El Instrument Control Toolbox de Matlab admite dos formas de comunicarse con un instrumento:

-interface objects

-device objects

La que he estado usando hasta ahora es el interface object, que lo que hace es abrir un tipo de comunicación (gpib, visa, etc). Luego es responsabilidad del usuario enviar los comandos adecuados al instrumento conectado ahí. Esto lo he hecho usando instrumentos y comandos gpib para los HP y la electrónica magnicon. Para la PXI-5922 la secuencia de instrucciones funciona, pero no tengo una lista de comandos que acepte la tarjeta y con la que pueda comunicarme!

pxi=visa('ni','PXI9::14::INSTR')

fopen(pxi)

str=sprintf('%s\n','\*TST?')

query(pxi,str)

Esa secuencia supuestamente manda un self-test, pero da error, no reconoce ese comando. Sin embargo, sí que se crea y se abre el instrumento. En realidad existen instrumentos ‘comand’ based e instrumentos ‘register’ based. Estos son por ejemplo las tarjetas digitalizadoras VXI. Así que si esta PXI-5922 es register based tendría que comunicarme con ella de otra forma. En el manual del Instrument control toolbox menciona funciones de lectura/escritura en memoria de alto y bajo nivel. Necesitaría la lista y significado de los registros de la tarjeta. Hay un ejemplo en que usa “vxi.MemorySpace” pero el objeto pxi creado como ‘visa’ no tiene ese miembro.

La otra opción es usar un device object. Ahí no sólo se crea el interface, sino que hay que pasarle el driver del instrumento. En teoría la librería ‘niScope’ incluye la PXI-5922, por lo que estaría soportada. Las primeras pruebas me dieron problemas porque no conseguía generar el driver. Al parecer Matlab necesita un compilador asociado y este es el problema que he venido teniendo desde hace tiempo y que me ha impedido usar también el Simulink. Finalmente, a fecha 23Oct17 consigo resolver este problema siguiendo las instrucciones de una web.

(<https://es.mathworks.com/matlabcentral/answers/95039-why-does-the-sdk-7-1-installation-fail-with-an-installation-failed-message-on-my-windows-system>). Tengo que desinstalar todas las redistributable versions de VS C++ anteriores a 2010. Luego instalo .NET4.7.1 (le cuesta un rato) (fichero: NDP471-KB4033344-Web.exe). Luego el SDK7.1 (fichero: winsdk\_web.exe) y finalmente un patch (fichero: VC-Compiler-KB2519277.exe). Según las instrucciones después debía reinstalar las redistributables, pero me daba error. No obstante, al ejecutar Matlab parece que él solo instaló alguna y al ejecutar el mex –setup ya fue capaz de reconocer un compilador!

Una vez resuelto este problema pruebo a crear un objeto con ese driver. Pruebo a crear un driver de tipo niScope y conectarlo a la dirección de la tarjeta. El niScope es un IVI driver genérico. Matlab requiere crear un Matlab Wrapper driver, que son ficheros .mdd. Luego se usa ese driver para comunicarse con el instrumento.

niscope=makemid('niScope','niscope.mdd')

pxi=icdevice('niscope.mdd','PXI9::14::INSTR')

Sin embargo, la instrucción connect(pxi) da error. Se puede ver el .mdd y crea prototipos de muchas funciones, pero no hay una de init(). No está claro si hay que completar el driver o debería funcionar ya out-of-the-box.

Con la tmtool se puede inspeccionar el hardware y software instalados y se ve que no aparece la tarjeta PXI-5922 en el slot3. Sólo aparece el PXI-backplane. ¿Puede estar ahí el problema? En teoría debería poder listarse todo el hardware disponible vía software:

info=instrhwinfo('visa','ni')

info.ObjectConstructorName

ans =

'visa('ni', 'ASRL1::INSTR');'

'visa('ni', 'ASRL3::INSTR');'

'visa('ni', 'ASRL5::INSTR');'

'visa('ni', 'ASRL10::INSTR');'

'visa('ni', 'GPIB0::4::INSTR');

Se listan los dispositivos disponibles, pero no aparece la PXI.

Hay también una opción para configurar directamente un scope, con la función ‘oscilloscope(), que permite por ejemplo listar availableResources y al hacerlo sale el PXI0:1:Backplane, pero no la tarjeta. Intento conectarme con el backplane pero da error. La clave está por tanto en intentar que reconozca automáticamente la tarjeta. A partir de ahí, esa clase ofrece métodos para configurar los canales y adquirir waveforms, etc.

La tmtool también lista el PXI0::1::BACKPLANE pero no el puente ni la tarjeta.

Instalo matlab2015a pero tampoco aparece la tarjeta en la tmtool.

De NI me contestan que ellos no dan soporte para Matlab y que tienen que ser ellos los que resuelvan el problema. Pruebo a ‘eliminar’(cambiar nombre) a los pxiesys.ini y pxisys.ini. Se listan con el NI-MAX las tarjetas PXI pero no en la estructura del chasis PXI. Si dejo el pxiesys sí que aparece la estructura del chasis con el PXIe-8375 dentro, pero no la 5922. Al añadir el pxisys.ini sí aparece esta tarjeta. La tmtool muestra en todos los casos el PXI0::1::BACKPLANE.

Update 16-01-2018

Tras conversación con Nicolás de electrónica, me comenta la función ‘daqregister()’ del daq toolbox de Matlab, que es necesario ejecutar para instalar una tarjeta de adquisición de datos nueva. Además hay que ejecutarlo como Administrador. Una vez instalada, se puede usar ya por cualquier usuario. Ejecuto daqregister(‘nidaq’) y unos cuantas opciones más, pero no encuentra ningún .dll. Ejecuto daqhwinfo y obtengo un mensaje de error diciendo que la ‘legacy interface’ no está soportada en mi plataforma, sólo la ‘session-based’. Parece ser que hay dos formas diferentes de comunicarse con tarjetas de adquisición de datos, ‘Legacy Interface’ y ‘Session-Based Interface’ y para W10 64-bits sólo está disponible la segunda. Hay instrucciones para configurar una sesión de adquisición. Primero ejecutar daq.getDevices para ver los dispositivos disponibles. Al hacerlo no encuentra ninguno. Hay una lista de los dispositivos PXI soportados y no está la 5922 (<https://es.mathworks.com/hardware-support/nidaqmx.html> ). Parece que no es lo mismo una tarjeta digitalizadora que una tarjeta de adquisición de datos, por ejemplo la PXI-4461, que se usa en adquisición de audio o acelerómetros.

Ejecuto la aplicación ‘VISA Interactive Control’ y sólo reconoce el Backplane PXI, el chasis NI PXIe-1071, igual que la tmtool, aunque el NIMAX reconoce tanto el puente NI PXIe-8375 como la tarjeta PXI-5922 en los slots correspondientes. Sigue sin resolverse por tanto el problema.