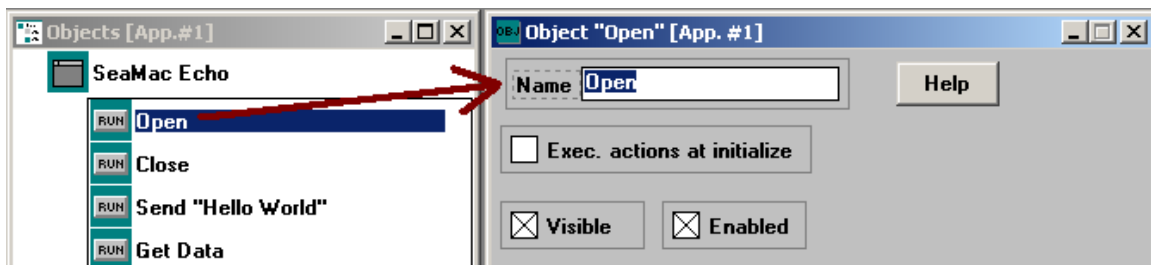
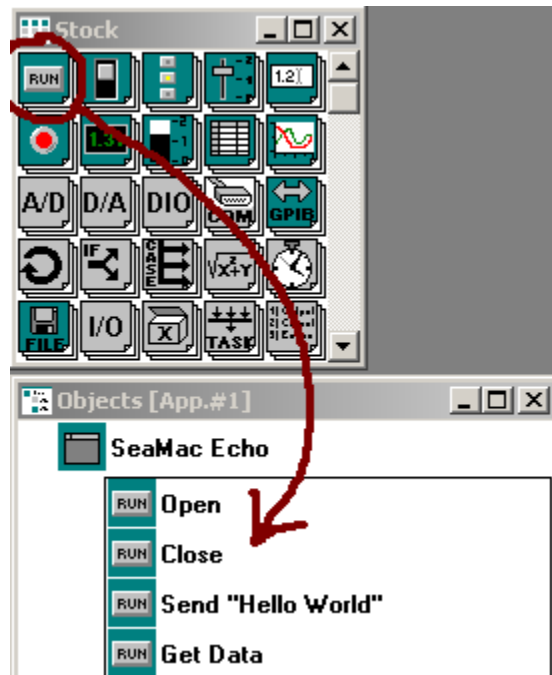


Installing a Sealevel Synchronous Card With TestPoint:

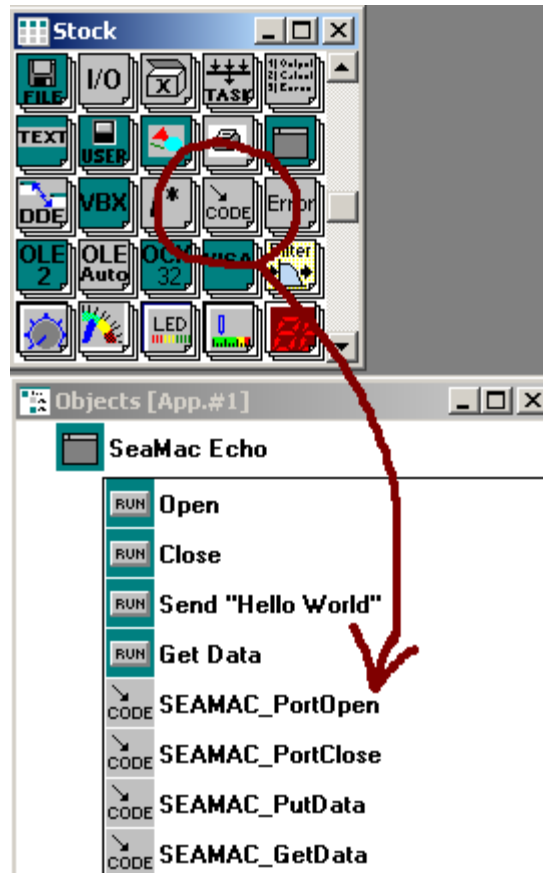
It is very easy to use any Sealevel Synchronous 85230 card with TestPoint. Install the card as per the directions in SeaMAC.hlp or your manual, and use the “SeaMAC Drivers” icon in the control panel to find out what port the card is installed as.

The method of attaching external APIs, (or any arbitrary .DLL,) to TestPoint is given in chapter 23 of the manual. For each call into the DLL, a separate "CODE" icon is brought into your list of objects. The "DLL filename" is SEAMAC32.DLL, and the "Subroutine Name" is any of the calls listed in the API. (Do not include the parentheses in the "Subroutine Name.")

Lets get started. First drag four “Pushbutton” objects to the objects list and name them as shown in the examples.



Now that we have the Buttons on the form lets add the code that makes it all work! Drag four “Code” objects to the object list.



Double clicking on one of the code objects will bring up the object property dialog box. Open all the code property dialog boxes and set up as per the examples below.

The image shows two overlapping windows from a software application. The top window is titled "Object 'SEAMAC_PortOpen' [App. #1]". It contains the following fields: "Name" (SEAMAC_PortOpen), "DLL Filename" (SEAMAC32.DLL), "Subroutine Name" (SEAMAC_PortOpen), "Argument Types" (word, byte, var dword), "Return Type" (dword), and a checked "Preload" checkbox. A "Help" button is located in the top right. Below the fields are tabs for "Settings", "Actions", "Comments", and "XRef".

The bottom window is titled "Object 'SEAMAC_PortClose' [App. #1]". It contains the following fields: "Name" (SEAMAC_PortClose), "DLL Filename" (SEAMAC32.DLL), "Subroutine Name" (SEAMAC_PortClose), "Argument Types" (dword), "Return Type" (dword), and a checked "Preload" checkbox. A "Help" button is located in the top right. Below the fields are tabs for "Settings", "Actions", "Comments", and "XRef".

Object "SEAMAC_PutData" [App. #1]

Name: SEAMAC_PutData [Help]

DLL Filename: SEAMAC32.DLL

Subroutine Name: SEAMAC_PutData

Argument Types: dword, var char, dword

Return Type: dword

Preload

Settings | Actions | Comments | XRef

Object "SEAMAC_GetData" [App. #1]

Name: SEAMAC_GetData [Help]

DLL Filename: SEAMAC32.DLL

Subroutine Name: SEAMAC_GetData

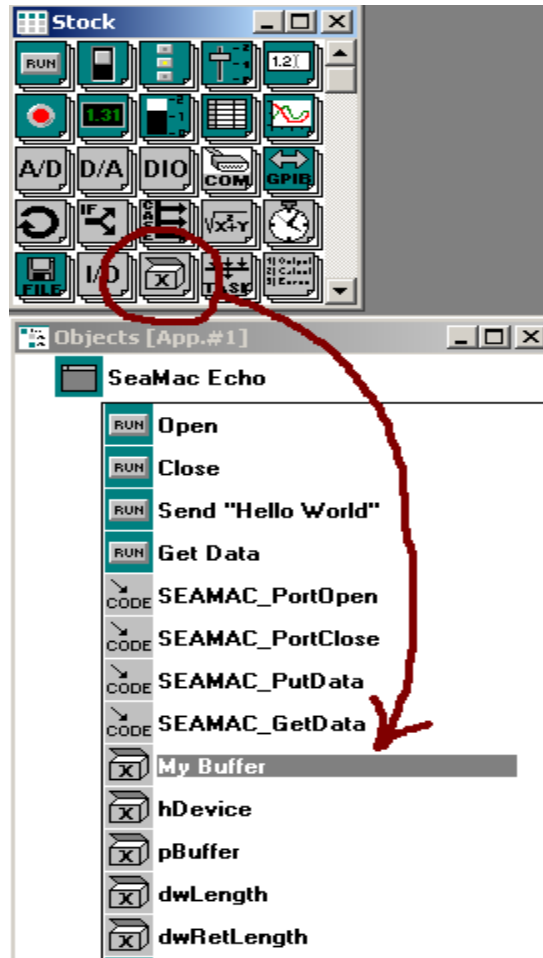
Argument Types: dword, var char, dword, var dword

Return Type: dword

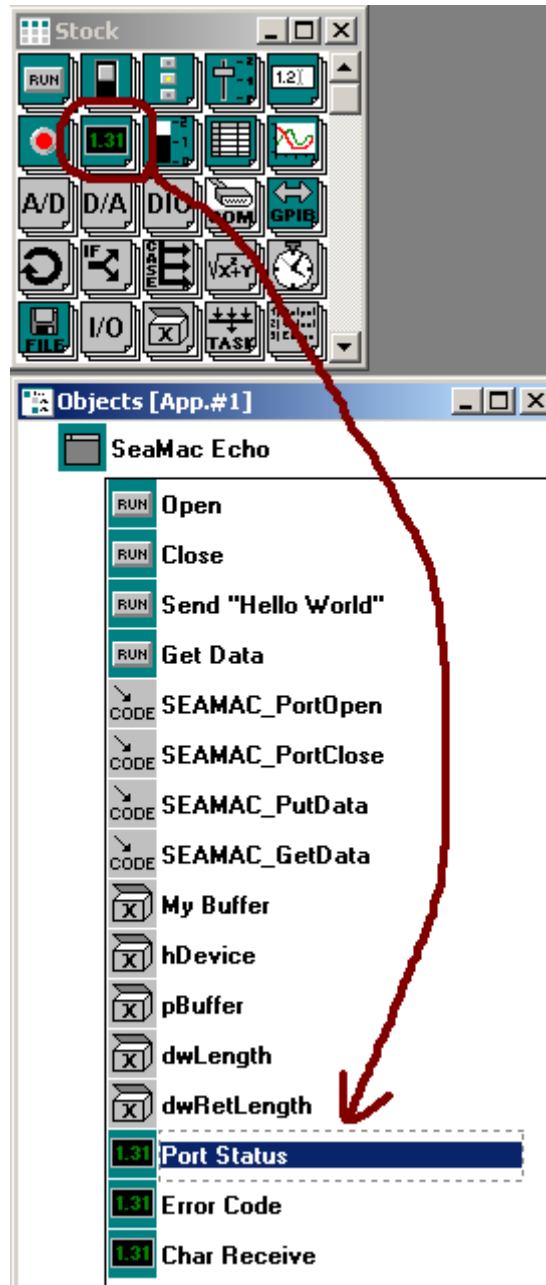
Preload

Settings | Actions | Comments | XRef

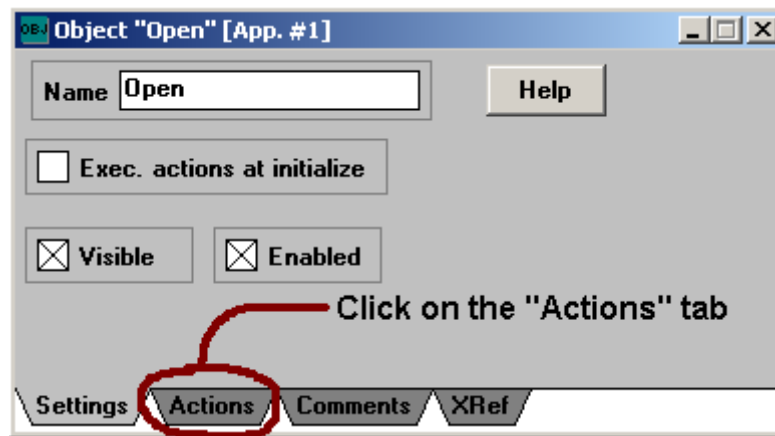
Now that the code properties are setup correctly we can setup a couple of “Containers” to store data. Drag five “Container” objects to the object list and name them as per example.



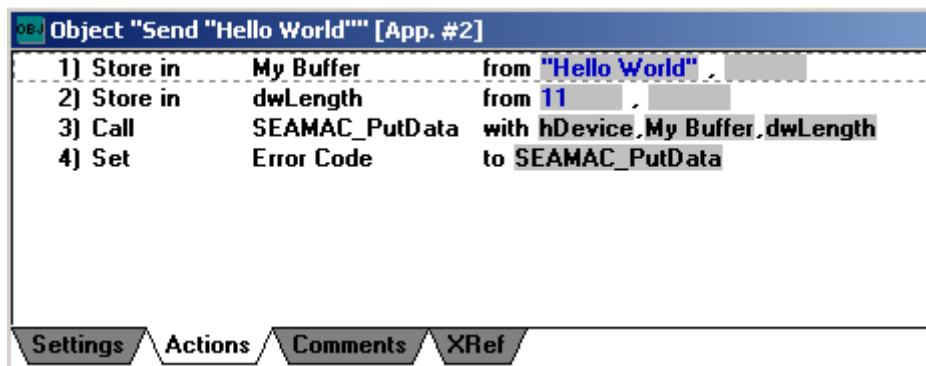
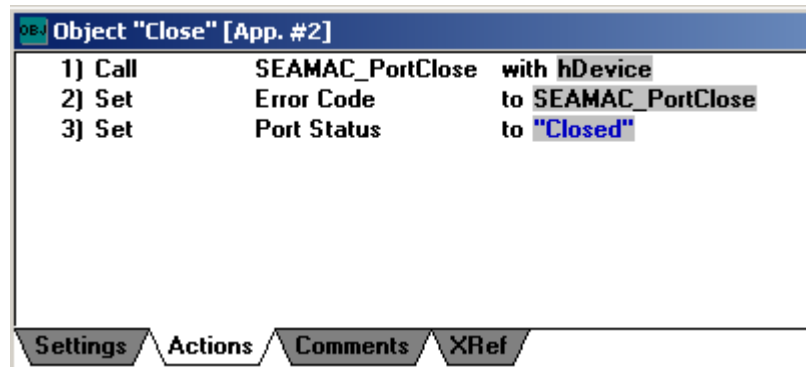
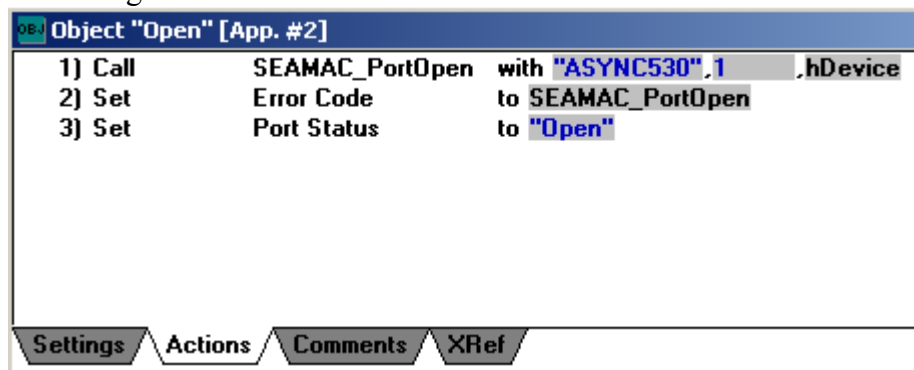
Lets add some “Display” objects so that we can view what is going on in our program. Drag three “Display” objects to the objects list and name as per example.

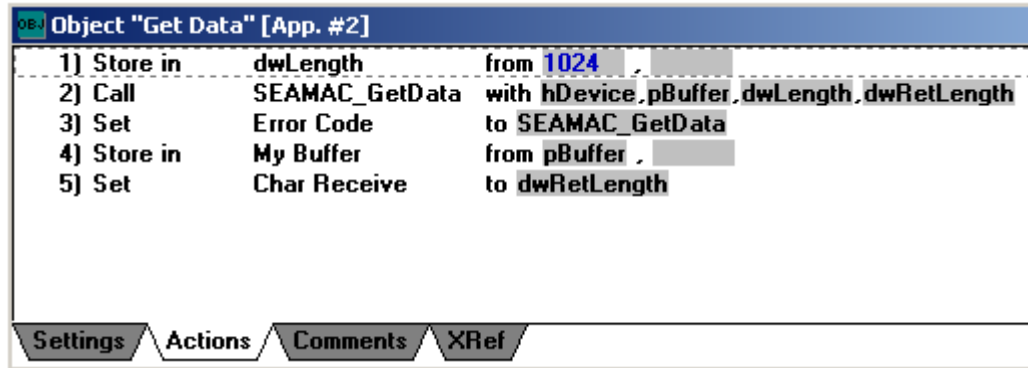


Lets get back to the “Pushbuttons”. Double clicking on the pushbutton icons in the objects list will bring up their property dialog box. Click on the “Actions” tab and setup the four pushbuttons as per examples.



Here are the settings for each button.





Now that we have everything set up correctly, lets do some cosmetic cleanup. Here is an example of how ours turned out.



The SeaMAC example distributed here, SeaMacEcho.TST, works with with the SeaMAC driver set to "Async" protocol. The "Get Data" function in the example assumes that an external loopback is in place, but is not necessary for running the driver.