SEALEVEL
SYSTEMS INCORPORATED

iFix with Sealevel I/O boards

SeaCOM

Asynchronous Serial communication capabilities are built into Visual Basic by using the Open, Close, Get, and Put statements. First, you "Open" COM1 or whatever Com# you may be using. Next it is your option to read data from the port by using the Get statement or send data out the port by using Put. An example of using the Open statement: (Open "COM1:9600,n,8,1" For Random As #1). That statement will open Com 1 at 9600 bits/second, no parity, 8 bits per character, 1 stop bit, for Random access, and file number 1. The put statement is what you would use to output a string of data to the Com port: (Put #1 , , out). "Out" being a string variable containing data. It does not have to be a string, but it works well as an example. Get and Close are similarly implemented. For complete documentation consult the Visual Basic reference guide.

SeaIO

In order to begin programming iFix with SeaIO, the first move after creating a new workspace or opening an existing one, is to enter the VB script editor for that project. In the picture below (Figure 1), we show an explorer window open to the VBTest example in SeaIO. In this directory is the SEAIO.GLO file; it contains all the Global variables and API links for the VBTest project. You have the option of creating your own .GLO file if this is not an adequate framework, but make sure to put into yours the section that has about twelve "Declare Function SeaIo_……" in our SeaIO.GLO. These "Declare Functions" will tell Visual Basic to include the functions included in the SeaIO API for use in your project. For a list of the API calls, view the included help files.

Figure 1 below shows where you can drag the SeaIO.GLO file into your project's tree and have it included as a module if you so wish.

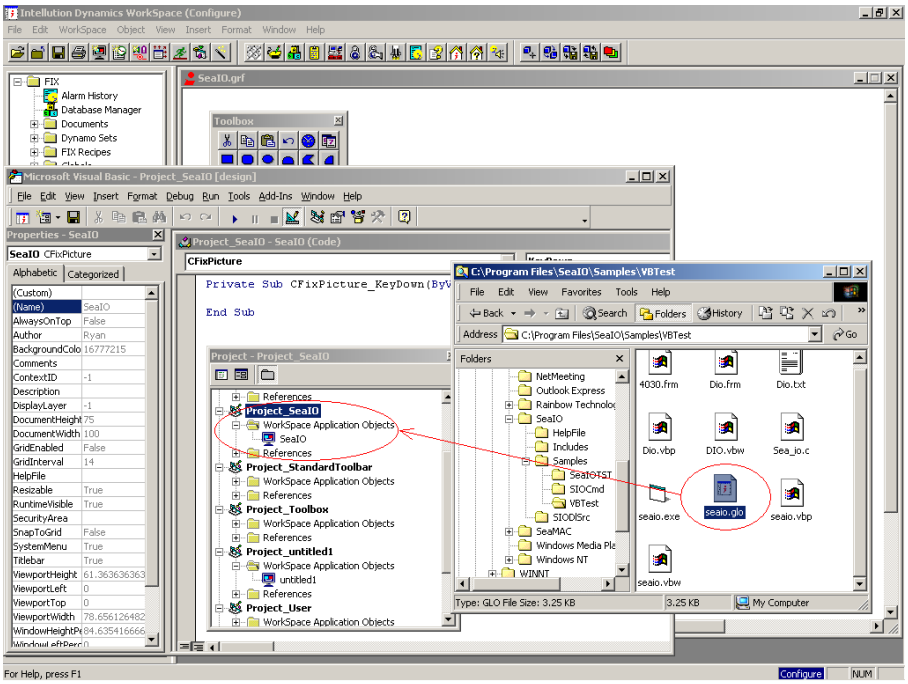

Figure 1

After putting the API calls into your globals, you can now start using the API calls to control a SeaIO card.

To physically start using the SeaIO cards within your project, the first thing that has to happen is a SeaIo_OpenDevice. This will open a specified SeaIO card and return a handle to the device which will be needed when we start using the card to control devices. The parameters for SeaIo_OpenDevice are as follows: SeaIo_OpenDevice(devnumber,Handle). Devnumber is the "installed device number". If you have 3 SeaIO cards installed, the first is "0", the second "1", the third is "2", and so on. Handle is a variable you create that will be used with the rest of the SeaIO API as a reference to the device you just opened.

You are now ready to close a relay, or read an input. Using SeaIo_RelayClose, you can cause a relay to close on the card. In the example supplied, the actions of opening the card, closing the card, closing a relay and opening a relay are all assigned to buttons within iFix. The parameters for SeaIo_RelayClose are: SeaIo_RelayClose(Handletocard, RelayNum). Where Handle is the handle to the card you just opened, and RelayNum is the number of the relay starting with 0.
Figure 2 shows the use of the SeaIo_OpenDevice call.

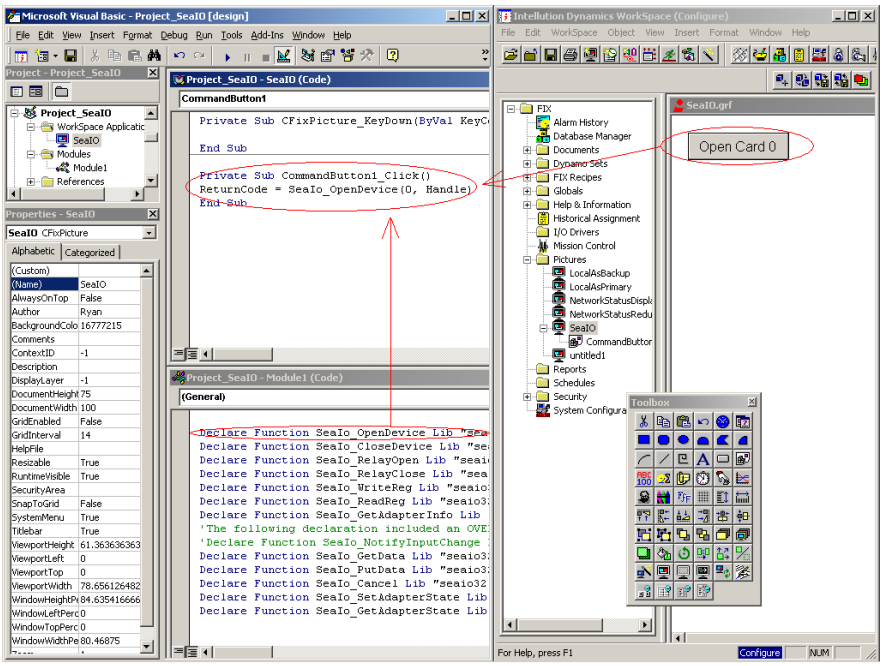

Figure 2

From this point it is a matter of where and how you want to use the software to control external devices.

In the example provided, SeaIO.grf, there are four buttons mapped to the opendevice, closedevice, closerelay and openrelay API calls. They give basic examples of how to use the SeaIO cards within iFix. Figure 3 is a picture of the completed SeaIO sample application in iFix. There are many more API calls at your disposal, and the help files give descriptions and the parameterization of each.

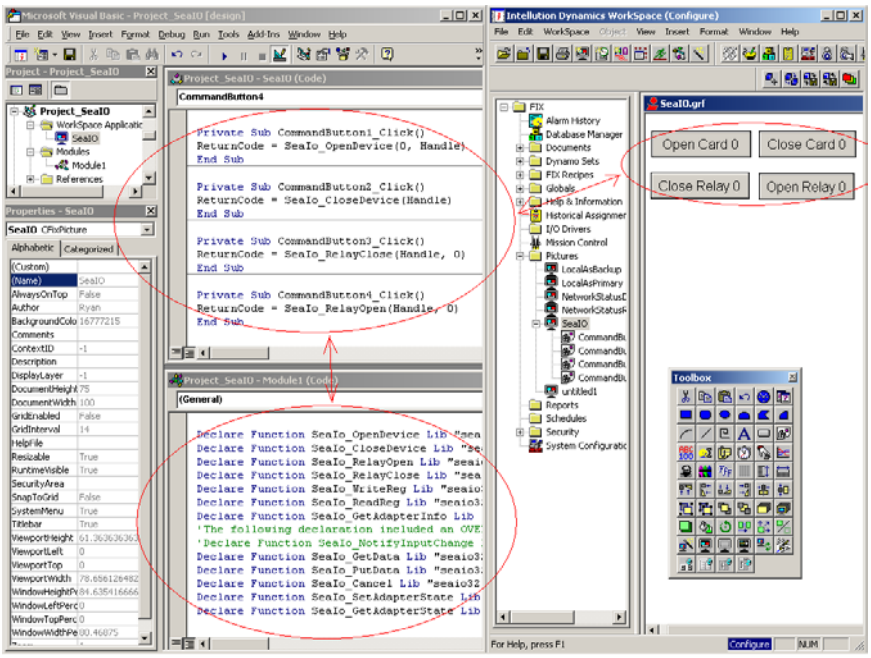

Figure 3

If you have your iFix engine running, you can double click on the file below to open up the SeaIO sample application.

SeaMAC

You can use the already written .GLO file we have located in the Samples\VBTerm for SeaMAC. You would insert it either by clicking and dragging the VBTerm.GLO into your project or by typing in the API linking manually as shown in the VBTerm.glo. After that point, you would use the SeaMAC_PortOpen to initially open the card. The parameter being: SeaMAC_PortOpen(4 , 0 , Handle). 4 Is the driver number. 0 for Asynchronous, 1 for HDLC530, 2 for MSBS530(Not implemented), 3 for HDLCC32, and 4 for the new WDM all-inclusive driver (new to SeaMAC rev 3+ packages). 0 is the port number. This number increments depending on how many cards you have in your system configured for X driver. Handle will be the handle to the port you will be opening for reference to other functions. From there you have the option to use SeaMAC_PutData or SeaMAC_GetData to send and receive data respectively. The SeaMAC_SetPortState will allow for dynamic reconfiguration of the card's behavior. Data rate, clocking methods, etc.; however, it is recommended that you do the SetPortState before you are in the middle of data transfer due to the possibility of data loss. The rest of the API functions are well documented in the Help files and can extend the functionality of the card beyond the scope of this document.