Pinnacle High Speed Logging

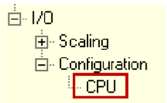
The text below refers to analog inputs (AIs/UIs) only. Digital are scanned at a much faster rate and therefore not an issue with high speed logging. Physical Outputs (analog and digital), unlike physical inputs, are written to as opposed being scanned which makes the rate that their values are changed impossible for controller to know. They are covered under the "Values that are being written to" section below.

The Pinnacle Controller is capable of logging data values to a file at up to 100 points per second (10ms rate). Data logging for physical inputs is independent of any control programs that are running (so program scan time is not an issue).

Values that are being 'written to' by a control program can also be logged at high speed, but will not show value updates any faster than the current program cycle time (which is the time it takes to execute all of the programs on the controller). For example, if you are logging a value written to by a control program at a 10 ms rate and the program scan time is 50ms, then the value will only show a change in the log file every 50ms, even though the log file is configured to log every 10ms. This will have the effect of showing the same value for 5 log entries in a row.

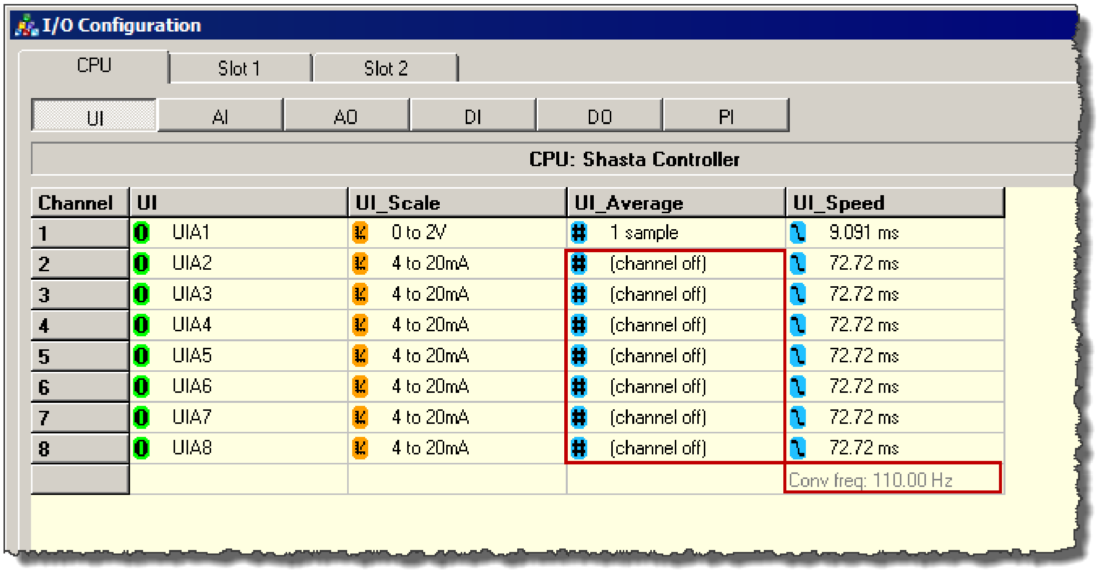
The logging speed is dependent however on how the physical inputs are configured. The controller will scan all of the inputs before updating register values, which takes time. That time is dependent on how many channels are enabled, and the UI Speed setting for each of the enabled channels. The final analog to digital conversion speed (the maximum speed that the values can be updated) is summed and displayed at the bottom of the speed column in Hertz (times per second).

Configuring the IO ("IO / Configuration / CPU" in ScadaBuilder)



The input configuration in ScadaBuilder affects the speed at which data can be logged effectively

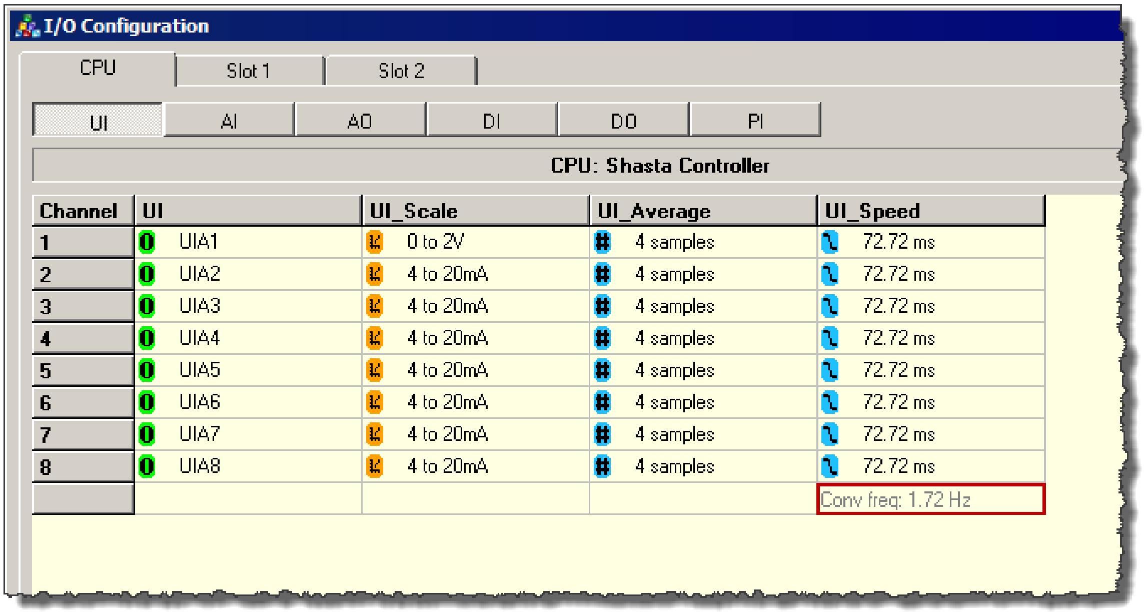
The configuration below has only the first UI channel enabled which still allows some room for noise filtering. Noise filtering is accomplished by setting the UI speed. In the example, sampling the input UI Speed at 9.091 ms gives the final conversion frequency of 110 Hz (over 10 times a second). The point here is that the UI Speed setting can be balanced so that the necessary filtering is being applied while still having a high enough conversion frequency to accommodate high speed logging. Setting the UI Speed to the lowest setting gives the fastest conversion speed, but may allow noise to affect the signal.



CPU: Shasta Controller

Channel	UI	UI_Scale	UI_Average	UI_Speed
1	UIA1	0 to 2V	1 sample	9.091 ms
2	UIA2	4 to 20mA	(channel off)	72.72 ms
3	UIA3	4 to 20mA	(channel off)	72.72 ms
4	UIA4	4 to 20mA	(channel off)	72.72 ms
5	UIA5	4 to 20mA	(channel off)	72.72 ms
6	UIA6	4 to 20mA	(channel off)	72.72 ms
7	UIA7	4 to 20mA	(channel off)	72.72 ms
8	UIA8	4 to 20mA	(channel off)	72.72 ms
				Conv freq: 110.00 Hz

This next example shows the all of the channels are enabled at the default UI speed setting. The conversion frequency is 1.72 Hz (1.72 times a second, or .581 ms per scan), so logging any UI values faster than twice a second would not work effectively. If you're trying to log at a 10 ms rate, then the conversion frequency should be at least that fast (10ms/100 Hz).



CPU: Shasta Controller

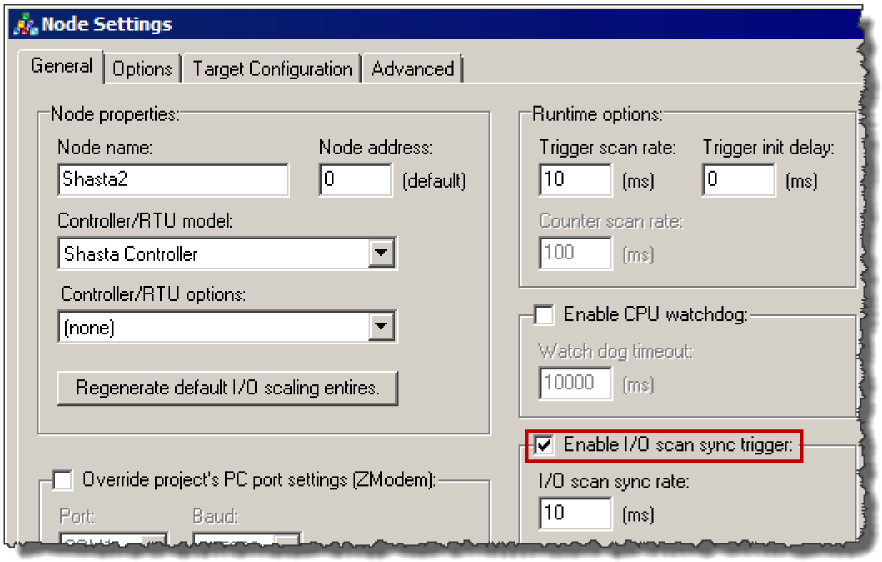
Channel	UI	UI_Scale	UI_Average	UI_Speed
1	UIA1	0 to 2V	4 samples	72.72 ms
2	UIA2	4 to 20mA	4 samples	72.72 ms
3	UIA3	4 to 20mA	4 samples	72.72 ms
4	UIA4	4 to 20mA	4 samples	72.72 ms
5	UIA5	4 to 20mA	4 samples	72.72 ms
6	UIA6	4 to 20mA	4 samples	72.72 ms
7	UIA7	4 to 20mA	4 samples	72.72 ms
8	UIA8	4 to 20mA	4 samples	72.72 ms
				Conv freq: 1.72 Hz

To summarize:

1. The rate at which you can log data effectively depends on how many inputs you are using, and the UI Speed setting for each.
2. If you are using high speed logging, then all unused UI channels should be disabled by selected "Channel off" from the average setting for each.
3. The channels that are being used should be set to UI Speed settings that give a high enough conversion frequency to allow for the desired logging rate, while also having a high enough setting to filter input noise effectively.
4. Higher UI Speed settings will produce smoother input values (with less affect due to signal noise), while lower values will produce a higher conversion frequency.
5. Note that the "UI Average" settings don't affect the conversion frequency, but can also be used to "smooth out" the input values by averaging a number of samples.

High Speed Logging Configuration

To enable high speed logging, check the box in the node settings for “Enable IO scan sync trigger”:



Node Settings

Node properties:

Node name: Shasta2

Node address: 0 (default)

Controller/RTU model: Shasta Controller

Controller/RTU options: (none)

□ Override project's PC port settings (ZModem):

Port:

Baud:

Runtime options:

Trigger scan rate: 10 (ms)

Trigger init delay: 0 (ms)

Counter scan rate: 100 (ms)

□ Enable CPU watchdog:

Watch dog timeout: 10000 (ms)

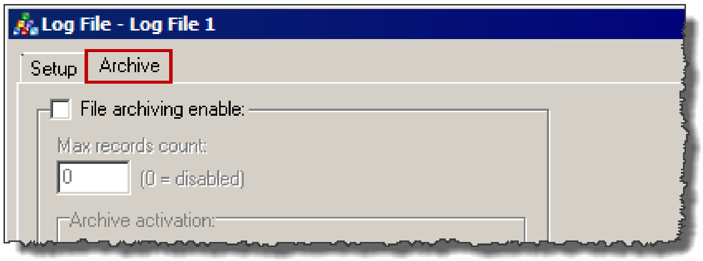
☑ Enable I/O scan sync trigger:

I/O scan sync rate: 10 (ms)

Note that the minimum rate is 10ms and logging at a smaller rate may not work properly.

Next, configure the log file settings. Refer to the ScadaWorks Reference (in the help section) for more details on how the log system is configured. This section mostly details how to accomplish high speed logging only.

When high speed logging is enabled, the event timing is configured on the archive tab of the log file instead of on the event configuration. Note that there is also an enable checkbox here that needs to be checked.

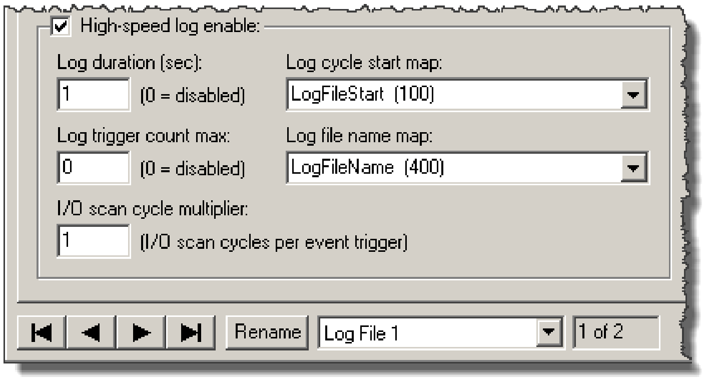


Log File - Log File 1

□ File archiving enable:

Max records count: 0 (0 = disabled)

Archive activation:



☑ High-speed log enable:

Log duration (sec): 1 (0 = disabled)

Log cycle start map: LogFileStart (100)

Log trigger count max: 0 (0 = disabled)

Log file name map: LogFileName (400)

I/O scan cycle multiplier: 1 (I/O scan cycles per event trigger)

Log duration – The length of time that the data will be logged. At the end of the duration the Log cycle start map Boolean will automatically be set to false.

Log trigger count max – This is used in conjunction with the Log duration and whichever is reached first will cause the logging to stop and set the Log cycle start map Boolean to false.

I/O scan cycle multiplier – The maximum log speed is a global setting in the node settings. The setting here allows you to have a different/unique setting for this log entry different from the global setting. It cannot be any faster, but it can be different. For example, setting the multiplier to 2 will give a log speed twice as slow as the global setting. So if the global setting is 10ms and this setting is 2, the speed for this log event will be 20ms.

Log cycle start map – This is mapped to a Boolean register that will start the log event (when the Boolean goes from false to true). The Boolean is automatically set to false once the duration setting is met. For example, if you set the duration to 5 sec and change the Boolean to true, the data will be logged for 5 seconds, then stop logging and set the Boolean to false. This allows you to control when the log event starts and stops from a trigger, local event, or program. Note that the Boolean must be set to true each time you wish to log data, so some mechanism (either trigger, local event, or line of program code) should be established to keep the logging going.

Log file name map (optional) – The file name that is applied to the file that is written is automatically stored in this message register. When high speed logging is used, a filename is automatically generated starting with the name you specify along with the applied extension (custom, lg1, or lg2) plus a time stamp with underscores between the parts. For example, if you put in “FastLog” as the filename then you’ll get something like “FastLog_2015_01_29_13_13_11.lg1” as the final name. The date stamp is formatted as such: “Year_Month_Day_Hour_Min_Sec”. The minimum time for the log duration is 1 second, so the timestamp only goes down to the seconds.

<u>Next, configure the log event.</u> The “Log activation” section is grayed out in this case because the timing is already set up. Configure the points you want logged, along with any text and delimiters as usual, and that’s all there is to it.

Testing:

To test the log file simply write a true value to the Log cycle start map Boolean and check the resulting file. If you map the log file name to a message register, then you can put it on a TUI or WUI, flip the Boolean, and wait for the filename to change. At that point the logging is finished and you can use an FTP client to grab the resulting file and look at it.