Tech Note

How the M1 Rule Engine Works

An event is any recognized system change, whether it be arm, disarm, an alarm, an input zone change or an output or light change. There are many more events than these. An event is something that occurred at a specific point in time. It may or may not be associated with a continuous state. For example, when a burglar alarm is armed, the instant it is armed is an event - a discrete point in time. That event is associated with a continuous state thereafter – the system is now armed. But when the clock changes to 1:00 PM , that discrete point in time (event) is not associated with any continuous state.

WHENEVER clauses in the rules refer to discrete events. AND clauses in the rules refer to continuous states. So WHEVEVER an event occurs AND some state exists at that moment, the THEN part of the rule is executed or "fired."

When an event occurs, it is put in a list of events. If two or more events occur near-simultaneously, they are both put in the list. The order of events in the list doesn't matter.

As long as the list is not empty, the M1 scans all the rules from the first to the last looking for any rule having a WHENEVER event that matches an event in the list. When a matching WHENEVER event is found, that rule's AND clauses are evaluated. If all the ANDS evaluate to true, the THEN statement is executed (the rule is "fired"). If two or more rules match an event in the list, those rules are processed in order from the first to the last. After all rules have been scanned, the events are removed from the list.

Rules can cause new events. If this happens, the new event is added to the list. When the initial scan completes, the original events are removed from the list, but the new events are retained. A second scan occurs to see if there are any WHENEVER events that match the new events.

An couple examples will help to clarify this…

Suppose we have the following (simplified) rules:

1 WHENEVER Area 1 is armed
  THEN Turn Light 1 on
2 WHENEVER Light 1 is turned on
  THEN Speak Light 1 Name
3 WHENEVER Area 1 is armed
  THEN Turn Light 2 on

Rules 1 and 3 could have been combined in to a single rule, but they are shown separately here to illustrate how the processing works. When someone arms area 1, the "Area 1 armed" event is put in the events list. The rules are scanned starting with the first rule, which matches. There are no AND clauses here, so the rule is fired and Light 1 is turned on. The "Light 1 on" event is placed in the list as a new event. Scanning continues for the "Area 1 armed" event, and we find that rule 3 matches it. So rule 3 is fired, and "Light 2 on" is placed as a new event in the list. The end of rules is reached an so the arm event is removed from the list, leaving the Light 1 on and Light 2 on events. The rules are then scanned again, from 1 to 3, to process these new events. Rule 2 matches one of the events in the list

so it is fired. There are no rules to handle Light 2 on, so nothing happens with it. When the second scan is complete, the events list is cleared.

If we were to swap rules 1 and 2 above, everything would still be processed correctly.

1 WHENEVER Light 1 is turned on
THEN Speak Light 1 Name
2 WHENEVER Area 1 is armed
THEN Turn Light 1 on
3 WHENEVER Area 1 is armed
THEN Turn Light 2 on

When the area is armed, the scan comes to rule 2 and fires it. Rule 3 would be fired also. Both rules place a new event in the list. So on the second scan, rule 1 matches one of the events in the list and it is fired. Because there is no rule to handle Light 2 turning on, nothing happens because of it, and both events are removed from the list.

Now let's look at an example that, on the surface, should work, but doesn't.

1 WHENEVER F1 on any keypad is pressed
AND Area 1 is disarmed
THEN arm Area 1
2 WHENEVER F1 on any keypad is pressed
AND Area 1 is armed
THEN disarm Area 1

You would think that pressing F1 would alternately arm and disarm the area as intended. But that's not what happens. When F1 is pressed, it's event goes into the list. Scanning begins. Rule 1's WHENEVER matches the event. Suppose the area is disarmed at the time – so the AND evaluates to true. The rule is then fired, arming the area. Scanning continues. Rule 2's WHENEVER also matches the F1 event. When it's AND clause is evaluated, it also evaluates to true, because the area was just armed by Rule 1. So rule 2 is also fired, disarming the area! That's not what we wanted. So how do we work around it? We need to momentarily "disable" rule 2. We can do that by setting some kind of "flag" for a short time until rule 2 is evaluated. "Phantom" lights or outputs are useful for this. A phantom output is any unused output in the system – any output that is not wired up. For our example, suppose the group of outputs 97-112 are not used or wired in the system. So let's use output 100, and re-write our rules with it.

1 WHENEVER F1 on any keypad is pressed
AND Area 1 is disarmed
THEN arm Area 1
THEN turn Output 100 on for 1 second
2 WHENEVER F1 on any keypad is pressed
AND Area 1 is armed
AND Output 100 is off
THEN disarm Area 1

Now, when rule 1 fires and arms the area, it turns Output 100 on for a second. Rule 2 is evaluated immediately and, even though the area is armed, Output 100 is on. So rule 2 is not fired. A second

later, the output turns off. If F1 is pressed then, what happens? Rule 1 is evaluated, but not fired, because the area is armed. Rule 2 is evaluated. Both AND clauses are true this time, so that rule is fired, disarming the area. In both cases, when the area is armed and again when it is disarmed, a new event (arm or disarm) is placed in the events list. Following the first scan, the older events are removed and the rules are scanned again to handle the new events. In our example, there are no rules to handle arm or disarm, so nothing happens and the second scan completes and removes the new events from the list.

There are a few caveats to be careful of when programming rules. One concerns time spans across midnight. Suppose we wanted to turn on a light when the front door opens between 9:00 PM and 6:00 AM. On the surface, the following rule would seem to do this:

1 WHENEVER Front Door Becomes not secure
AND the time is later than 9:00 PM
AND the time is earlier than 6:00 AM
THEN turn Light 1 on

This rule will never fire, because the time cannot be both later than 9:00 PM and earlier than 6:00 AM. Midnight marks a new day. So time spans later than 9:00 PM run from 9:00 PM to midnight. Likewise, time spans earlier than 6:00 AM run from midnight to 6:00 AM. To make this work correctly, we need two rules as follows:

1 WHENEVER Front Door Becomes not secure
AND the time is later than 9:00 PM
THEN turn Light 1 on
2 WHENEVER Front Door Becomes not secure
AND the time is earlier than 6:00 AM
THEN turn Light 1 on

Also remember that “later than” does not include the time specified. “Later than 9:00 PM” means 9:01 PM until midnight. Similarly for “earlier than.” The same holds true when comparing other values with greater than and less than.

For more information:
Visit www.elkproducts.com/M1-rules for rule writing tips and examples
Watch our rules webinar at http://youtu.be/LP7E3Lt8yFQ
Check out our support forum: www.elkproducts.com/M1-forum