

Programming with Multiple Digital F/T Sensors

Issues:

- NI Serial interface can use a lot of CPU time if interrupts occur too frequently, which slows down system.
- Inefficient user code can cause buffer overrun if processing one sample takes longer than the time for the sensor to send out another sample.

Recommendations:

- Set FIFO receive buffer level in Measurement & Automation Explorer as high as possible without causing buffer overrun – a good level is 115, which allows room for one 13-byte streaming sample to come in before the 128-byte hardware buffer fills up.
- Run the sensors at 19200 or 115200 baud. This limits the amount of streaming data, and can greatly reduce the amount of time spent processing serial interrupts.
- Use multithreaded code, e.g. one thread or process for each sensor. This is especially helpful on computers with multiple processor cores.
- Test your code for performance. If you find you need to improve your code's performance, some common methods are:
 - Avoid creating new objects during the processing of each sample (try filling in the fields of existing objects instead).
 - Cache commonly used values (such as the calibration matrix).
 - “Unroll” loops that iterate several times by decreasing the number of loop iterations and doing more processing during each iteration.
- Use a faster computer, or a computer with multiple processor cores (you will need to write multithreaded or multiprocessing code to make use of the multiple cores).