Interactive Logging with FlukeView® Forms

FlukeView Forms Technical Note

Fluke developed an Event Logging function allowing the Fluke 89-IV and the Fluke 189 models to profile the behavior of a signal over time without requiring a great deal of memory to record redundant data. This approach monitors the signal and records the highest, lowest, and average values and the start and end times of the period over which the values are essentially the same.

Beginning with FlukeView Forms version 1.5, can now interactively log events directly from a meter to the PC, in real time, with the meter connected to the PC during the logging session. This can be done for a number of different meters (including the 89-IV and 189), and the event capture being done by the PC (with the option of having it performed on the 189/89-IV meters as well.)

This technical note explains the enhancements to the event logging functionality incorporated into FlukeView Forms interactive logging, beginning with version 1.5 of FlukeView Forms.

Differences between Event Logging and Data Logger

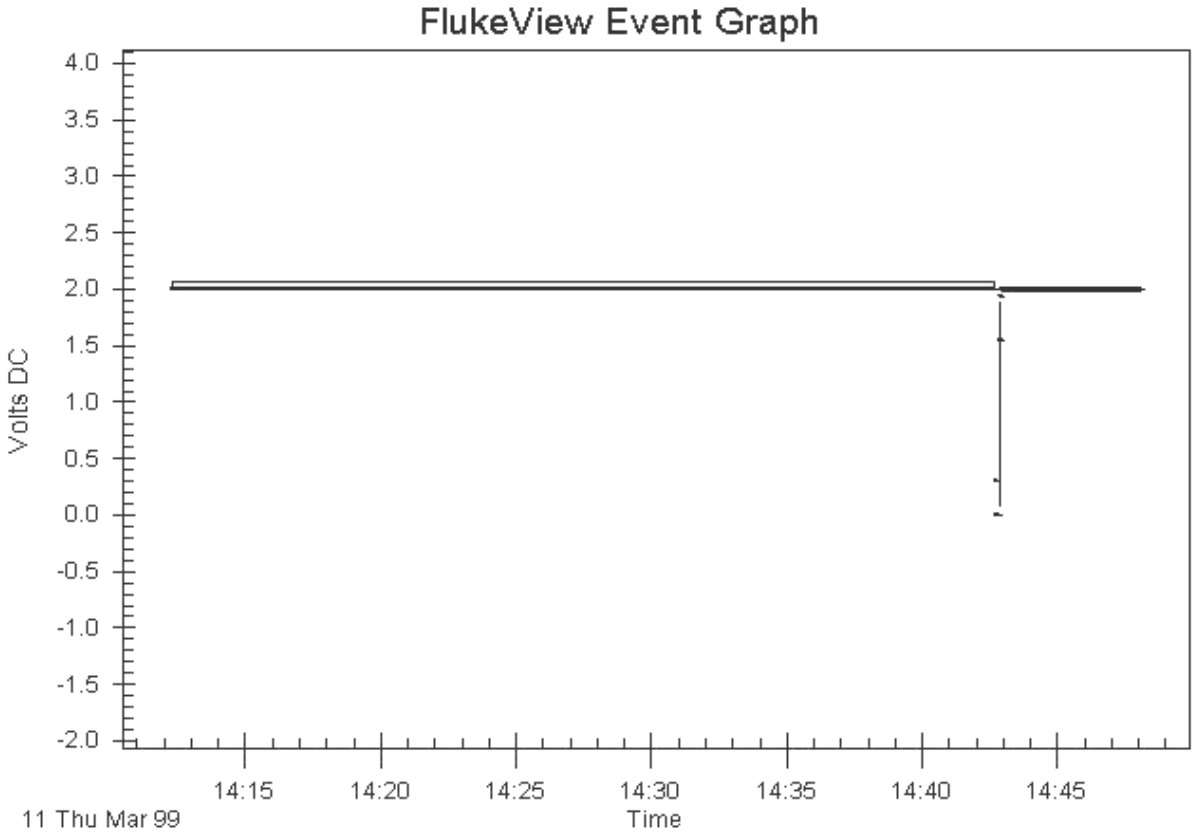
Event logging attempts to reduce the amount of stored data to only that which is necessary to capture the normal and/or abnormal behavior of the signal.

Typically, the goal of a data logger is to sample an input signal at a rate sufficient enough to be able to track something of interest that you expect to be contained within the signal. This often means that it is desirable to sample the signal as fast as possible so you "don’t miss anything." The problem with this approach is you need a large storage place for the fast data sampling that is taking place. You can also end up with a lot of redundant data that is considered "normal" and not of interest. You must wade through the normal data to find the exceptional data (or lack thereof) that you are interested in seeing.

With Event Logging, the input signal is monitored continuously and data is stored when the input changes significantly or when an interval amount of time has passed. The data includes the highest, lowest and average values seen during that event and the start and stop timestamps which define the time period.

An Example - Viewing what Event Logging Data Looks Like

FlukeView Forms is able to display the data in table or graph form. An example is shown below:



Logged Readings Table

	Start Time	Duration	High	Average	Low	Description	Stop Time
1	3/11/99 2:12:17 PM	0:30:27.3	2.0625 V DC	1.9962 V DC	1.9898 V DC	Stable	3/11/99 2:42:44 PM
2	3/11/99 2:42:44 PM	0:00:00.9	1.8022 V DC	0.3092 V DC	0.0058 V DC	Unstable	3/11/99 2:42:45 PM
3	3/11/99 2:42:45 PM	0:00:07.6	0.0050 V DC	-0.0000 V DC	-0.0005 V DC	Stable	3/11/99 2:42:53 PM
4	3/11/99 2:42:53 PM	0:00:01.3	1.8836 V DC	1.5379 V DC	0.0600 V DC	Unstable	3/11/99 2:42:54 PM
5	3/11/99 2:42:54 PM	0:00:01.1	1.9625 V DC	1.9319 V DC	1.8865 V DC	Stable	3/11/99 2:42:55 PM
6	3/11/99 2:42:55 PM	0:05:10.7	1.9986 V DC	1.9974 V DC	1.9642 V DC	Logging Stop	3/11/99 2:48:06 PM
7							

Looking at this data, you can see that the logging session started at 2:12:17 PM and ended at 2:48:06 PM with an elapsed time around 36 minutes. You can also see that the signal being measured was stable at 2 volts for most of the logging session. However, there were 2 occurrences of instability detected by the meter during an 11 second period just before 2:43 PM.

The graph gives a visual indication of what happened around that time. What is interesting about this example is that the table of data shows that only 6 "events" needed to be stored in the meter memory over the entire 36 minutes of logging to capture the essence of what happened.

What this illustration shows is that with event logging, only a very small amount of memory was needed to store the information. To capture the same data using traditional data logging would require a 1 second sample rate over the entire 36 minutes which would have produced 2160 records of data.

Defining a significant event

When you describe the signal you are logging, you might describe noise levels, small 'normal variations' - possibly a drift, and some change in the signal level which you would consider interesting, or significant. In practice, any change less than the 'significant' level is 'uninteresting' and can be characterized by a block representing the span over which the signal varied before something interesting happened. FlukeView Forms allows you to specify a fixed input variation, or a percentage change that defines the beginning of something possibly of interest. With the percent of reading selection, you can also specify the minimum value to prevent reacting to noise levels on small signals.

To demonstrate how to define a significant event, go to the FlukeView Forms **Logging Window**. It provides an interactive tool that models how events will be logged, based on how you've defined what a significant event is.. Under the **Setup** tab, you can display a **Simulation graph** which simulates how the current event capture settings on the **Logging Window** would perform event logging on example data. Click **With Setup Data** to toggle between viewing the example data, and the events that would be produced during a logging session with these settings.

For more details, see the *Interactive Logging and the FlukeView Forms Logging Window* section later in this document.

Introduction to Event Logging, Stable and Unstable Periods, and Logging Interval

FlukeView Forms records data to describe the input behavior over a period of time - an event data record. Let's assume that we started logging readings that were all significantly the same. The first readings in this period set the boundaries defining the highest and lowest possible values which could be allowed in this period - the size of the "event window", if you will. This event will be logged, and a new event started, when either a reading is received that is outside this window size, or it is time for another 'interval' record.

- When the event is logged due to the input changing outside the window, its duration or **period** (stop time - start time) is checked, and if it too short, the event is labeled as an '**unstable**' input event, or unstable input data. If the duration was not too short, the event is labeled as "**stable**".

- If the event is closed because it was time for another 'interval' record, the event is labeled as **'interval'** data or an interval event.

- Regardless what caused the event to close, each event record consists of:
  - start time stamp
  - stop time stamp
  - highest actual value in the period
  - lowest actual value in the period
  - average value (excluding overloads) during the period
  - a description of the event - stable / unstable / interval / etc.

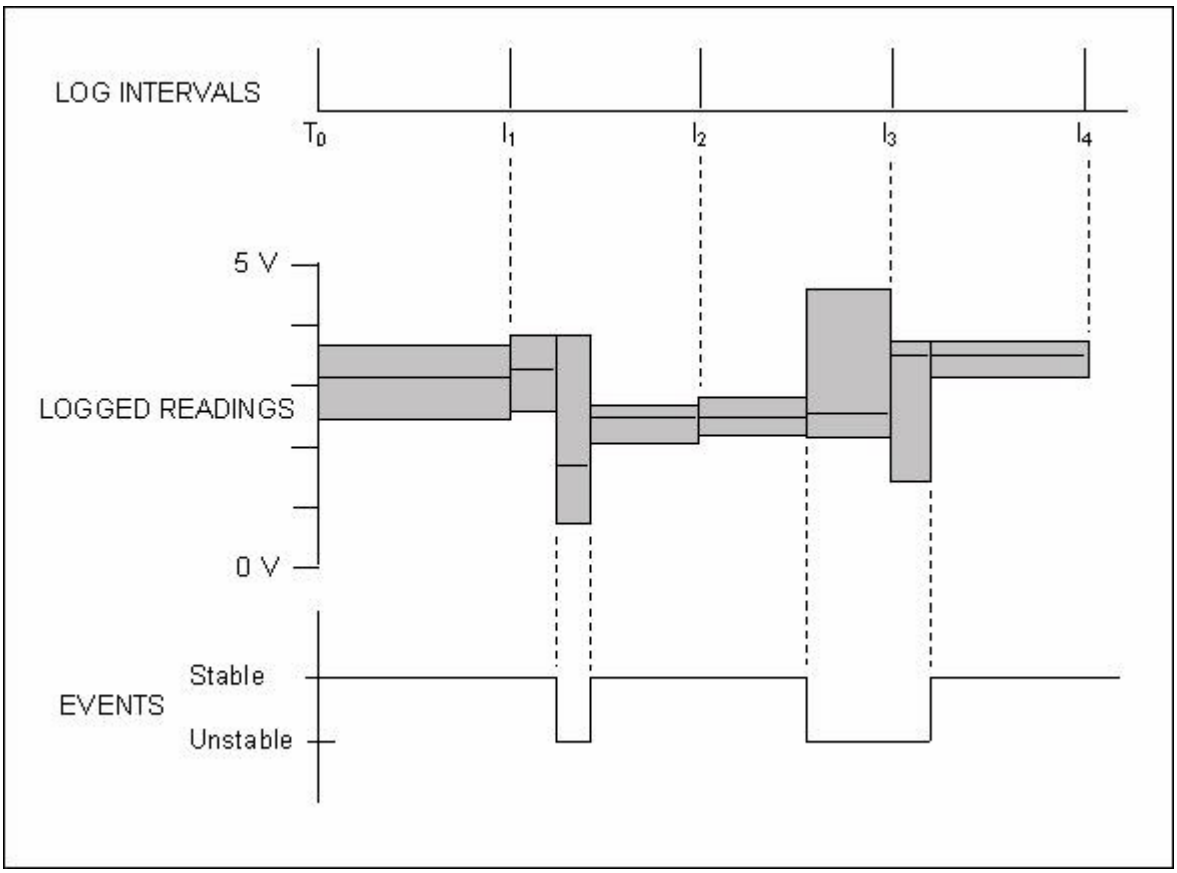
Why set a logging interval?

The number one reason to use a logging interval - do you have a requirement to show the average signal values in fixed increments? Setting the interval to that increment will insure that you can quickly show a table with that fixed duration. This is true even if there were input events during the interval. FlukeView Forms logged readings table has a right click menu where you may choose to show "All" / "Interval" / (input) "Events".

Another reason to set a logging interval concerns signal drift. Since the window size is determined in the first few readings of an event, a signal may drift out of the window size, resulting in a reading labeled as an input event, and displayed a block of data where the duration is how long it took to drift across the event threshold. In such a case, setting an interval will break the drift into more discrete chunks so that you can see the effect better, and because the window size is reset at each interval, result in the input events caused by ’significant’ changes rather than drift. FlukeView Forms provides some special ’event logging’ form objects (an amplitude/duration scatter graph and duration histogram) which process non interval data to show input event duration views.

Event Logging with input events and interval events

The following diagram illustrates a mixture of interval event and input events and how it will appear graphically in the FlukeView Forms software:



At the top of the diagram is a time line indicating when the end of one logging interval takes place and the start of another logging interval begins. The logging session starts at time T0. Intervals are shown ending at times I1, I2, I3, and I4. At the bottom of the diagram is a time line showing when the meter is logging stable or unstable signals. The logging session starts with a stable signal, then shows 2 periods of an unstable signal occurring.

In the center of the diagram, a graph is depicted which looks like a Logged Readings Graph in the FlukeView Forms software. The dashed lines indicate when the Fluke 89 / 189 will log for the preceding time period the high, low, and average measurement and the ending time stamp. The FlukeView Forms software constructs a box, whose vertical height represents the high and low measurements for a given logging period and whose horizontal length represents the period duration. The average for that period is shown as a straight line through the box. An explanation for each box in the above diagram follows:

Box #	Explanation
1	The first box is generated because a logging interval (I1) came to an end.
2	A stable signal went unstable, the box will represent the stable period that has come to a close.
3	The signal became stable again, data is logged about the unstable period.
4	A second logging interval (I2) has expired. (This interval has been split into 3 pieces because of some events.)
5	A stable signal went unstable.
6	The third logging interval (I3) expired.
7	The unstable period ended.
8	A fourth logging interval (I4) has ended.

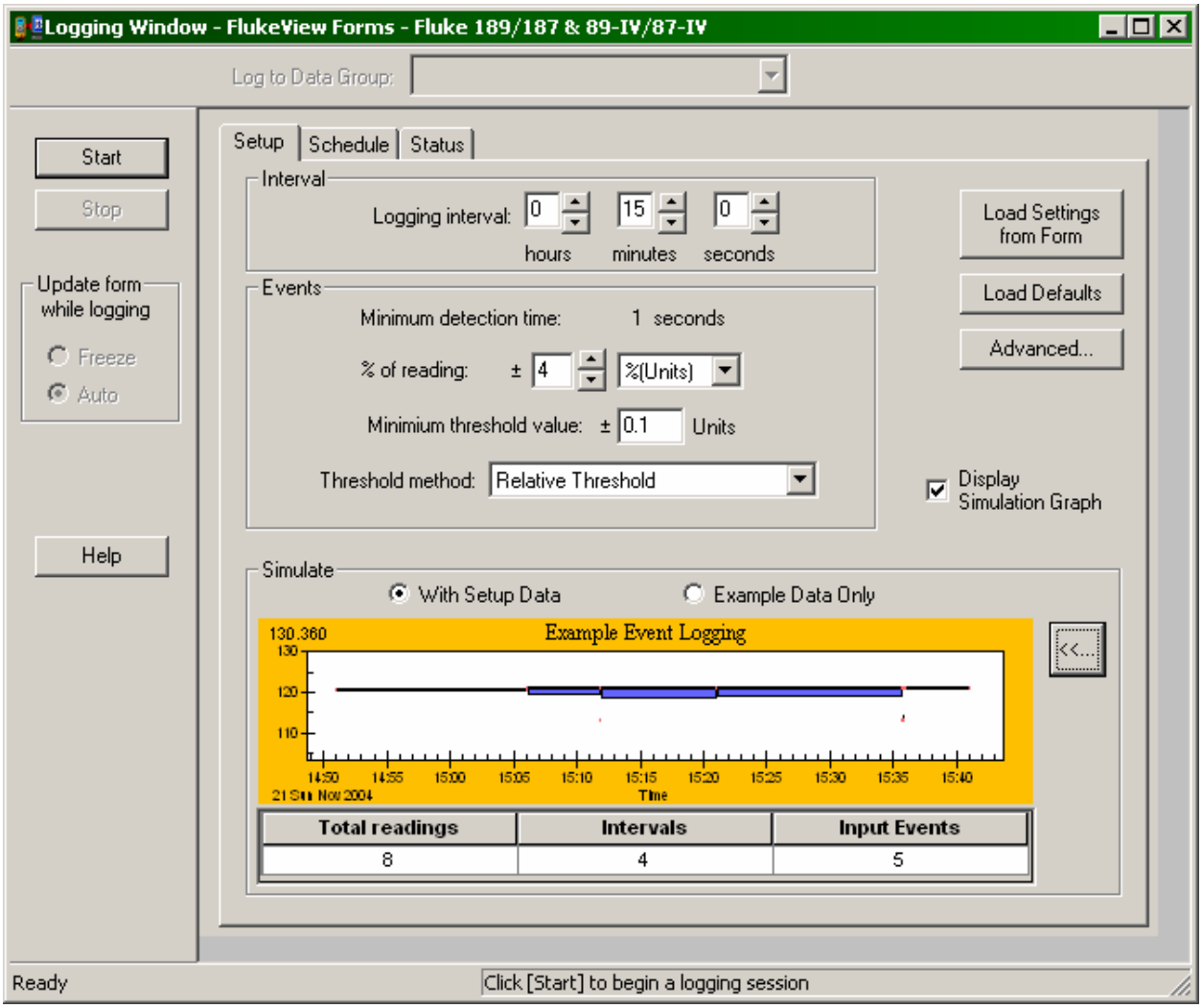
Looking at this diagram shows that the logging intervals get added to the logging results at regular periods of time. The events also get added to the logging results whenever a transition to or from a stable state occurs.

Note that when an interval expires, it will "split up" any period of stability (or instability) and cause a new logging period (box) to be started. FlukeView Forms logged readings table and logged readings graph have the capability to show just the interval data or just the event data if this is preferred. You can do this by placing the mouse pointer over the Logged Readings Table, clicking the right mouse key and selecting show data. While you are doing this, you also might want to try right clicking while the mouse pointer in over one of the Logged Readings Graphs and selecting View.

Configuring Interactive Logging in the FlukeView Forms Logging Window

So far, we've talked about how event logging is performed – either by the 89-IV or 189 meter internally, or by FlukeView Forms when doing interactive logging. You should have an understanding of the terminology, and how event logging is performed.

Now we'll examine the FlukeView Forms **Logging Window** – the tool in FlukeView Forms that configures and monitors interactive event logging – and describe how different settings in the **Logging Window**'s **Setup** tab affect the way events are captured during your interactive logging session.



A comprehensive overview of the **Logging Window** can be found in the *FlukeView Forms Users Manual*, in the *Interactive Logging* section. We will review only the **Setup** tab of the **Logging Window** here, as it relates to the concepts of event logging we've discussed in this document.

- **Interval**: The **Interval** section is where you specify what the logging interval will be for your interval events.

- **Events**: The **Events** (i.e., input events) section specifies the "event capture" window for events – i.e. how significant a change in the input signal will cause the current event to be closed, and a new event started.

    - **Minimum detection time**: The minimum amount of time for this meter to be able

to detect a new event (when **Threshold method** is not set to **Fluke 189 and 89-IV Settings**.

- **Threshold method**: Specifies whether the amount of change in the input signal that determines a new event is:
  - **Fixed Threshold**: a fixed value (e.g., "±2 mV DC")
  - **Relative Threshold**: a percentage of the current signal (e.g., a change in signal greater than 4% of the current signal)
  - **Fluke 189 and 89-IV Settings**: (Fluke 89-IV and 189 meters only) Indicates the meter uses its own event capture window to determine when new events occur. (I.e., the meter performs event logging, not the PC.)

  For each **Threshold method** setting, the **Events** section displays controls for setting the event capture window. Any changes made when the **Threshold method** is set to **Fluke 189 and 89-IV Settings** will cause the logging threshold to be set on the meter.

- **Simulate**: When **Display Simulation Graph** is checked, this displays the Simulation Graph. This shows how your current settings in the **Interval** and **Events** section (as well as **Advanced** settings) would affect example data from a simulated logging session.

  The example data can be changed by clicking the "<<" button; this brings up a dialog where you can specify the nominal input, the drift, noise, and a 'significant change' in order to make the example data more like the expected signal. The 'example data' reflects 3000 readings over a 50 minute period.

The rest of this technical note will discuss advanced features of the FlukeView Forms interactive logging; specifically:

- Data compression
- Limit bands
- Dual logging of both primary and secondary readings

Further data compression: Enabling Interval Data and Input Event data

The default is to enable recording both interval data and input event data. With properly set input threshold size, and an appropriate interval, the data storage should be relatively small. Both the logged reading graph and logged reading table can be set to post process the data to show the data as if you had enable only one or the other , interval or event data.

However, if you elect to store only one or the other type, this ability is lost. To disable the logging of either interval data or input event data, click on the **Advanced** button; this brings up the **Advanced Settings** dialog. The dialog is divided into three sections; unchecking either **Enable interval data** or **Enable input events** in the middle section will disable the collection of interval data or input event data, respectively.

Disabling both interval data and input event data results in combining all the data into a single compound event.

However, this feature becomes valuable when in used in conjunction with limit bands.

Limit Bands

There are times when you may be interested in data within a range of values, and not interested in the data outside of this range. FlukeView Forms supports setting different compression rules within three separate *bands* (ranges of values). Thus you could choose to record only interval data in one band, both interval and input event data in another, and elect to combine both interval and input events together (by disabling both) in the third.

1. From the **Logging Window**, setup tab, press the **Advanced** button. This brings up the **Advanced Settings** dialog, which shows the three separate bands.

2. Check **Enable high limit band** and set the high limit value. Enable/disable the data to record in the band. At this point there are two bands defined by the high limit value.

3. Check **Enable low limit band** and set a low limit value. Enable/disable the data to record in the band. At this point there are now three active bands.

When the data crosses from the mid band into the high or low band, the event in progress for the mid band is closed, and labeled a 'limit band'. A new event is started in the applicable band, and the compression rules modified as set.

Limit Band events are counted as input events as they are caused by the input changing.

Since it is possible for the data to 'bounce around' the limit value, FlukeView Forms uses the **Minimum threshold value** setting on the **Logging Window** dialog (with **Threshold method** set to **Relative Threshold**) as a hysteresis value. The data must drop by this far into the previous band before it closes the event.

Dual Logging of Primary and Secondary Readings

Some meters can display two readings at the same time. These meters have both primary and secondary displays - they can show two functions of the same input signal on the meter display at the same time.

For example, the meter could be set to display the AC voltage reading of the signal it is measuring, while also displaying the frequency (Hz) reading as well, below the V AC reading on the meter display.

If you interactively log from a meter that is configured to display two readings, FlukeView Forms can interactively log both readings at the same time (with the exception of the Fluke 789 ProcessMeter™). Readings on the meter's primary display will be referred to as "primary readings", and readings displayed on the secondary display will be referred to as "secondary readings".

To do this, you need to do the following:

- Configure the meter so that both readings appear in both the primary and secondary display of the meter.
- On the FlukeView Forms **Logging Window**, **Setup** tab, if input event logging is enabled under the **Events** section, set the **Threshold method** to either a **Relative** or **Fixed Threshold**. Do not set to it to **Fluke 189 or 89-IV Settings** - this will only log the primary reading, and not the secondary reading.
- Also for event logging, you will need to choose which meter function is used to determine when input events begin or end. Events recorded for both primary and secondary readings will begin and end at the same time. When they begin or end depends on whether you set the threshold for input events to the meter function of the primary or the secondary readings.

  For instance, if your meter was displaying both AC voltage and frequency, and you wanted input events to begin and end depending on the AC voltage readings, you would select "V AC" in the **Events** section of the Logging Window, to the right of the **% of reading** setting or **Threshold value** (depending on what the **Threshold method** is set to). If you selected "Hz" instead, the events will begin and end when the frequency exceeds the relative or fixed threshold you've set.

We refer to the meter function used for determining when events begin and end as the "Master" function, and the other function as the "Slave" function.

Now begin interactive logging. You will notice that any Logged Readings Tables have two rows each for each event - the first row displaying the reading for the Master function, and the second row display the reading for the Slave function.

Also, the Logged Readings Graph is displaying traces of both meter functions side-by-side - the graph is split horizontally, and shows a trace in each half. The top half shows the logged readings for the Master function, and the bottom half shows the logged readings for the Slave function.

You can choose to display only one of the traces on the graph by right-clicking on the Logged Readings Graph, and selecting **Select Traces/Groups**.. Select only the traces you want to display.

Also, if you prefer to see both traces overlapped on the same graph, rather than side-by-side in a "split view", right-click on the Logged Readings Graph. Select **Graph View** and choose **Overlap**. To return to the "split view", select **Split** instead.