

There are 3 ways to read/write channels to/from instruNet:

1) High Speed Digitize

The instruNet DSP collects data from a specific group of channels at a specific rate. Data is placed into ram buffers, 4bytes per point, in 32bit floating point form. The instruNet DSP is running so fast, that other instruNet activity, from that DSP, is blocked out [e.g. `GetChannelValue_iNet()` and `SetChannelValue_iNet()` cannot run]. In some cases, adding multiple DSP cards is necessary, so that one can do the >1KHz High Speed Digitize on one card, and the slower work on another. When digitizing at >1Ks/sec/ch, you are running High Speed Digitize. High Speed Digitize is implemented with the following C subroutines: `DigitizeListOfChannels()` and `GetDigitizedSegment()`.

2) Medium Speed Digitize

This is similar to High Speed, described above, except that the instruNet DSP acquires digitize data via internal interrupts, and can therefore be interrupted by `GetChannelValue_iNet()` and `SetChannelValue_iNet()` while digitizing. The internal interrupts occur sort of randomly, with plenty of idle time inbetween. This method is used when the sample rate is approximately < 1Ks/sec, and is implemented with the following C subroutines: `DigitizeListOfChannels()` and `GetDigitizedSegment()`.

3) Scalar Read/Write

This method involves reading and writing single values, as needed, via the `GetChannelValue_iNet()` and `SetChannelValue_iNet()` routines. These routines run properly, except if a High Speed >1Ks/sec/ch Digitize is currently running, in which case, an error code is returned.

The `GET_DIGITIZE_STATUS(netNum)` C macro returns the current status of a DSP card as `notDigitizing`, `digitizingHighSpeed`, or `digitizingMediumSpeed`.

For more information on the differences between high speed digitize and single value read/write with DasyLab, please see Application Note #109.