# 01 07 Event-Schedule-Logic 801701

## Application program usage

| | |
|---|---|
| Manufacturer: | Siemens |
| Product family: | Controller |
| Product type: | Controller |
| Name: | Event-Schedule-Logic Controller N 350 |
| Order-No.: | 5WG1 350-1AB01 |

## Functional description

The Event-Schedule-Logic Controller N350 is a DIN rail mounted device.

In a compact unit the module offers
– Event programs,
– Schedule programs (weekly scheduler) and
– Logic functions
for binary input and output signals.

Up to ten event programs are available. For each event program up to ten event actions may be activated. An event program is triggered via an associated event object. The event trigger type may be chosen from this list:
– Reception of any telegram (0 or 1)
– Reception of 1
– Reception of 0
– Change from 0 to 1
– Change from 1 to 0
The value sent (0 or 1) can be defined per event action. The delay of an event action with respect to the time of the event trigger may also be defined.

The weekly scheduler provides a total of 100 schedules for 20 time controlled channels. Each schedule switches a time object on the minute at a pre-defined time on one or several days of the week.
The schedules are executed based on a controller-internal clock which must be synchronized at least once a day with a master time source. The 4-channel time switch REG 372 (order number: 5WG1 372-3EY01), the 4-channel time switch with DCF77 REG 372/02 (order number: 5WG1 372-3EY02), the ISDN gateway N147 (order number: 5WG1 147-1AB01), or the IP Interface AP146 (order number: 5WG1 146-3AB01) are available as master time clock or time source.

Ten logic gates, each with up to six inputs and one output, are available. Each gate's logic may be selected from this list: AND, OR, NAND, NOR. Individual logic gate inputs may be inverted. If the configurable send condition, i.e. send on each reception or only on change at output, is fulfilled then the send filter determines whether any output value or only 0 or only 1 is sent.

With the ETS (EIB Tool Software) the application program is selected, its parameters and addresses are assigned appropriately, and downloaded to the Event-Schedule-Logic Controller N 350.

## Application Examples

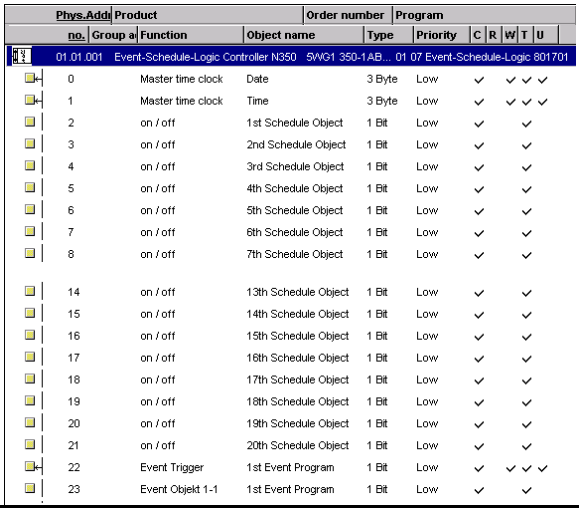
- Indoor and outdoor lighting control applications
- Lighting control dependent on outdoor light level and weekly schedule (opening hours)
- Lighting control scenes with dimming in conjunction with a scene controller
- Timer based lighting control
- Control of shutters, blinds, and shades
- Individual schedules for automated comfort (heating, lighting, shading...)
- Programming for different life styles and user profiles (scene control)
- Irrigation control / water storage control
- Garage Door Control

Schedule Control examples
- Every day the lighting of company garage is automatically turned on at 06:00 and turned off at 22:00.
- Monday through Friday the entrance door lighting of a house is switched on at 18:30 and switched off at 06:00
- Monday through Friday the bed room shutter is raised at 06:00 and closed at 21:30. On Saturday and Sunday it is opened at 08:30 and closed at 22:00.

Event Control examples
- Switch lighting of garage on and open motorized garage door – but only after authorized entrance.
  Switch lighting of garage off and close motorized garage door – after a specified delay time has passed and the motion detector has signalled an empty space
- Switch lighting of a house entrance on if the ambient light falls below a certain level (this requires a binary light level detector output)
  Switch lighting of a house entrance off if the ambient light level rises above a certain level (this requires a binary light level detector output)

## 01 07 Event-Schedule-Logic 801701

### Communication objects

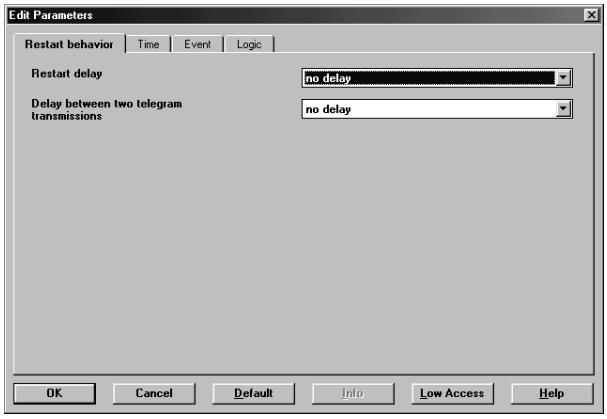


| Phys.Addr | Product | | Order number | Program | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| no. | Group a | Function | Object name | Type | Priority | C | R | W | T | U |
| 01.01.001 | | Event-Schedule-Logic Controller N350 | SWG1 350-1AB... | | 01 07 Event-Schedule-Logic 801701 | | | | | |
| 0 | | Master time clock | Date | 3 Byte | Low | ✓ | | ✓ | ✓ | ✓ |
| 1 | | Master time clock | Time | 3 Byte | Low | ✓ | | ✓ | ✓ | ✓ |
| 2 | | on / off | 1st Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 3 | | on / off | 2nd Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 4 | | on / off | 3rd Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 5 | | on / off | 4th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 6 | | on / off | 5th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 7 | | on / off | 6th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 8 | | on / off | 7th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 14 | | on / off | 13th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 15 | | on / off | 14th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 16 | | on / off | 15th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 17 | | on / off | 16th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 18 | | on / off | 17th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 19 | | on / off | 18th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 20 | | on / off | 19th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 21 | | on / off | 20th Schedule Object | 1 Bit | Low | ✓ | | | ✓ | |
| 22 | | Event Trigger | 1st Event Program | 1 Bit | Low | ✓ | | ✓ | ✓ | ✓ |
| 23 | | Event Objekt 1-1 | 1st Event Program | 1 Bit | Low | ✓ | | | ✓ | |

| Obj | Function | Object name | Type | Flag |
|---|---|---|---|---|
| **0** | Master time clock | Date | 3 Byte | C WTU |
| This object must be connected with the group address for the date sent by the master time clock. The schedule control is not started until date and time are received from the master time clock. | | | | |
| **1** | Master time clock | Time | 3 Byte | C WTU |
| This object must be connected with the group address for the time sent by the master time clock. The schedule control is not started until date and time are received from the master time clock. | | | | |
| **2** | On / Off | 1st Schedule Object | 1 Bit | C T |
| Schedule control for the 1st schedule control channel is executed via the group address assigned to this 1st scheduler control object. | | | | |
| … | | | | |
| **21** | On / Off | 20th Schedule Object | 1 Bit | C T |
| Schedule control for the 20th schedule control channel is executed via the group address assigned to this 20th scheduler control object. | | | | |
| **22** | Event Tigger | 1st Event Program | 1 Bit | C WTU |
| The 1st Event Program is triggered via this object. | | | | |
| **23** | Event Object 1-1 | 1st Event Program | 1 Bit | C T |
| The 1st Event Action of the 1st Event Program is sent to the bus via the group address assigned to this object. | | | | |
| **24** | Event Object 1-2 | 1st Event Program | 1 Bit | C T |
| The 2nd Event Action of the 1st Event Program is sent to the bus via the group address assigned to this object. | | | | |
| … | | | | |
| **32** | Event Object 1-10 | 1st Event Program | 1 Bit | C T |
| The 10th Event Action of the 1st Event Program is sent to the bus via the group address assigned to this object. | | | | |
| **33** | Event Trigger | 2nd Event Program | 1 Bit | C WTU |
| The 2nd Event Program is triggered via this object. | | | | |
| **34** | Event Object 2-1 | 2nd Event Program | 1 Bit | C T |
| The 1st Event Action of the 2nd Event Program is sent to the bus via the group address assigned to this object. | | | | |
| **35** | Event Object 2-2 | 2nd Event Program | 1 Bit | C T |
| The 2nd Event Action of the 2nd Event Program is sent to the bus via the group address assigned to this object. | | | | |
| … | | | | |
| **43** | Event Object 2-10 | 2nd Event Program | 1 Bit | C T |
| The 10th Event Action of the 2nd Event Program is sent to the bus via the group address assigned to this object. | | | | |
| … | | | | |
| **131** | Event Object 10-10 | 10th Event Program | 1 Bit | C T |
| The 10th Event Action of the 10th Event Program is sent to the bus via the group address assigned to this object. | | | | |
| **132** | Input | A 1st Gate | 1 Bit | C WTU |
| This is Input A of the 1st Logic Gate. | | | | |
| **133** | Input | B 1st Gate | 1 Bit | C WTU |
| This is Input B of the 1st Logic Gate. | | | | |
| **134** | Input | C 1st Gate | 1 Bit | C WTU |
| This is Input C of the 1st Logic Gate. | | | | |
| **135** | Input | D 1st Gate | 1 Bit | C WTU |
| This is Input D of the 1st Logic Gate. | | | | |
| **136** | Input | E 1st Gate | 1 Bit | C WTU |
| This is Input E of the 1st Logic Gate. | | | | |
| **137** | Input | F 1st Gate | 1 Bit | C WTU |
| This is Input F of the 1st Logic Gate. | | | | |
| **138** | Output | 1st Gate | 1 Bit | C T |
| This is Output of the 1st Logic Gate. | | | | |
| **132** | Input | A 2nd Gate | 1 Bit | C WTU |
| This is Input A of the 2nd Logic Gate. | | | | |
| … | | | | |
| **200** | Input | F 10th Gate | 1 Bit | C WTU |
| This is Input F of the 10th Logic Gate. | | | | |
| **201** | Output | 10. Gate | 1 Bit | C T |
| This is Output of the 10th Logic Gate. | | | | |

# 01 07 Event-Schedule-Logic 801701

## Parameter Restart Behavior




| Parameter | Einstellungen |
|---|---|
| **Restart delay** | **No delay**<br>2 Seconds<br>5 Seconds<br>10 Seconds<br>30 Seconds<br>1 Minute<br>2 Minutes<br>5 Minutes<br>10 Minutes |
| The controller is functional after the restart delay time has elapsed. | |
| **Delay between two telegram transmissions** | **No delay**<br>0.2 Seconds<br>0.5 Seconds<br>1 Second<br>2 Seconds |
| This parameter provides for the distribution of bus communication load at restart. This is achieved by distributing request telegrams. | |

**Behavior on bus voltage restoration**
After an initialization time of approximately 2 seconds and a configurable startup delay on restart the N 350 is operational again.
On restart all event trigger inputs are set to 0. The controller fetches the current values from the bus. If an event trigger input is set to 1 during restart and the event trigger is set to „change from 0 to 1“ or „reception of 1“ then the event program is triggered and executed.
On restart all logic gate inputs are set to 0. The controller fetches the current input values from the bus. The logic gate sends the result of the logic function to the bus.
On restart the device gets the time from a master clock. Until the synchronized time is available all schedule functions are blocked.

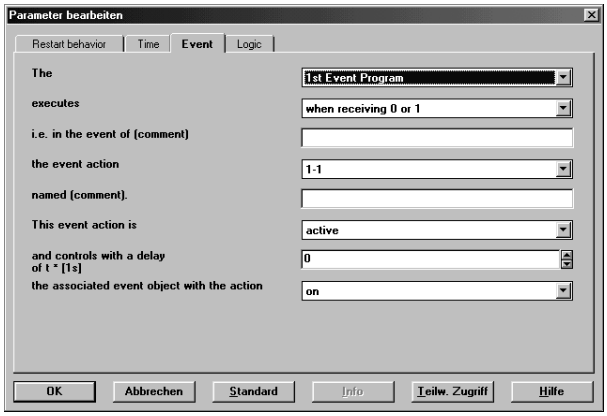
## Parameter Time


| Parameter | Einstellungen |
|---|---|
| **The** | **1st Schedule**<br>2nd Schedule<br>3rd Schedule<br>.<br>.<br>99th Schedule<br>100th Schedule |
| This parameter determines the schedule number. | |
| **named (comment)** | |
| Enter a comment for documentation purposes here. | |
| **is** | **inactive**<br>active |
| Activate / deactivate a single schedule here. | |
| **and is executed on** | <br>Monday<br>Tuesday<br>Monday and Tuesday |
| **and on** | <br>Wednesday<br>Thursday<br>Friday<br>Wednesday and Thursday<br>Wednesday and Friday<br>Thursday and Friday<br>Wednesday, Thursday and Friday |
| **and on** | <br>Saturday<br>Sunday<br>Saturday and Sunday |
| These parameters determine on which days a schedule is active. | |

## 01 07 Event-Schedule-Logic 801701

| | |
|---|---|
| **at (hour)** | 0 – 23<br>**Default 0** |
| **(minute)** | 0 – 59<br>**Default 0** |
| These parameters determine the time of execution of a schedule.. | |
| **and controls** | **1st Schedule Object**<br>2nd Schedule Object<br>3rd Schedule Object<br>4th Schedule Object<br>5th Schedule Object<br>6th Schedule Object<br>7th Schedule Object<br>8th Schedule Object<br>9th Schedule Object<br>10th Schedule Object<br>11th Schedule Object<br>12th Schedule Object<br>13th Schedule Object<br>14th Schedule Object<br>15th Schedule Object<br>16th Schedule Object<br>17th Schedule Object<br>18th Schedule Object<br>19th Schedule Object<br>20th Schedule Object |
| The controller offers 20 Schedule Objects or Schedule Channels. Via this parameter a schedule can be assigned to any of these Schedule Objects. | |
| **with the action** | **On**<br>Off |
| This parameter determines the schedule action as on or off. | |

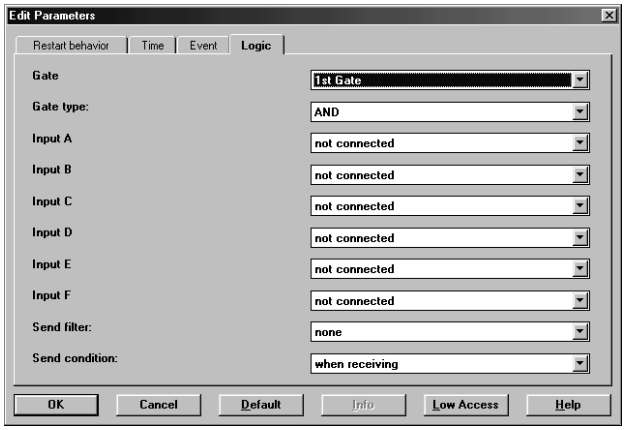
## Parameter Event




| Parameter | Einstellungen |
|---|---|
| **The** | **1st Event Program**<br>2nd Event Program<br>3rd Event Program<br>4th Event Program<br>5th Event Program<br>6th Event Program<br>7th Event Program<br>8th Event Program<br>9th Event Program<br>10th Event Program |
| The Event Program number is selected here. | |
| **executes** | **when receiving 0 or 1**<br>when receiving 1<br>when receiving 0<br>when changing from 0 to 1<br>when changing from 1 to 0 |
| This parameter determines what triggers the Event Program. | |
| **i.e. in the event of (comment)** | |
| Enter a comment for documentation purposes here. | |
| **the event action** | **1-1**<br>1-2<br>1-3<br>1-4<br>1-5<br>1-6<br>1-7<br>1-8<br>1-9<br>1-10 |
| Each event program may consist of up to 10 event actions. The event action for the current event program is selected here. | |
| **named (comment).** | |
| Enter a comment for documentation purposes here. | |
| **This event action is** | **inactive**<br>active |
| Activate or deactivate the event action with this parameter. | |
| **and controls with a delay of t * [1s]** | 0 – 6500<br>**default 0** |
| A delay of the event action is determined by this parameter. If a running event program is triggered again the current execution is halted and the event program started over a-gain. | |
| **the associated event ob-ject with the action** | **On**<br>Off |
| This parameter determines the event action as on or off. | |

## 01 07 Event-Schedule-Logic 801701

### Parameter Logic




| Parameter | Einstellungen |
|---|---|
| **Gate** | **1st Gate**<br>2nd Gate<br>3rd Gate<br>4th Gate<br>5th Gate<br>6th Gate<br>7th Gate<br>8th Gate<br>9th Gate<br>10th Gate |
| One of the ten logic gates is selected here. | |
| **Gate type:** | **AND**<br>OR<br>NAND<br>NOR |
| The gate type is selected here. | |
| **Input A** | **Not connected**<br>direct<br>inverted |
| **Input B** | **Not connected**<br>direct<br>inverted |
| **Input C** | **Not connected**<br>direct<br>inverted |
| **Input D** | **Not connected**<br>direct<br>inverted |
| **Input E** | **Not connected**<br>direct<br>inverted |
| **Input F** | **Not connected**<br>direct<br>inverted |

These parameters determine if and how a logic gate input is used for the logic.

‚Not connected': This input is not used by the logic gate.

‚direct': The value of this input is used for the logic gate. A ‚1' telegram is interpreted as a logic ‚1', a ‚0' telegram as a logic ‚0' input to the gate logic.

‚inverted':The value of this input is used for the logic gate. The input value is inverted for the logic gate. A ‚1' telegram is interpreted as a logic ‚0', a ‚0' telegram as a logic ‚1' input to the gate logic.

| | |
|---|---|
| **Send filter** | **None**<br>only when 1at the output<br>only when 0at the output |

This parameter determines which output values are sent onto the bus.

„None" means that both output values 0 and 1 are sent.

only when 0at the output

„Only when 1at the output" means that only the output value ‚1' of the logic gate output is sent.

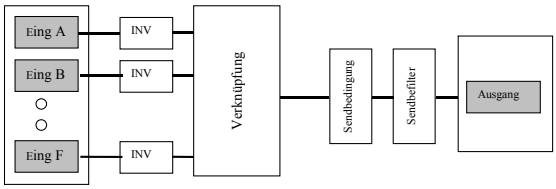
„Only when 0at the output" means that only the output value ‚0' of the logic gate output is sent.

| | |
|---|---|
| **Send condition** | **Only on change of output**<br>when receiving |

This parameter determines under which condition the result of the logic gate is sent.

‚Only on change of output': The output value is only sent when the value of the output changes either from ‚1' to ‚0' or from ‚0' to ‚1' AND the send filter above allows the transmission.

‚when receiving': The output value is sent when any input receives a new telegram AND the send filter above aloows the transmission.



**01 07 Event-Schedule-Logic 801701**

## Room for Notes





Siemens AG
Automation and Drives Group
Electrical Installation Technology
Postfach 10 09 53, D-93009 Regensburg