Sequence of Events Recording (SER) Device Selection Guide

Summary
Critical power applications require precision timing for monitoring and control. Sequence of Events Recording (SER) systems are an essential part of an Electrical Power Monitoring System (EPMS), with time-stamping accuracy of 1ms required for system diagnostics and root-cause analysis.

Stand-alone Sequence of Events Recorders are specifically designed to provide reliable event recording. Other devices, such as meters, relays and PLCs are equipped with I/O, but not all are capable of performing the same level of SER functionality.

This document describes the five (5) main criteria for an effective Sequence of Events Recording device and provides a checklist of features to help the reader determine the appropriate SER device for a given project and budget.

Introduction

Sequence of Events Recording (SER) systems are an essential part of Electrical Power Monitoring Systems (EPMS) for "critical power" applications, such as data centers, hospitals, and micro-grids. Unlike the simpler radial systems found in commercial buildings, these complex power networks are characterized by multiple power sources and parallel paths to serve loads. A single disturbance can initiate a series of events in rapid succession. Sequence of Events Recording systems answer the questions "what happened and when?" [1] Without the ability to reconstruct an incident in the proper sequence, root-cause analysis is impossible. Sometimes the information provides critical details about an outage, but it can also warn in advance about conditions of increased risk and prevent an outage. Because power system events can occur within milliseconds, a time accuracy of 1ms is the generally-accepted standard for sequence of events recording. [2]

SER Devices

Sequence of Events Recorders monitor the status of up to 32 high-speed input channels and record a time-stamped event for each state change (Off-to-On or On-toOff). Other devices with general-purpose I/O, such as relays, meters and PLCs may offer SER features. Regardless of hardware platform, several capabilities are required for sequence of events recording:

1. Precision time reference (internal clock accuracy of 100 microseconds or better)
2. High-speed response by discrete inputs (1 kHz sampling min.)
3. Event filtering (filter, debounce, and chatter functions)
4. Data logging to on-board non-volatile memory (8000 events or more)
5. Network communications for access to event records (Ethernet, Modbus TCP)

Unless a device meets all of these criteria, it may not be suitable for sequence of events recording. These are explained in more detail in the following sections.

Event	Date and Time	Channel	Event Type	Status	Time Quality
6	01/20/2021 23:03:10.431	System Event	Setup Changed	---	3:Bad (no sync)
5	01/20/2021 23:02:50.728	System Event	Power On	---	3:Bad (no sync)
4	02/11/2021 16:48:26.679	System Event	Power Fail	---	3:Bad (no sync)
3	02/11/2021 16:47:02.359	System Event	Event Log Cleared	---	3:Bad (no sync)
2	02/11/2021 16:47:01.116	System Event	Power On	---	3:Bad (no sync)
1	02/11/2021 16:44:02.409	System Event	Firmware Upgraded	---	3:Bad (no sync)

Sample event log from Sequence of Events Recording system

PRECISION TIME

To achieve millisecond time-stamping, the accuracy of the SER device's internal time clock needs to be an order of magnitude better (<100 microseconds). Therefore, the first requirement for an SER device is that its onboard clock have provisions for synchronizing to a known accurate time source, such as a GPS time signal. [3] The most common precision time protocols used by power system devices are IRIG-B and DCF77. These are described in detail in Technical Notes TN-102 and TN-103 from Trystar. (See the "For More Information" section at the end of this document.)

NTP (Network Time Protocol) is also supported by some devices for obtaining time/date over an Ethernet network, but these are typically not sufficient to ensure 1ms accuracy of event time stamps. IEEE-1588 or PTP Protocol, can achieve much higher accuracy over an Ethernet network, but is not yet widely supported by power system devices.

HIGH-SPEED INPUTS

SER devices have "high-speed" inputs, with defined response times of 1ms or less. To achieve reliable time-stamping with 1ms accuracy, the SER device must monitor the status of input channels and take a "snapshot" of all inputs simultaneously at the start of every millisecond interval (1 kHz sampling).

Alternatively, an interrupt-driven scheme (report by exception) is sometimes used by meter and relay processors. However, the response time required to service the interrupt can vary, since it depends on processor loading, prioritization of tasks, etc. This can result in an unacceptable degree of uncertainty, perhaps not for the first event, but to accurately time-stamp subsequent events in rapid succession.

For reliable event recording, the response time of an SER device must be considered deterministic, not adversely affected by other demands on the processor. The device must be capable of detecting every single transition event within the 1ms window.

In addition to status-change events, it is often useful to have time-stamped voltage and current waveform data associated with the events. To accomplish this, a "Trigger output" contact is provided which can be configured to close when a given input channel changes state. This in turn triggers a waveform capture (WFC) feature of an advanced power meter.

EVENT FILTERING

In a practical SER application, filtering is required to ensure meaningful and reliable event recording. Without proper filtering, momentary spikes due to noise would be recorded as events, and single (real) events could be recorded multiple times. An SER device must include three different user-configurable filter functions which define the method it uses to process events. The names may vary by manufacturer, but the following three functions must be implemented:

Filter— The amount of time for which the state of a digital input point must remain detected in a valid "on" or "off" condition before it is considered to be in that position for event-recording purposes. This pre-event filter complements a post-event function called "debounce."

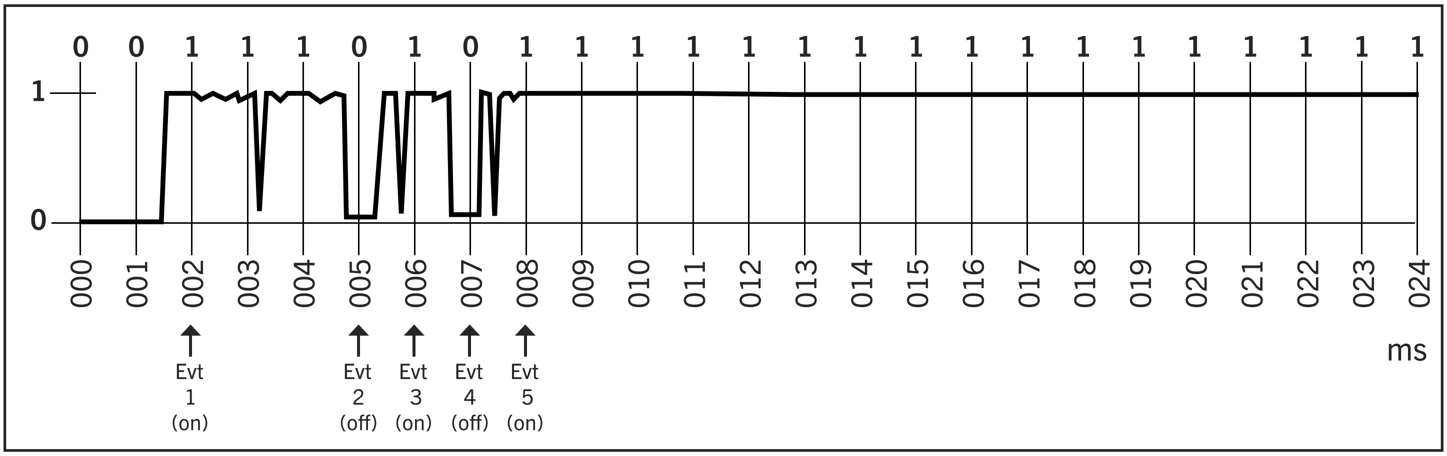
Debounce—period of time post-event during which event processing for a given input is suspended to avoid recording multiple events associated with the same state change. As the name suggests, this function filters out normal signal excursions ("bounce") which accompany the closure of mechanical contacts.

Chatter—number of repeated events within a given time period, when exceeded, event processing is suspended on that channel.

Both the filter and debounce functions are used to distinguish real status change events from unintended "noise" from normal relay contact operation, while the chatter function ensures that the event log is not filled with superfluous events caused by a defective ("chattering") relay contact. The following sections illustrate each of these functions in more detail.

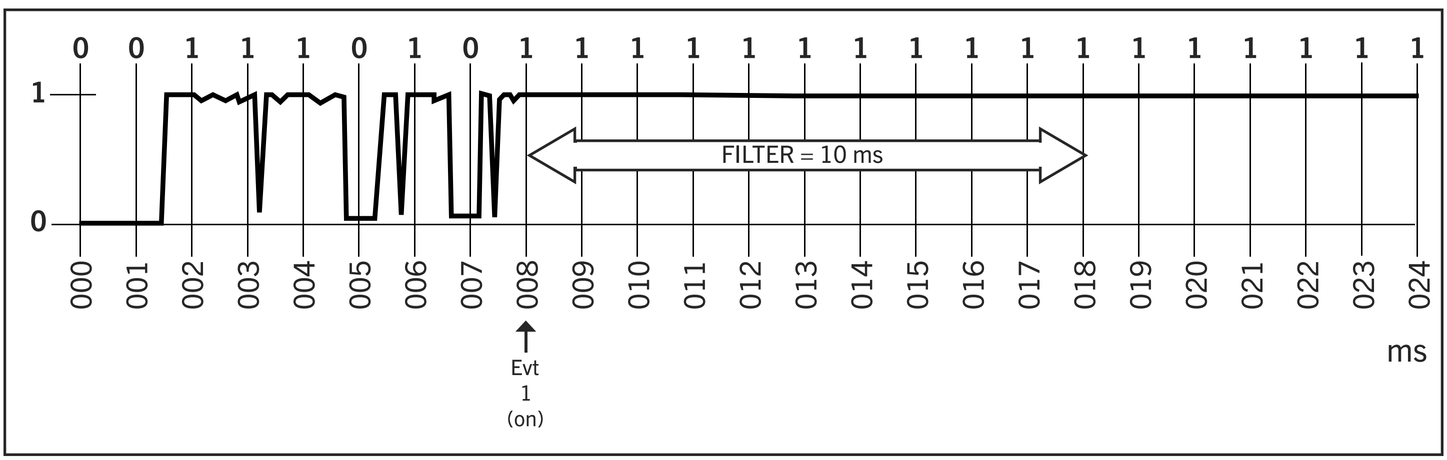
Filtering Momentary Spikes

Without proper filtering, momentary spikes caused by noise or interference would be recorded as events. Consider the example shown in the figure below. The graph shows the signal as seen by the input, and the numbers across the top show the status as determined by the device (0=off and 1=on).



Filter = 0 ms (no special filtering)

Without any filtering, the first spike is recorded as an event (Evt 1, time-stamp= 002 ms). The next dip (at 003 ms) is so short it is not detected as a new event. However, at 005ms, Evt 2 is recorded, followed immediately by successive events 3, 4 and 5—even though these are probably all due to a single contact closure.

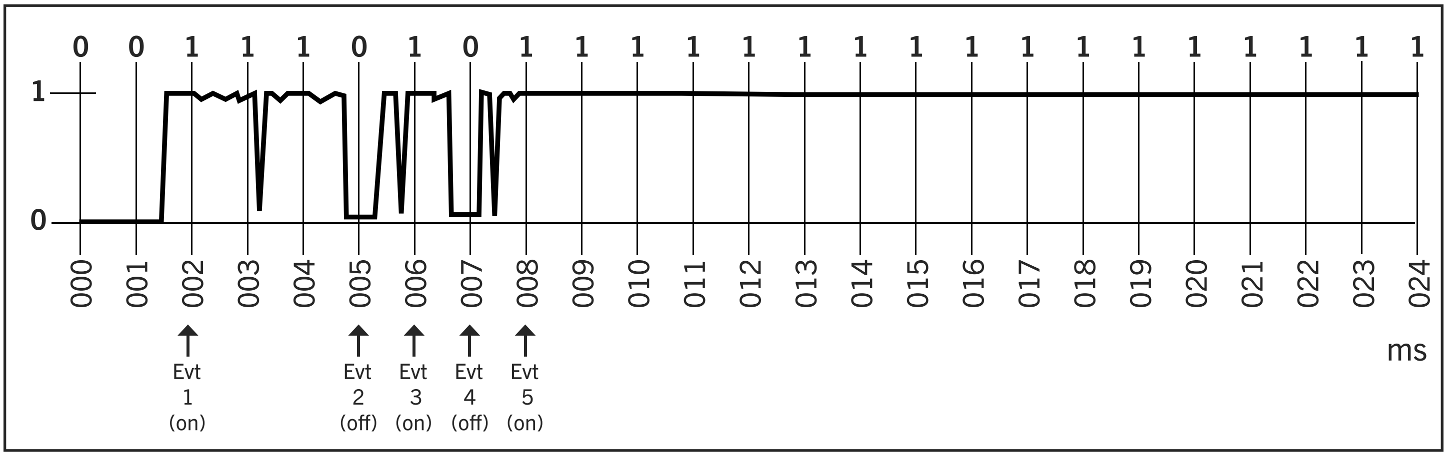


Filter = 10 ms

Next, we'll apply a 10ms filter to the same signal. Because none of the early spikes detected lasts longer than the filter setting of 10ms, these are all ignored for the purposes of event recording. At 018ms, the new state (1=on) has persisted for a period of time exceeding the filter setting of 10ms; therefore, the device records this as Event 1, with time-stamp of 008ms (since that is when the filtered condition began).

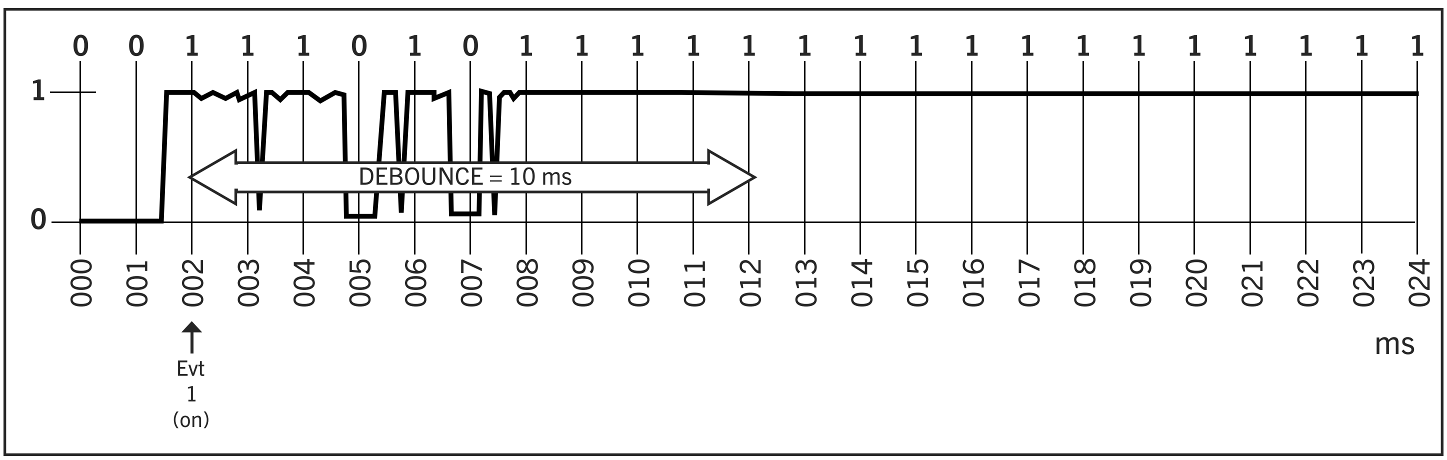
Debounce Function

Returning to our previous unfiltered example, the graph shows the signal applied to the input, and the status (0=off and 1=on) is indicated across the top.



Filter = 0 ms (no special filtering)

Instead of using a pre-event filter, let's apply a post-event filter, called "debounce." In this scenario, the event detection is very sensitive, and afterwards, event recording is suspended for the duration of the debounce setting. (See graph below.)

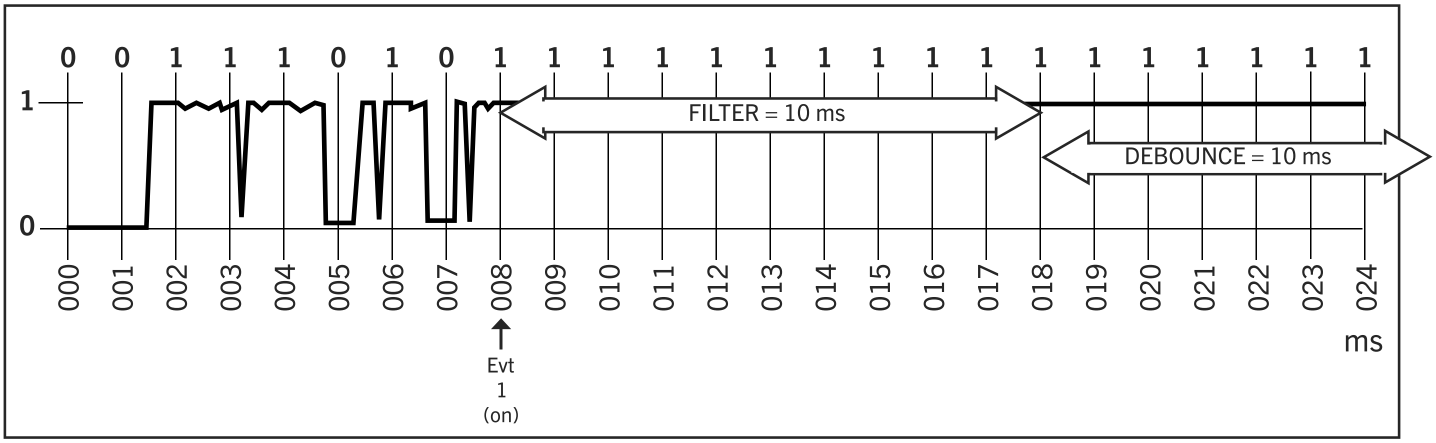


Debounce function = 10 ms

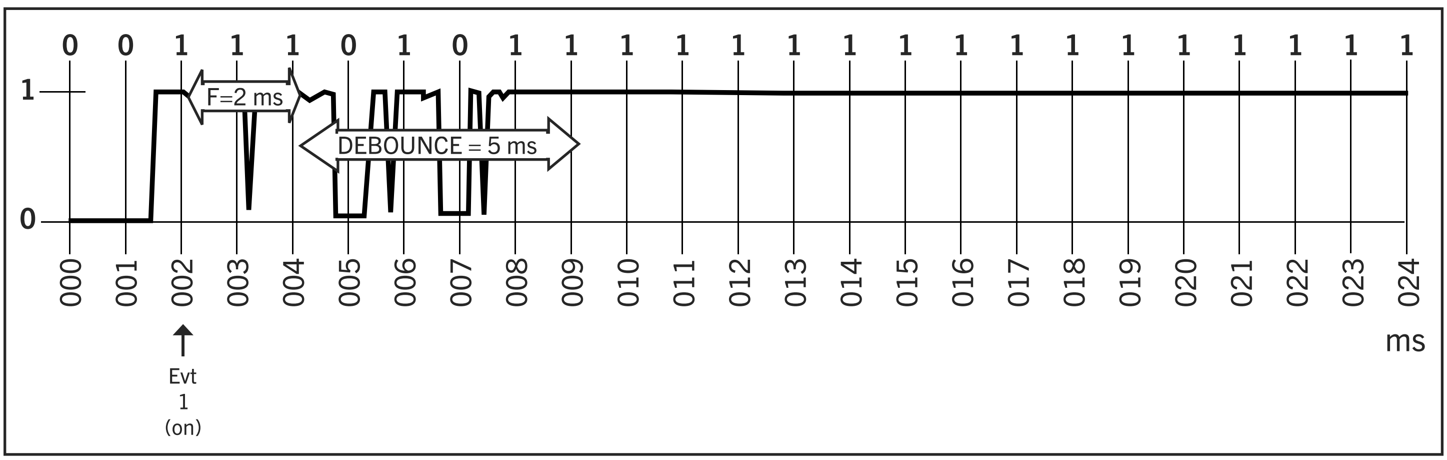
Event 1 is recorded at the very first detected change in state (time = 002ms), and the debounce filter "desensitizes" event recording on that channel for the 10ms following the event. The other dips and spikes were likely caused by the "bouncing" of the contact, and so these are—in this case—appropriately ignored.

This avoids multiple events being recorded for the same contact closure, and has the advantage of time-stamping the event record at the earliest time at which the change is detected. However, as mentioned previously, without (pre-event) filtering, the unit is vulnerable to recording "nuisance events" due to momentary spikes.

If the filter and debounce functions are both increased (e.g., 10ms or higher), then contact bounce may be filtered out, but the time stamp for the event will be delayed until the starting point at which the input reached its steady state (remained in the new state throughout the filter time period), not necessarily the true contact closure.



Filter = 10ms / Debounce = 10 ms



Filter = 2ms / Debounce = 5 ms

The optimal combination of settings would be to set the filter as low as possible to screen out momentary spikes and set the debounce long enough to ride-through contact bounce. Consider the example above in which the filter is set to 2 ms and debounce is set to 5 ms. In this case, the event is detected and time-stamped at 002ms, and the contact bounce is filtered out.

The choice of filter and debounce settings depends on the expected characteristics of the monitored contacts and a trade-off between highly-accurate time-stamping and the risk of recording spurious events.

A conservative setting for both functions might be 20ms. However, depending on the application, it may be desirable to adjust one or both of these values per input channel to achieve optimal results. Most SER devices support such flexibility.

Chatter Count Function

In addition to the filter and debounce functions, a chatter count function is provided to further guard against superfluous events being recorded and filling the event log. Per the 2007 IEEE Standard for SCADA and Automation Systems [4], the definition of chatter function is as follows:

"Chatter function—a facility that is used to disable a digital input point if the number of state changes of that point during a defined time interval is excessively high."

Typically, SER devices support values for chatter-count setting from 0 to 255 (where 0 indicates the function is disabled). If the number of state changes exceeds this value within a given period (e.g., one minute), the SER device suspends event processing for that input, and an event is recorded in the event log to indicate this condition. Event recording is suspended for that channel until the input is considered stable (the chatter count for a subsequent one-minute interval drops to less than the user-specified value).

ON-BOARD NON-VOLATILE MEMORY

Supervisory software may access the data from each SER every 15 minutes, every hour or even once a day. Since numerous events can occur in a very short period of time, an SER device must have its own on-board, non-volatile memory and be sized sufficiently to store hundreds or even thousands of event records—typically 8000 events or more, stored "First In, First Out"(FIFO).

In an SER event log, each event record must contain at least the following data:

- Event number (index number)
- Time/Date stamp (referenced to UTC, with ms resolution)
- Channel name (user-configurable text)
- Event type (e.g., status change, time-sync lock/fail, setup change, chatter, etc.)
- Status (Off-to-On or On-to-Off—with user-configurable descriptive text)
- Time quality (indication of time-stamp fidelity)
- Sequence number (unique serial number, even if FIFO log "rolls over")

In addition, an alternate set of data logs can be useful where status of input/output channels must be reported as a group.

For example, to generate test reports for an Emergency Power Supply System (EPSS), it is necessary to report the states of each device in the generator test process. While it might be possible to deduce the state of one member of a test group by scanning the entire event log, a simpler method is to record the states of all group members each time any member changes state. This greatly simplifies report generation and ensures reliable results.

NETWORK COMMUNICATIONS

Thanks to distributed processing and highly-accurate time-stamping of events, SER devices do not need to be scanned rapidly by supervisory software. Instead, SER devices can be polled periodically for new events and these can be uploaded to a master database.

For best results, SER devices must integrate with other electrical power monitoring system (EPMS) devices using the same protocol, typically over an Ethernet local area network using Modbus TCP. SER devices should support a minimum set of Modbus function codes, including FC 20 for efficient file record access.

A pre-defined Modbus register list allows pre-engineered "native device drivers" to be developed for standard EPMS software. Multiple masters must be supported. Proprietary schemes (sometimes used with PLCs) in which event records are erased after being read by the master are not permitted.

An embedded web server in the SER device is used for setup and monitoring, often before the EPMS software is operational. Indeed, these web pages can facilitate commissioning of the EPMS software itself and contribute to overall system reliability.

SER DEVICES

Sequence of Events Recorders

Sequence of Events Recorders are stand-alone devices designed specifically for event recording, and so these typically meet all criteria for sequence of events recording discussed previously.

Trystar Sequence of Events Recorders from Trystar are available with two I/O options:

- SER-32e or SER-3200—32 Inputs
- SER-2408—24 Inputs and 8 Outputs

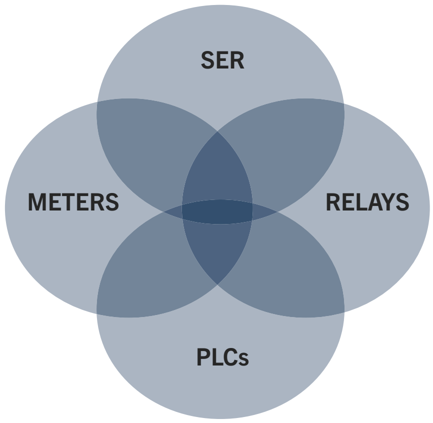
A summary checklist of all features is shown in the table below.

SER DEVICE FEATURE CHECKLIST

SER Device Function	Requirement	Trystar SER-32e (32 inputs)	Trystar SER-3200 or 2408 (32 or 24 Inputs / 8 Outputs)
Precision Time-Sync (clock accuracy 100 µsec)	IRIG-B	✓	✓
	DCF77	✓	✓
	Other (SER inter-device)	✓	✓
High-speed I/O response	Time-stamp accuracy of 1ms	✓ ①	✓ ①
	Trigger output	✓	✓
Event filtering	Filter (0 to 65535 ms)	✓	✓
	Debounce (0 to 65535 ms)	✓	✓
	Chatter (0 to 255)	✓	✓
Data logging	Non-volatile memory	✓ ②	✓ ②
	Time/date, Channel, Type, Status, Time Quality, Serial No.	✓ ③	✓ ③
	EPSS reporting data logs	✓ ④	✓ ④
Network communications	Ethernet (10/100BaseTx)	✓	✓
	Modbus TCP	✓	✓
	Web server	✓	✓
	Multiple masters supported	✓	✓

1. A dedicated events processor ensures reliable 1ms response for I/O
2. Event log stores up to 8192 events.
3. Channel name, off-text and on-text are all user-configurable.
4. Total of 16 data log groups for EPSS test reporting (in addition to main SER event log).

SUMMARY



Stand-alone Sequence of Events Recorders are specifically designed for high-speed recording of status changes, time-stamped to 1ms. Devices such as meters, relays and PLCs may also include SER functionality, in addition to their primary functions. All SER devices should satisfy the following selection criteria:

1. Precision time reference (internal clock accuracy of 100 microseconds or better)
2. High-speed response by discrete inputs (1 kHz sampling min.)
3. Event filtering (filter, debounce, and chatter functions)
4. Data logging to on-board, non-volatile memory (8000 events or more)
5. Network communications for access to event records (Ethernet, Modbus TCP)

The best way to judge a device's suitability for SER performance is to consult the manufacturer's literature or test results pertaining to the criteria above. Whether to use dedicated SER devices or built-in SER features of meters or relays will depend on project specifications, performance goals and budget constraints.

References

1. Kennedy, Robert A., P.E., "GPS Time Synchronization: How precision timing and sequence of events recording will make the Smart Grid even smarter," Electrical Construction & Maintenance (EC&M) magazine, August 19, 2011, pp. 18-20.

http://ecmweb.com/computers-amp-software/gps-time-synchronization

2. Brown, PE, Bill, and Mark Kozlowski, "Power System Event Reconstruction Technologies for Modern Data Centers," Square D Critical Power Competency Center. Aug. 2006.

http://www.criticalpowernow.com

3. Dickerson, Bill, P.Eng., Arbiter Systems, Inc. "Time in the Power Industry: How and Why We Use It."

http://www.arbiter.com/solutions/event-time-and-reconstruction.php

4. "IEEE Standard for SCADA and Automation Systems," IEEE Std C37.1-2007 (Revision of IEEE Std C37.1-1994).

TRYSTAR®

15765 Acorn Trail / Faribault, MN 55021 / 507.333.3990
Document Number: 0016TN0326 / March 2026 /