Setting up Channels with End User Software

When working with Testpoint, HPVee, C, and Visual Basic; one is often faced with the question of how to set up instruNet channels (e.g. specify thermocouple). Several methods for managing channel settings are listed below. DasyLab does not have this issue, since the DasyLab software automatically saves channel settings within their .dsb file.

1) Save/Load Settings from instruNet World

Within instruNet World, one can set up all sensors in the Network page and press the SAVE button to save these to a .prf file on disk, or the OPEN button to pull in settings from a .prf file. Also, one can press STORE or RESTORE to save/load these settings from a standard file in the system and not specify a file name. And, one can press buttons from their program (e.g. HpVee) by calling the Press_instruNet_Button() dll subroutine. For example, Press_instruNet_Button(9) does a RESTORE and Press_instruNet_Button(10) does a STORE.

2) Set Fields with a Subroutine

The SetFieldValue_iNet() dll subroutine can set any of the settings within instruNet. For details on this, please see files "INET_INT.H" and "INET_INT.H", or Application Note #33.

3) instruNet World PLUS (iW+) startup program

If your computer has been licensed for instruNet World PLUS (iW+) and an instruNet Script file by the name of "iNetLoad.ibs" exists in the "…\program files\instruNet\End User\" directory, then it will be executed when the instruNet driver is loaded by your instruNet application program (e.g. DASYLab, LabVIEW, instruNet World, etc).   For examples of script files that set up channels, see file "…\ program files \ instruNet \ instruNet World+ \ Examples \ Set up channels, digitize, spool to disk.iBs".

4) Call a subroutine to get and set channel settings.

There is a dll function that one can call from your program (e.g. TestPoint) to get the channel settings inside iNet (it returns a block of binary data) after you had set them up manually from instruNet World, you can then save this binary block w/ by your program, and then when you run your program again, you can call an iNet dll routine and pass this binary block back to the iNet dll to set up your channels. This is done via dll routines Get_instruNet_Network_State() and Set_instruNet_Network_State(). For details, please see Application Note #35.