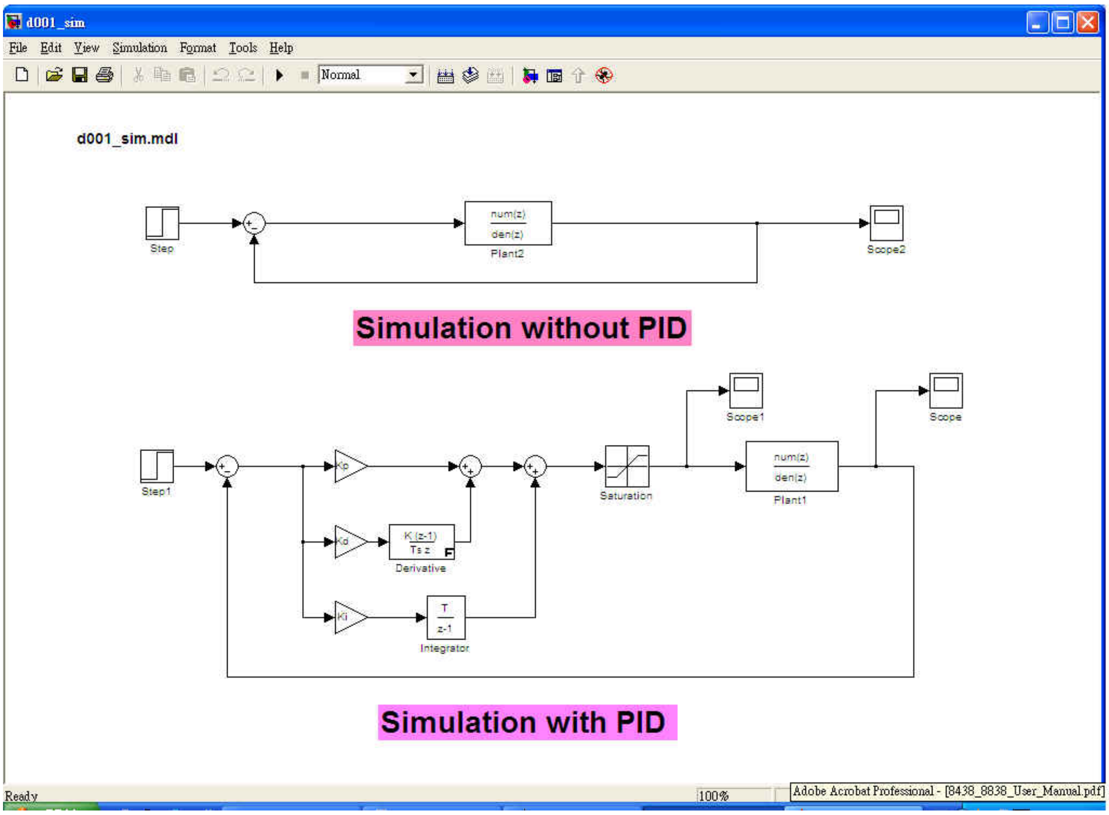


Then click 🗋 on the Library Browser's toolbar (Windows only). Simulink opens a new model window, and then you can start to construct your own control model on the blank area of the model window.



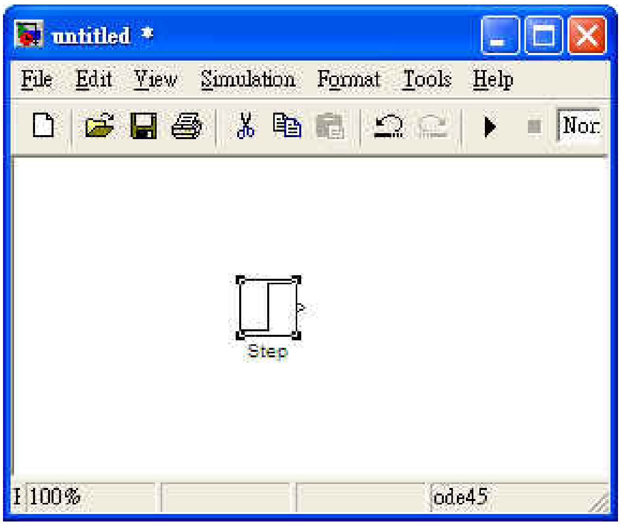


Here, we offer a simple PID controller model to you, and show you how to create a model. The completed block diagram of the model looks like the figure below.

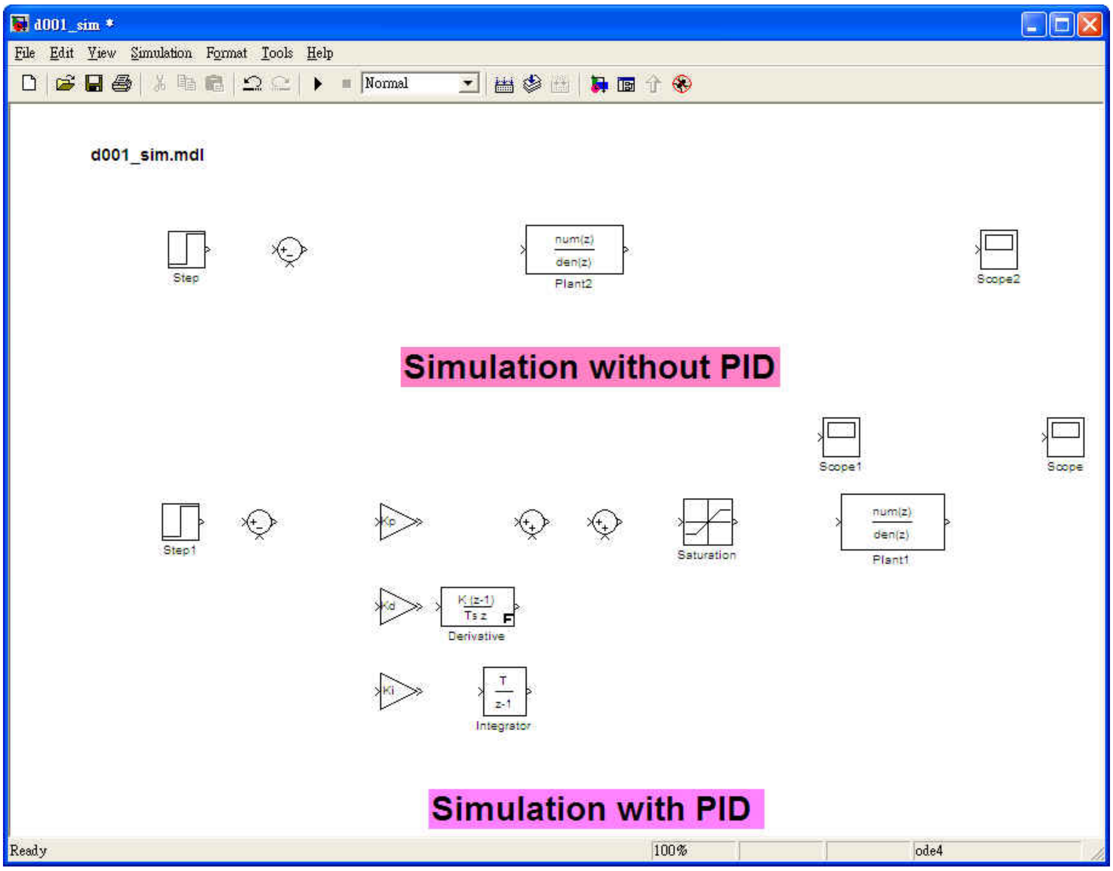


To create this model, you need to copy blocks to the model from the Simulink block libraries. For example, to copy a *Step* block to the model window, follow these steps:

1. First expands the Library Browser tree to display the blocks in the ***Sources library***. Do this by clicking the *Sources* node to display the *Sources* library blocks.
2. Choose the *Step* block. Then drag the *Step* block from the browser and drop it to the model. Now the model window should look like the figure below.



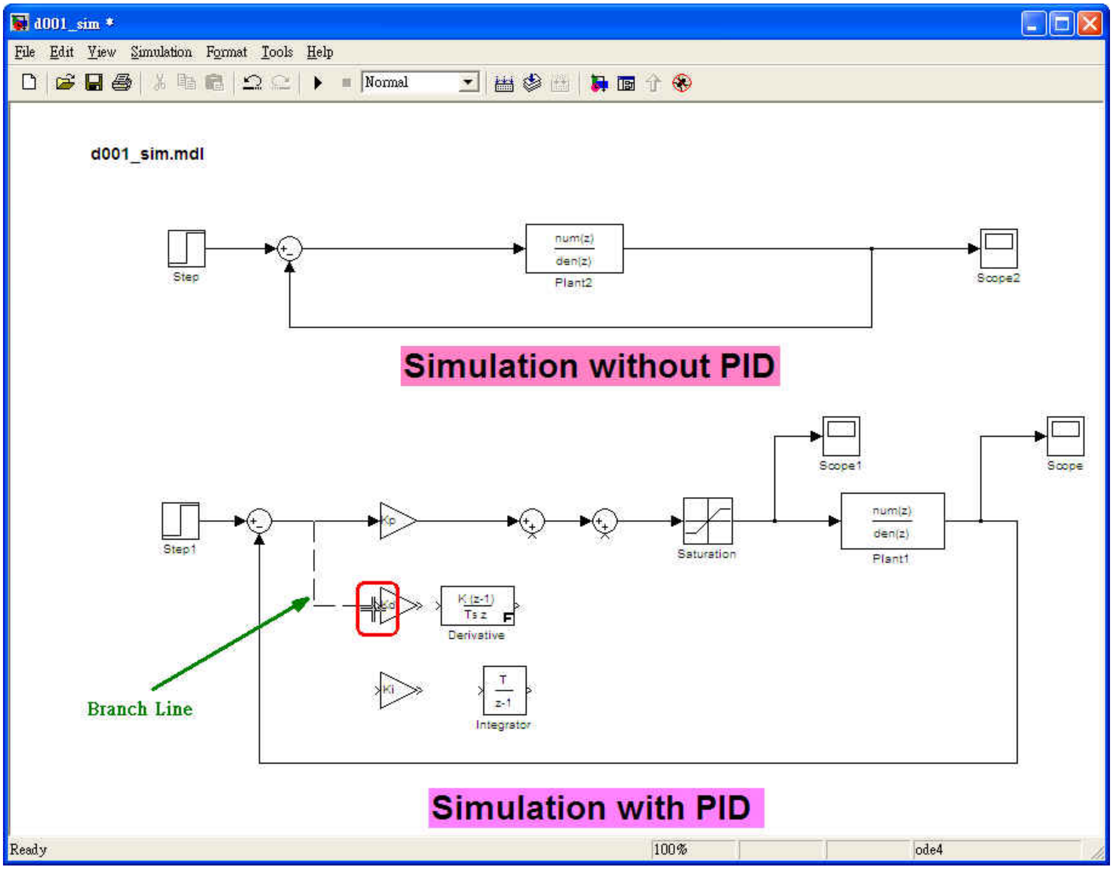
Copy the rest of the blocks in a similar method from other libraries to the model. After all the blocks are copied to the model, it should look like the figure below.



It's time to connect these blocks. To connect one block with another, just select one block, and position the mouse cursor over the output port of the block. Notice that the cursor shape changes to crosshairs. Click the left mouse button and move the cursor to the input port of another block and then the blocks are connected.

However, we also need to connect one existed line to the input port of another block. This line is called ***branch line***. Drawing a branch, please follow these steps as follows：

1. First, put the cursor on the existing line.
2. Click the right mouse button and then drag the cursor to the input port of another block.
3. Release the mouse button when the cursor shape changes to double-lined crosshairs as shown in the figure below.

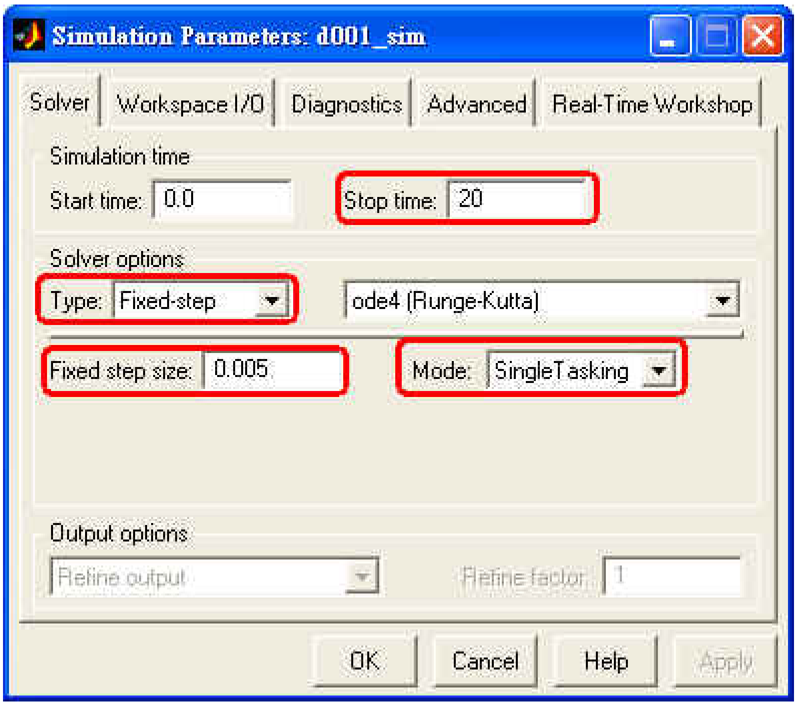


After finish the connection of these blocks, the model should look like the model at the beginning of this section.

Now we need to set the simulation parameters by choosing *Simulation parameters* from the *Simulation* menu. Set the options as follows:

- Stop time：***20***
- Solver options：***Fixed-step***
- Fixed step size：***0.005***
- Mode：***Single Tasking***

Here, we set Solver options to **Fixed-step** and Mode to **Single Tasking**. The reason is that the RTW Embedded Coder does not support variable step time yet. To compare the result of simulation with experiment fairly, the setting is suggested.



Simulation Parameters: d001_sim

Solver | Workspace I/O | Diagnostics | Advanced | Real-Time Workshop

Simulation time

Start time: 0.0

Stop time: 20

Solver options

Type: Fixed-step

ode4 (Runge-Kutta)

Fixed step size: 0.005

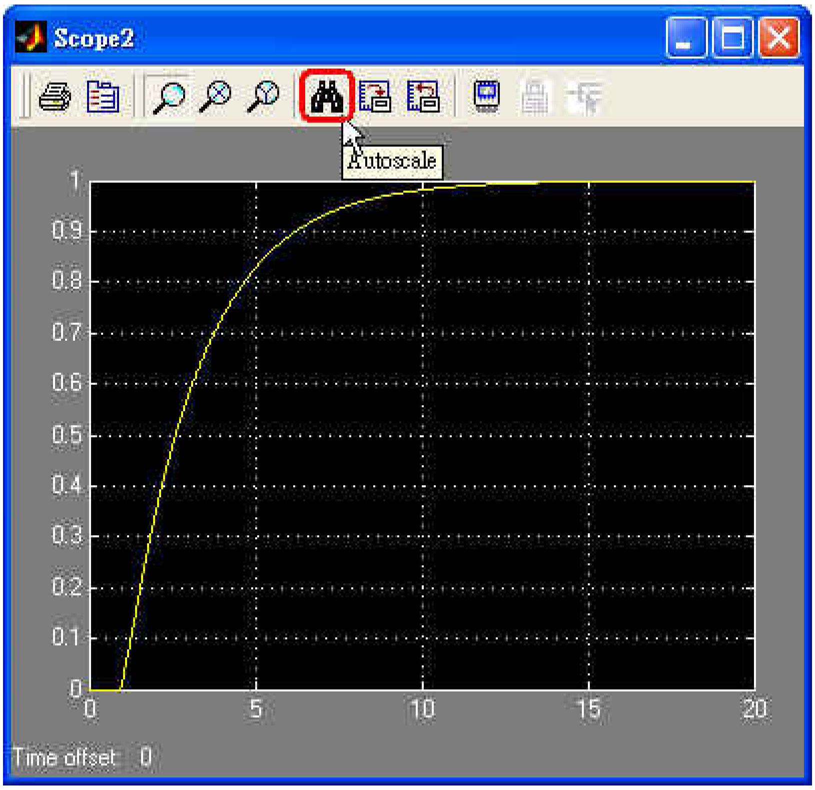
Mode: SingleTasking

Output options

Refine output

Refine factor: 1

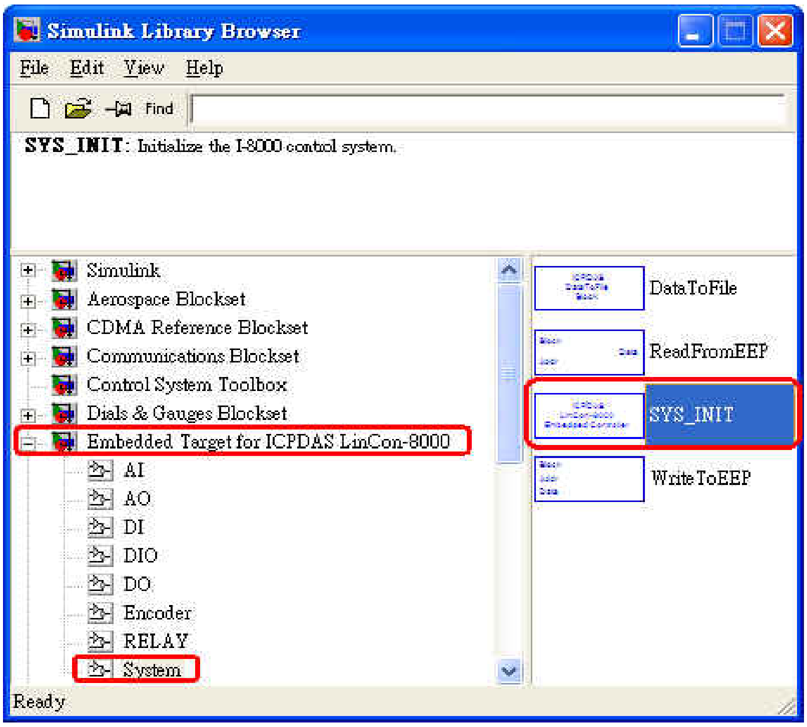
Click the “OK” button to set the options. To start the simuloation of the model by clicking the “***Start simulation***” button ▸ on the toolbar of the model window. Then the result of simulation can be watched by double clicking the scope block. Press the “**Autoscale**” button in the “Scope” window and then it will show the appropriate scale. (Refer to the figure below)



3.2. Simulink model combines ICPDAS driver

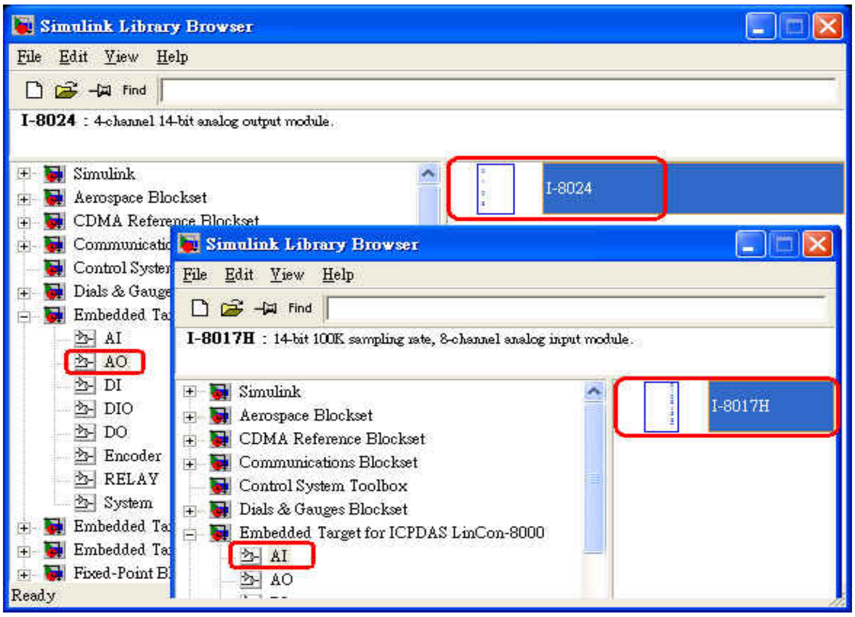
If the result of simulation is satisfied, users need to replace the “plant block” with “**I-8000 modules driver blocks of ICPDAS**”. To add I-8000 driver blocks to the model, please follow these steps：

Step 1: Add the “**SYS_INIT**” block from the System block library.



SYS_INIT: Initialize the I-8000 control system.

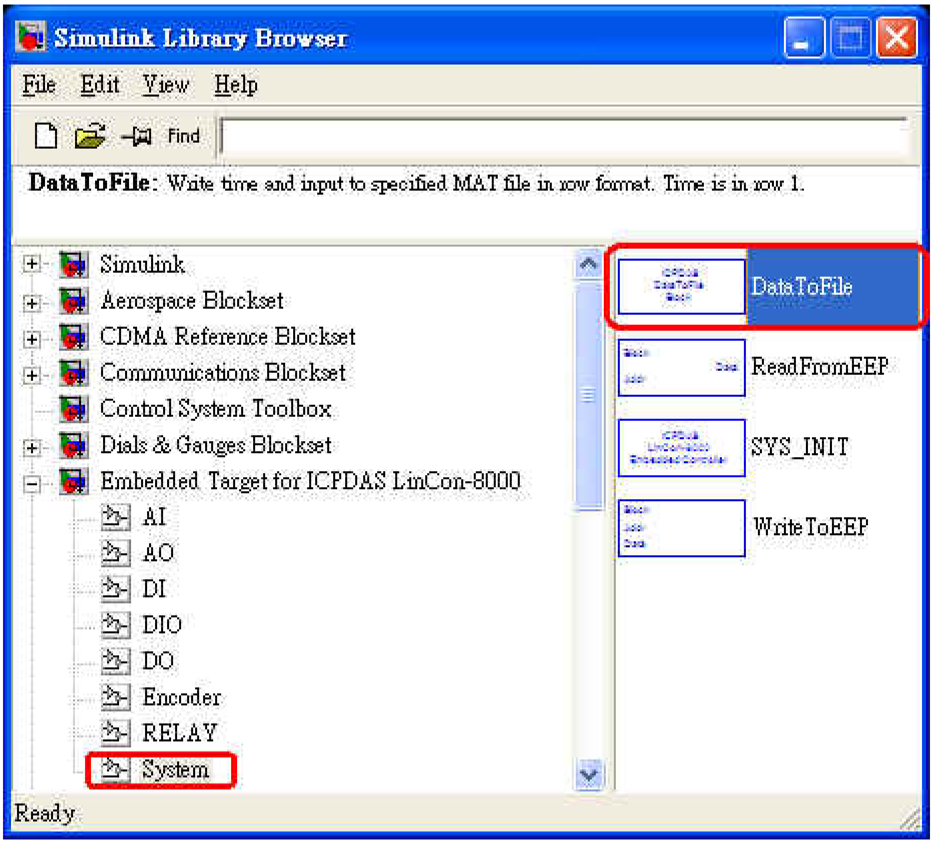
Step 2: Remove the Plant subsystem block and copy an **I-8024** and an **I-8017H** block from the AI and AO library respectively.



I-8024 : 4-channel 14-bit analog output module.

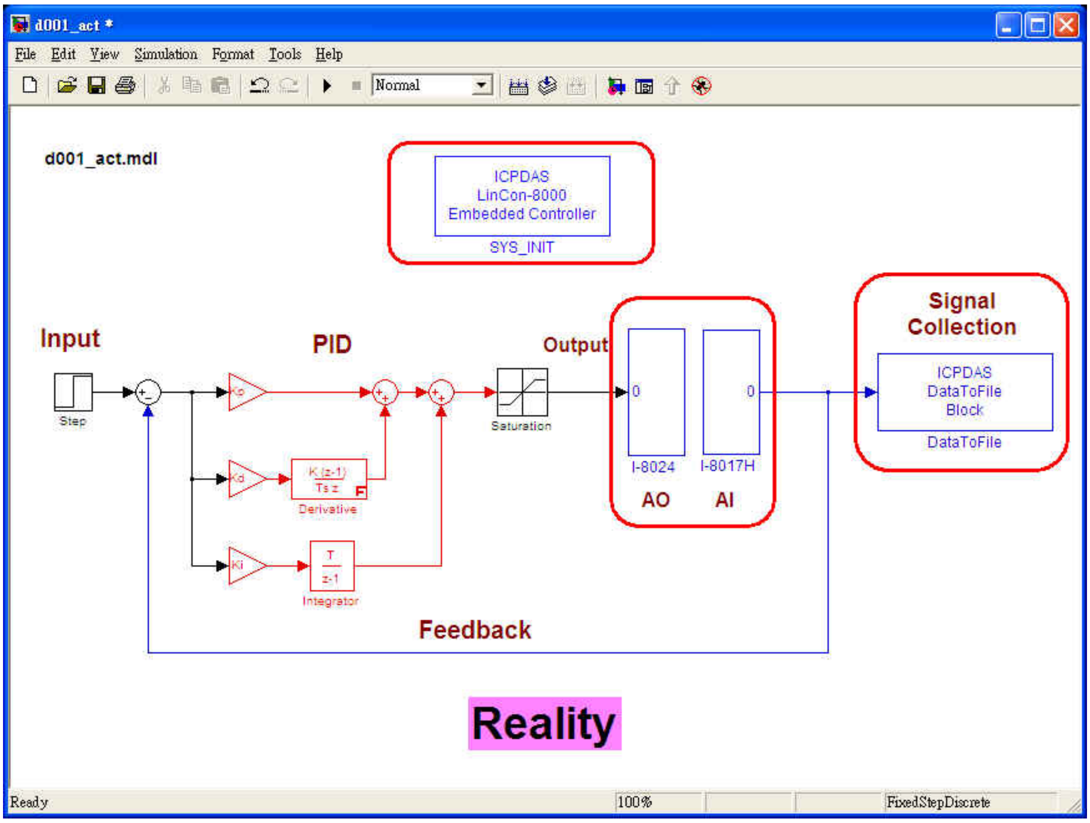
I-8017H : 14-bit 100K sampling rate, 8-channel analog input module.

Step 3: Replace the “Scope” block with the “**DataToFile**” driver block.



DataToFile: Write time and input to specified MAT file in row format. Time is in row 1.

Step 4: Connect the blocks as shown in the following figure.

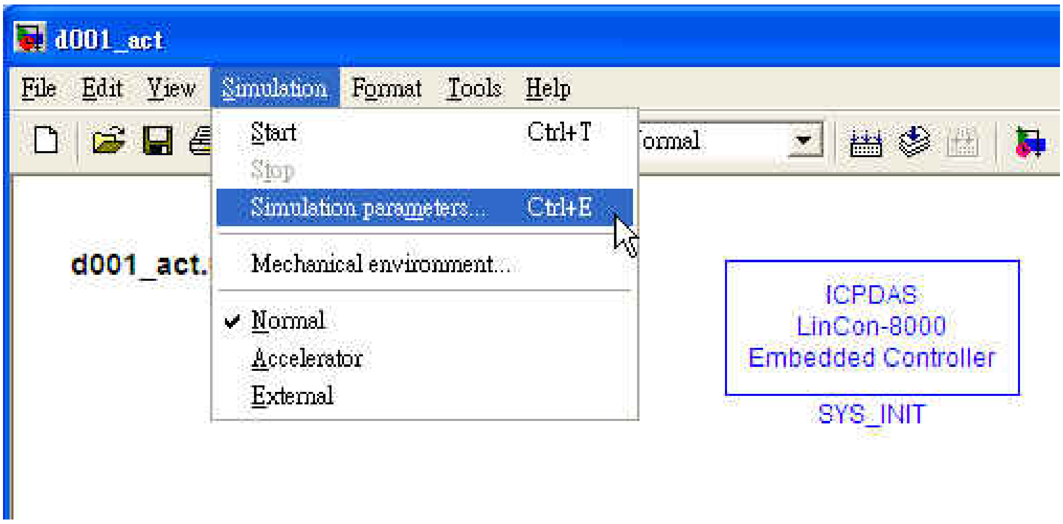


After the above steps are done, the model with the ICPDAS I-8000 driver blocks is ready for built into the executable file.

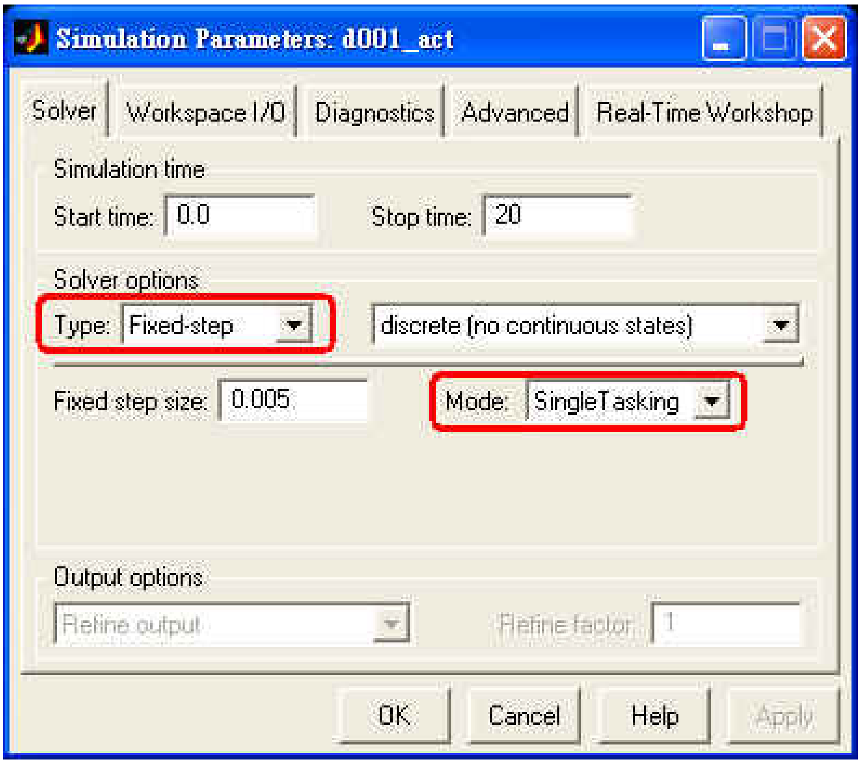
3.3. Build the Model to Executable file by RTW

In this section, we will introduce how to convert the **model** created in the previous section into an **.exe** file by RTW. Please follow these steps：

Step 1: Click the “Simulation / Simulation parameters”



Step 2: On the dialog box that displays, select Type as *Fixed-step*, Mode as *Single Tasking* in the “Solver options” field.



Simulation time

Start time: 0.0 Stop time: 20

Solver options

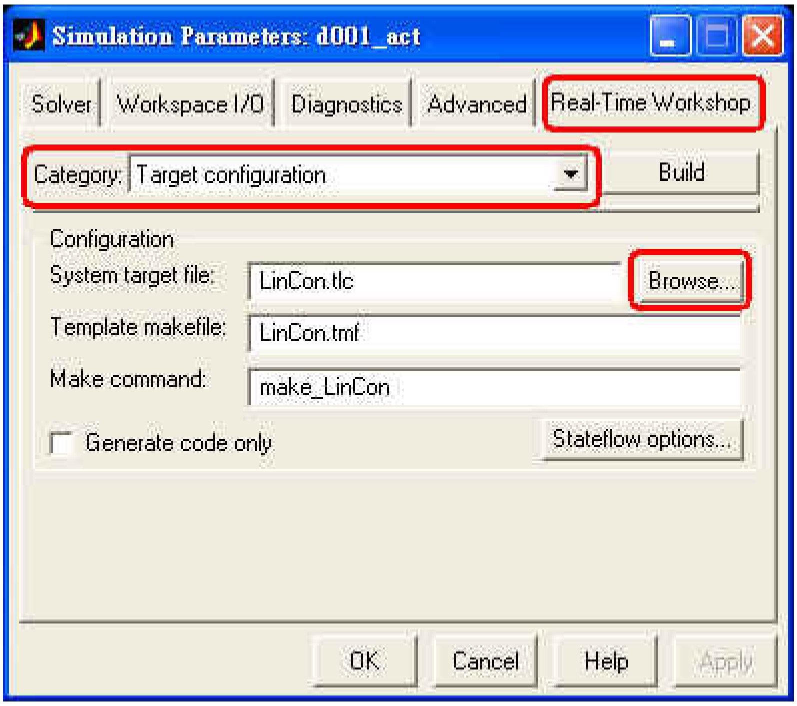
Type: Fixed-step discrete (no continuous states)

Fixed step size: 0.005 Mode: SingleTasking

Output options

Refine output Refine factor: 1

Step 3: Then click the "**Real-Time Workshop**" tab and select "**Target configuration**" item from "Category" field. Then click the "**Browse**" button to open the "System Target File Browser" window.



Category: Target configuration

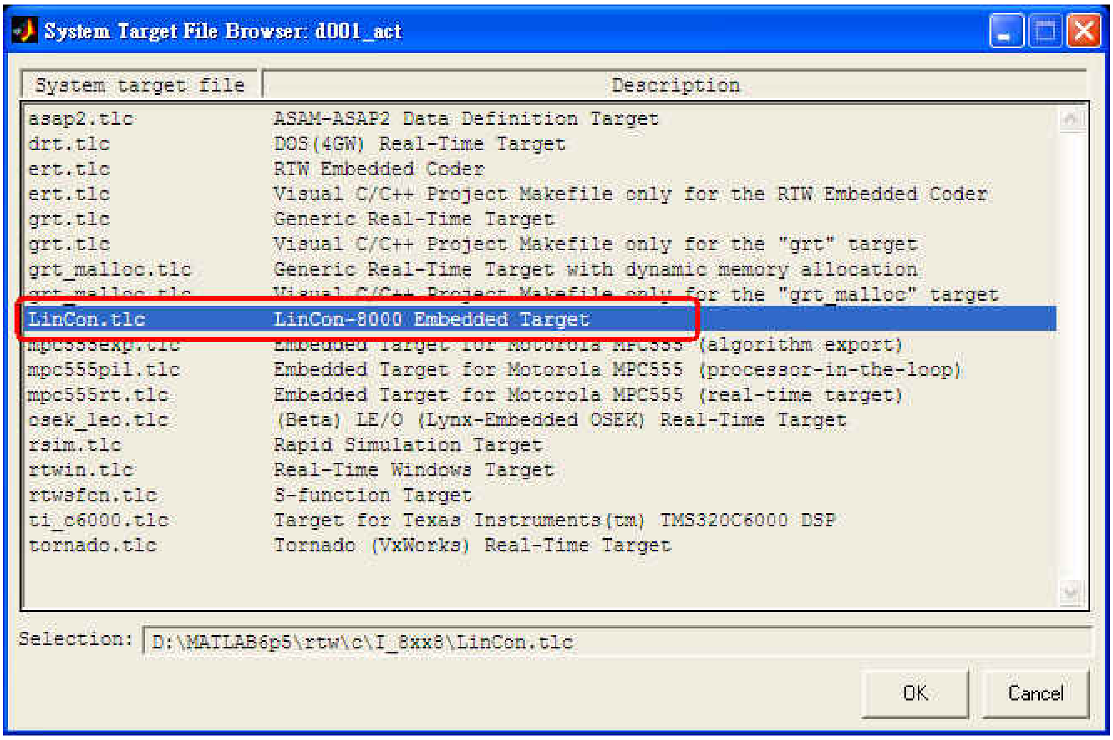
System target file: LinCon.tlc

Template makefile: LinCon.tmf

Make command: make_LinCon

□ Generate code only

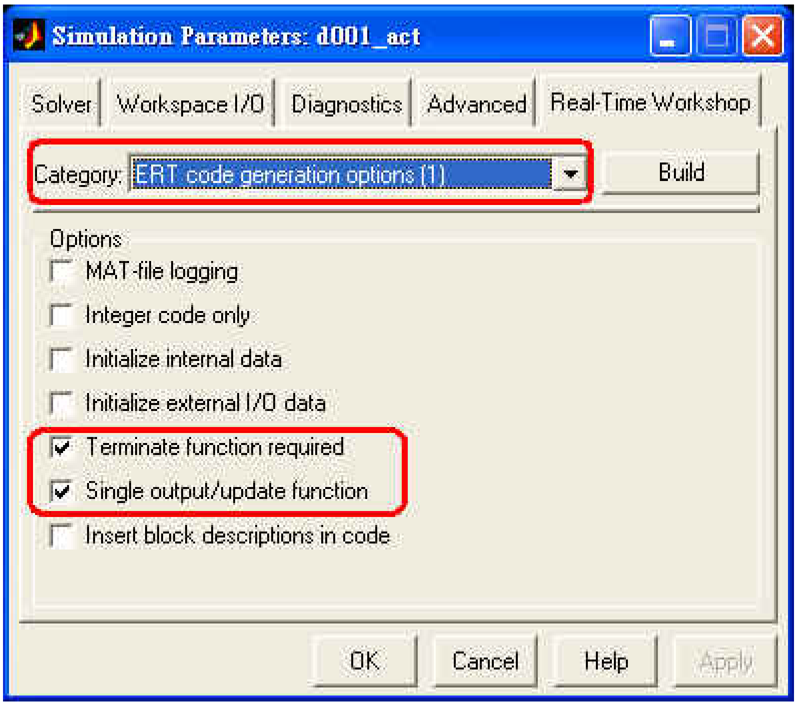
Step 4: On the "**System Target File Browser**" dialog, select the "**LinCon-8000 Embedded Target**" option and then click the OK button.



System target file	Description
asap2.tlc	ASAM-ASAP2 Data Definition Target
drt.tlc	DOS(4GW) Real-Time Target
ert.tlc	RTW Embedded Coder
ert.tlc	Visual C/C++ Project Makefile only for the RTW Embedded Coder
grt.tlc	Generic Real-Time Target
grt.tlc	Visual C/C++ Project Makefile only for the "grt" target
grt_malloc.tlc	Generic Real-Time Target with dynamic memory allocation
grt_malloc.tlc	Visual C/C++ Project Makefile only for the "grt_malloc" target
LinCon.tlc	LinCon-8000 Embedded Target
mpc555exp.tlc	Embedded Target for Motorola MPC555 (algorithm export)
mpc555pil.tlc	Embedded Target for Motorola MPC555 (processor-in-the-loop)
mpc555rt.tlc	Embedded Target for Motorola MPC555 (real-time target)
osek_leo.tlc	(Beta) LE/O (Lynx-Embedded OSEK) Real-Time Target
rsim.tlc	Rapid Simulation Target
rtwin.tlc	Real-Time Windows Target
rtwsfcn.tlc	S-function Target
ti_c6000.tlc	Target for Texas Instruments(tm) TMS320C6000 DSP
tornado.tlc	Tornado (VxWorks) Real-Time Target

Selection: D:\MATLAB6p5\rtw\c\I_8xx8\LinCon.tlc

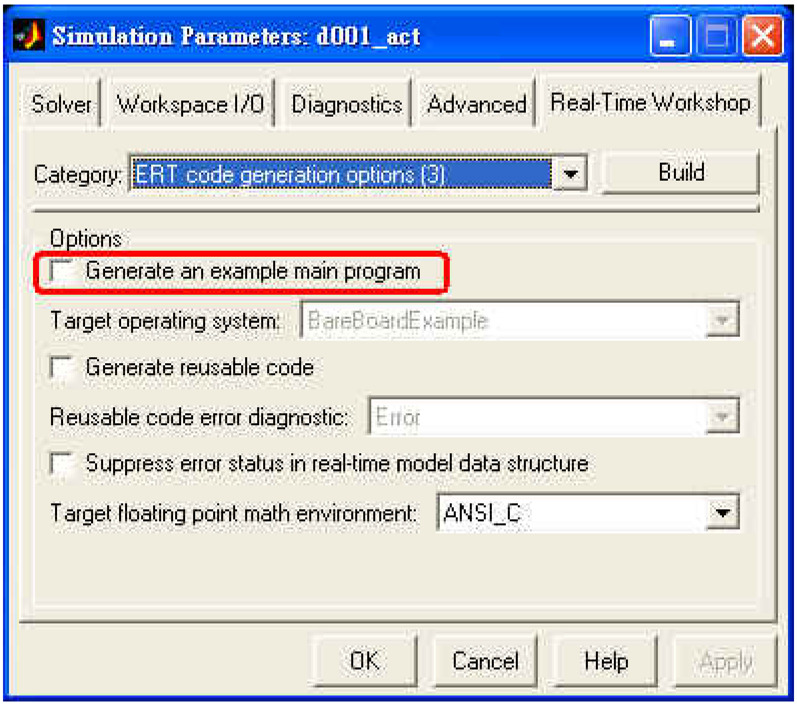
Step 5: Select "**ERT code generation options (1)**" (for MATLAB 6.5) in the Category field. Check the **Terminate function required** and **Single output/update function** options on the pane.



Category: ERT code generation options (1)

Options
- ☐ MAT-file logging
- ☐ Integer code only
- ☐ Initialize internal data
- ☐ Initialize external I/O data
- ☑ Terminate function required
- ☑ Single output/update function
- ☐ Insert block descriptions in code

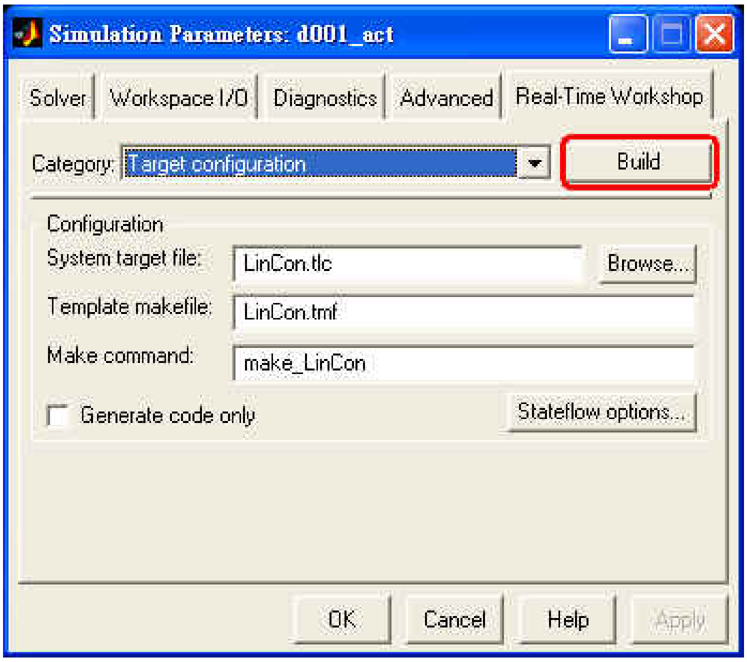
Step 6: For MATLAB 6.5, you have to select "**ERT code generation options (3)**" from the Category field. Cancel "**Generate an example main program"** option.



Category: ERT code generation options (3)

Options
- ☐ Generate an example main program
- Target operating system: BareBoardExample
- ☐ Generate reusable code
- Reusable code error diagnostic: Error
- ☐ Suppress error status in real-time model data structure
- Target floating point math environment: ANSI_C

Step 7: When the above steps finished, click the "**Build**" button to start to convert the model to executable file.



Category: Target configuration

Configuration

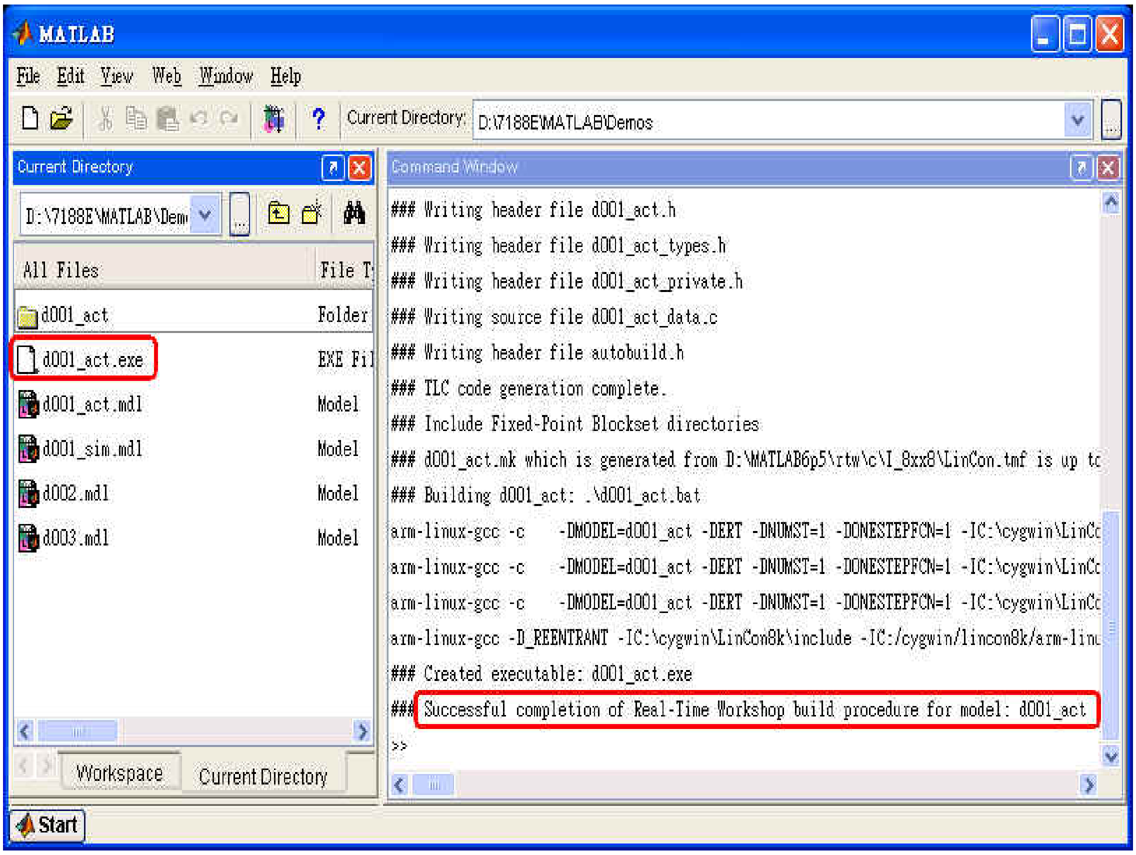
System target file: LinCon.tlc

Template makefile: LinCon.tmf

Make command: make_LinCon

□ Generate code only

Step 8: After the process completed successfully, the message in the MATLAB command window looks like as the following figure.



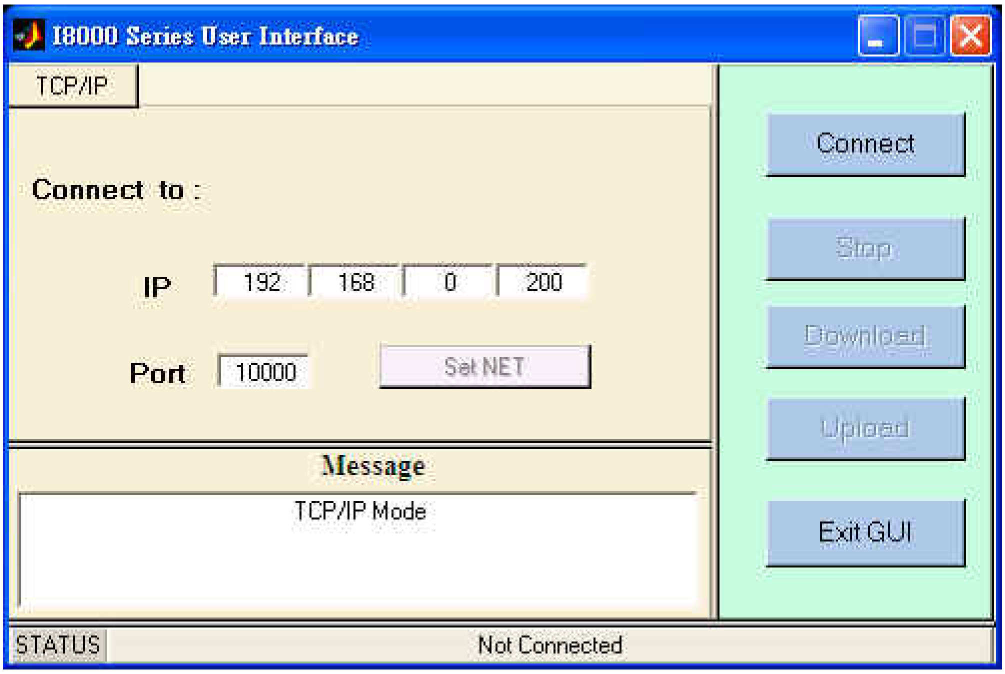
Current Directory: D:\7188E\MATLAB\Demos

All Files	File T
d001_act	Folder
d001_act.exe	EXE Fil
d001_act.mdl	Model
d001_sim.mdl	Model
d002.mdl	Model
d003.mdl	Model

```
### Writing header file d001_act.h
### Writing header file d001_act_types.h
### Writing header file d001_act_private.h
### Writing source file d001_act_data.c
### Writing header file autobuild.h
### TLC code generation complete.
### Include Fixed-Point Blockset directories
### d001_act.mk which is generated from D:\MATLAB6p5\rtw\c\I_8xx8\LinCon.tmf is up to
### Building d001_act: .\d001_act.bat
arm-linux-gcc -c    -DMODEL=d001_act -DERT -DNUMST=1 -DONESTEPFCN=1 -IC:\cygwin\LinCo
arm-linux-gcc -c    -DMODEL=d001_act -DERT -DNUMST=1 -DONESTEPFCN=1 -IC:\cygwin\LinCo
arm-linux-gcc -c    -DMODEL=d001_act -DERT -DNUMST=1 -DONESTEPFCN=1 -IC:\cygwin\LinCo
arm-linux-gcc -D_REENTRANT -IC:\cygwin\LinCon8k\include -IC:/cygwin/lincon8k/arm-linu
### Created executable: d001_act.exe
### Successful completion of Real-Time Workshop build procedure for model: d001_act
>>
```

3.4. Program downloading & data uploading

After the build process is completed, the executable file will be generated automatically. Then the GUI communication interface will appear later. So users can download the .exe file to the LinCon-8000 embedded controller through it. Or users can type "**gui8000**" at the MATLAB prompt, and the GUI communication interface will also appear. The GUI figure shows as below：



I8000 Series User Interface

TCP/IP

Connect to :

IP 192 168 0 200

Port 10000

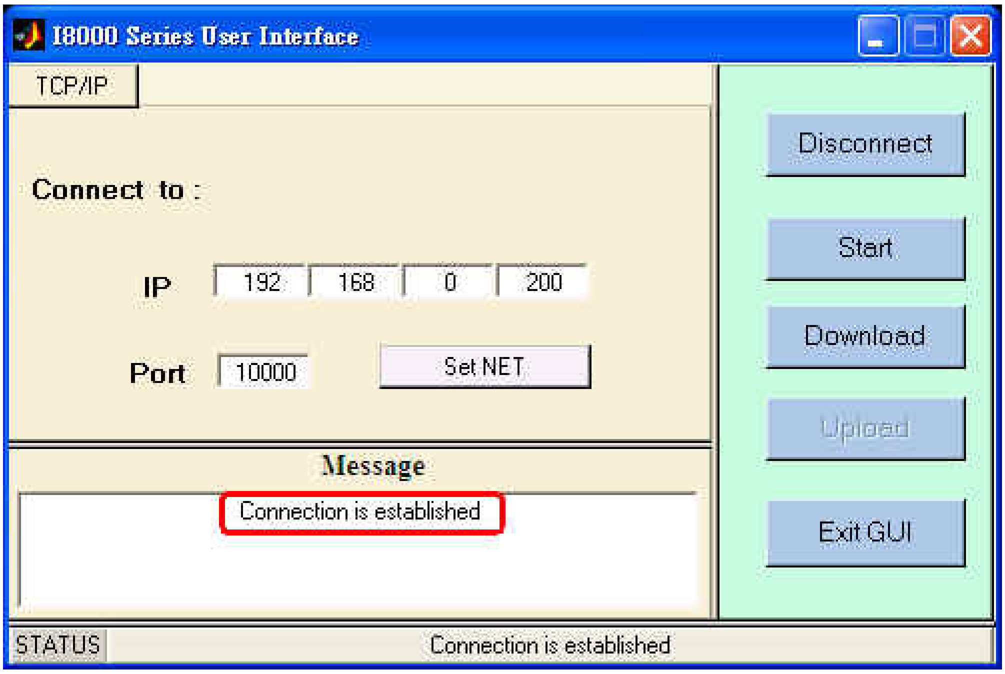
Message

TCP/IP Mode

STATUS Not Connected

Please follow these steps to download the program to the LinCon-8000 target system.

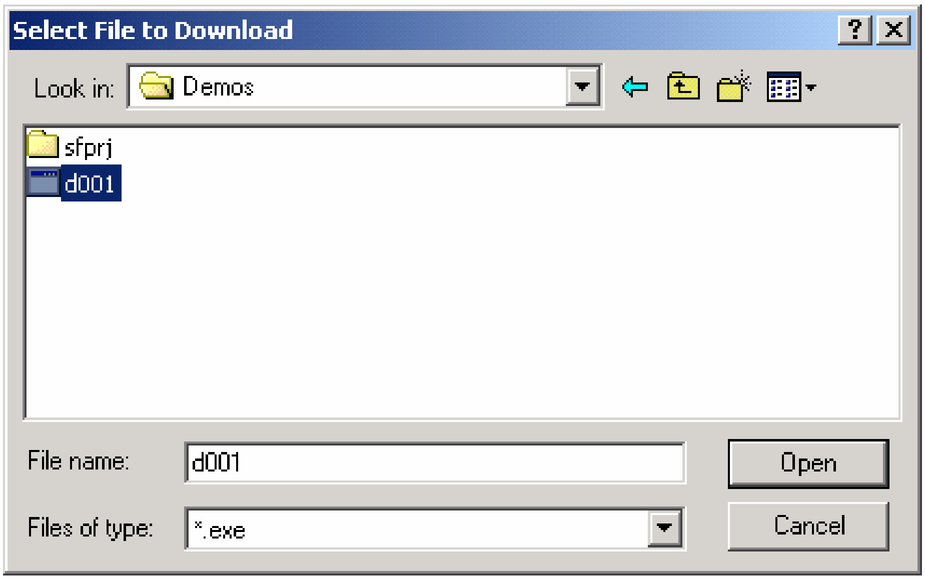
Step 1: Type "**gui8000**" at the MATLAB prompt and the GUI will appear. Then type the "**LinCon IP address** (like 192.168.0.200)" in IP field of GUI and **port number is "10000" (default)** for LinCon TCP Server which is placed in the **/mnt/hda/matlab/mat_server.exe** of LinCon). Then click the "***connect***" button and the message of "Connection is established" will show up. **(Note : Remember to setup network of PC and the LinCon-8000 in the same LAN. Then the connection between "Matlab TCP Client" and "LinCon TCP Server" will be established.)**



Connection is established

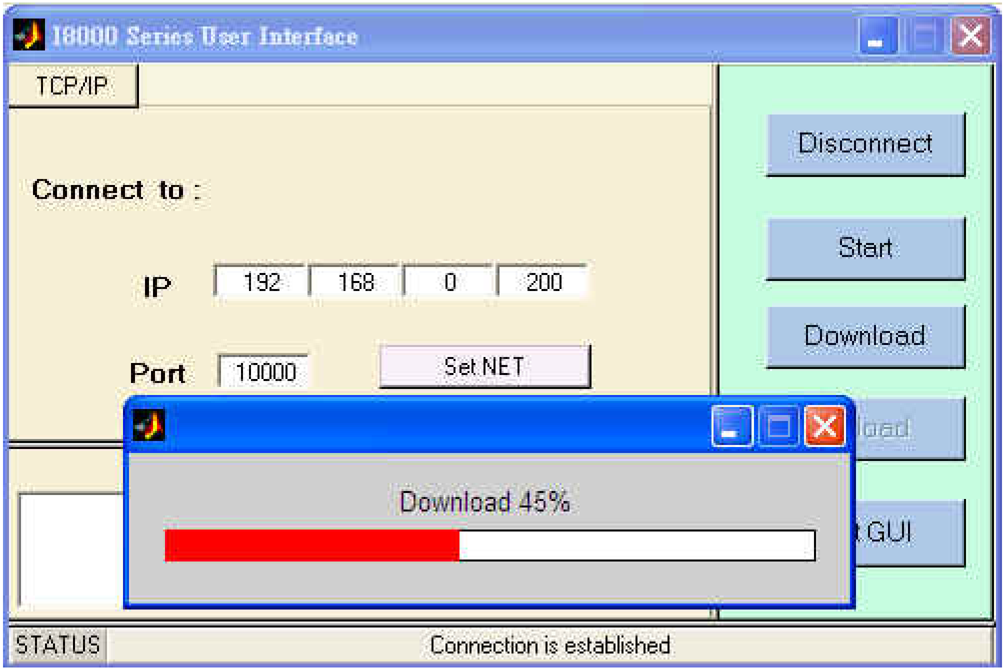
STATUS Connection is established

Step 2: Click the "***Download***" button and select the file you want to download to the LinCon-8000. The download progress will show the percentage of completion. After the execution file in PC is downloaded to the LinCon, it will be saved with the fixed name – **matapp** in the path : **/mnt/hda/matlab/**.

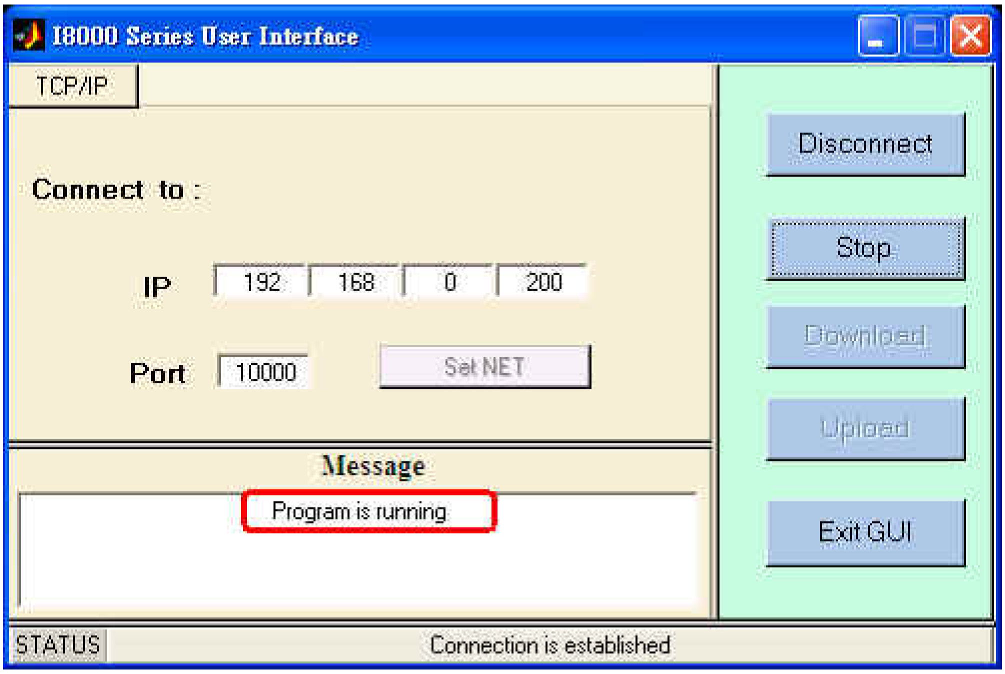


File name: d001

Files of type: *.exe



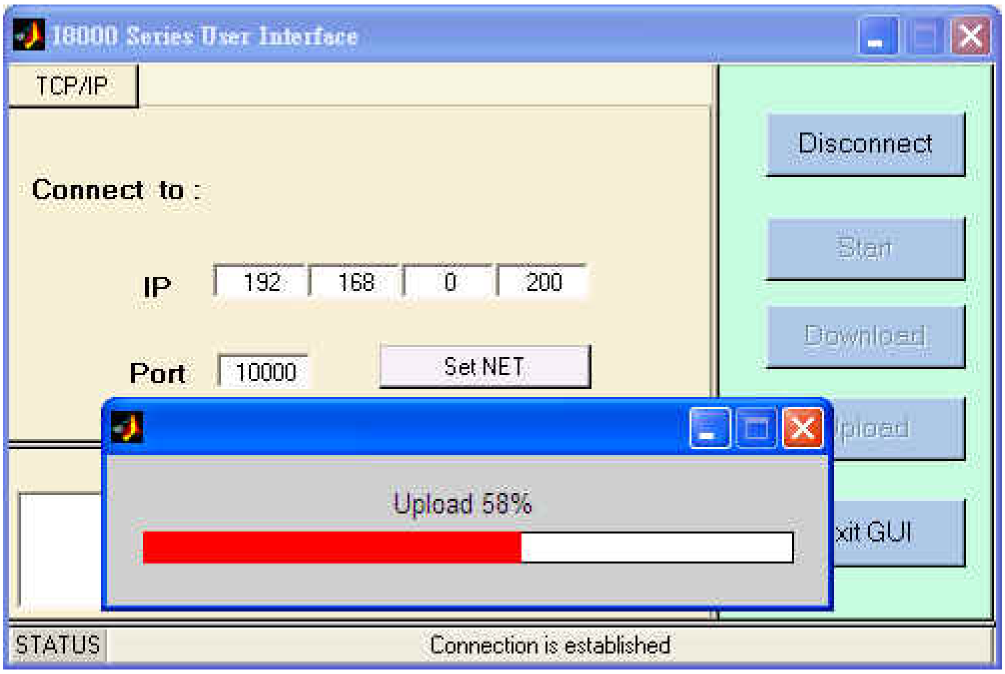
Step 3: If users want to execute the downloaded program, just click the "***Start***" button. In the meanwhile, the message of "Program is running" will show up.



Program is running

Connection is established

Step 4: After the program is executed successfully, users can click the "***Stop***" button to stop the program. After the program stopped, users can click the "***Upload***" button to upload the data from LinCon-8000. The upload data will be saved in a file -- "**tcpdata.mat**" (It is fixed) and it will be placed in the current working directory.



Connect to :

IP 192 168 0 200

Port 10000

Upload 58%

STATUS Connection is established

Step 5: To close the GUI communication interface, users can click the "***Exit GUI***" button or ☒ on the right-upper corner of GUI.

3.5. Working with Stateflow

What is Stateflow?

Stateflow is a graphical design and development tool for control and supervisory logic. The advantage of Stateflow are as follows：

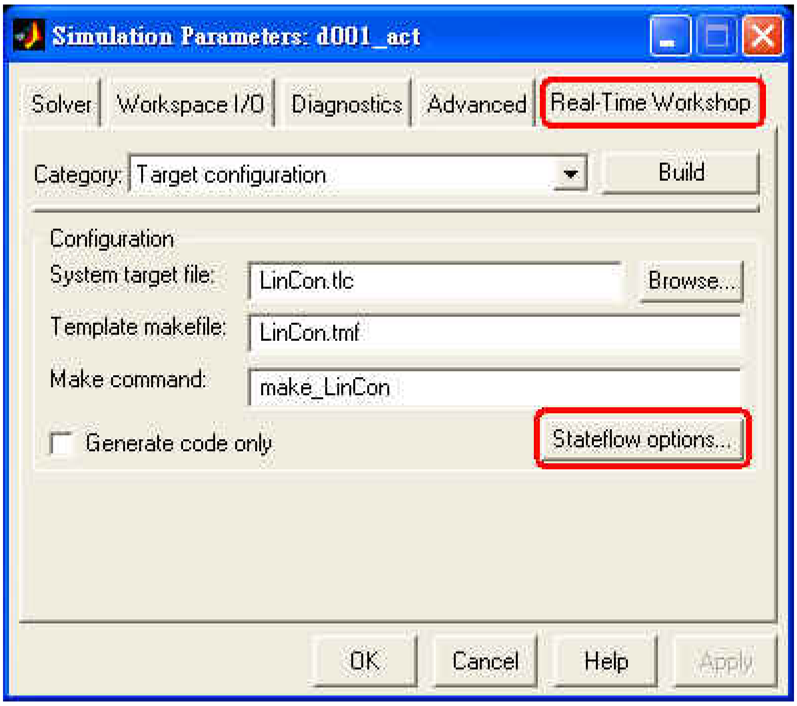
- Visually model and simulate complex reactive systems based on finite state machine theory.
- Design and develop deterministic, supervisory control systems.
- Easily modify your design, evaluate the results, and verify the system's behavior at any stage of your design.

To get more information about the Stateflow, please visit the website http://www.mathworks.com/products/stateflow.

How to work with Stateflow

Here just explain the difference from the section 3.4. Other parts are all the same. To configure the Stateflow Coder options, please follow these steps：

Step 1: On the "***Simulation Parameters***" dialog box, click the "*Real-Time Workshop*" tab and then click "***Stateflow options***" button.



Simulation Parameters: d001_act

Category: Target configuration

Configuration

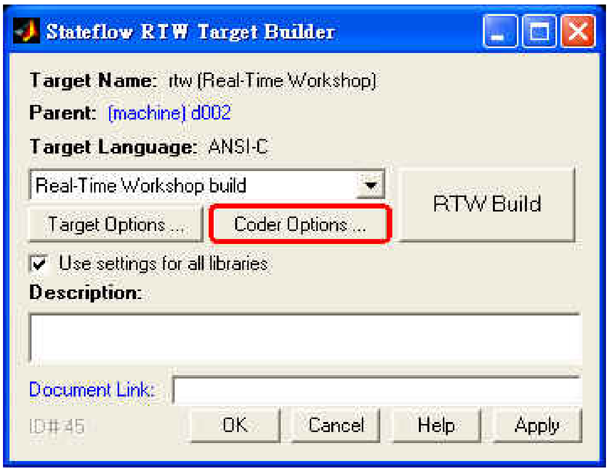
System target file: LinCon.tlc

Template makefile: LinCon.tmf

Make command: make_LinCon

□ Generate code only

Step 2: When the "***Stateflow RTW Target Builder***" dialog box appears, click the "***Coder Options…***" button.



Stateflow RTW Target Builder

Target Name: rtw (Real-Time Workshop)

Parent: [machine] d002

Target Language: ANSI-C

Real-Time Workshop build

☑ Use settings for all libraries

Description:

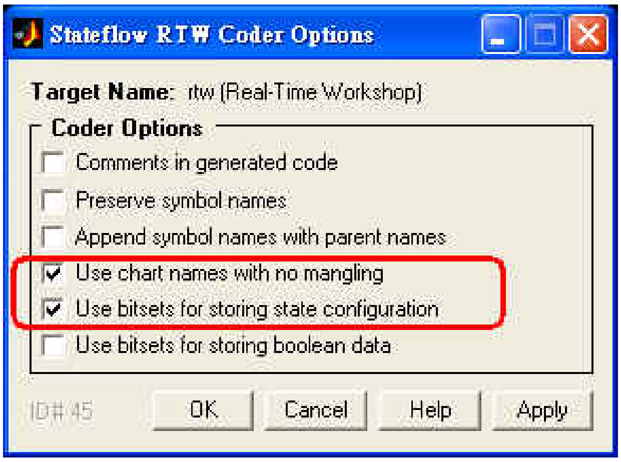
Document Link:

ID# 45

Step 3: Check the two following options and then click "OK" button to apply the setting.

(1) Use chart names with no mangling

(2) Use bitsets for storing state configuration



Stateflow RTW Coder Options

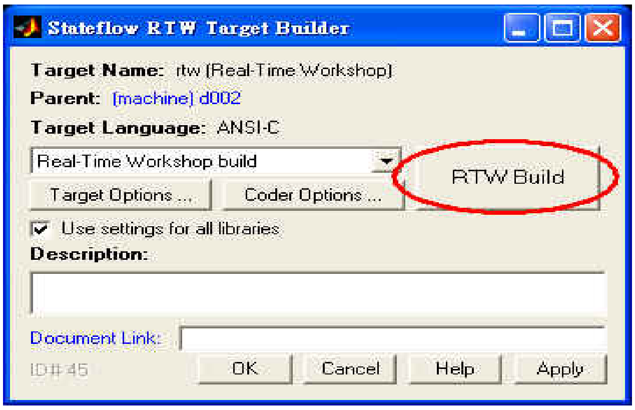
Target Name: rtw (Real-Time Workshop)

Coder Options

- ☐ Comments in generated code
- ☐ Preserve symbol names
- ☐ Append symbol names with parent names
- ☑ Use chart names with no mangling
- ☑ Use bitsets for storing state configuration
- ☐ Use bitsets for storing boolean data

ID# 45

Step 4: Click the "***RTW Build***" button to start the build process. When the build process is completed successfully, an executable file will be created in the current directory.



Stateflow RTW Target Builder

Target Name: rtw (Real-Time Workshop)

Parent: [machine] d002

Target Language: ANSI-C

Real-Time Workshop build

☑ Use settings for all libraries

Description:

Document Link:

ID# 45

4. Demos

In the following sections, we will introduce the usage of the I-8000 driver blocks by a series of demos, including DI, DO, AI, AO and Relay.

4.1. DI & DO Modules

Introduction

Digital I/O interfaces are frequently used for the control system. In this section, we will guide users how to use the DI and DO driver block by an I-8053 module and an I-8057 module.

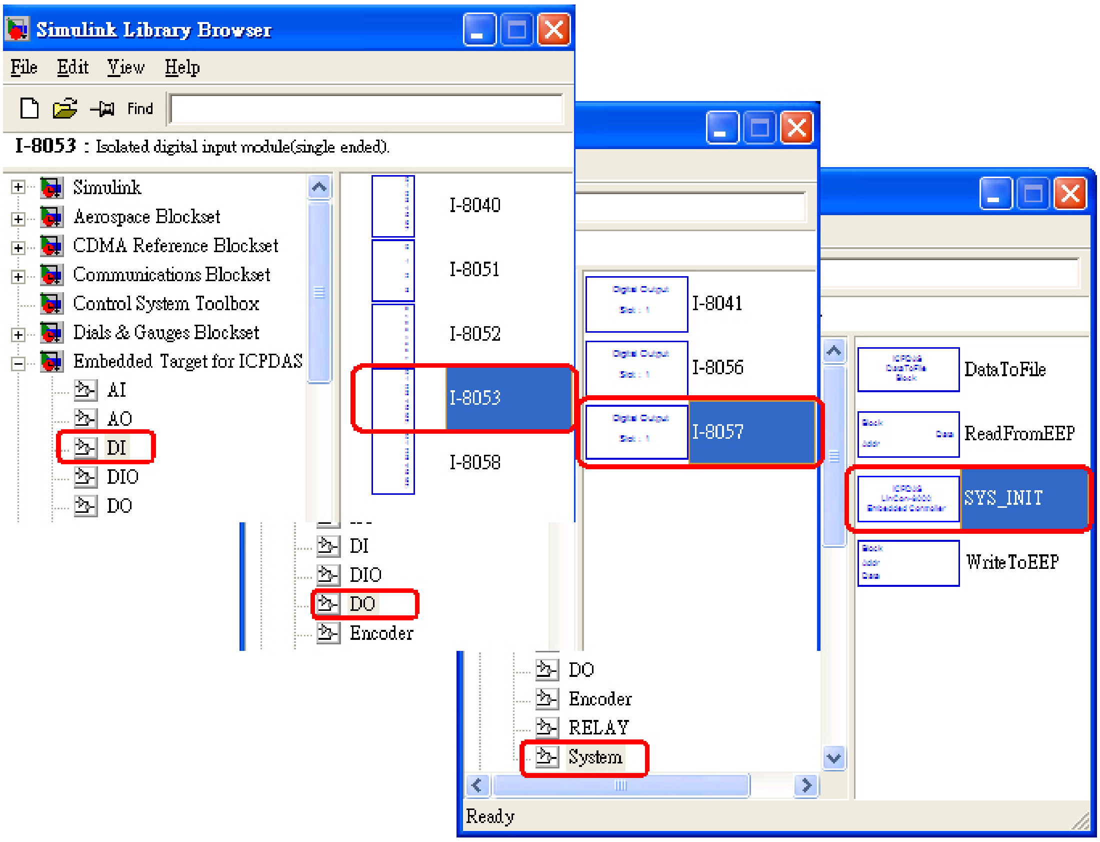
The I-8057 driver block expects a value smaller than 65535 from its input port. This is because I-8057 module is a 16-channel digital input module. If you assign a value 5 to the I-8057 module, then the channel 0 and channel 2 of the I-8057 module will be set ON.

The I-8053 driver block will output a value 1 from the individual port if the corresponding channel of the module acquires a digital input. Otherwise, it will output a value 0.

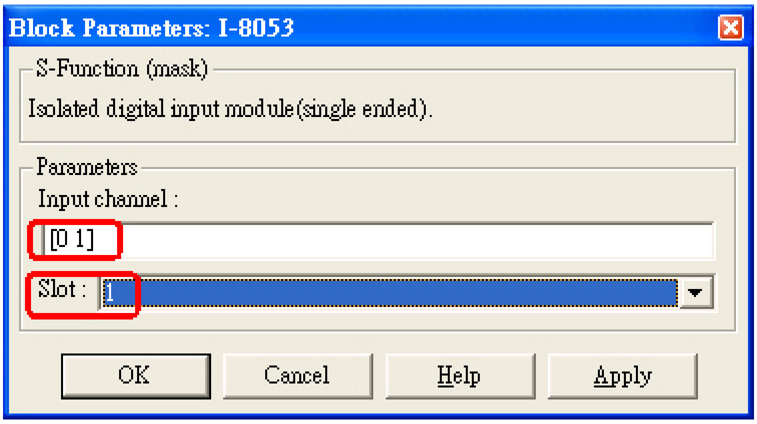
Demonstration

This demonstration uses 2 DO channels of I-8057 module and 2 DI channels of I-8053 module to test the digital input and digital output function. The digital output channels are connected to digital input channels. The following steps describe how to create and implement the experiment of digital input/output modules.

Step 1: Create a new model window and copy SYS_INIT, I-8053, and I-8057 blocks from the System, DI, and DO library to the model window respectively.



Step 2: Double-click on the I-8053 block and type [0 1] in the “Input channel” field. And select the slot number of the I-8053 module that is according to the position of 8053 in the LinCon-8000.



Block Parameters: I-8053

S-Function (mask)

Isolated digital input module(single ended).

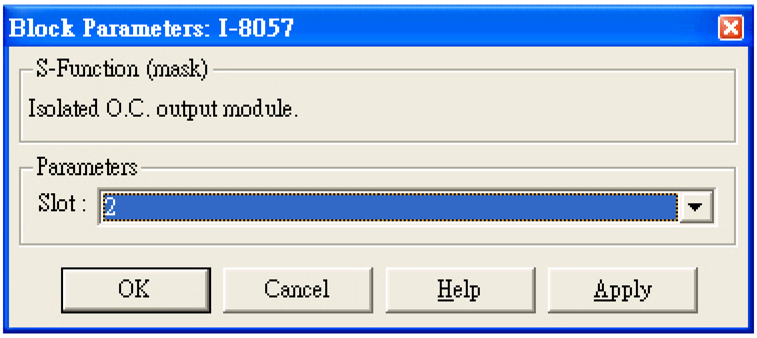
Parameters

Input channel :

[0 1]

Slot : 1

Step 3: In a similar method, open the I-8057 dialog box and select the slot number where I-8057 module is mounted in the LinCon-8000.



Block Parameters: I-8057

S-Function (mask)

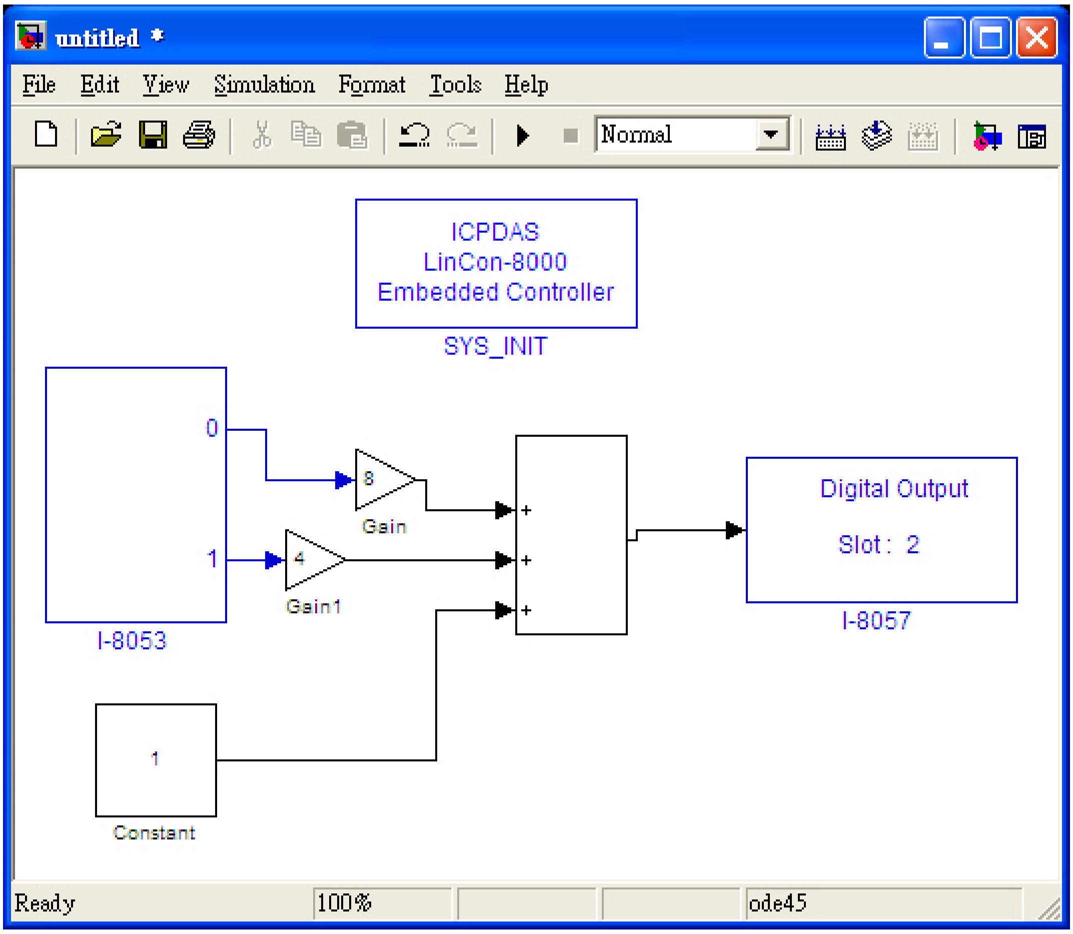
Isolated O.C. output module.

Parameters

Slot : 2

Step 4: Copy two "*Gain*" and one "*Sum*" blocks to the model from the "*Simulink / Math Operations*" library.

Step 5: Connect all the blocks as shown in the following figure.



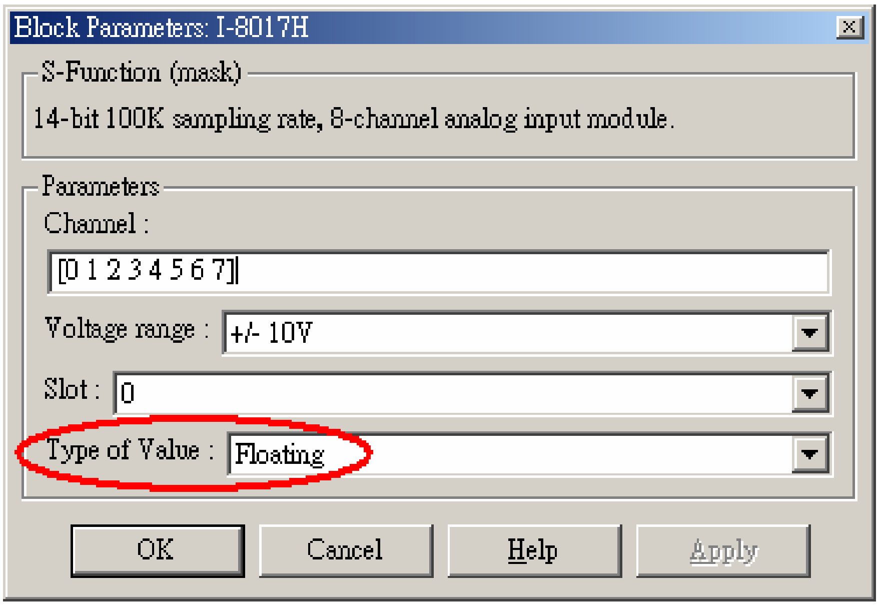
Step 6: Build the model to executable file. The process are all the same as section 3.3.

4.2. AI Modules

In this section, we will introduce how to use the AI driver block, and a demo model will be presented.

Floating or Hex Input Type

The AI driver block supports two types of input values, ***floating*** and ***hex***. When the input type is set to *floating*, the output port will output a value, which is the same with the voltage measured from the corresponding channel. If the input type is set to *hex*, the output port will output a value between *-8192* and *8191* (This value is affected by the *Voltage range*; ex. When the *voltage range* is set to +/-10V, the output port will output a value *-4096* while the corresponding channel acquires a -*5V* analog input.).



Block Parameters: I-8017H

S-Function (mask)

14-bit 100K sampling rate, 8-channel analog input module.

Parameters

Channel : [0 1 2 3 4 5 6 7]

Voltage range : +/- 10V

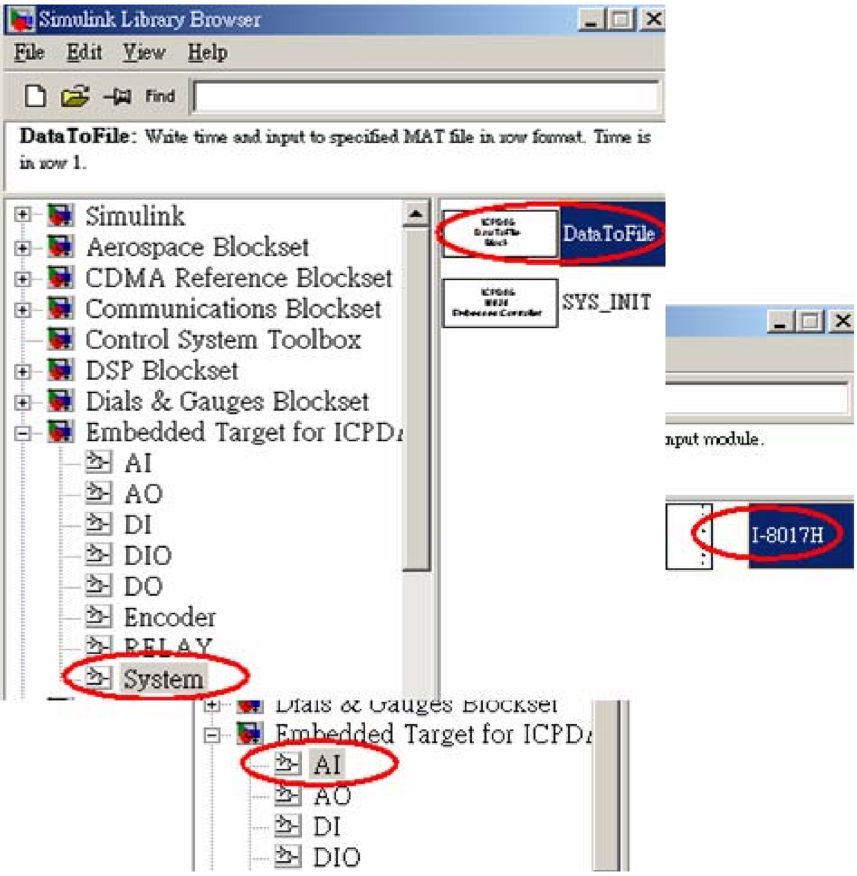
Slot : 0

Type of Value : Floating

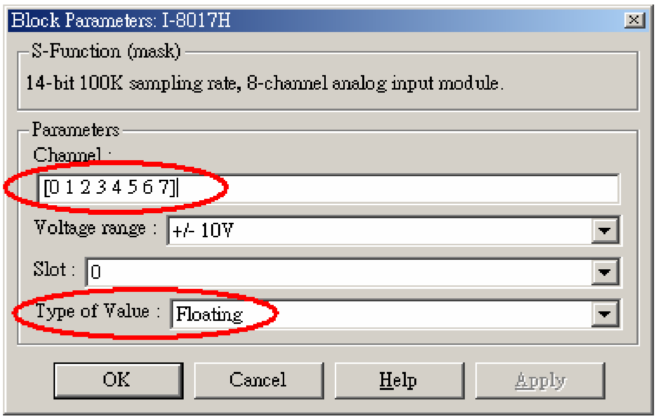
Demonstration

This demonstration uses 1 *I-8017H* and 1 *DataToFile* driver block to test the analog input function. By using the *DataToFile* driver block, the data acquired from the I-8017H module can be uploaded to PC for analysis. The following steps describe how to create and implement the experiment.

Step 1: Create a new model window and copy *SYS_INIT*, *I-8017H*, and *DataToFile* blocks from the *System* and *AI* library to the model window respectively.



Step 2: Double-click on the **I-8017H block**, and then the dialog box appears. In this demo, we use all the channels, so enter [0 1 2 3 4 5 6 7] in the "Channel" field. Then set voltage range as +/-10V, and specify the slot where I-8017H module is mounted. Finally, set *Type of Value* to *Floating*.



Block Parameters: I-8017H

S-Function (mask)

14-bit 100K sampling rate, 8-channel analog input module.

Parameters

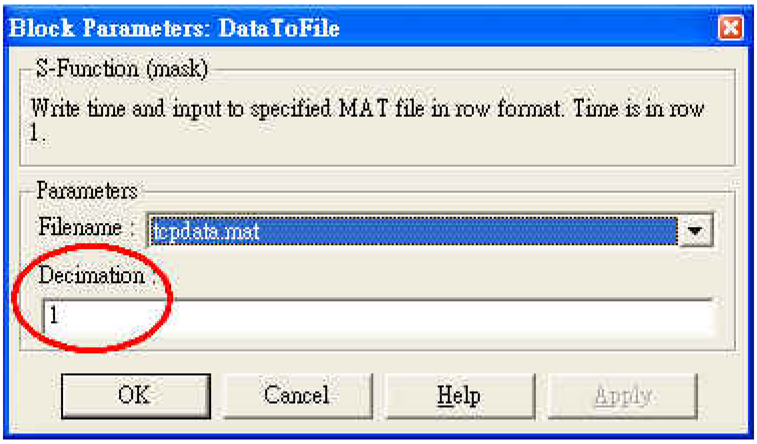
Channel : [0 1 2 3 4 5 6 7]

Voltage range : +/- 10V

Slot : 0

Type of Value : Floating

Step 3: In a similar manner, open the DataToFile dialog box and the filename is fixed – **tcpdata.mat**. Enter **1** in the "Decimation" field, and then the analog input signal will be recorded in every sampling time interval.



Block Parameters: DataToFile

S-Function (mask)

Write time and input to specified MAT file in row format. Time is in row 1.

Parameters

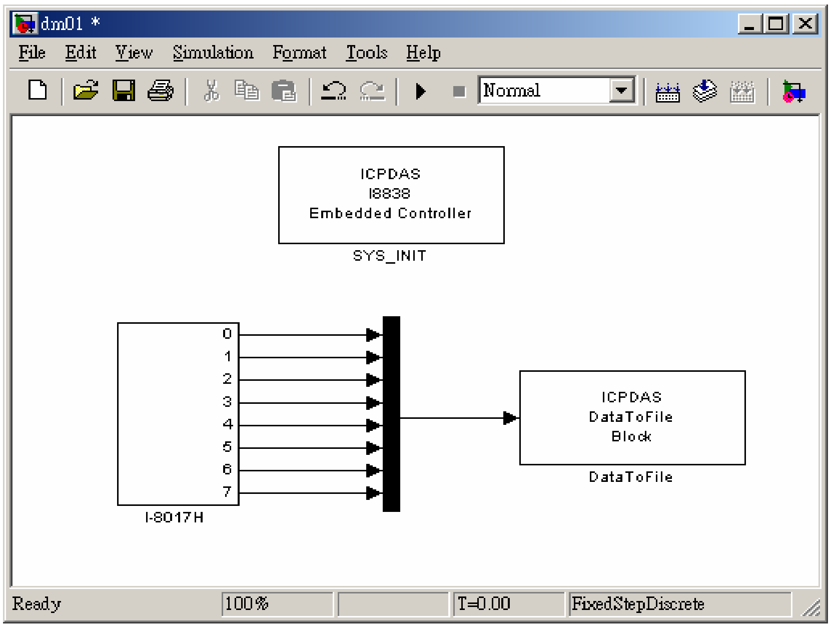
Filename : tcpdata.mat

Decimation :

1

Step 4: Copy one Mux block to the model window from the **Simulink/Signal Routing** (for MATLAB 6.5).

Step 5: Connect all the blocks as shown in the figure below.



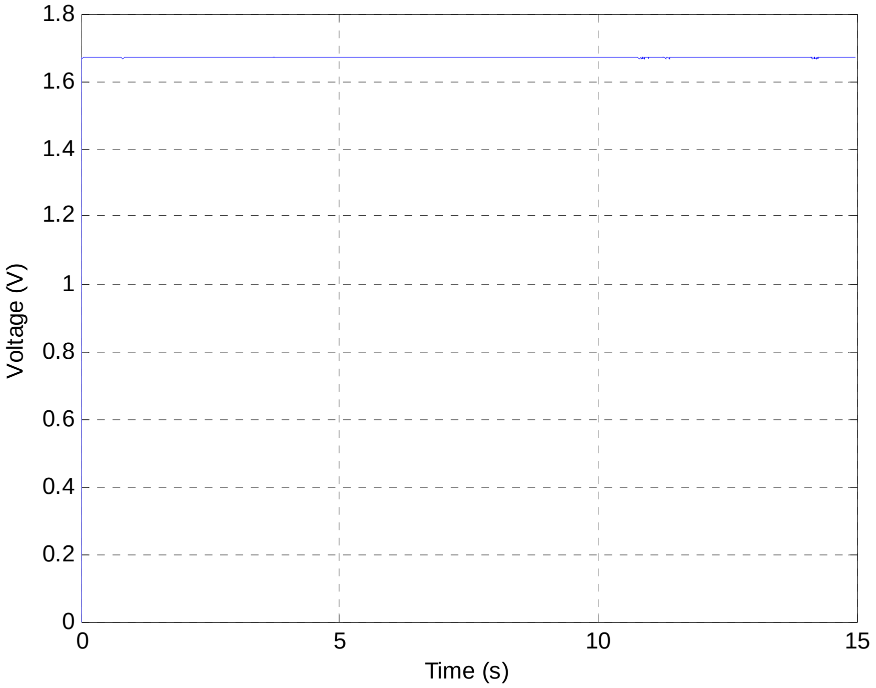
Step 6: Click **Simulation parameters** from the *Simulation* menu on the model window. When the dialog box appears, configure the RTW options (refer to section 3.3). And then click **Build** button to start the build process.

Step 7: When the build process ends successfully, download the control program generated to the LinCon-8000 and upload the experiment data for further analysis (refer to section 3.4).

Step 8: If the stop time that you specified is up, you can start the data uploading. After the uploading process is completed, use the built-in MATLAB scripts to plot the result. It would look like the figure below.

For Example :

plot (tcpdata(1, :), tcpdata(2, :))

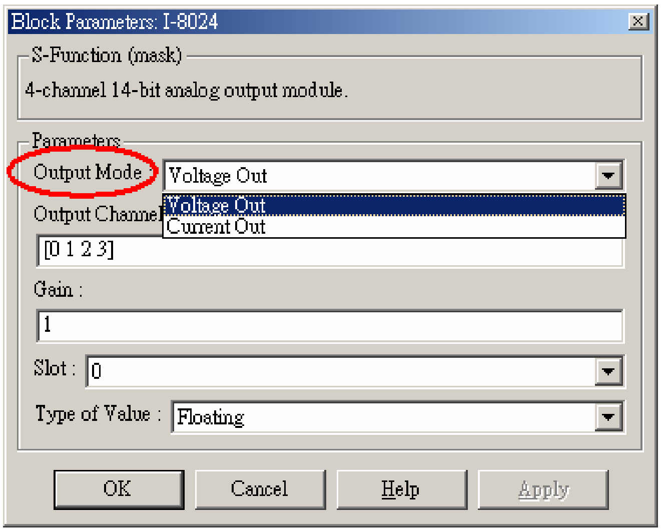


4.3. AO Modules

This section attempts to make you familiar with the usage of the I-8024 module, which is a 4-channel and 14-bit analog output module. In the beginning of this section, we will give you a detailed introduction of the I-8024 driver block. And then a demonstration will be presented in the rest of this section.

Introduction to I-8024 driver block

On the dialog box of the I-8024 driver block, you can select the outputs of the I-8024 module as either “Voltage Out” or “Current Out”. If you select “Voltage Out” in the “Output Mode” field, an output range between +10V and –10V will be available; otherwise, a current between 0 and 20 mA will be instead.


Block Parameters: I-8024

S-Function (mask)

4-channel 14-bit analog output module.

Parameters

Output Mode: Voltage Out

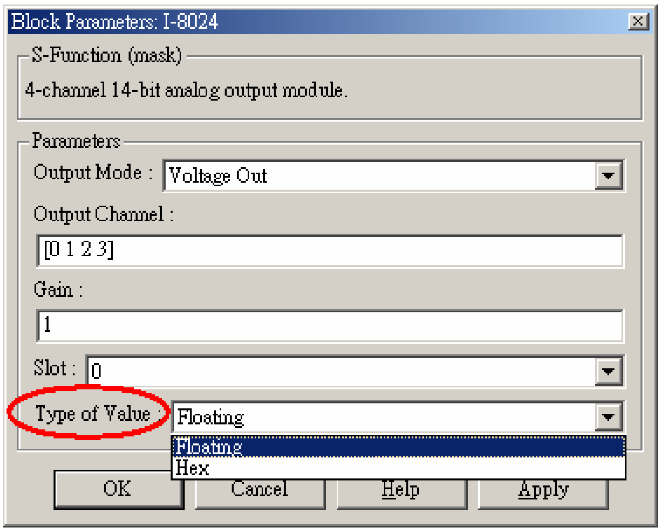
Output Channel: [0 1 2 3]

Gain : 1

Slot : 0

Type of Value : Floating

Furthermore, the input type of the I-8024 driver block can be either “Floating” or “Hex”.



Block Parameters: I-8024

S-Function (mask)

4-channel 14-bit analog output module.

Parameters

Output Mode : Voltage Out

Output Channel : [0 1 2 3]

Gain : 1

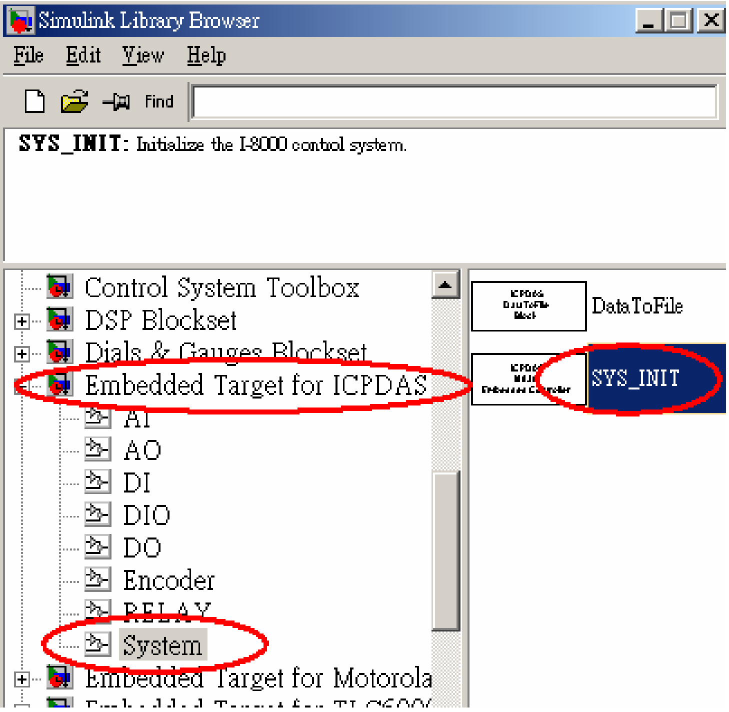
Slot : 0

Type of Value Floating

Demonstration

In this demo, we use 1 channel of the I-8024 module and 1 channel of the I-8017H module. Then channel 0 of I-8024 module is physically connected to channel 0 of I-8017H module. After the physical wire connection is done, send a sine wave to the input port of the I-8024 driver block. The following figures describe the steps by which this demo is created and implemented.

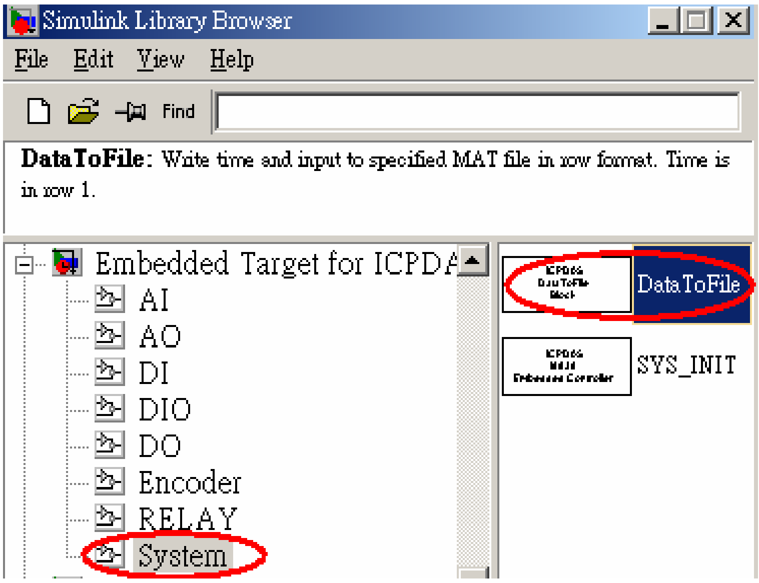
Step 1: Create a new model in Simulink and insert a SYS_INIT block from the System block library.



Simulink Library Browser

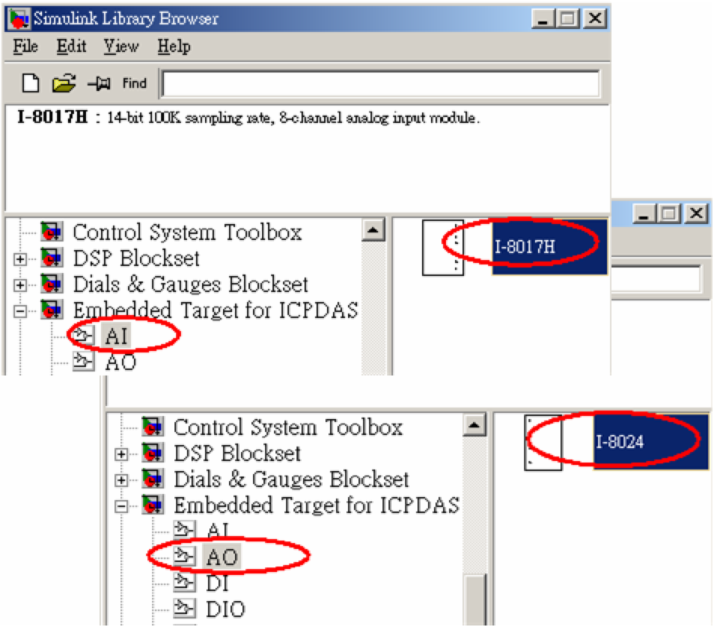
SYS_INIT: Initialize the I-8000 control system.

Step 2: Copy the DataToFile block from the System block library to the model.



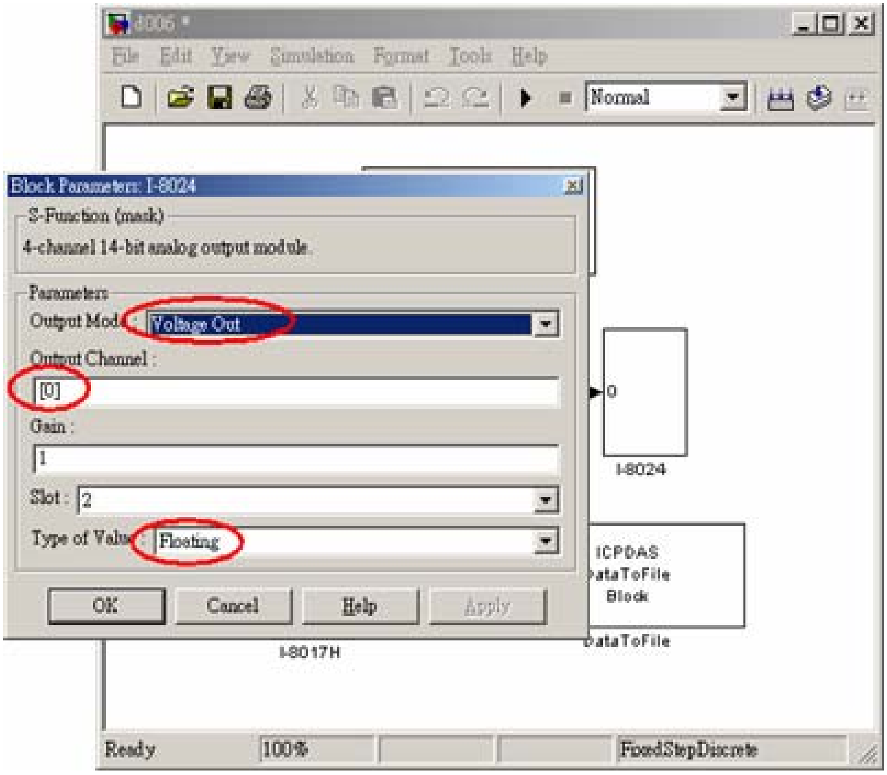
DataToFile: Write time and input to specified MAT file in row format. Time is in row 1.

Step 3: In a similar manner, insert an I-8024 block and an I-8017H block separately from the AO and AI block library.



I-8017H : 14-bit 100K sampling rate, 8-channel analog input module.

Step 4: Double click on the I-8024 block to setup the AO module. Here we use channel 0 of the I-8024 module, which is mounted on slot 2 of the I-8xx8 embedded controller. And the output mode is set to “Voltage Out”; the type of value is selected as “Floating”.



Block Parameters: I-8024

S-Function (mask)

4-channel 14-bit analog output module.

Parameters

Output Mode : Voltage Out

Output Channel : [0]

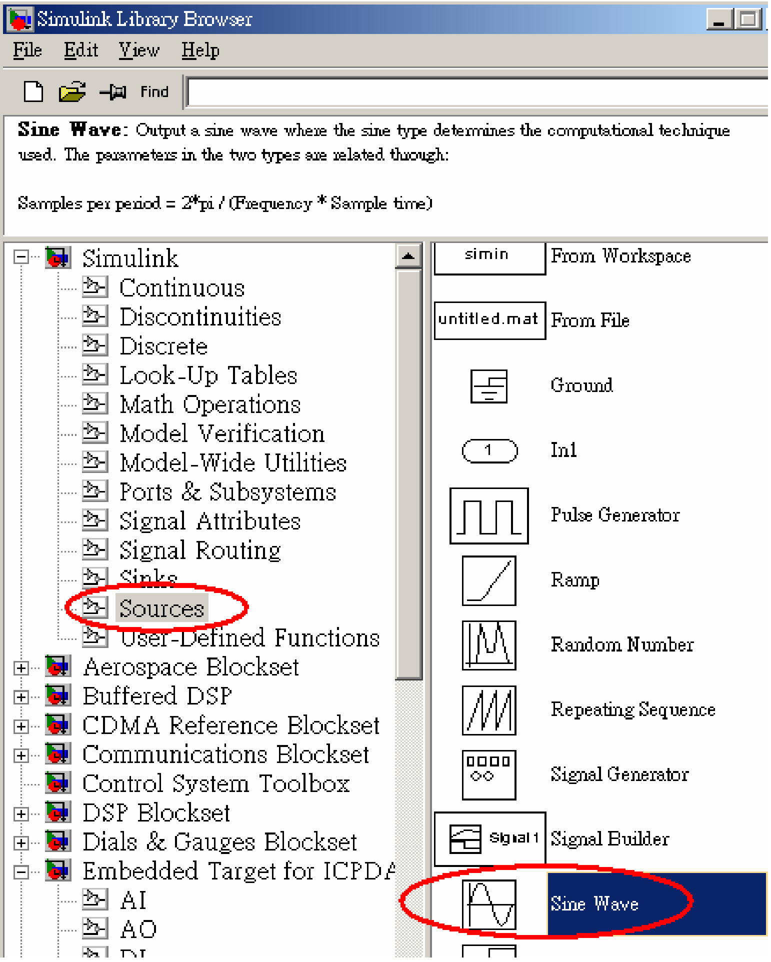
Gain : 1

Slot : 2

Type of Value : Floating

Step 5: As to the setting of SYS_INIT and I-8017H, please refer to the section 4.1 and 4.2.

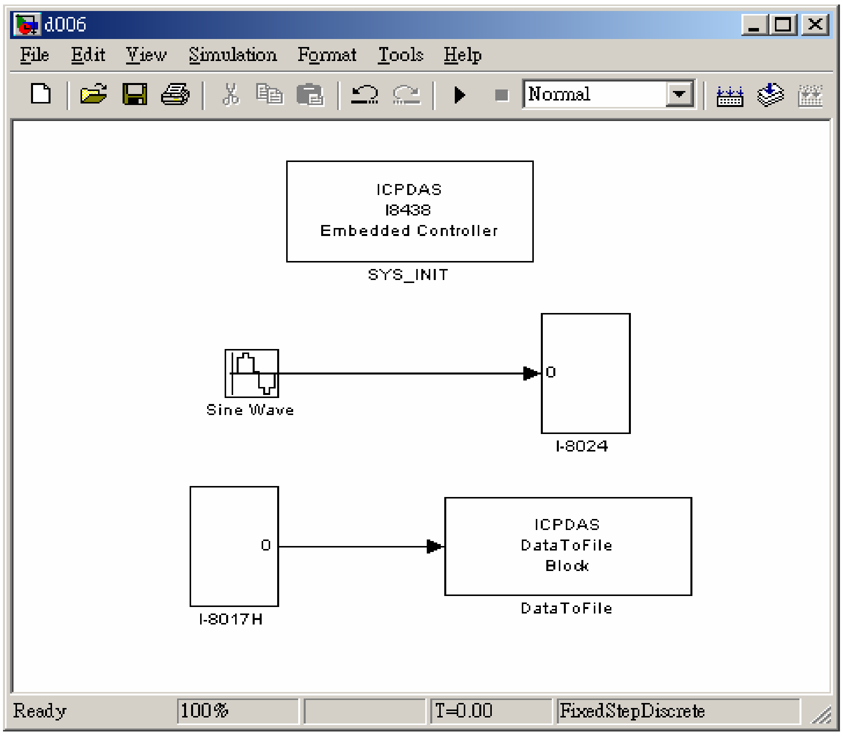
Step 6: Add a Sin Wave block from the Simulink\Sources library.



Sine Wave: Output a sine wave where the sine type determines the computational technique used. The parameters in the two types are related through:

Samples per period = 2*pi / (Frequency * Sample time)

Step 7: Connect all blocks as shown in the figure below.



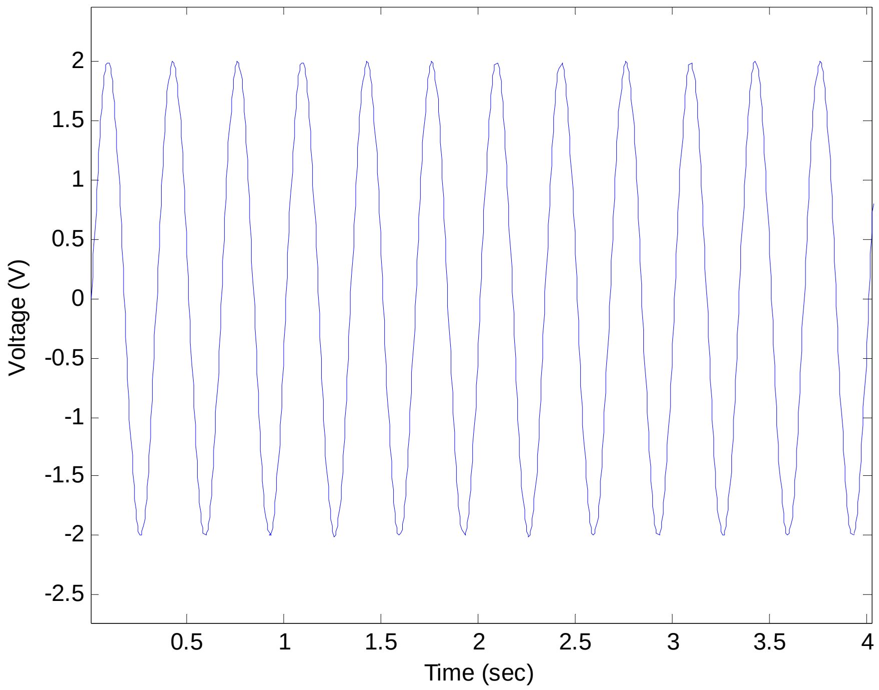
Step 8: Click Simulation parameters from the Simulation menu to open the Simulation Parameters dialog box. Then configure the RTW options (refer to section 3.3) and press the Build button to start the build process.

Step 9: When the build process ends successfully, download the .exe file generated to the LinCon-8000 and run (refer to section 3.4).

Step 10: If the stop time that you specified is up, you can start the data uploading. After the uploading process is completed, use the built-in MATLAB scripts to plot the result. It would look like the figure below.

For Example:

```
plot ( tcpdata(1, :), tcpdata(2, :) )
```

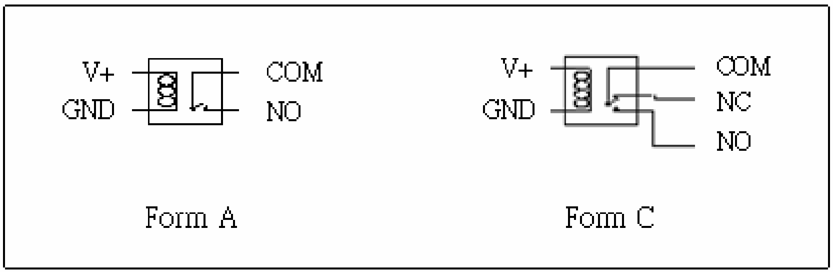


4.4. Relay Modules

In the area of automatic control, the relay component is frequently used as a voltage-activated switch, consisting of a bi-directional switched connections and a switching input. The I-8000 series module MATLAB Driver also provides the relay driver block to facilitate your design. In the following subsection we demonstrate the usage of this block.

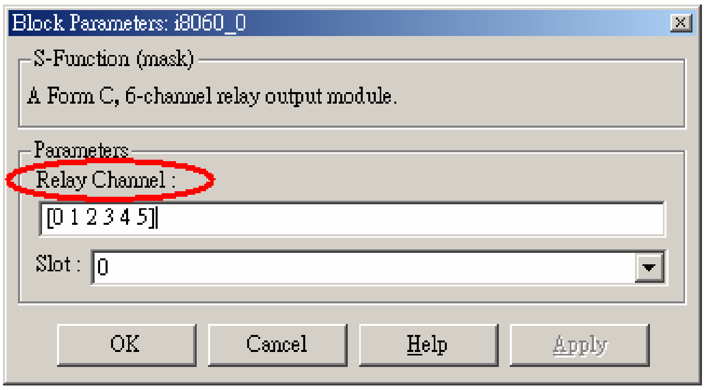
Introduction to Relay driver block

Basically, the relay modules provided by ICP DAS are categorized into two forms, form A and form C. The figure below shows the difference between them.



The relay of **form A** is represented by the I-8064, and the relay of **form C** is represented by the I-8060. No matter which form a relay is, however, the usage of the driver block is the same. The individual input port of the driver block expects a value greater than 0 to activate the corresponding physical relay output. In other words, if you assign a value 1 to the channel of the relay driver block, the connection of the relay will be switched to NO.

On the dialog box of the relay driver block shown in the figure below, you can select the channels, which are used by entering a row vector in the field of "Relay Channel". And remember to choose the correct slot where the relay module is mounted.



A Form C, 6-channel relay output module.

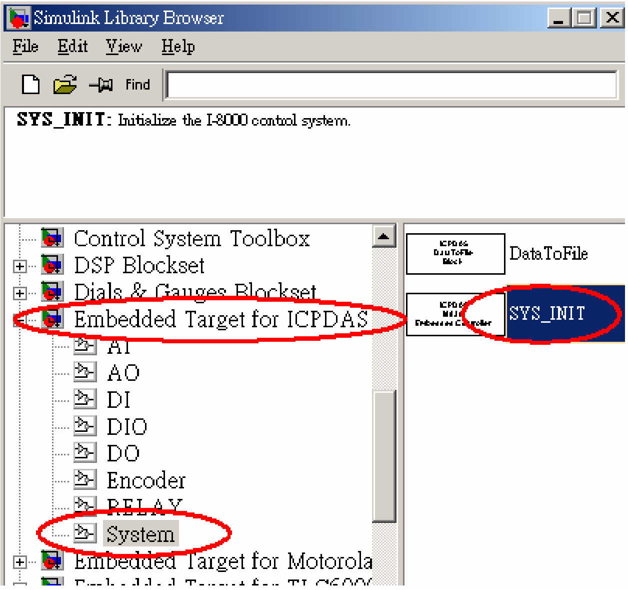
Relay Channel : [0 1 2 3 4 5]

Slot : 0

Demonstration

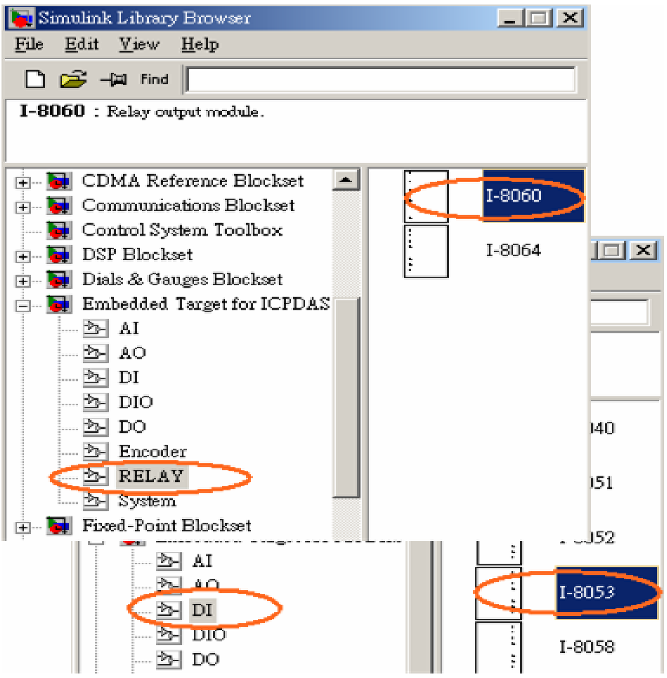
This demo uses an I-8060 relay module and an I-8053 DI module to setup the experiment. Channel 0 of the I-8053 module is connected to a limit switch. When channel 0 of the I-8053 module is activated, it will cause the channel 0 of the I-8060 module switch to NO. The following steps describe how this demo is created and implemented.

Step 1: Create a new model in Simulink and insert a SYS_INIT block from the System block library.



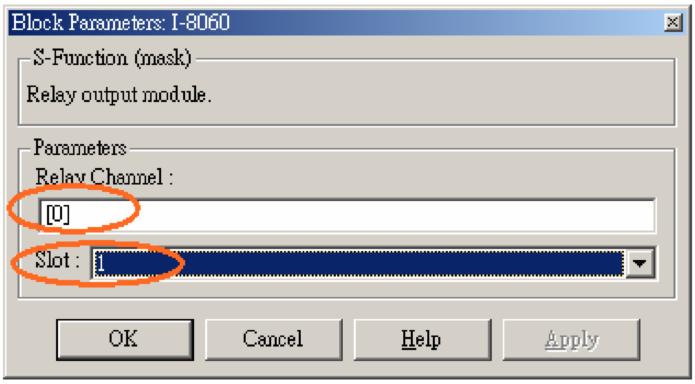
SYS_INIT: Initialize the I-8000 control system.

Step 2: Copy an I-8053 block and an I-8060 block separately from the DI and RELAY block library.



I-8060 : Relay output module.

Step 3: Double click on the I-8060 block to setup the relay module. Here we use channel 0 of the I-8060 module, which is mounted on slot 1 of the I-8x38 embedded controller.



Block Parameters: I-8060

S-Function (mask)

Relay output module.

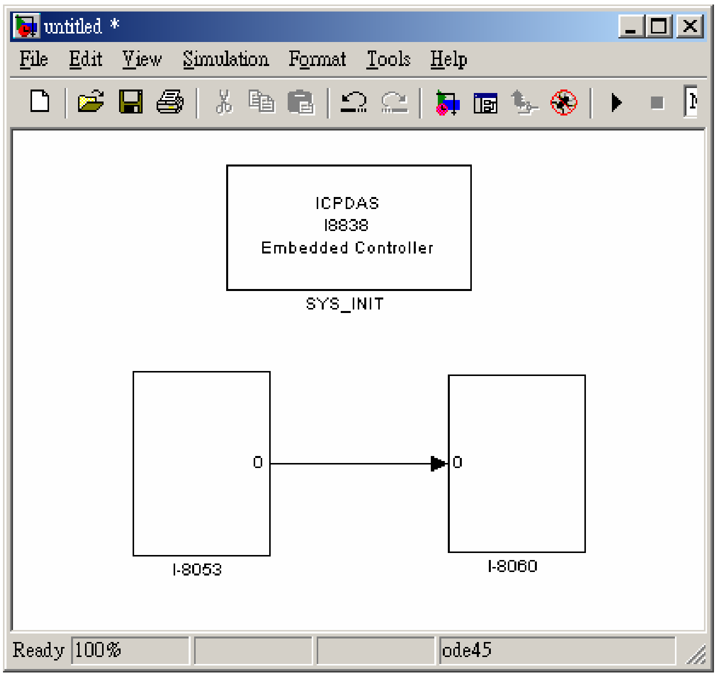
Parameters

Relay Channel : [0]

Slot : 1

Step 4: Configure the I-8053 block. (Please refer to the section 4.1.)

Step 5: Connect all blocks as shown in the figure below.



Step 6: Click Simulation parameters from the Simulation menu to open the Simulation Parameters dialog box. Then configure the RTW options (refer to section 3.3) and press the Build button to start the build process.

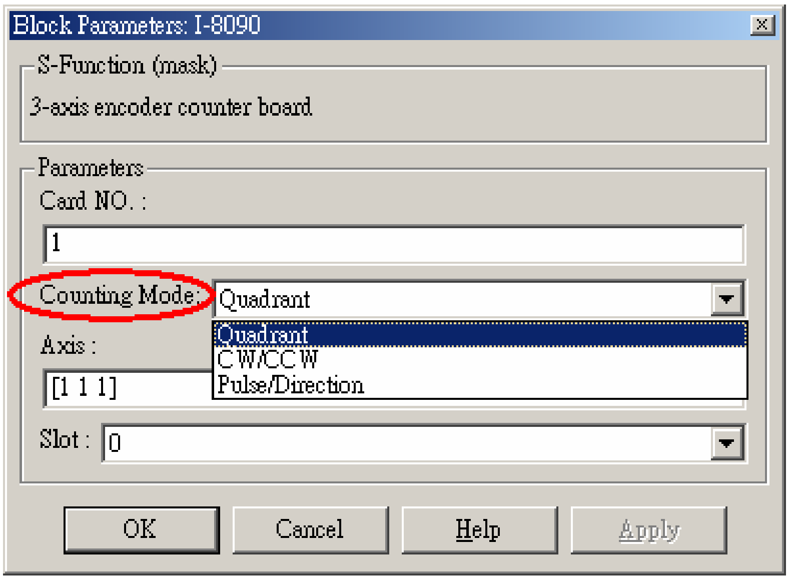
Step 7: When the build process ends successfully, download the .exe file generated to the LinCon-8000 and run it (refer to section 3.4).

4.5. Encoder Module

In most applications of motion control, encoder input modules are used to accept signals from encoders on the motors. Therefore, the I-8000 series module MATAB Driver also provides the driver block for I-8090, which is a 3-axis encoder counter board on I-8000 platform. By using the I-8000 series module MATLAB Driver, you can design your motion controller easily and quickly. In the following paragraphs, we will show you the usage of the I-8090 driver block.

Introduction of I-8090 driver block

On the dialog box of I-8090 driver block, you can select the counting mode of the I-8090 among ***Quadrant***, ***CW/CCW***, and ***pulse/direction***. The setting in this field must match with your wire connections. Otherwise, the measurement of data will not be correct. To get more information about wire connections, please refer to the I-8090/8091 user's manual.



Block Parameters: I-8090

S-Function (mask)

3-axis encoder counter board

Parameters

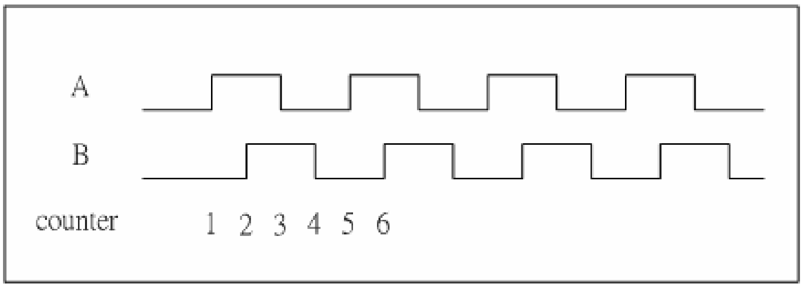
Card NO. : 1

Counting Mode: Quadrant

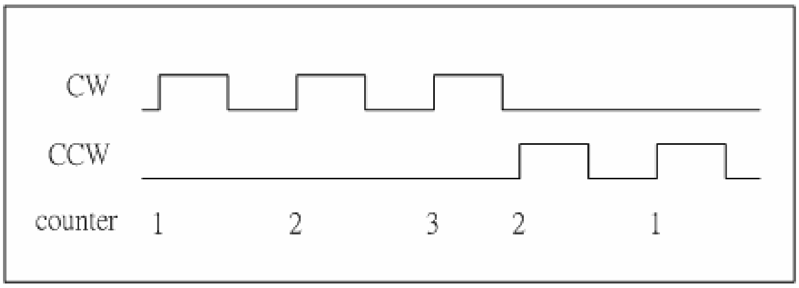
Axis : [1 1 1]

Slot : 0

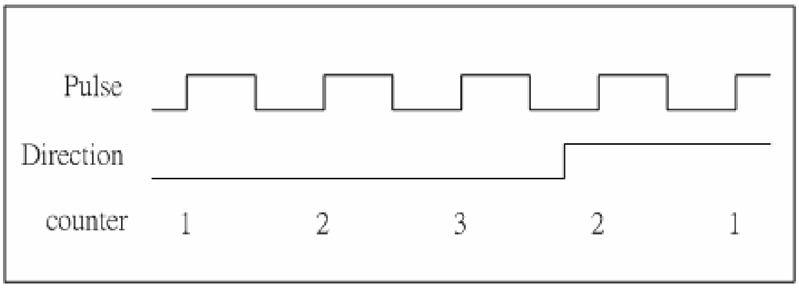
The difference among quadrant, CW/CCW, and pulse/direction counting modes is shown in the following figures.



Quadrant counting mode

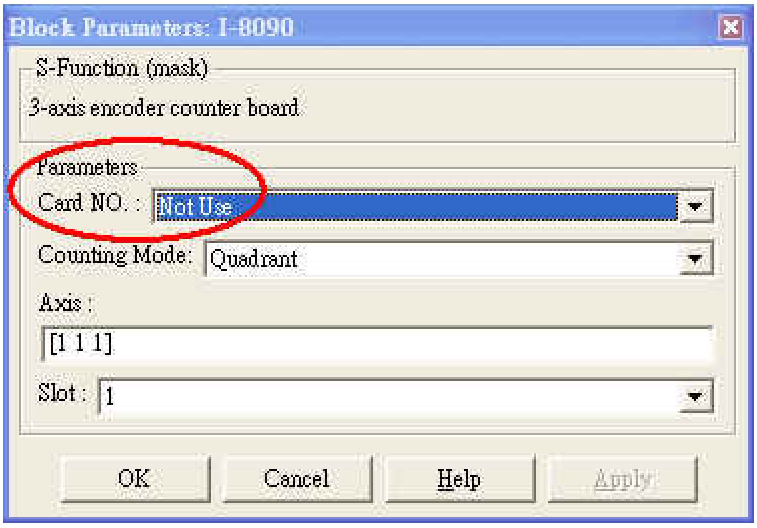


CW/CCW counting mode



Pulse/Direction counting mode

The “Card NO” field on the dialog of I-8090 driver block is not used in the LinCon-8000 and users don’t care this field..


Block Parameters: I-8090

S-Function (mask)

3-axis encoder counter board

Parameters

Card NO. : Not Use

Counting Mode: Quadrant

Axis :

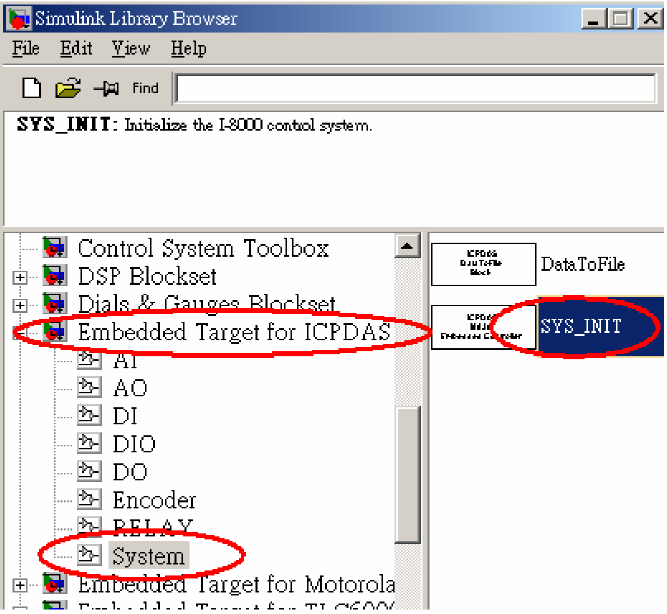
[1 1 1]

Slot : 1

Demonstration

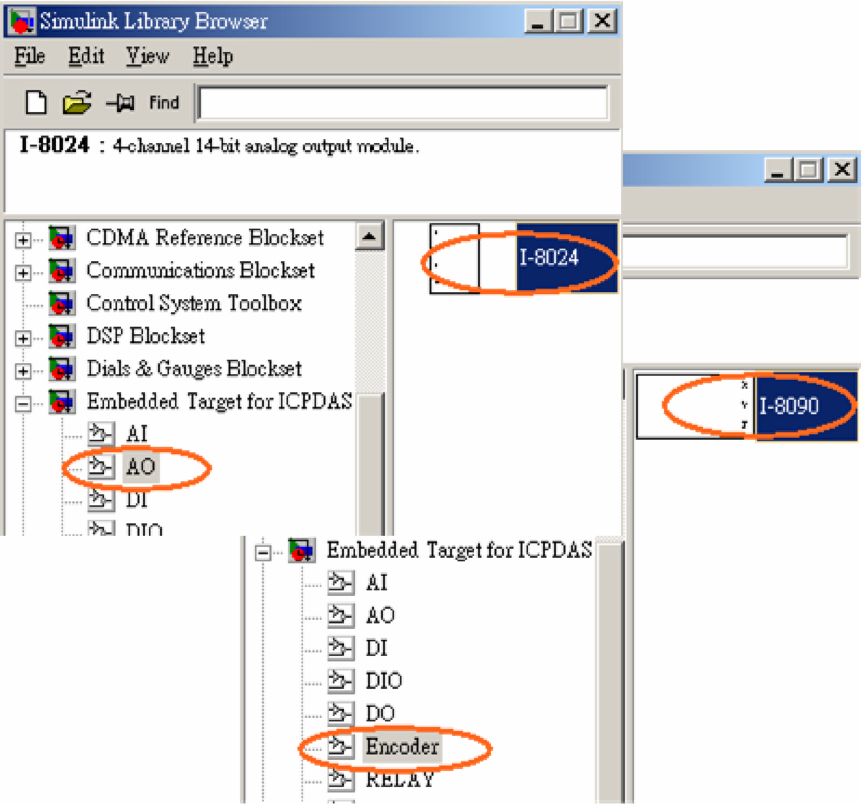
This demo uses an I-8024 AO module to drive the motor and an I-8090 encoder input module to accept the signals from the encoder of the motor. And we also add a DataToFile block to the model for data recording. Here, we will show you the procedure step by step.

Step 1: Create a new model in Simulink and insert a SYS_INIT block from the System block library.



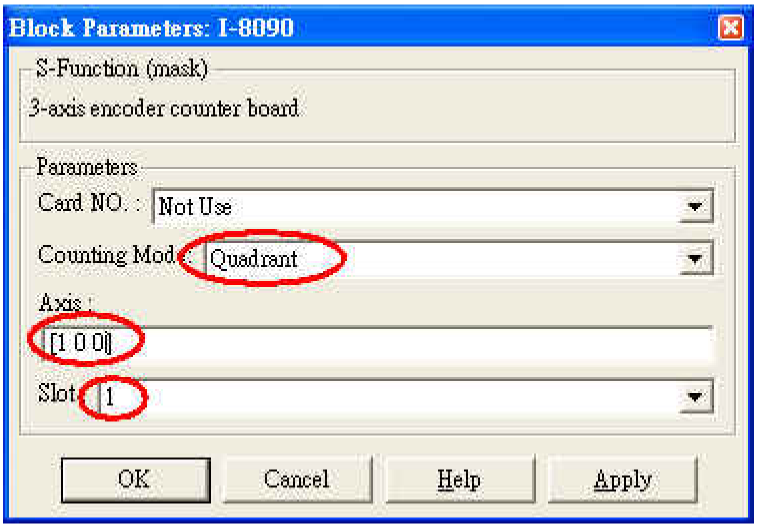
SYS_INIT: Initialize the I-8000 control system.

Step 2: Insert an I-8024 block and an I-8090 block separately from the AO and Encoder block library.



I-8024 : 4-channel 14-bit analog output module.

Step 3: Double click on the I-8090 block to setup the encoder module. Here we use X-axis of the I-8090 module, which is mounted on slot 1 of the I-8xx8 embedded controller. And select the counting mode as "Quadrant".



Block Parameters: I-8090

S-Function (mask)

3-axis encoder counter board

Parameters

Card NO. : Not Use

Counting Mode: Quadrant

Axis :

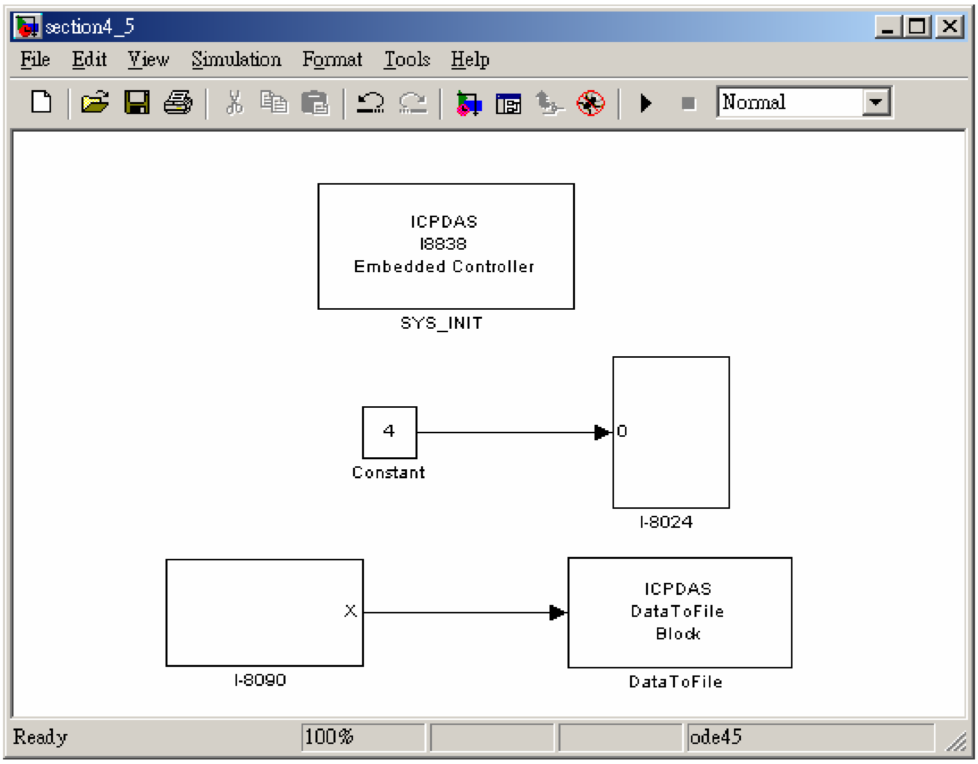
[1 0 0]

Slot: 1

Step 4: Configure the I-8024 block. (Please refer to the section 4.1 and 4.3.)

Step 5: Add a DataToFile block from the System block library, and use it to store the data of the encoder. At the same time, insert a Constant block to send the voltage command to the I-8024 block.

Step 6: Connect all blocks as shown in the following figure.



Step 7: Click Simulation parameters from the Simulation menu to open the Simulation Parameters dialog box. Then configure the RTW options

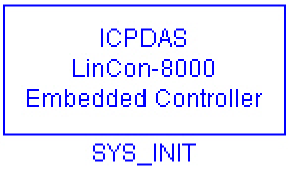
(refer to section 3.3) and press the Build button to start the build process.

Step 8: When the build process ends successfully, download the .exe file generated to the LinCon-8000 and run it (refer to section 3.4).

5. Matlab Driver Block Reference

This section presents detailed descriptions and usage of all blocks in the MATLAB Driver block library.

SYS_INIT



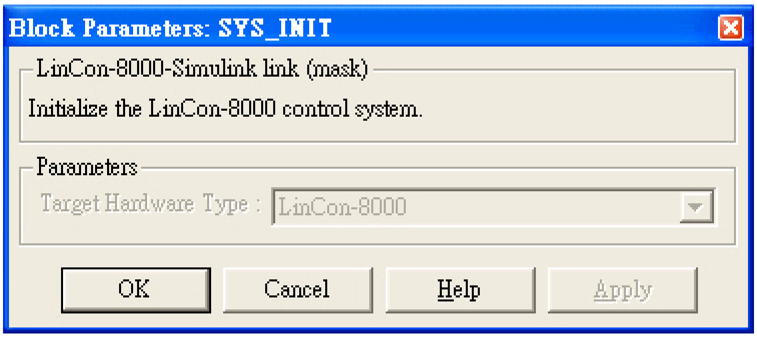
Description:

Initialize the LinCon-8000 control system.

Library:

System

Dialog Box:



Block Parameters: SYS_INIT

LinCon-8000-Simulink link (mask)

Initialize the LinCon-8000 control system.

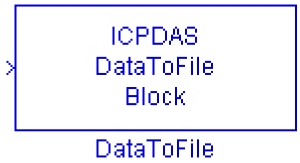
Parameters

Target Hardware Type : LinCon-8000

Driver Block Parameters:

Target Hardware Type - The type of the LinCon-8000 control system is fixed and users doesn’t setup it.

DataToFile



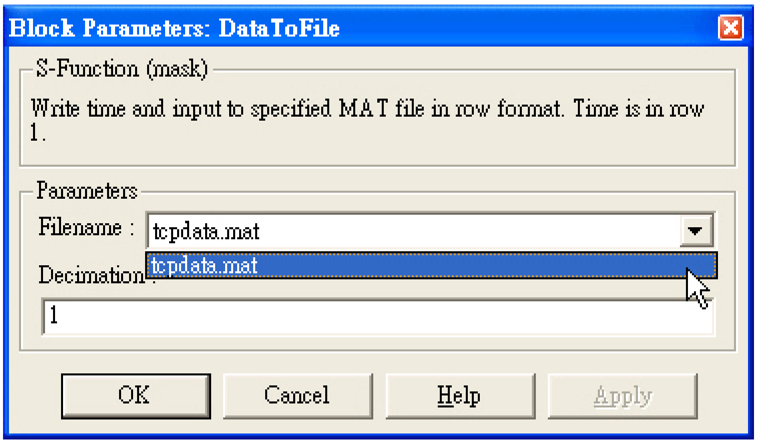
Description:

Write time and input to specified MAT file in row format. Time is in row 1.

Library:

System

Dialog Box:



S-Function (mask)

Write time and input to specified MAT file in row format. Time is in row 1.

Parameters

Filename : tcpdata.mat

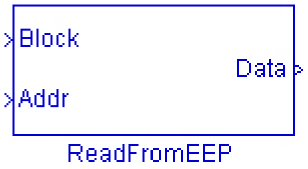
Decimation : 1

Driver Block Parameters:

Filename - The field is fixed and users can't modify it. The default filename is "tcpdata.mat". The file will be stored in the current working directory after uploading.

Decimation - A decimation factor. The default value is 1. This parameter allows you to write data at every n-th sample, where n is the decimation factor.

ReadFromEEP



Function:

Read data from EEPROM

Library:

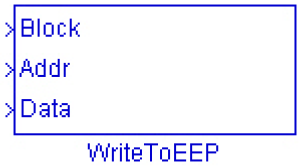
System

Description:

In this block, there is no parameter for users to set. To use this block properly, what users have to do is to assign the intended value to input ports, Block and Addr. And the data read from eeprom will be exported to the “Data” port. More details of these ports are given as follows：

1. **Block –** This parameter specifies the number of the eeprom’s block, from which you want to read. For LinCon-8000, its range is between 0 and 255.
2. **Addr –** A value between 0 and 64 is acceptable. In LinCon-8000, the eeprom is divided into 256 blocks, and each block is 64 bytes. If you want to obtain the data of Block 1, Offset 60., you would assign 1 to “Block” and 60 to “Addr”,
3. **Data –** It will output a value between 0 and 255 according to the setting of Block and Addr number.

WriteToEEP



Function:

Write data to EEPROM

Library:

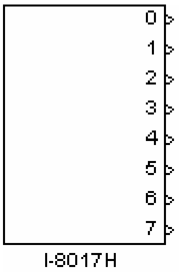
System

Description:

This block is used to write data to EEPROM, one byte per loop. To operate this block correctly, you need to keep some details in mind. They are given as follows:

1. **Block –** In the LinCon-8000 control system, the eeprom is divided into 256 blocks. The number of the block is counted from 0 to 255.
2. **Addr –** In each block of the eeprom, it is 64 bytes in size. So users can assign a value between 0 and 63 to this port.
3. **Data –** The data that you want to write to the eeprom. A value between 0 ~ 255 is meaningful.

I-8017H



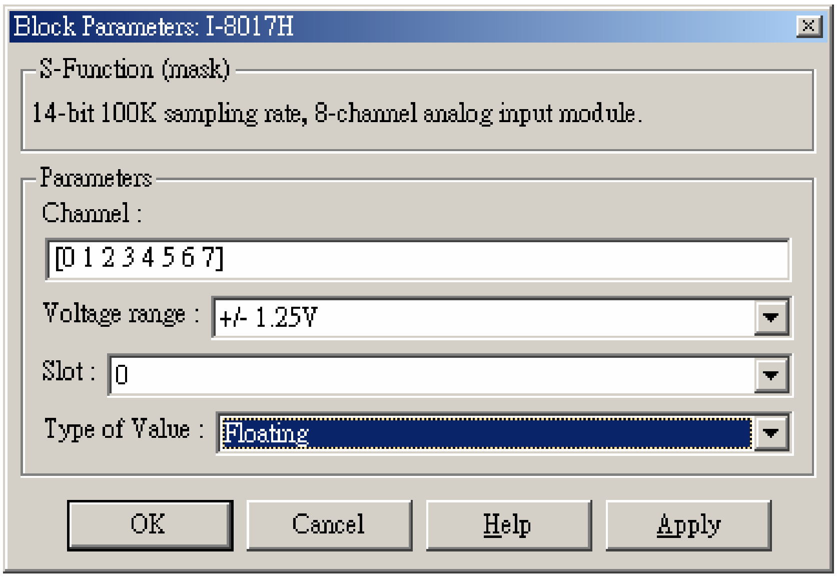
Description:

8-channel Isolated Analog Input Module.

Library:

AI

Dialog Box:



14-bit 100K sampling rate, 8-channel analog input module.

Channel : [0 1 2 3 4 5 6 7]

Voltage range : +/- 1.25V

Slot : 0

Type of Value : Floating

Driver Block Parameters:

Channel - Enter numbers between 0 and 7. This block allows the selection of analog input lines in any order. The number of elements defines the number of analog inputs used. For example, to use the first 8 analog inputs, enter [0 1 2 3 4 5 6 7]

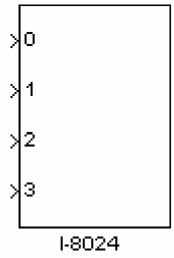
Voltage range - The I-8017H, AI module, provides 4 ranges of the input voltage, and they are +/-10V, +/-5V, +/-2.5V, and +/-1.25V.

Slot - The number of the slot where I-8017H module is located. For example, choose 2 from the popup list if you have mounted an I-8017H module on slot 2.

Type of Value - Floating or Hex is available. The table below presents the list of ranges of the input voltage.

Type of Value	Hardware Input	Block Output Value
Floating	-10 ~ 10V	-10 ~ 10
Hex	-10 ~ 10V	-8192 ~ 8191
Floating	-5 ~ 5V	-5 ~ 5
Hex	-5 ~ 5V	-8192 ~ 8191
Floating	-2.5 ~ 2.5V	-2.5 ~ 2.5
Hex	-2.5 ~ 2.5V	-8192 ~ 8191
Floating	-1.25 ~ 1.25V	-1.25 ~ 1.25
Hex	-1.25 ~ 1.25V	-8192 ~ 8191

I-8024



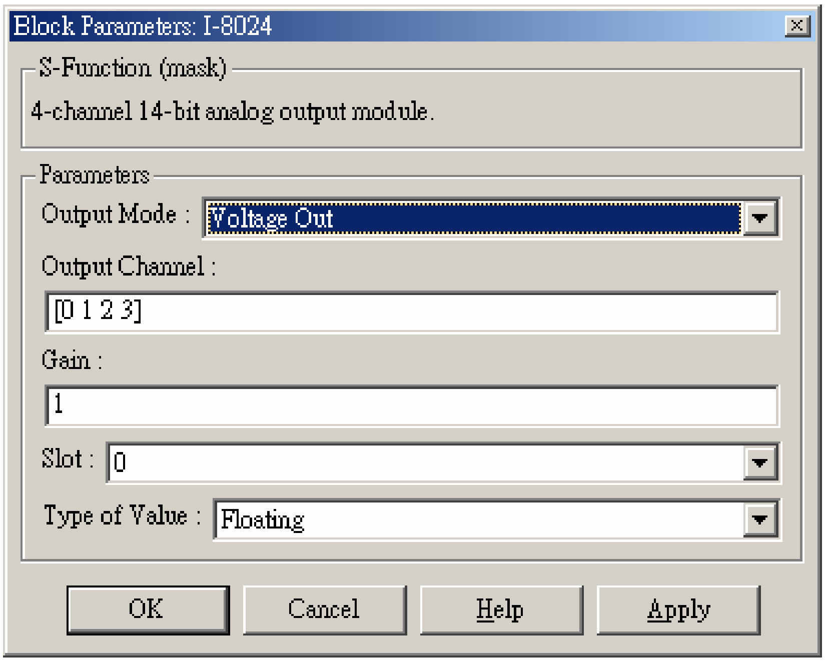
Description:

4-channel Isolated Analog Output Module.

Library:

AO

Dialog Box:



Block Parameters: I-8024

S-Function (mask)

4-channel 14-bit analog output module.

Parameters

Output Mode : Voltage Out

Output Channel :

[0 1 2 3]

Gain :

1

Slot : 0

Type of Value : Floating

Driver Block Parameters:

Output Mode – *Voltage Out* or *Current Out* is available.

- Voltage Output -> +/-10V
- Current Output -> 0~20mA

Output channel - Enter numbers between 0 and 3. This block allows the selection of analog output lines in any order. The number of elements defines the number of analog outputs used. For example, to use the first 4 analog outputs, enter [0 1 2 3]

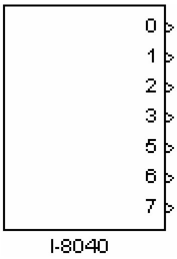
Gain - A multiplier. The default value is 1.

Slot - The number of the slot where I-8024 module is located. For example, choose 2 from the popup list if you have mounted an I-8024 module on slot 2.

Type of Value - Floating or Hex is available. The following table presents the difference between them.

Type of Value	Block Input Data Range	Hardware Output
Floating	-10 ~ 10 (Voltage Out) 0 ~ 20 (Current Out)	-10 ~ 10V (Voltage Out) 0 ~ 20mA (Current Out)
Hex	0 ~ 16383 (Both)	-10 ~ 10V (Voltage Out) 0 ~ 20mA (Current Out)

I-8040



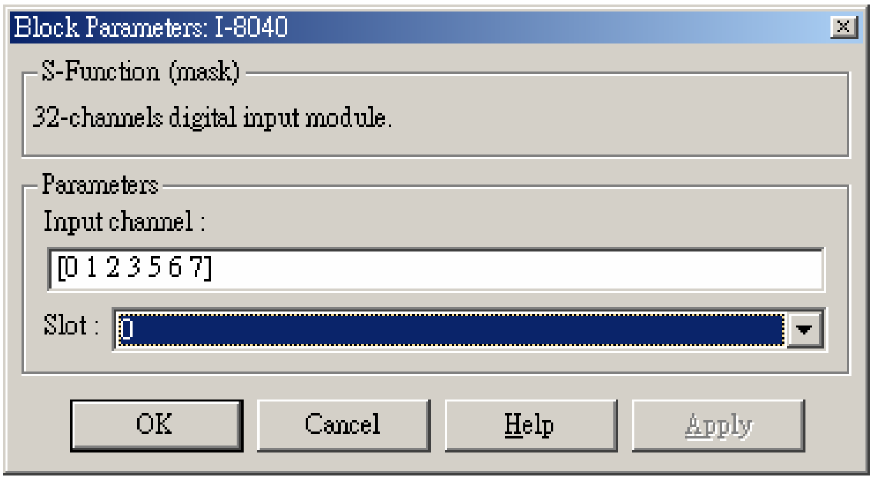
Description:

32-channel Isolated Digital Input Module.

Library:

DI

Dialog Box:


Block Parameters: I-8040

S-Function (mask)

32-channels digital input module.

Parameters

Input channel :

[0 1 2 3 5 6 7]

Slot : 0

Driver Block Parameters:

Input channel - Enter numbers between 0 and 31. This block allows the selection of individual digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 8 digital inputs, enter

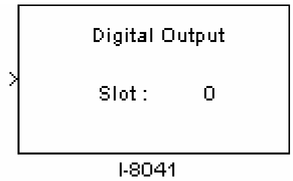
[0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8040 module is mounted. For example, choose 2 from the popup list if you have mounted an I-8040 module on slot 2.

Scaling Input to Output:

Hardware Input	Block Output Value
Below 3.5V	= 0
3.5V ~ 30V	= 1

I-8041



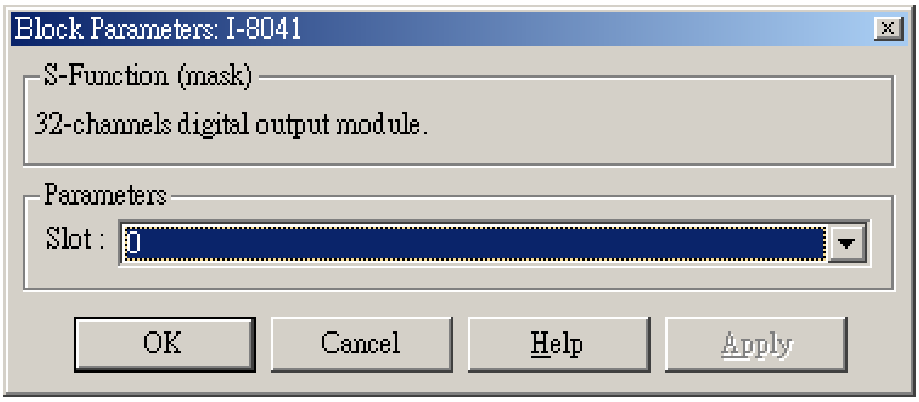
Description:

32-channel Isolated Digital Output Module.

Library:

DO

Dialog Box:


Block Parameters: I-8041

S-Function (mask)

32-channels digital output module.

Parameters

Slot : 0

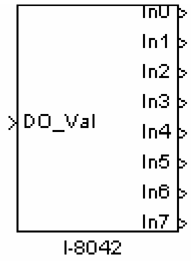
Driver Block Parameters:

Slot - The number of the slot where I-8041 module is located. For example, choose 2 from the popup list if you have mounted an I-8041 module on slot 2.

Scaling Input to Output:

BLOCK INPUT VALUE	HARDWARE OUTPUT
0	All channels are off.
1	Ch0 is on, and the others are off.
2	Ch1 is on, and the others are off.
…	…
2^32 - 1	All channels are on.

I-8042



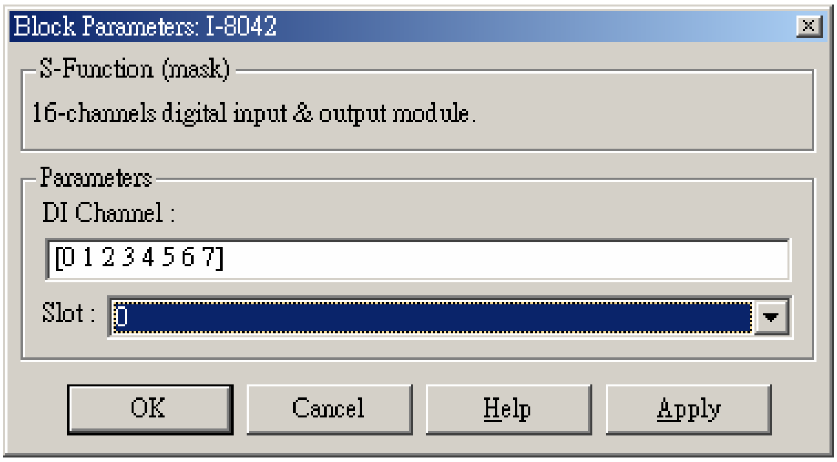
Description:

16-channel Isolated Digital Input & 16-channel Isolated Digital Output Module.

Library:

DIO

Dialog Box:



Block Parameters: I-8042

S-Function (mask)

16-channels digital input & output module.

Parameters

DI Channel :

[0 1 2 3 4 5 6 7]

Slot : 0

Driver Block Parameters:

DI Channel - Enter numbers between 0 and 15. This block allows the selection of digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 8 digital inputs, enter [0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8042 module is located. For example, choose 2 from the popup list if you have mounted an I-8042 module on slot 2.

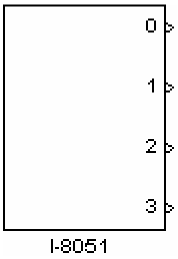
Scaling Input to Output (digital input):

Hardware Input	Block Output Value
Below 3.5V	= 0
3.5V ~ 30V	= 1

Scaling Input to Output (digital output):

Block Input Value	Hardware Output
0	All channels are off.
1	Ch0 is on, and the others are off.
2	Ch1 is on, and the others are off.
…	…
65535	All channels are on.

I-8051



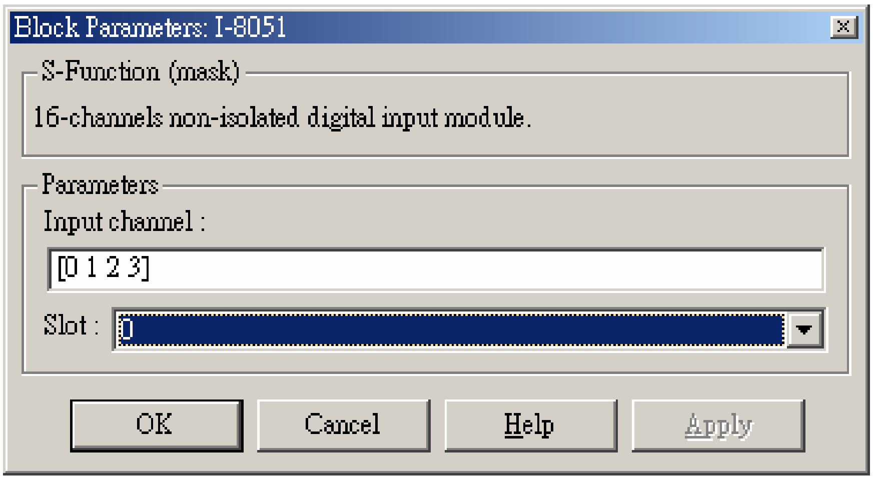
Description:

16-channel Non-isolated Digital Input Module.

Library:

DI

Dialog Box:



S-Function (mask)

16-channels non-isolated digital input module.

Parameters

Input channel :

[0 1 2 3]

Slot : 0

Driver Block Parameters:

Input channel - Enter numbers between 0 and 15. This block allows the selection of individual digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 8 digital inputs, enter

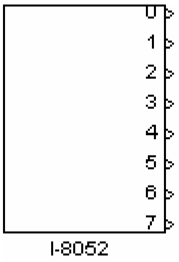
[0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8051 module is mounted. For example, choose 2 from the popup list if you have mounted an I-8051 module on slot 2.

Scaling Input to Output:

Hardware Input	Block Output Value
Below 1V	= 0
3.5V ~ 30V	= 1

I-8052



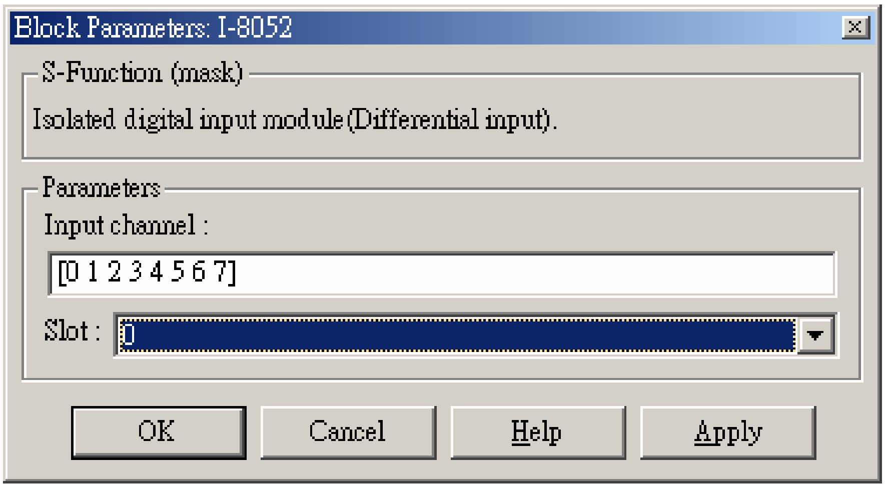
Description:

8-channel Isolated Digital Input Module (Differential input).

Library:

DI

Dialog Box:


Block Parameters: I-8052

S-Function (mask)

Isolated digital input module (Differential input).

Parameters

Input channel :

[0 1 2 3 4 5 6 7]

Slot : 0

Driver Block Parameters:

Input channel - Enter numbers between 0 and 7. This block allows the selection of individual digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 8 digital inputs, enter

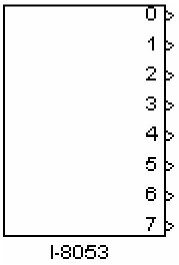
[0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8052 module is located. For example, select 2 from the popup list if you have mounted an I-8052 module on slot 2.

Scaling Input to Output:

Hardware Input	Block Output Value
Below 1V	= 0
3.5V ~ 30V	= 1

I-8053



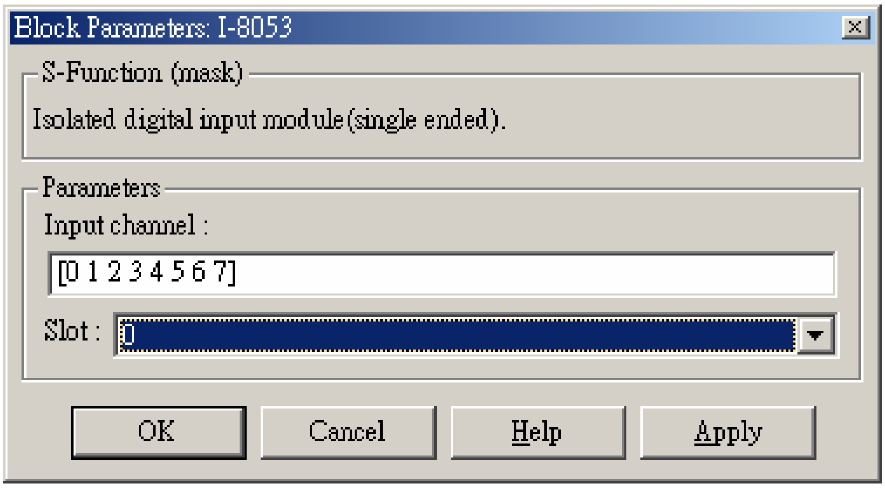
Description:

16-channel Isolated Digital Input Module (single ended).

Library:

DI

Dialog Box:


Block Parameters: I-8053

S-Function (mask)

Isolated digital input module (single ended).

Parameters

Input channel :

[0 1 2 3 4 5 6 7]

Slot : 0

Driver Block Parameters:

Input channel - Enter numbers between 0 and 15. This block allows the selection of individual digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 8 digital inputs, enter

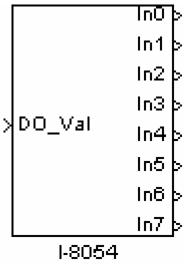
[0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8053 module is located. For example, select 2 from the popup list if you have mounted an I-8053 module on slot 2.

Scaling Input to Output:

Hardware Input	Block Output Value
Below 1V	= 0
3.5V ~ 30V	= 1

I-8054



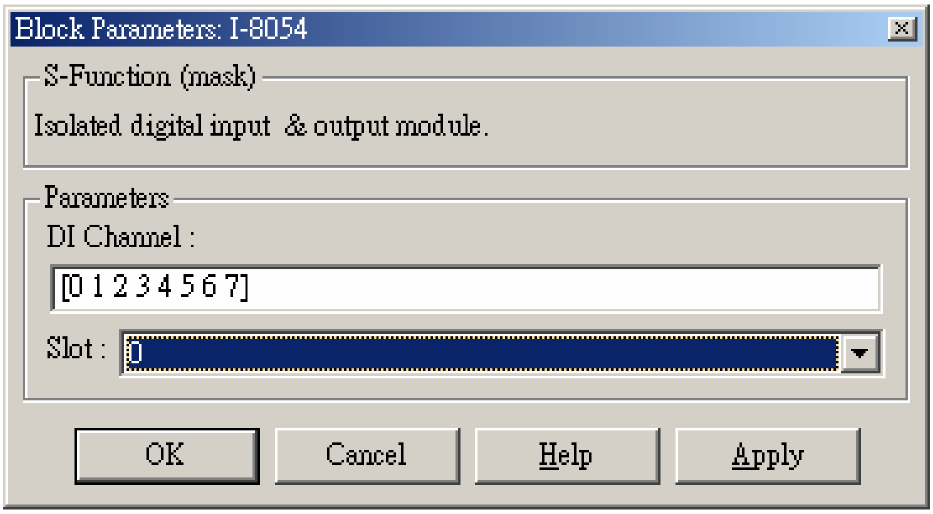
Description:

16-channel Isolated Digital I/O Module.

Library:

DIO

Dialog Box:



Block Parameters: I-8054

S-Function (mask)

Isolated digital input & output module.

Parameters

DI Channel :

[0 1 2 3 4 5 6 7]

Slot : 0

Driver Block Parameters:

DI Channel - Enter numbers between 0 and 7. This block allows the selection of digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 8 digital inputs, enter [0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8054 module is located. For example, choose 2 from the popup list if you have mounted an I-8054 module on slot 2.

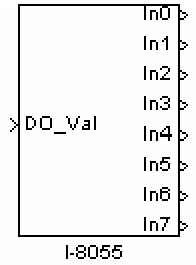
Scaling Input to Output (digital input):

Hardware Input	Block Output Value
Below 1V	= 0
3.5V ~ 30V	= 1

Scaling Input to Output (digital output):

Block Input Value	Hardware Output
0	All channels are off.
1	Ch0 is on, and the others are off.
2	Ch1 is on, and the others are off.
…	…
255	All channels are on.

I-8055



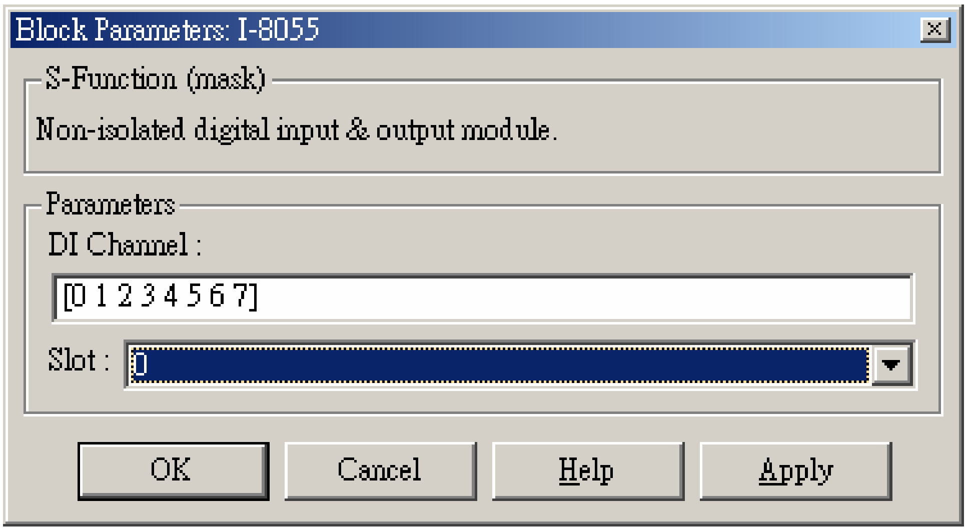
Description:

16-channel Non-isolated Digital I/O Module.

Library:

DIO

Dialog Box:



S-Function (mask)

Non-isolated digital input & output module.

Parameters

DI Channel :

[0 1 2 3 4 5 6 7]

Slot : 0

Driver Block Parameters:

DI Channel - Enter numbers between 0 and 7. This block allows the selection of digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 8 digital inputs, enter [0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8055 module is located. For example, choose 2 from the popup list if you have mounted an I-8055 module on slot 2.

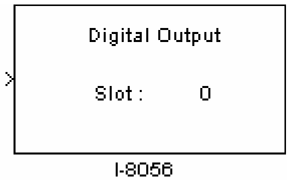
Scaling Input to Output (digital input):

Hardware Input	Block Output Value
Below 1V	= 0
3.5V ~ 30V	= 1

Scaling Input to Output (digital output):

Block Input Value	Hardware Output
0	All channels are off.
1	Ch0 is on, and the others are off.
2	Ch1 is on; the others are off.
…	…
255	All channels are on.

I-8056



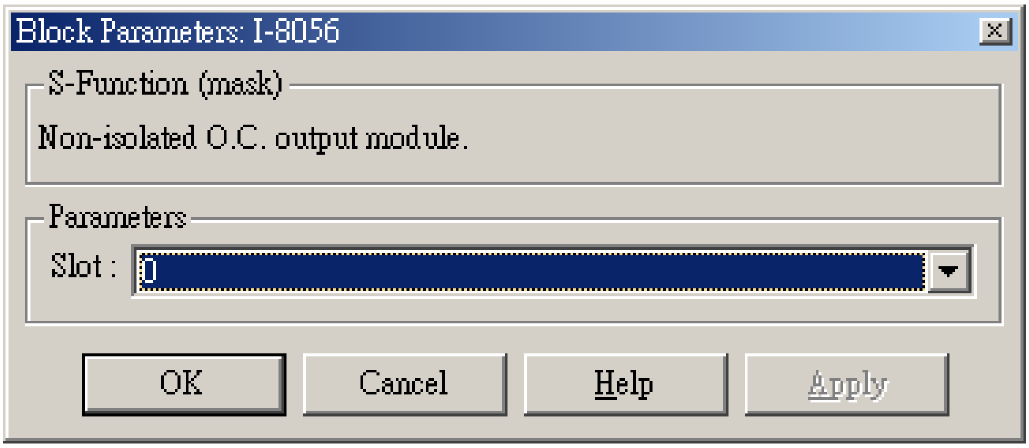
Description:

16-channel Non-isolated O.C. Output Module.

Library:

DO

Dialog Box:


Block Parameters: I-8056

S-Function (mask)

Non-isolated O.C. output module.

Parameters

Slot : 0

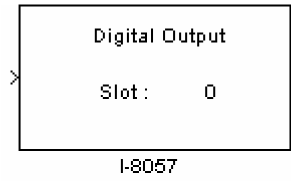
Driver Block Parameters:

Slot - The number of the slot where I-8056 module is located. For example, choose 2 from the popup list if you have mounted an I-8056 module on slot 2.

Scaling Input to Output:

BLOCK INPUT VALUE	HARDWARE OUTPUT
0	All channels are off.
1	Ch0 is on, and the others are off.
2	Ch1 is on, and the others are off.
…	…
65535	All channels are on.

I-8057



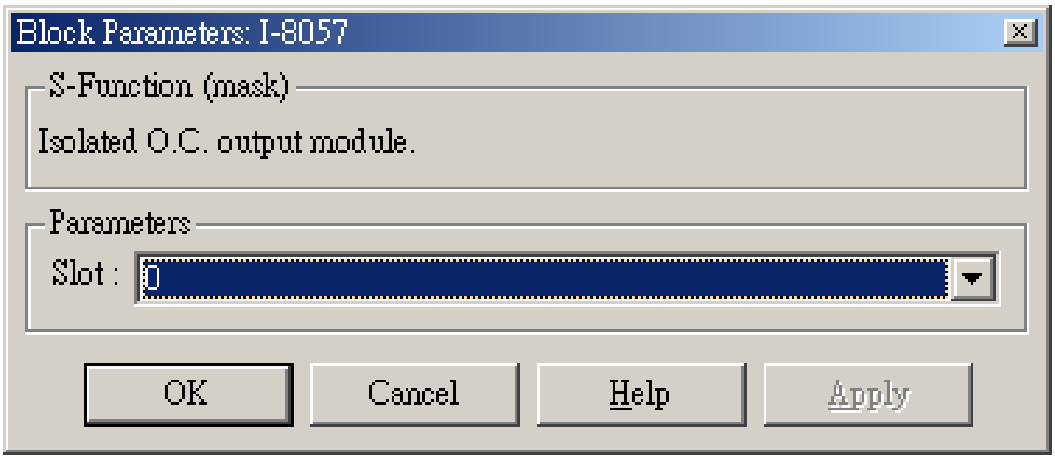
Description:

16-channel Isolated O.C. Output Module.

Library:

DO

Dialog Box:


Block Parameters: I-8057

S-Function (mask)

Isolated O.C. output module.

Parameters

Slot : 0

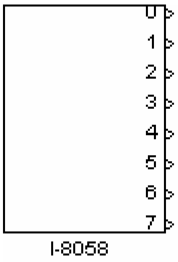
Driver Block Parameters:

Slot - The number of the slot where I-8057 module is located. For example, choose 2 from the popup list if you have mounted an I-8057 module on slot 2.

Scaling Input to Output:

BLOCK INPUT VALUE	HARDWARE OUTPUT
0	All channels are off.
1	Ch0 is on, and the others are off.
2	Ch1 is on, and the others are off.
…	…
65535	All channels are on.

I-8058



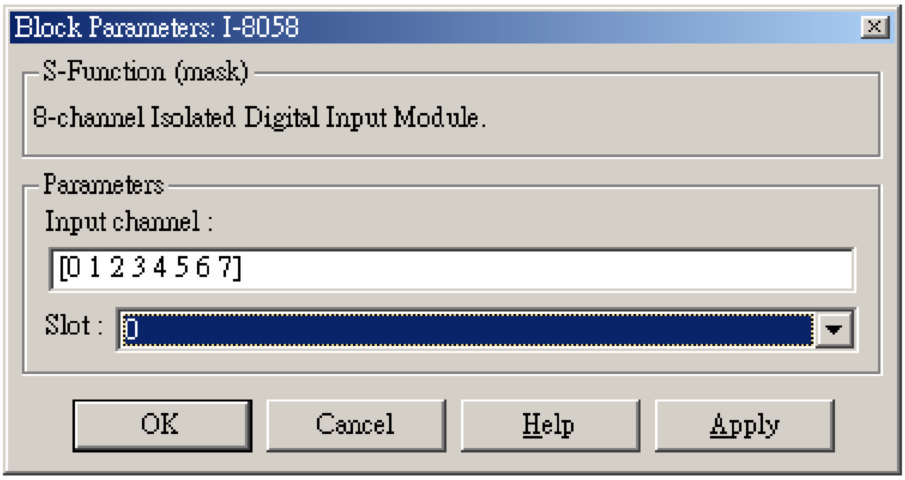
Description:

8-channel Isolated Digital Input Module.

Library:

DI

Dialog Box:



S-Function (mask)

8-channel Isolated Digital Input Module.

Parameters

Input channel :

[0 1 2 3 4 5 6 7]

Slot : 0

Driver Block Parameters:

Input channel - Enter numbers between 0 and 7. This block allows the selection of individual digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 8 digital inputs, enter

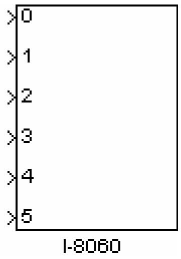
[0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8058 module is located. For example, choose 2 from the popup list if you have mounted an I-8058 module on slot 2.

Scaling Input to Output:

Hardware Input	Block Output Value
AC/DC 30V max.	= 0
AC/DC 80V mini	= 1

I-8060



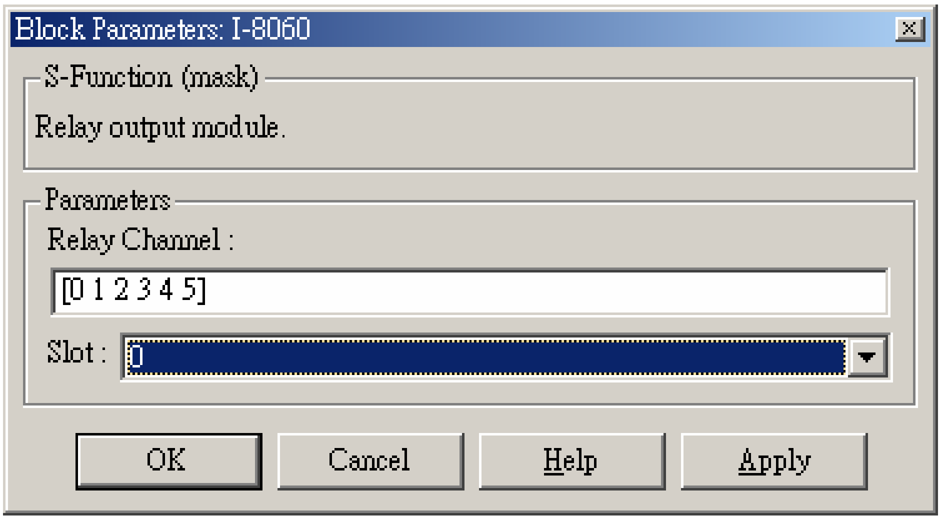
Description:

6-channel Relay Output Module.

Library:

RELAY

Dialog Box:


Block Parameters: I-8060

S-Function (mask)

Relay output module.

Parameters

Relay Channel :

[0 1 2 3 4 5]

Slot : 0

Driver Block Parameters:

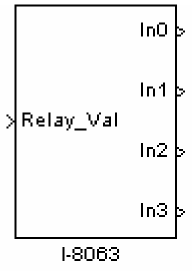
Relay Channel - Enter numbers between 0 and 5. This block allows the selection of relay output lines in any order. The number of elements defines the number of relay outputs used. For example, to use the first 6 relay outputs, enter [0 1 2 3 4 5]

Slot - The number of the slot where I-8060 module is located. For example, choose 2 from the popup list if you have mounted an I-8060 module on slot 2.

Scaling Input to Output:

Block Input Value	Hardware Output
> 0	Switch to NO
<=0	Switch to NC

I-8063



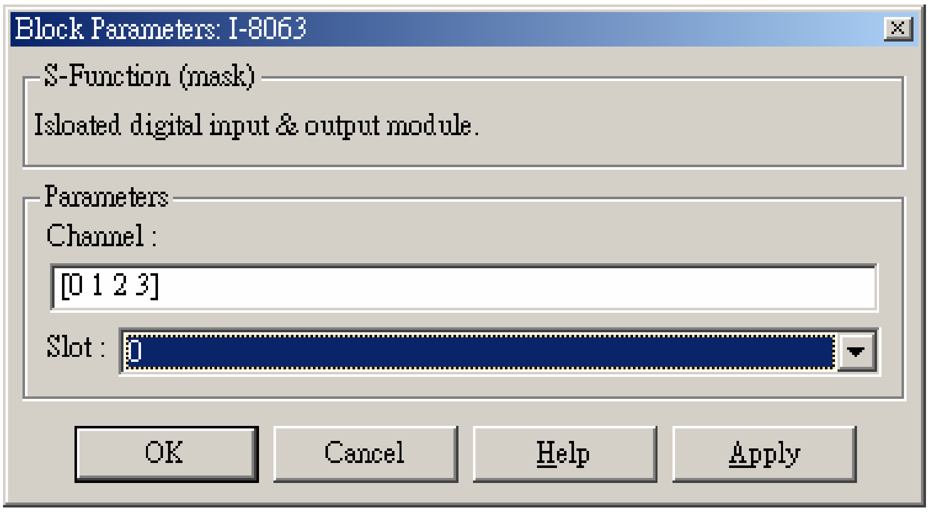
Description:

8-channel Isolated Digital I/O Module.

Library:

DIO

Dialog Box:



Isloated digital input & output module.

Channel :

[0 1 2 3]

Slot : 0

Driver Block Parameters:

DI Channel - Enter numbers between 0 and 3. This block allows the selection of digital input lines in any order. The number of elements defines the number of digital inputs used. For example, to use the first 4 digital inputs, enter [0 1 2 3]

Slot - The number of the slot where I-8063 module is located. For example, choose 2 from the popup list if you have mounted an I-8063 module on slot 2.

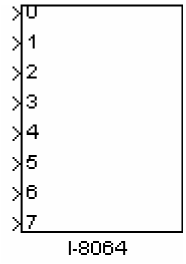
Scaling Input to Output (digital input):

Hardware Input	Block Output Value
Below 1V	= 0
3.5V ~ 30V	= 1

Scaling Input to Output (relay output):

Block Input Value	Hardware Output
0	All channels are switched to NO.
1	Ch0 is switched to NO, and the others remain NC.
2.	Ch1 is switched to NO, and the others remain NC.
…	…
7	All channels are switched to NC.

I-8064



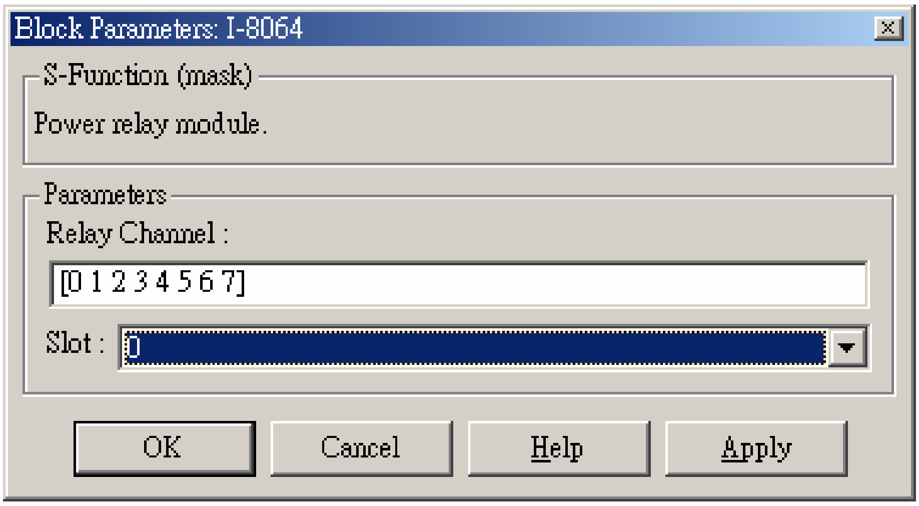
Description:

8-channel Power Relay Output Module.

Library:

RELAY

Dialog Box:


Block Parameters: I-8064

S-Function (mask)

Power relay module.

Parameters

Relay Channel :

[0 1 2 3 4 5 6 7]

Slot : 0

Driver Block Parameters:

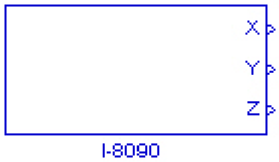
Relay Channel - Enter numbers between 0 and 7. This block allows the selection of relay output lines in any order. The number of elements defines the number of relay outputs used. For example, to use the first 8 relay outputs, enter [0 1 2 3 4 5 6 7]

Slot - The number of the slot where I-8064 module is located. For example, choose 2 from the popup list if you have mounted an I-8064 module on slot 2.

Scaling Input to Output:

Block Input Value	Hardware Output
> 0	Switch to NO
<=0	Switch to off

I-8090



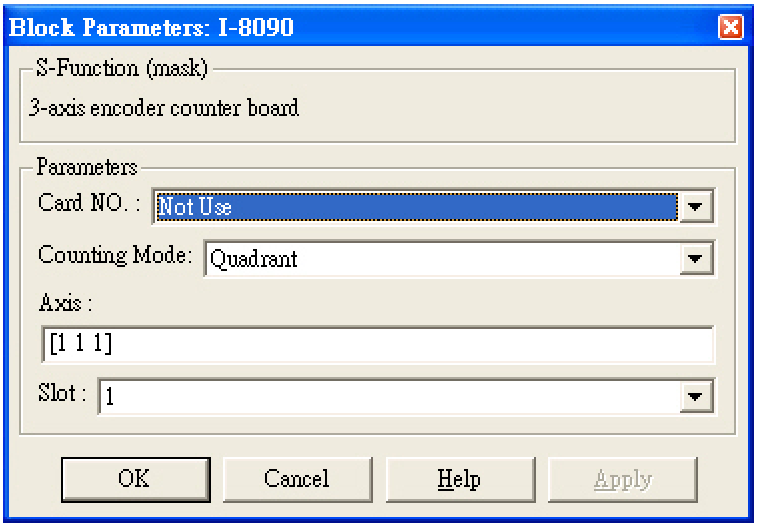
Function:

3-axis Encoder Input Module.

Library:

Encoder

Dialog Box:



Block Parameters: I-8090

S-Function (mask)

3-axis encoder counter board

Parameters

Card NO. : Not Use

Counting Mode: Quadrant

Axis :

[1 1 1]

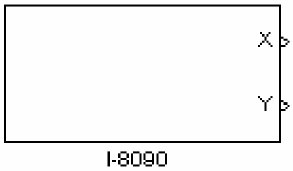
Slot : 1

Driver Block Parameters:

Card NO - The number is not used for LinCon-8000.

Counting Mode - *Quadrant*, *CW/CCW*, and *pulse/direction* is available. This setting depends on the wire connections of the I-8090 encoder module.

Axis - The selected axis vector. The default value is [1 1 0]. Each of the elements in the vector is used to determine which axis is selected or not. For example, if we enter [1 1 0] in the "Axis" field, the I-8090 driver block should look like as follows:

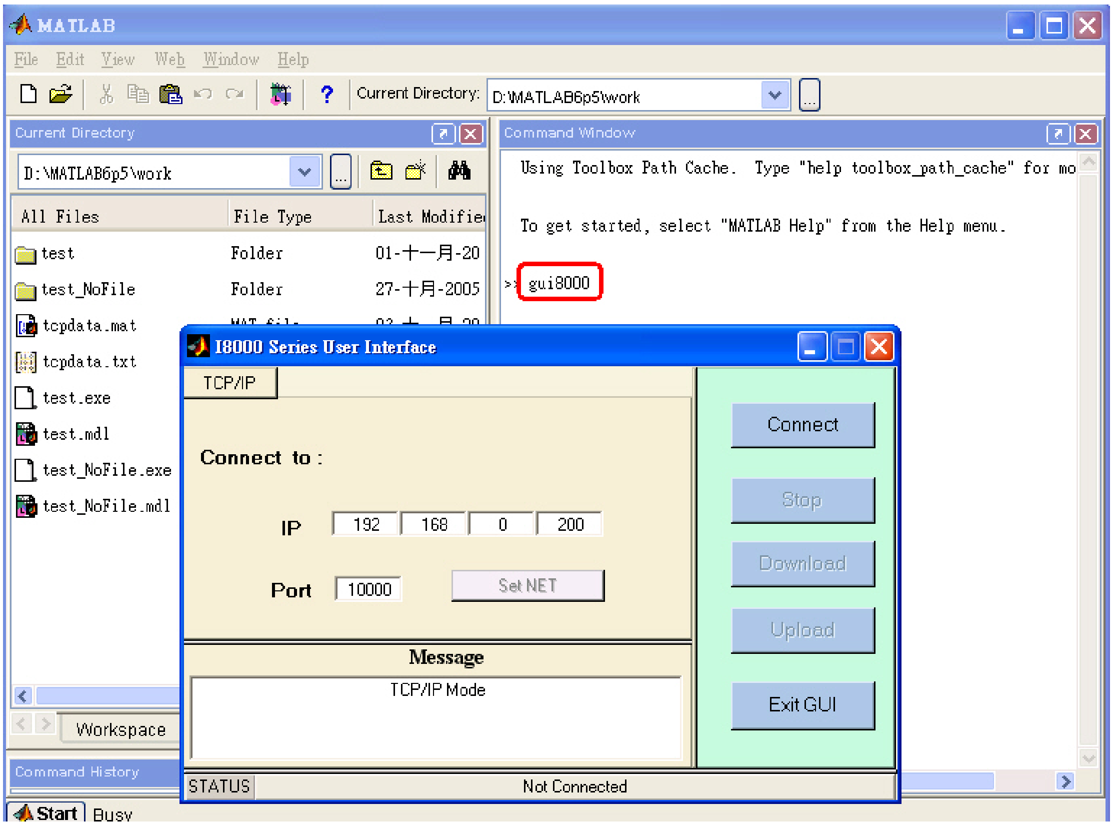


Slot - The number of the slot where I-8090 module is located. For example, choose 2 from the popup list if you have mounted an I-8090 module on slot 2.

Appendix A. Change the IP of LinCon-8000

Through the GUI under MATLAB, users can change the IP, Mask and Gateway of LinCon-8000 embedded controller conveniently. We will demonstrate how to complete this work.

Step 1: Start MATLAB, and then enter **gui8000** at the MATLAB prompt.



Current Directory: D:\MATLAB6p5\work

All Files	File Type	Last Modifie
test	Folder	01-十一月-20
test_NoFile	Folder	27-十月-2005
tcpdata.mat		
tcpdata.txt		
test.exe		
test.mdl		
test_NoFile.exe		
test_NoFile.mdl		

```
Using Toolbox Path Cache.  Type "help toolbox_path_cache" for mo

To get started, select "MATLAB Help" from the Help menu.

>> gui8000
```

Connect to :

IP 192 168 0 200

Port 10000

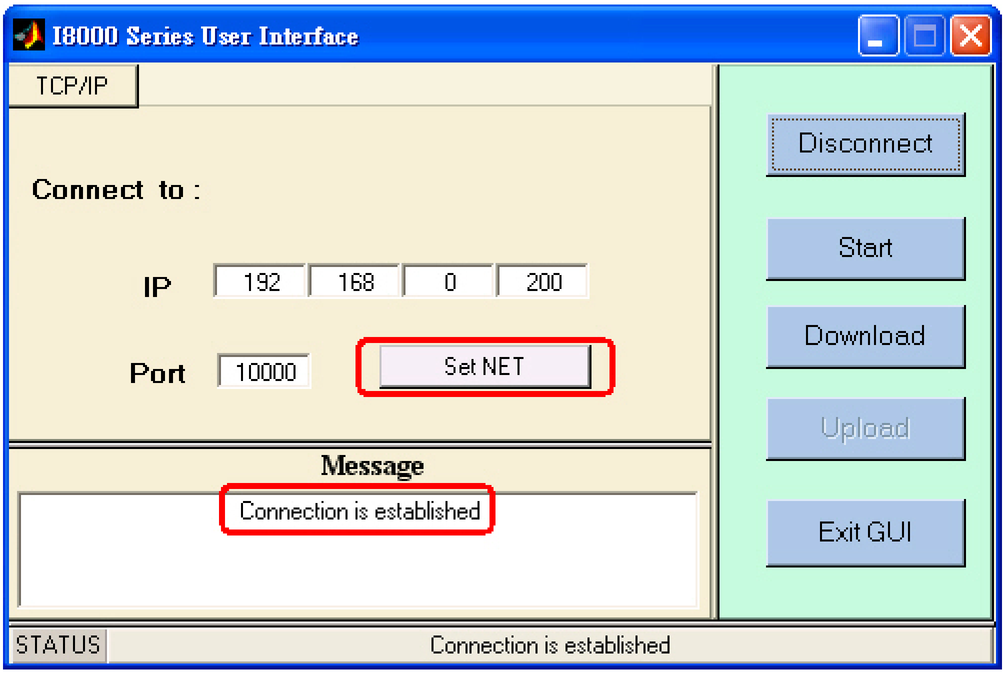
Message

TCP/IP Mode

STATUS Not Connected

Step 2: Enter **LinCon IP address** (like : 192.168.0.200) in the field ‘IP’ and **10000** in the field ‘Port’ respectively. Then click **Connect** to establish a connection with LinCon-8000.

Step 3: When the connection was established successfully, the grayed button **Set NET** is enabled.



Connect to :

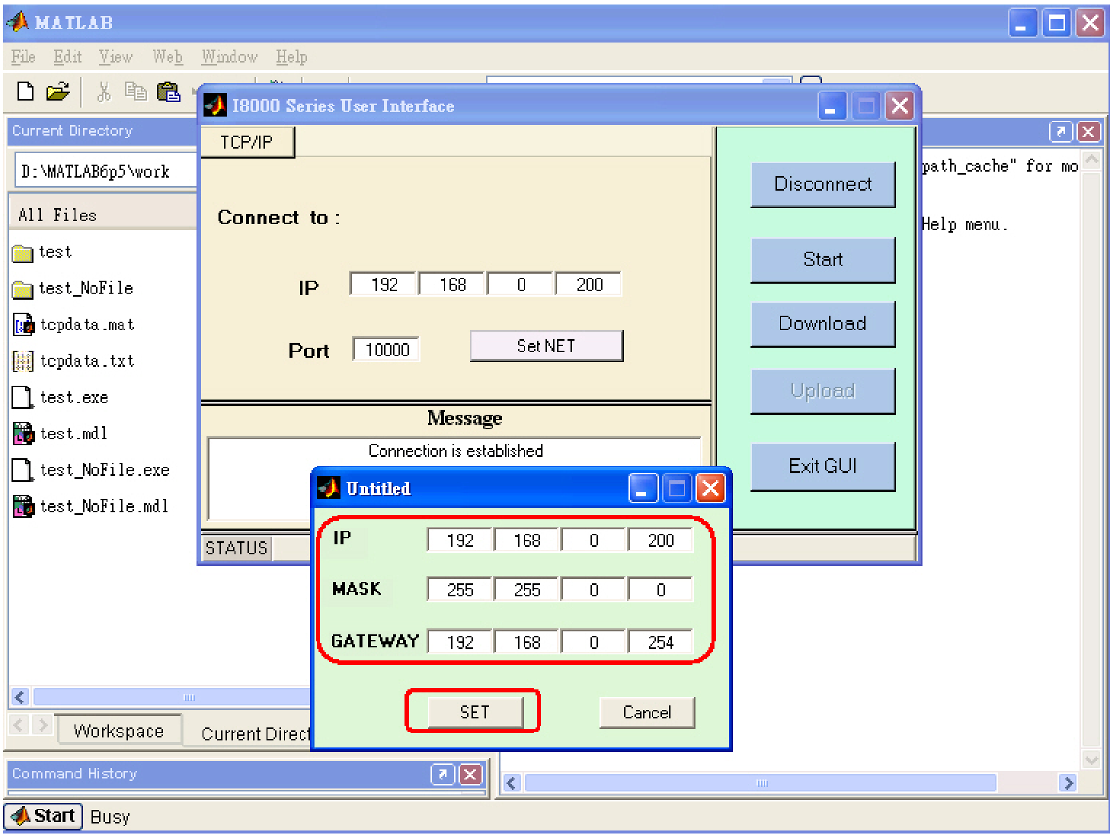
IP 192 168 0 200

Port 10000

Connection is established

STATUS Connection is established

Step 4: Click SetNET to open the setting window as shown in the figure below. Enter intended ip, mask, gateway in the field ‘IP’, ‘MASK’, and ‘GATEWAY’ respectively. Then press **SET** to change the setting.



D:\MATLAB6p5\work

path_cache" for mo

Help menu.

Connect to :

IP 192 168 0 200

Port 10000

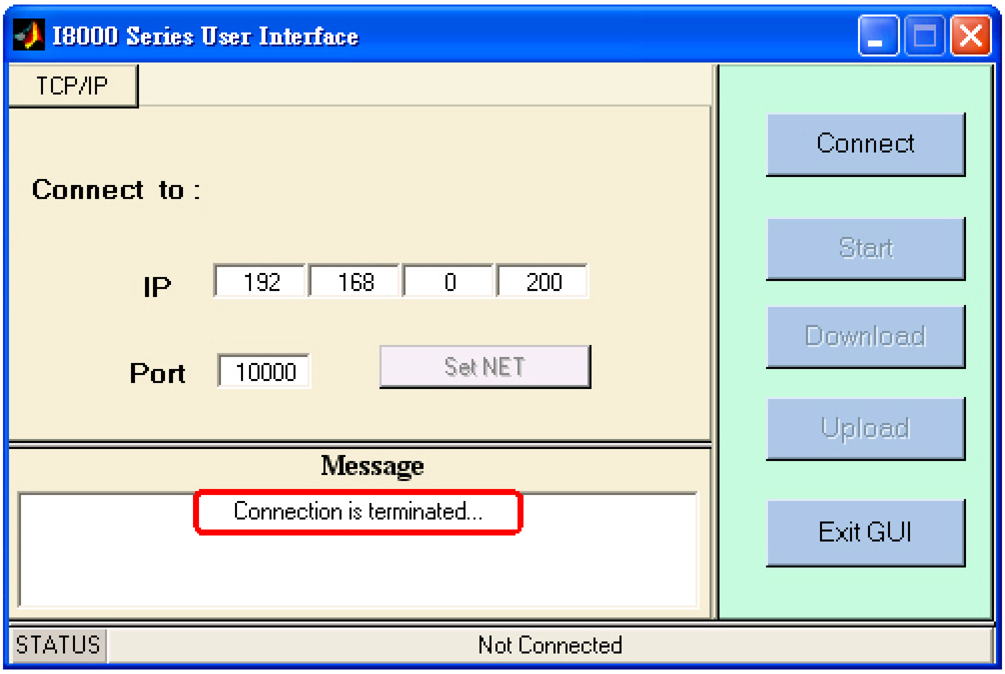
Connection is established

IP 192 168 0 200

MASK 255 255 0 0

GATEWAY 192 168 0 254

Step 5: If the process was completed successfully, the connection will be terminated. Users will need to do the connection again.



Connection is terminated...

Not Connected