

Can I use more than 32 I/O in my ISaGRAF project if I don't have ISaGRAF-256 or ISaGRAF-L?

If you have an ISaGRAF-256 or ISaGRAF-L workbench (Version 3.4x or 3.5x) with a dongle, you don't need to use the skill described in this section.

A very useful feature of the ISaGRAF Workbench program is the ability to create "directly represented (or internal)" variables. Internal variables are program variables that can be used in an ISaGRAF program, but they are not physically connected to any of the input or output variables. There are four versions of the ISaGRAF Workbench program available with the I-8xx7 controller system: ISaGRAF-32, ISaGRAF-80, ISaGRAF-256, and ISaGRAF-L. The number after "ISaGRAF" represents the number of I/O variables that are allowed with that particular ISaGRAF Workbench program.

The ISaGRAF Workbench program comes with a hardware protection device (dongle) that plugs directly into your development computers parallel port. Every time you compile a program in ISaGRAF the hardware protection device is read to make sure that you are not trying to connect to more program variables than are allowed with your particular copy of the ISaGRAF Workbench program that you purchased with your I-8xx7 controller system.

These "directly represented (henceforth called "internal") variables can be used in lieu of your real world inputs and outputs so you can create additional program variables that do not count against the amount of ISaGRAF program variables. The only "caveat emptor" to these internal variables is that you must follow a strict programming scheme to program and access these internal variables, and they are more complicated to create than the regular input and output variables. **For a professional programmer, recommend to purchase an ISaGRAF-256 workbench rather than an ISaGRAF-80 or ISaGRAF-32 workbench for programming on I-8xx7, I-7188EG/XG & Wincon-8xx7 controllers.**

Single Type Internal Variable Programming Scheme:

For single-typed board: "s" is the slot No, "c" is the channel No.

%IXs.c	free channel of a boolean input board,	ex. %IX2.3
%QXs.c	free channel of a boolean output board,	ex. %QX0.2
%IDs.c	free channel of an integer input board,	ex. %ID3.1
%QDs.c	free channel of an integer output board,	ex. %QD2.4
%ISs.c	free channel of a message input board,	ex. %IS3.1
%QSs.c	free channel of a message output board,	ex. %QS2.4

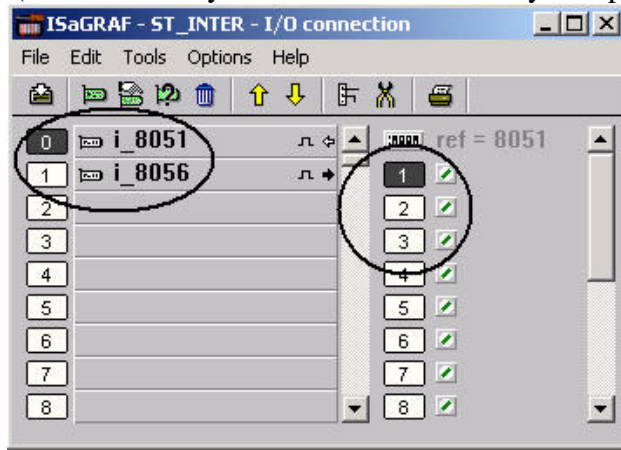
Complex Type Internal Variable Programming Scheme:

For complex board: "s" is the slot No, "b" is the index of the single board within the complex equipment. "c" is the channel No.

%IXs.b.c	free channel of a boolean input board,	ex. %IX2.3.2
%QXs.b.c	free channel of a boolean output board,	ex. %QX0.2.1
%IDs.b.c	free channel of an integer input board,	ex. %ID3.1.3
%QDs.b.c	free channel of an integer output board,	ex. %QD2.4.3
%ISs.b.c	free channel of a message input board,	ex. %IS3.3.1
%QSs.b.c	free channel of a message output board,	ex. %QS2.1.4

An Internal Variable Program Example

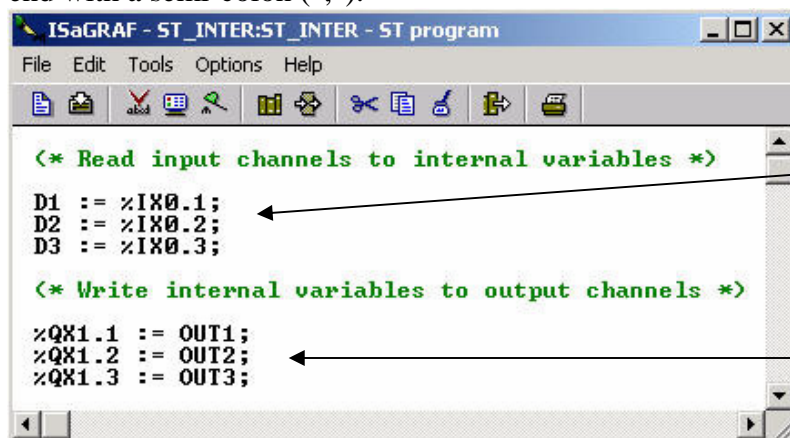
Create a new project for an ISaGRAF ST program, and then create a link to the I/O boards that are specified in the window below. Declare three input variables called "D1", "D2", & "D3" for the I-8051 board located at slot 0, and then create three output variables called "OUT1", "OUT2", & "OUT3" for an I-8056 board located at slot 1. This time set each of their respective attributes to "internal" instead of input or output (this means they are not connected to any real physical I/O).



Create A New "ST" Program



Double click on the "ST_Inter" that is highlighted and the "ISaGRAF ST Program" window will open. Type in the program code displayed in the window below EXACTLY as shown. Remember, each line MUST end with a semi-colon (";").



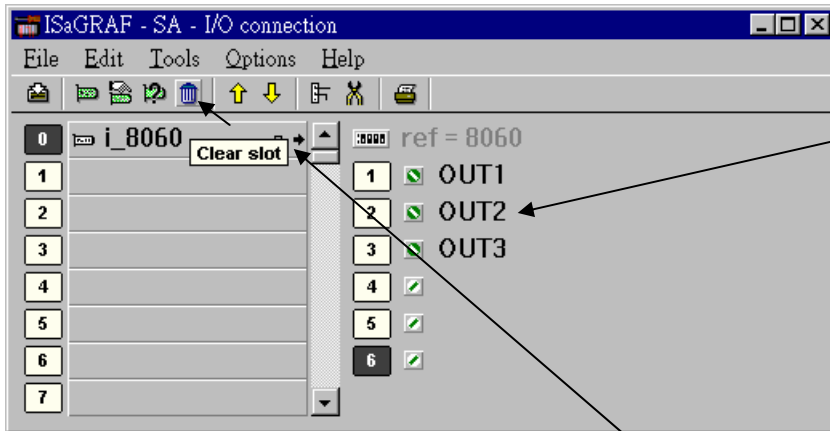
D1 := %IX0.1 ;
D2 := %IX0.2 ;
D3 := %IX0.3 ;

%QX1.1 := OUT1 ;
%QX1.2 := OUT2 ;
%QX1.3 := OUT3 ;

Now we can use the internal variables D1 through D3 and OUT1 through OUT3 that have been created in other programs in the same project. The newly created internal variables will generate input and output actions to the associated channels in this ST program.

IMPORTANT NOTE:

If once the input or output attributed variables have been connected to an connected IO board or complex equipment, and if they would like to be replaced by Directly represented variables, these input or output attributed variables have to be re-attributed to “internal” and the board or equipment **must be re-connected to the slot.**



If you wish to replaced these variables by directly represented variables, re-attributed them to “internal” attribution in the “dictionary” window.

Clear slot and re-connect again.

IMPORTANT NOTE

If you enable the compiler option of upload, option “**Comments for not connected I/O channels**” must be chosen if “Directly represented variables” is used in this project (refer to section 9.2).