ALARM ABSOLON

genesis

Bezpečnostní systém
Kontrola vstupu
Domácí automatizace
Automatizace budov

Programovací manuál pro skripty

Verze 2.0 (Build 80)

genesis

Manual summary

Table of contents

Page left BLANK

Revision history

Revision	Date	By	Description
1.00	December 2001	J. Hola	Base release of Script manual for Build 80
1.01	April 2002	J. Hola	Update 'THIS' Input, Area, Door, Clock Added more scripts
1.02	May 2002	J. Hola	Add Sample fo Alarm and Alert Scripts Dialler Commands

About this manual

This manual contains information that is to be read from beginning to end as well as reference only material.

Underlined words have associated sections in the manual. In the electronic PDF version, they can be clicked with the left mouse button to go to the associated topic.

Knowledge Base

You should have sound knowledge of Alarm and Access control functionality. We recommend your attendance to our training to ensure understanding and improvement in your knowledge of this product.

We recommend, when system is installed, that is fully tested for all functions, to ensure correct operation.

We reserve the rights, to change or modify this product without any further notice.

Scripting

The scripting section is broken into a tutorial and a reference section.

For standard installations, the technician should be able to use standard scripts and so will not have to understand the operation of scripts. Most technicians, however, will require an increasing knowledge of the scripting service. It is recommended that technicians read the tutorial before attempting any scripts.

We have taken all care to supply you with working examples of scripts. Each example is as demonstration ONLY and we cannot take responsibility of its correct operation. Please ensure full system test is done, to ensure correct operation for your installation site.

Quick set up guide

The Script icon on the tool bar allows you to enter Script programming menu.

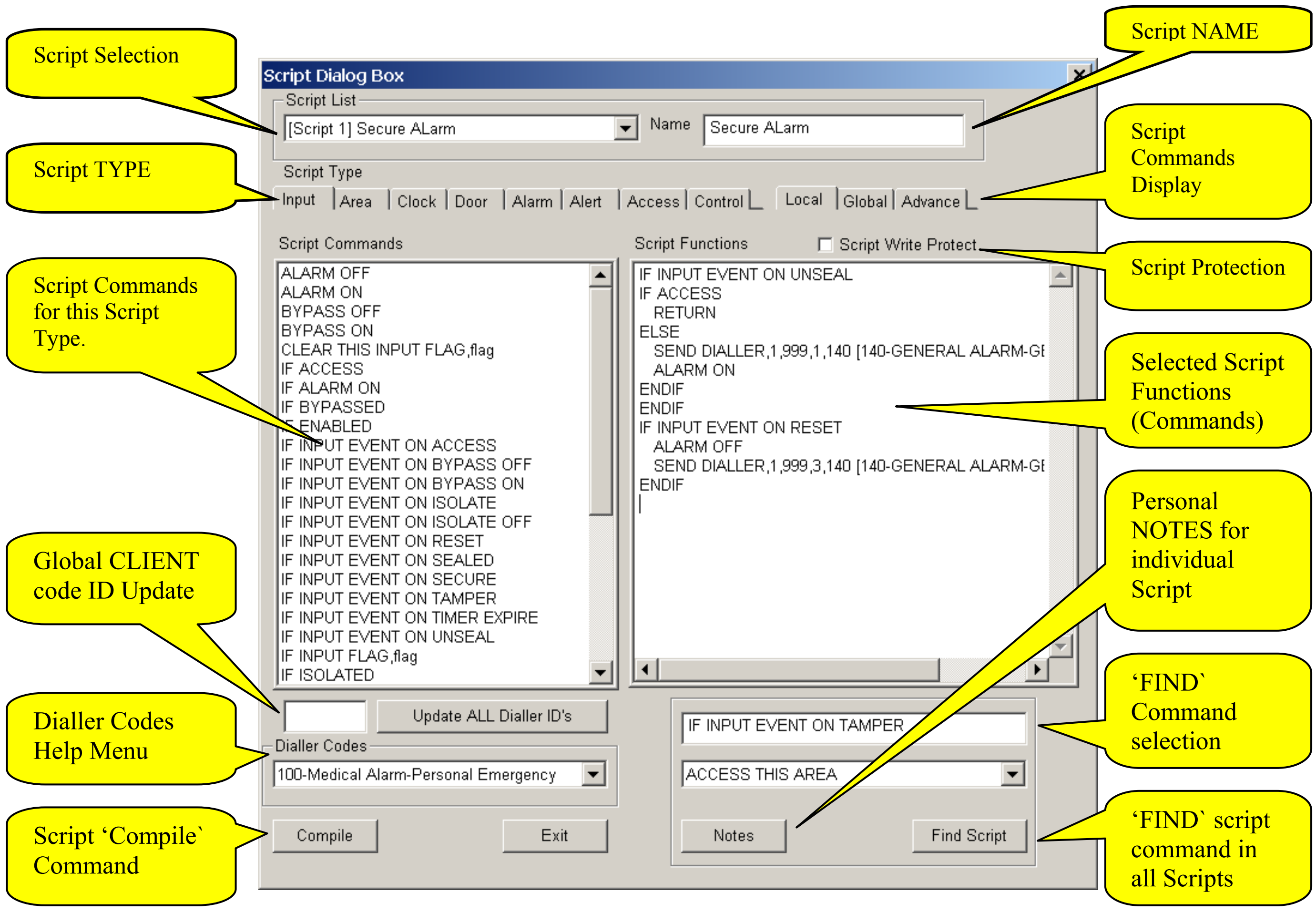
Create additional scripts to customise the system for the site.

Scripts

Scripts make it possible for the system to do the most demanding of applications as well as quickly achieving common requirements.

The script language is a very simple programming language. Anyone familiar with languages such as C, Basic, Pascal, Delphi or Java will quickly understand the scripting language. No previous programming experience is required to start creating your own scripts.

The Script Programming screen



Script List: [Script 1] Secure ALarm

Name: Secure ALarm

Script Commands:

```
ALARM OFF
ALARM ON
BYPASS OFF
BYPASS ON
CLEAR THIS INPUT FLAG,flag
IF ACCESS
IF ALARM ON
IF BYPASSED
IF ENABLED
IF INPUT EVENT ON ACCESS
IF INPUT EVENT ON BYPASS OFF
IF INPUT EVENT ON BYPASS ON
IF INPUT EVENT ON ISOLATE
IF INPUT EVENT ON ISOLATE OFF
IF INPUT EVENT ON RESET
IF INPUT EVENT ON SEALED
IF INPUT EVENT ON SECURE
IF INPUT EVENT ON TAMPER
IF INPUT EVENT ON TIMER EXPIRE
IF INPUT EVENT ON UNSEAL
IF INPUT FLAG,flag
IF ISOLATED
```

Script Functions:

```
IF INPUT EVENT ON UNSEAL
IF ACCESS
   RETURN
ELSE
   SEND DIALLER,1,999,1,140 [140-GENERAL ALARM-GE
   ALARM ON
ENDIF
ENDIF
IF INPUT EVENT ON RESET
   ALARM OFF
   SEND DIALLER,1,999,3,140 [140-GENERAL ALARM-GE
ENDIF
```

Dialler Codes: 100-Medical Alarm-Personal Emergency

IF INPUT EVENT ON TAMPER

ACCESS THIS AREA

Script Selection

The script list displays all the scripts within the system (Between 100 – 300, depending of system configuration). The description is displayed next to the list.

Script Name

User selectable Script name.

Script TYPE

The script type specifies what type of script it is. The script type determines what the script can be linked to. It also determines what functions are available and what they do. Most functions, however, are available for all types and have the same purpose.

Script Commands Selection

Local Selection displays only LOCAL script, while Global, displays all available Scripts within the Script Function command window.

Script Commands

The Script Commands Selection displays all the available commands. Double Click with the left mouse button will copy the selected command into the Script Functions window. If the selected command requires additional information then a dialog box will appear requesting the information. For example, the "SECURE AREA,area" command requires the area and the operator will be able to select the area from a list of areas.

Script Function

The Script Functions (right display) is the actual script. This will be discussed in more detail.

Global Client Code ID update

Since the client ID is embedded into the scripts, the "Update all Dialler ID's" command makes it possible to do a bulk change to the system. It will find every Dialler command and set the client ID to the entered value.

Compile

The compile function performs syntax check and conversion of script functions into code recognized by the system.

Find Script

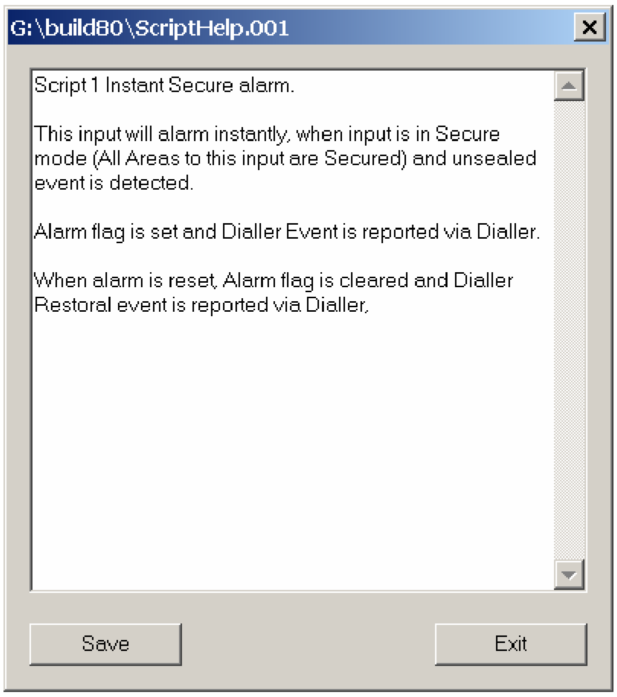
The Find Script box at the bottom of the screen makes it possible to find all the scripts that accesses an areas, output, door or input. This is an invaluable tool for trying to determine what is controlling a particular point.

Note that the find script box cannot determine what triggers it. The individual inputs or devices must be viewed to determine what is using the script.

Notes

Personal Help screen, allows description to be attached to each script. It is a good practice to use this help menu, not only for your self in the future, but also for your colleagues, to quickly pick up the understanding and functionality of the Script.
Each Script Help file is stored in the current directory using the name 'ScriptHelp.xxx`, where xxx is the Script number.

Example:



Script 1 Instant Secure alarm.

This input will alarm instantly, when input is in Secure mode (All Areas to this input are Secured) and unsealed event is detected.

Alarm flag is set and Dialler Event is reported via Dialler.

When alarm is reset, Alarm flag is cleared and Dialler Restoral event is reported via Dialler,

Tutorial

The following examples introduce the essential ideas of programming with scripts. They should be read in order because each example assumes knowledge presented in previous ones.

This tutorial will take you through the process of creating new scripts and how to modify them. It is assumed that you know how to setup the rest of the system (e.g. devices, clocks, areas and users).

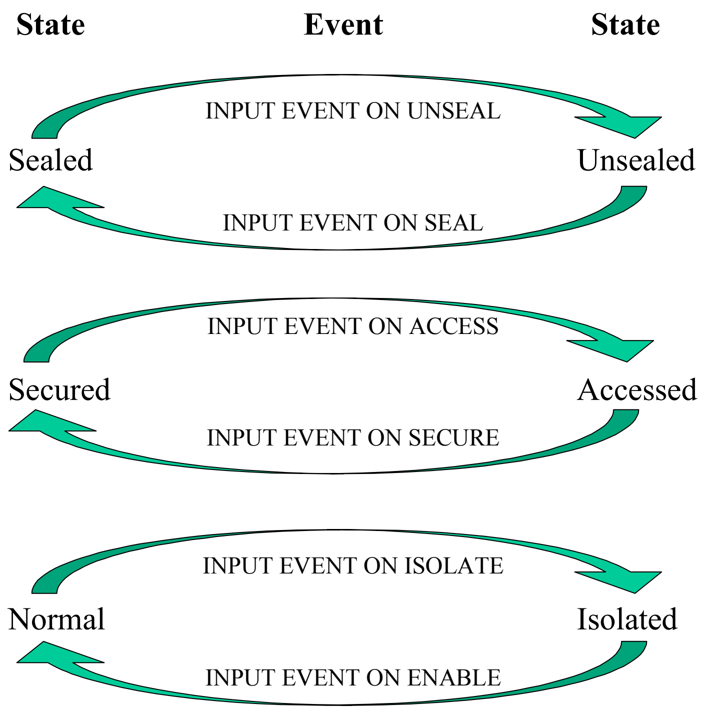
Event driven code

Genesis uses an event driven format for its scripts. That is, a script is run when some condition has changed within the system.

This simplifies the program because it is on changes that other actions need to occur. If the code were to be run periodically, the technician would need to manually check all the inputs and check for changes.

For example, an input has the following states (among others):
- Sealed and unsealed
- Secured and accessed
- Isolated or de-isolated

This results in the following relationships:



When the input changes state (for example from secured to access) then the associated event is triggered (INPUT EVENT ON ACCESS in this example).

This method means that it is possible to modify the system to do exactly what is required. The following examples indicate the flexibility.

The INPUT EVENT ON UNSEAL could be used to turn on the air conditioning. In this case the input does not go into alarm.

Another input could use the INPUT EVENT ON UNSEAL to then test to see if it is secured. If it is, then it turns its alarm status on.

A third input could use the INPUT EVENT ON UNSEAL to actually turn the alarm status off. It is a key switch used after a duress alarm.

In these three examples, each script uses the same event but decides what to do with it. The first example was independent of the alarm system, the second triggered the alarms and the third cleared the alarms.

The power of Genesis comes from its ability to allow the installer to associate a script with nearly every possible event. The system is under their complete control.

Within a script it is possible to check the status of other inputs, areas, clocks etc. This makes it possible to write code that is responding to changes, but will take into account the state of other parts of the system. The key to remember is that an event is required to start a script, but the system can then respond to the trigger, the status of something, or both.

Each SCRIPT, when EVENT is activated, may include *HIDDEN* Input, Area, Timer, User, Device, Group or Door number, allowing common SCRIPT to be allocated to multiple INPUTS, TIMERS, AREAS, Devices, GROUPS or DOORS, where identical command function is required.

SCRIPT TYPES

To provide maximum flexibility of the system, majority of the ACTIONS are controlled within the SCRIPTS. It depends on the SCRIPT functions, how the systems will response to individual event. It is important that the Technician fully understands the effect of each script command
It is recommended; total system test is performed to confirm proper operation of individual commands within the scripts as well proper operation of the full system.
Each Script includes number of required EVENTS, CONTROL and TEST functions, but remembers, you have a total control of the functionality, which adds the benefit of Scripts into your hand.

Flow control

Flow control refers to how the controller will pass through a script. In particular, how it will decide which lines to run and which lines to skip.

Up until now, the IF and ENDIF statements have been regularly used. There is a third option, which is the ELSE statement.

Example: AREA TYPE SCRIPT

```
IF AREA EVENT ON ACCESS
OUTPUT CONTROL,2,2,0010 [AIRCON]
IF CLOCK VOID,2
OUTPUT CONTROL,1,2,0010 [LIGHTS]
ENDIF
ENDIF
IF AREA EVENT ON SECURE
IF CLOCK VOID,1
OUTPUT CONTROL,1,1,0
ENDIF
IF CLOCK VOID,3
OUTPUT CONTROL,2,1,0
ENDIF
ENDIF
```

Please note:

1) Each command within the script starts with the **'EVENT`** statement (in the example above, indicated in red color). Within each EVENT we can include additional statements, such as controls or test to a system condition.
2) Each 'EVENT` is treated as a separate command.
3) Don't use the 'ELSE' command on 'EVENT' statement, unless you are expert and you fully understand Scripts.
4) You always use the 'TEST` and 'CONTROL' commands enclosed within the 'EVENT` statement.
5) Each 'IF'` command **MUST** end with 'ENDIF' command.

If the area is accessed then turn the air conditioning OUTPUT on. Then, if CLOCK 2 is void (outside work hours), turn the lights OUTPUT on.
If the area is armed then if CLOCK 1 is void (when the lights are on automatically) turn the lights OUTPUT off. Also, if CLOCK 3 is void (when the air conditioning is on automatically) then turn the air conditioning OUTPUT off.

Note that the last two CLOCK checks are both done when the area is armed. They are independent of each other, but both rely on the AREA EVENT ON SECURE to be true.

The above grouping technique is recommended when creating scripts or trying to decipher some one else's script.

The 'IF` statement will allow this script to continue on the lines after, or skip it to the matching 'ENDIF'. If the result is true then the following lines will run. If the result is false, the following lines are skipped. The ENDIF determines how many lines are run or skipped.

The ELSE statement provides an alternative. For example, in the morning a person may decide, "If it is sunny then wear a hat else take an umbrella". The 'IF` statement results in a choice of two possible actions.

If, however, there is more than two possible conditions, then multiple IF statements are required. For example, the area script has four possible triggers. To determine which one occurred, the script could be either

Example ‘a`:

```
IF AREA EVENT ON ACCESS
<<a>>
ENDIF
IF AREA EVENT ON SECURE
<<b>>
ENDIF
IF AREA EVENT ON ALARM
<<c>>
ENDIF
IF AREA EVENT ON RESET
<<d>>
ENDIF
```

Or
Example ‘b’:

```
IF AREA EVENT ON ACCESS
<<a>>
ELSE
IF AREA EVENT ON SECURE
<<b>>
ELSE
IF AREA EVENT ON ALARM
<<c>>
ELSE
IF AREA EVENT ON RESET
<<d>>
ENDIF
ENDIF
ENDIF
ENDIF
```

Both scripts achieve exactly the same result. In fact they are both correct and both have their supporters. The most important point is to ensure that the code makes sense to other people. Sometimes it is advisable to write scripts using more lines, less complicated, as it is easier to follow. I would recommend the example ‘a’ over the example ‘b’.

Logic

Associated with flow control is the need to make decisions based upon multiple factors.

In Genesis, the aim was to keep the scripting simple and easy to understand. It was decided to limit the IF statements to a single test. It is possible, however, to achieve complex results using this structure.

The **AND** statement is easily achieved. An AND statement means that the result is true only if both tests are true.

“If it is raining AND I am going outside, then get an umbrella” is an example. An umbrella is not required if it is not raining and I am going outside, if it is raining but I am not going outside, or if it is not raining and I am not going outside. It must be raining, and I must be going outside before I need an umbrella.

Nesting IF statements achieves an AND statement. For example,

```
IF INPUT EVENT ON UNSEAL
IF AREA IN ACCESS,1
IF AREA IN ACCESS,2
<<a>>
ENDIF
ENDIF
ENDIF
```

The code <<a>> is only run if both areas 1 and 2 are in access (disarmed). If either is secure then <<a>> will not be reached.

The **OR** statement is also easily achieved but requires some code duplication. The OR statement means that the result is true if either test is true.

“If it is raining or cold then get a coat” is an example. The only time a coat is not required is when it is not raining and it is not cold. As soon as it is raining or cold or both, a coat is required.

To achieve an OR statement, duplicate the code in separate IF statements. For example,

```
IF INPUT EVENT ON UNSEAL
IF AREA IN ACCESS,1
<<a>>
ENDIF
IF AREA IN ACCESS,2
<<a>>
ENDIF
ENDIF
```

The code <<a>> has been duplicated within both IF statements. Within the script editor this can be easily achieved by using the copy (Ctrl-C) and paste (Ctrl-V) functions.
It is possible to achieve more complex conditions such as an exclusive OR (XOR) where one or the other, but not both, must be true. For further assistance contact your local distributor.

Security Functions

The tutorials so far have focused on simple logic and building automation applications. This section focuses on the Genesis as an alarm system.

Genesis does not provide automatic alarm functionality immediately. The standard scripts are sufficient to achieve most security requirements, but they must still be assigned to inputs. For standard installations the assigning of scripts is very quick and simple.

The power of Genesis, however, is that the technician can make Genesis act like an alarm system, access control system or building automation system, or any combination.

It is possible to expand the standard alarm functionality or to remove it. For example, an input could be used to release a door. In this case a change in state is not an alarm, while a tamper alarm should still be reported. Scripts make it possible.

Script Inputs Type

Inputs have a number of states. The most important ones are:

- Sealed (closed) or unsealed (open)
- Access or secure
- Alarm or normal
- Tamper or Seal or Unseal
- Isolate on or Isolate off
- Bypass on or Bypass off

An input changes from sealed to unseal. The ‘IF INPUT EVENT ON UNSEAL` event is triggered. Now what? Is this an alarm, or is it a switch to turn something on?

The standard alarm input script processes it like this:

```
IF INPUT EVENT ON UNSEAL
      IF SECURED
           IF ALARM ON
             RETURN
           ENDIF
      OUTPUT CONTROL,1,2,0600
      SEND DIALLER,1,1234,1,140
      ALARM ON
      ENDIF
ENDIF
```

If it was the UNSEAL event, than system will continue commands below. In this case it was. If the input is not secured then nothing happens. That is, all the areas associated with the input must be secure before additional responses will occur. If the point is already in alarm then return and do nothing. If it is not in alarm then turn the siren on and send alarm to the Dialler. Turn the alarm latch on for the point.

This script demonstrates how every response action the script determines. Removing the output command can generate silent alarms. Adding clock checks can provide restricted hours for the siren. Since multiple scripts are possible, some inputs can respond one way, and others another way.

The ‘SEND DIALLER’ Dialler command is very common. Its format can be found in the reference section. At this stage, just assume that it is making contact with the monitoring company with detailed information on the situation.

The tamper code is very similar:

```
IF INPUT EVENT ON TAMPER
IF TAMPER
    RETURN
ENDIF
    TAMPER ON
    SEND DIALLER,1,1234,1,137
    OUTPUT CONTROL,1,2,0100
ENDIF
```

If the input tamper event has just occurred then check if the point is already in TAMPER. If it is then do nothing. If not, however, then set the tamper flag send alarm to Dialler and turn on the siren. The system responds to a tamper in almost the same way it responds to a detector activating. Once again, however, every aspect of the response is under the technician’s control.

The input receives a reset. The alarm has been responded to. Now what?

```
IF INPUT EVENT ON RESET
IF ALARM ON
    SEND DIALLER,1,1234,3,140
    ALARM OFF
ENDIF
IF TAMPER
    SEND DIALLER,1,1234,3,137
    TAMPER OFF
ENDIF
ENDIF
```

If the input is in alarm or in tamper, then the monitoring station is notified of the reset. The alarm and tamper flags are turned off.

Areas

Areas are simpler then inputs and have less states:
- Access or secure
- Alarm or secure
- Unsealed or seal
- Tamper or secure
- Isolate ON or Isolate OFF
- Bypass ON or Bypass OFF

The access/secure state is under the control of the user and scripts. It can be controlled from a RAS, reader, PC or by a script triggered by anything.

The alarm/normal state, however, is automatically determined by the system. An area is in alarm if one of its inputs is in alarm (ALARM ON flag set), or in tamper (TAMPER ON flag set). Note that there is a difference between an input being unsealed and in alarm. When an input becomes unsealed it is up to the associated input script to turn the input's alarm flag on.
The unsealed status is activated, when first input in the area is unsealed, and is cleared, when last input returns to seal status. It applies for the Isolated and Bypassed events.

Event flow

Ultimately there is a strong relationship between areas and inputs. Actions with one will automatically result in events for the other.

When an area is secured (for example by the user from RAS), the flow of events is:
- If the area is unsealed, cancel and display the unsealed inputs, otherwise
- For each input in the area, trigger the 'IF INPUT EVENT ON SECURE` if all the areas associated with the input are now secure.
- Trigger the 'IF AREA EVENT ON SECURE` for the area.

The second step may need clarifying. For example, area 1 contains two inputs. Input 1 is only in area 1 and it is secure. The 'IF INPUT EVENT ON SECURE` is triggered. Input 2, however, is also in areas 2, 3 and 4. The 'IF INPUT EVENT ON SECURE` will only be trigger for this input if areas 2, 3 and 4 are also secure. If any of them are not secure, the event will not be triggered.

Input to an AREA Relation ship:

If an input becomes unsealed then:
- 'IF INPUT EVENT ON UNSEAL` is triggered.
- If 'ALARM ON' function is activated, and this is first alarm in the area, then.

- 'IF AREA EVENT ON ALARM` will be triggered.

Note: The 'IF AREA EVENT ON ALARM'` event is triggered by first input in the area issuing the 'ALARM ON' command.
While 'IF AREA EVENT ON TAMPER' event is triggered by first input in the area issuing the 'TAMPER ON' command.

If the area is then accessed (disarmed) then the input events are triggered:
- If an input was in alarm the 'IF INPUT EVENT ON RESET` is generated,
- The 'IF INPUT EVENT ON ACCESS` is triggered
- If the area was in alarm the 'IF AREA EVENT ON ` is triggered,
- The 'IF AREA EVENT ON ACCESS` is triggered.

This open structure ensures that the system is flexible. That means that it can be modified to suit most applications.

The standard scripts ensure that normal alarm functionality occurs. That is, everything discussed here happens automatically, without the technician having to ensure the associations discussed.

The states and events are shown in the following diagram. For each transition the top line indicates the transition from left to right, while the bottom line is the transition from right to left.

Absolute and self addressing

Many of the commands provide for either absolute or self-addressing.

With absolute addressing, the address and node, area number etc are specified. There is only one possible match for the command and it will do exactly the same action no matter what point triggers the script.

For example, ACCESS AREA, 3 will always access area number 3. This command can be used in any type of script.

Self-addressing provides the ability to run a command on the point that triggered the event. Note that the command must be of the same type as the event.

For example, ACCESS THIS AREA will access the area that triggered the (area) script. If three areas were associated with the script then any one of them could be accessed. The result depends upon which area triggered the script.

It is possible to mix absolute and self addressing within the same script, but self addressing must only occur for commands of the same type as the script type.

Self addressing is a very powerful method to re-use scripts. For example, the standard input script contains the following code:

```
IF INPUT EVENT ON UNSEAL
IF SECURED
IF ALARM ON
```

In this example, the first and third line both refer to the input that triggered the event. The one script can then be applied to multiple inputs.

INPUT EVENT SCRIPT TYPE

Input scripts are always allocated to an INPUTS ONLY.

The changes in input status generates the following EVENTS:

IF INPUT EVENT ON ACCESS

Input status has changed to ACCESS MODE. This EVENT is automatically generated, when ***first Area***, which includes this input, has changed into access mode,

IF INPUT EVENT ON SECURE

Input status has changed to SECURE MODE. This EVENT is automatically generated, when ***last Area***, which includes this input, changed into secure mode,

IF INPUT EVENT ON BYPASS OFF

The Bypass Off command has been requested. Please note: Input will not remove the BYPASS status, until 'BYPASS OFF' script command is issued.

IF INPUT EVENT ON BYPASS ON

The Input Bypass command been requested. Please note: Input will NOT BE set to BYPASS status, until 'BYPASS ON' script command is issued. When input is in Bypass status, the SEALED, UNSEALED and TAMPER EVENTS are automatically suppressed.

IF INPUT EVENT ON ISOLATE ON

Isolate Input command has been requested for this input (e.g. User has requested Input to be isolated). Please note: System will NOT set the input ISOLATE flag, until 'ISOLATED ON' script command is issued. It is up to the SCRIPT commands to test the ISOLATION FLAG status and provide appropriate action. **INPUT CAN NOT BE ISOLATED**, while INPUT is in **SECURE MODE**.

IF INPUT EVENT ON ISOLATE OFF

The Isolate Off command has been requested. Please note: Input ISOLATE STATUS will NOT BE REMOVED, until the 'ISOLATED OFF' script command is issued

IF INPUT EVENT ON RESET

The Input Reset command is issued ONLY, when the 'ALARM ON` or 'TAMPER ON' flag is SET and the AREA, which includes this input, has changed to ACCESS mode. Normally the 'ALARM OFF' or 'TAMPER OFF' command is used within this EVENT.

IF INPUT EVENT ON SEAL

Input has changed into SEALED condition. This event is automatically generated by the system.

IF INPUT EVENT ON UNSEAL

Unsealed status has been detected. This event is automatically generated by the system.

IF INPUT EVENT ON TAMPER

Tamper status detected this events are automatically generated by the system. Normally the 'TAMPER ON` command is used to LATCH ON the TAMPER condition for this event

IF INPUT EVENT ON TIMER EXPIRE

ANY associated Input Timer(s) has expired. ***IF INPUT TIMER EXPIRED,timer*** can be used to test to find out, which timer has generated this event.

AREA EVENT SCRIPT TYPE

Area scripts are always allocated to an AREAS ONLY.

The changes in the Area status are generated automatically by the system and the events are as following:

IF AREA EVENT ON ACCESS

This event is generated, when an AREA is changed into ACCESS MODE. Usually, this event is used to trigger the ***'SEND DIALLER'*** command to report AREA opening message.

IF AREA EVENT ON SECURE

This event is generated, when an AREA is changed into SECURE MODE. This event is normally used to trigger the ***'SEND DIALLER'*** command to report AREA closing message.

IF AREA EVENT ON ALARM

This event is generated, when the 'ALARM ON' command has been issued on the input and this was the FIRST ALARM in this AREA Usually, this event is used to generate AREA ALARMS, such as Sirens and Strobe Lights.

IF AREA EVENT ON TAMPER

This event is generated, when the 'TAMPER ON' command has been issued on the input and this was the FIRST TAMPER ALARM in this AREA Usually, this event is used to generate AREA ALARMS, such as Sirens and Strobe Lights, on any TAMPER alarms.

IF AREA EVENT ON RESET

This event is generated, when any inputs in this Area includes the 'ALARM ON' flag set and the AREA is changed to ACCESS mode or AREA RESET command has been issued. Usually, this event is used to STOP previously generated AREA ALARMS.

The following AREA EVENTS can be used to provide USER with common status indication of the AREASTATUS on the MIMIC PANEL, RAS or on special indicator.

IF AREA EVENT ON BYPASS OFF

This event is generated, when the 'BYPASS OFF' command has been issued on previously 'BYPASSED' input and this was the last BYPASSED input in this AREA

IF AREA EVENT ON BYPASS ON

This event is generated, when the 'BYPASS ON' command has been issued on input and this was the FIRST BYPASSED input in this AREA

IF AREA EVENT ON ISOLATE OFF

This event is generated, when the 'ISOLATE OFF' command has been issued on previously 'ISOLATED' input and this was the LAST ISOLATED input in this AREA

IF AREA EVENT ON ISOLATE ON

This event is generated, when the 'ISOLATE ON' command has been issued on an input and this was the FIRST ISOLATED input in this AREA

IF AREA EVENT ON SEAL

This event is generated, when last input in this AREA has changed its status to sealed condition

IF AREA EVENT ON UNSEAL

This event is generated, when first input in this AREA has changed its status to unsealed condition.

CLOCK EVENT SCRIPT TYPE

Clock scripts are always allocated to CLOCKS ONLY.

The changes in the Clock status are generated automatically by the system and the events are as following:

IF CLOCK EVENT TO VALID

This Event is generated, when Clock has changed the TIME as specified within the START window of the Clock. This Event can be used to automatically lock or unlock a Door(s), turn Lights on or off, and set AREA status to Access or Secure Mode, start Sprinklers and many more time related event controls.

IF CLOCK EVENT TO VOID

This Event is generated, when Clock has changed the TIME as specified within the STOP window of the Clock This Event can be used to automatically lock or unlock a Door(s), turn Lights on or off, and set AREA status to Access or Secure Mode, start Sprinklers and many more time related event controls.

DOOR EVENT SCRIPT TYPE

Door scripts are always allocated to DOORS ONLY.
The changes in the Door status are generated automatically by the system and the events are as following:

IF DOOR EVENT ON CLOSED

This Event is generated, when Door has returned from DOTL, FORCED or TAMPER ALARM back to normal status. This event normally would be used to reset any previously generated alarms from DOTL, FORCED or DOOR TAMPER events.

IF DOOR EVENT ON DOTL

This Event is generated, when Door has been left open, after valid entry, for longer period than is programmed in the DOOR Record. In this event, normally you generate an external alarm, when DOTL alarm has been generated.

IF DOOR EVENT ON FORCED

This Event is generated, when Door has been FORCED to open, without valid access authorization. In this event, normally you generate external alarm, when FORCED alarm has been generated, such as unauthorized entry.

IF DOOR EVENT ON TAMPER

This Event is generated, when Door input has been tampered with.(see EOL programming for TAMPER detection on Door Input). In this event, normally you generate external alarm, when TAMPER alarm has been generated.

IF DOOR EVENT ON TIMER EXPIRE

This Event is generated, when Door Script Timer has expired. In this event can be used to return Door to Normal, reset alarms, switch off lights, or any special controls. Within the '**IF DOOR EVENT ON TIMER EXPIRE**` event, ***'IF DOOR TIMER EXPIRED,timer'*** test command can be used to test which timer has generated this '**IF DOOR EVENT ON TIMER EXPIRE**` event.

ALARM EVENT SCRIPT TYPE

Alarm Script is always allocated to UNIT, such as MU, EU, RAS, TDC and OC Device.

The Alarm Script Events are generated automatically by the system and the events are as following:

IF ALARM EVENT ON,1 [[001] UNIT ONLINE/OFFLINE]

IF ALARM EVENT OFF,1 [[001] UNIT ONLINE/OFFLINE]

These events are generated, when Device unit lost or restored communication with the MASTER. Normally you will use this event area to report Device TROUBLE to monitoring station or generate indication of the problem to the user.

IF ALARM EVENT OFF,2 [[002] DURESS ALARM]

IF ALARM EVENT ON,2 [[002] DURESS ALARM]

These events are generated, when USER has activated or reset the DURESS code at the RAS device. Normally you will use this event to report DURESS to monitoring station or generate indication or appropriate action in event of DURESS condition. Please note: Each DURESS condition MUST BE RESET at the RAS, where the DURESS code has been ACTIVATED.

IF ALARM EVENT OFF,3 [[003] TAMPER ALARM]

IF ALARM EVENT ON,3 [[003] TAMPER ALARM]

These events are generated, when DEVICE TAMPER (Cabinet) has generated or restored. Normally, in this area you would provide Dialler reporting of cabinet Tamper alarm or restore, as well any special indication or action needed in this event.

IF ALARM EVENT OFF,4 [[004] MAINS ALARM]

IF ALARM EVENT ON,4 [[004] MAINS ALARM]

These events are generated, when MAINS supply power is not present to the device for longer than 60 seconds. These events are applicable for the MU and EU device only. Normally, in this area you would provide Dialler reporting of MAINS failure and restore, as well any indication or action needed in this event.

IF ALARM EVENT OFF,5 [[005] BATTERY ALARM]

IF ALARM EVENT ON,5 [[005] BATTERY ALARM]

These events are generated, when BATTERY voltage falls below selected voltage level, and restore when voltage returns above selected voltage. Normally, in this area you would provide Dialler reporting of BATTERY low and restore, as well any indication or action needed in this event.

IF ALARM EVENT OFF,6 [[006] POWER FUSE ALARM]

IF ALARM EVENT ON,6 [[006] POWER FUSE ALARM]

These events are generated, when POWER FUSES (**F1 or F2**) has changed its status. Normally, in this area you would provide Dialler reporting of FUSE Fault/Restore as well any indication or action needed in this event. This event is applicable for MU and UE Devices ONLY

IF ALARM EVENT OFF,7 [[007] SIREN FUSE ALARM]

IF ALARM EVENT ON,7 [[007] SIREN FUSE ALARM]

These events are generated, when SIREN FUSES (**F3 or F5**) has changed its status. Normally, in this area you would provide Dialler reporting of FUSE Fault/Restore as well any indication or action needed in this event.

IF ALARM EVENT OFF,8 [[008] STROBE FUSE ALARM]

IF ALARM EVENT ON,8 [[008] STROBE FUSE ALARM]

These events are generated, when STROBE FUSES (**F4 or F6**) has changed its status. Normally, in this area you would provide Dialler reporting of FUSE Fault/Restore as well any indication or action needed in this event.

IF ALARM EVENT OFF,9 [[009] DIALLER ALARM

IF ALARM EVENT ON,9 [[009] DIALLER ALARM]

These events are generated, when DIALLER has not established communication with the monitoring station and has reached Maximum retries as programmed in the Dialler programming menu. Normally, in this area you would provide indication of Dialler FAULT or selection of backup line. ***Do not use here any commands to send DIALLER MESSAGES!***

IF ALARM EVENT OFF,10 [[010] SIU ALARM]

This provision is for future Securitel Error event and is not presently in use.

ACCESS SCRIPT TYPE

Access Script is always allocated to USER GROUP.

Access script type does not have any EVENTS and Script is activated, every time any valid user code or card has been presented in the system.

CONTROL SCRIPT TYPE

Control Script does not have any EVENTS and is used by all other scripts to utilize any common actions.
This Script can include commands such as ‘OUTPUT CONTROL` or any GLOBAL Commands, common to multiple Scripts.
You can pass value (1-255) from calling script. This value can be tested on, allowing multiple functions in the control script.

Script reference

The following reference section explains each of the script commands in detail. Each command will have the following information:

Command name: As shown in the Script Commands Section

Purpose:	A brief explanation of what the command does.
Type:	A function returns information and typically start with the if command. A procedure does an action and does not return a response.
Available for:	Specifies what script type the command is available for.
Description:	A detailed description of what the script does and how it may be useful.
Parameters:	What additional information the command requires to operate.
See also:	Other related or similar commands.
Example:	An example of how the command is used.

LOCAL SCRIPT COMMANDS

Local Scripts are Script Commands, which do not require separate entry of an input or an Area number. System automatically substitutes this number, depending which input or area this script is called from.

System Single commands are recognized as a Local script, as they do not require additional data input.

Local Script Commands for INPUT SCRIPT types are:

INPUT EVENT SCRIPTS:

IF INPUT EVENT ON ACCESS
IF INPUT EVENT ON BYPASS OFF
IF INPUT EVENT ON BYPASS ON
IF INPUT EVENT ON ISOLATE ON
IF INPUT EVENT ON ISOLATE OFF
IF INPUT EVENT ON RESET
IF INPUT EVENT ON SEAL
IF INPUT EVENT ON SECURE
IF INPUT EVENT ON TAMPER
IF INPUT EVENT ON TIMER EXPIRE
IF INPUT EVENT ON UNSEAL

INPUT FLAG CONTROLS:
ALARM OFF
ALARM ON
BYPASS OFF
BYPASS ON
ISOLATE OFF
ISOLATE ON
TAMPER OFF
TAMPER ON

THIS INPUT FLAG CONTROLS: (8 flags (1-8) per each Input)
CLEAR THIS INPUT FLAG, flag
SET THIS INPUT FLAG, flag
IF THIS INPUT FLAG, flag

INPUT TIMER CONTROL: (up to 7 timers per each Input (1-7))
START THIS INPUT TIMER, timer, time
STOP THIS INPUT TIMER, timer
IF THIS INPUT TIMER ON, timer

INPUT TIMER TEST CONTROL:
IF THIS INPUT TIMER EXPIRED, timer

INPUT TEST CONTROL:
IF ACCESS
IF ALARM ON
IF BYPASSED
IF ENABLED
IF ISOLATED
IF SEALED
IF SECURED
IF TAMPER
IF UNSEALED

FUNCTION RETURN COMMAND:
RETURN

Local Script Commands for AREA SCRIPT types are:

AREA EVENT SCRIPTS:

IF AREA EVENT ON ACCESS
IF AREA EVENT ON ALARM
IF AREA EVENT ON BYPASS OFF
IF AREA EVENT ON BYPASS ON
IF AREA EVENT ON ISOLATE OFF
IF AREA EVENT ON ISOLATE ON
IF AREA EVENT ON RESET
IF AREA EVENT ON SEAL
IF AREA EVENT ON SECURE
IF AREA EVENT ON TAMPER
IF AREA EVENT ON UNSEAL

AREA CONTROL COMMANDS:
ACCESS THIS AREA
RESET THIS AREA
SECURE THIS AREA

AREA FLAG CONTROL: (8 flags (1-8) per each Area))
CLEAR THIS AREA FLAG, flag
SET THIS AREA FLAG, flag

AREA TIMER CONTROL: (up to 7 timers per each Areas (1-7))
START THIS AREA TIMER, timer, time
STOP THIS AREA TIMER, timer

AREA TIMER TEST CONTROL:
IF THIS AREA TIMER ON, timer

AREA TEST CONTROL:
IF AREA IN ACCESS CONDITION
IF AREA IN SECURE CONDITION

IF AREA IN ALARM CONDITION
IF AREA IN TAMPER CONDITION

IF AREA IN BYPASSED CONDITION
IF AREA IN ISOLATED CONDITION

IF AREA IN SEALED CONDITION
IF AREA IN UNSEALED CONDITION

IF THIS AREA FLAG, flag

Local Script Commands for CLOCK SCRIPT types are:

CLOCK EVENT SCRIPTS:
IF CLOCK EVENT TO VALID
IF CLOCK EVENT TO VOID

CLOCK TEST CONTROL:
IF CLOCK VALID, clock
IF CLOCK VOID, clock

Local Script Commands for DOOR SCRIPT types are:

DOOR EVENT SCRIPTS:
IF DOOR EVENT ON CLOSED
IF DOOR EVENT ON DOTL
IF DOOR EVENT ON FORCED
IF DOOR EVENT ON TAMPER
IF DOOR EVENT ON TIMER EXPIRE

DOOR FLAG CONTROL: (8 flags (1-8) per each Door)
SET THIS DOOR FLAG, flag
CLEAR THIS DOOR FLAG, flag
DOOR TEST CONTROL:
IF DOOR EQUAL, door
IF THIS DOOR FLAG, flag
IF DOOR RANGE, door1, door2

DOOR TIMER CONTROL: (up to 7 timers per each Door (1-7))
START THIS DOOR TIMER, timer, time
STOP THIS DOOR TIMER, timer
IF THIS DOOR TIMER ON, timer

DOOR TIMER TEST CONTROL:
IF THIS DOOR TIMER EXPIRED, timer

DOOR CONTROL COMMANDS:
LOCK ALL ON THIS DOOR
LOCK THIS DOOR ENTRY
RELEASE THIS DOOR
SECURE THIS DOOR
UNLOCK THIS DOOR

Local Script Commands for ACCESS SCRIPT types are:

ACCESS TEST CONTROL:

IF ACCESS DOOR EQUAL, door
IF ACCESS DOOR RANGE, door1, door2
IF GROUP EQUAL, group
IF GROUP RANGE, group1, group2
IF USER EQUAL, user
IF USER RANGE, user1, user2

ACCESS FLAGS CONTROL: (8 flags (1-8) per each Group and User)

RESET THIS GROUP FLAG, flag
SET THIS USER FLAG, flag
RESET THIS USER FLAG, flag
SET THIS USER FLAG, flag
IF THIS GROUP FLAG, flag
IF THIS USER FLAG, flag

Global Script Commands

Global scripts allow test and control of specific input, door, area or group. Global scripts can be included in any script type within the system.

AREA GLOBAL CONTROL:

ACCESS AREA, area
SECURE AREA, area
RESET AREA, area
CLEAR AREA FLAG, area, flag
SET AREA FLAG ON, area, flag
START AREA DELAY TIMER, area, timer, duration
STOP AREA DELAY TIMER, area, timer
IF AREA DELAY TIMER ON, door, timer
IF AREA IN ACCESS, area
IF AREA IN ALARM, area
IF AREA IN SECURE, area
IF AREA IN TAMPER, area
IF AREA IS BYPASSED, area
IF AREA IS ISOLATED, area
IF AREA IS SEALED, area
IF AREA IS UNSEALED, area

OUTPUT GLOBAL SCRIPTS:

BUZZER CONTROL, device, action, time
OUTPUT CONTROL, output, action, time
OLIST CONTROL, olist, action, time
IF OUTPUT ON, output

INPUT GLOBAL SCRIPTS:

BYPASS INPUT, input
REMOVE BYPASS, input
ISOLATE INPUT, input
REMOVE ISOLATION, input
SET INPUT FLAG, input, flag
CLEAR INPUT FLAG, input, flag
IF INPUT FLAG ON, input, flag
INPUT ALARM OFF, input
INPUT ALARM ON, input
IF INPUT DELAY TIMER ON, input, timer
START INPUT DELAY TIMER, input, timer, duration
STOP INPUT DELAY TIMER, input, timer

GLOBAL CLOCK SCRIPTS:

SET CLOCK FLAG, clock, flag
CLEAR CLOCK FLAG, clock, flag
ENABLE CLOCK, clock
DISABLE CLOCK, clock
IF CLOCK VALID, clock
IF CLOCK VOID, clock

GLOBAL DOOR SCRIPTS:

SET DOOR FLAG, door, flag
IF DOOR FLAG ON, door, flag
CLEAR DOOR FLAG, door, flag
SECURE DOOR, door
LOCK DOOR ENTRY, door
LOCK DOOR ENTRY, door
RELEASE DOOR, door
UNLOCK DOOR, door
IF DOOR DELAY TIMER ON, door, timer

GLOBAL SCRIPTS

CALL SCRIPT, script, param
IF CONTROL VALUE EQUAL, value
CHAIN, script
SEND RAS MESSAGE, ras, message
SEND COMM MESSAGE, unit, port, message
SEND DIALLER, dialler, subscriber, type, code
SEND SECURITEL**,** priority, event, value
ELSE
ENDIF
RETURN
TRACE ON
TRACE OFF
IF TIME IS, condition, hours, minutes

IF INPUT EVENT ON BYPASS OFF

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when BYPASS EVENT has been requested.

Parameters: Nil

See also: IF INPUT EVENT ON BYPASS ON, INPUT EVENT SCRIPTS

Example:
The following input script removes the BYPASS flag.

Example:

```
IF INPUT EVENT ON BYPASS OFF      // Bypass off command
        BYPASS OFF                // Remove Bypass
ENDIF
```

IF INPUT EVENT ON BYPASS ON

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when BYPASS EVENT has been requested.

Parameters: Nil

See also: IF INPUT EVENT ON BYPASS OFF, INPUT EVENT SCRIPTS
Example:
The following input script removes the BYPASS flag.

Example:

```
IF INPUT EVENT ON BYPASS ON        // Bypass on command
        BYPASS ON                  // Set Bypass
ENDIF
```

IF INPUT EVENT ON ISOLATE OFF

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when ISOLATE EVENT has been requested.

Parameters: Nil

See also: IF INPUT EVENT ON ISOLATE ON, INPUT EVENT SCRIPTS

Example:
The following input script removes the ISOLATE flag.

Example:

```
IF INPUT EVENT ON ISOLATE OFF     // Isolate off command
        ISOLATE OFF               // Remove Isolation
ENDIF
```

IF INPUT EVENT ON ISOLATE ON

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when ISOLATE EVENT has been requested.

Parameters: Nil

See also: IF INPUT EVENT ON ISOLATE OFF, INPUT EVENT SCRIPTS

Example:
The following input script removes the BYPASS flag.

Example:

```
IF INPUT EVENT ON ISOLATE ON        // Isolation command
        ISOLATE ON                  // Set Isolation
ENDIF
```

IF INPUT EVENT ON RESET

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when RESET EVENT has been requested.

Parameters: Nil

See also: ALARM ON, ALARM OFF, TAMPER OFF, TAMPER ON, INPUT EVENT SCRIPTS

Example:
The following input script removes the ALARM ON flag, when RESET is generated.

Example:

```
IF INPUT EVENT ON RESET             // Reset command
        ALARM OFF                   // Remove ALARM ON
ENDIF
```

IF INPUT EVENT ON SEAL

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when INPUT HAS RETURNED TO SEAL status from either unsealed or tamper condition.

Parameters: Nil

See also: IF INPUT EVENT ON UNSEAL, INPUT EVENT SCRIPTS

Please note: When last input in an Area is sealed, system automatically will generate the 'IF AREA EVENT ON SEAL'.

Example:
Timer is started on unsealed status, and if timer expires, alarm is generated. If input seals prior timer expire, alarm is not generated. This example can be used as delayed input

Example:

```
IF INPUT EVENT ON UNSEAL              // Unseal command
        START THIS INPUT TIMER, 1, 60         // Start 60 SECONDS timer
ENDIF
IF INPUT EVENT ON SEALED              // seal command
        STOP THIS INPUT TIMER, 1              // Stop timer
ENDIF
IF INPUT EVENT ON TIMER EXPIRE        // Timer expired
        ALARM ON                      // Generate ALARM
ENDIF
IF INPUT EVENT ON RESET               // Reset command
        ALARM OFF                     // Reset ALARM
ENDIF
```

IF INPUT EVENT ON UNSEAL

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when INPUT HAS CHANGED TO UNSEAL status.

Parameters: Nil

See also: IF INPUT EVENT ON SEAL, INPUT EVENT SCRIPTS

Please note: When FIRST input in an Area is unsealed, system automatically will generate the 'IF AREA EVENT ON UNSEAL'.

Example:
Timer is started on unsealed status, and if timer has expired, alarm is generated. If input seals prior timer expire, alarm is not generated. This example can be used as delayed input

Example:

```
IF INPUT EVENT ON UNSEAL           // Unseal command
        START THIS INPUT TIMER, 1, 60        // Start 60 SECONDS timer
ENDIF
IF INPUT EVENT ON SEALED           // seal command
        STOP THIS INPUT TIMER, 1             // Stop timer
ENDIF
IF INPUT EVENT ON TIMER EXPIRE     // Timer expired
        ALARM ON                   // Generate ALARM
ENDIF
IF INPUT EVENT ON RESET            // Reset command
        ALARM OFF                  // Reset ALARM
ENDIF
```

IF INPUT EVENT ON ACCESS

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when INPUT HAS changed to ACCESS mode.

Parameters: Nil

See also: IF INPUT EVENT ON SECURE, INPUT EVENT SCRIPTS

Example:

Timer is started on unsealed status, and if timer expires, alarm is generated. If input seals prior timer expire, alarm is not generated. This example can be used as delayed input

Example:

```
IF INPUT EVENT ON ACCESS                // Access command
        STOP THIS INPUT TIMER, 1                // Stop Exit timer
        STOP THIS INPUT TIMER, 2                // Stop Entry timer
ENDIF
IF INPUT EVENT ON SECURE                // Secure command
IF ENABLED                              // If Input is ENABLED
        START THIS INPUT TIMER, 1, 60           // Start Exit timer
ENDIF
ENDIF
IF INPUT EVENT ON UNSEAL
IF ENABLED                              // Input in Normal
IF SECURED                              // Input is Secured
IF THIS INPUT TIMER ON, 1
        RETURN
ENDIF
IF THIS INPUT TIMER  ON, 2
        RETURN
ELSE
        START THIS INPUT TIMER, 2, 30           // start entry timer
ENDIF
ENDIF
ENDIF
ENDIF
IF INPUT EVENT ON TIMER EXPIRE          // Timer expired
IF THIS INPUT TIMER EXPIRED, 1                  // Exit Expired ?
IF UNSEALED                             // Is input still unsealed ?
        ALARM ON                        // Generate ALARM
        SEND DIALLER,1,999,1,140        // Send LWS Dialler alarm
ENDIF
IF THIS INPUT TIMER EXPIRED,2                   // was is exit timer
        ALARM ON                        // Generate ALARM
        SEND DIALLER,1,999,1,140        // Send Dialler alarm
ENDIF
ENDIF
IF INPUT EVENT ON RESET                 // Reset command
        ALARM OFF                       // Reset ALARM
        SEND DIALLER,1,999,3,140        // Send Dialler Reset
ENDIF
```

IF INPUT EVENT ON SECURE

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when INPUT HAS changed to ACCESS mode.

Parameters: Nil

See also: IF INPUT EVENT ON ACCESS, INPUT EVENT SCRIPTS

Example:
Timer is started on unsealed status, and if timer expires, alarm is generated. If input seals prior timer expire, alarm is not generated. This example can be used as delayed input

Example:

```
IF INPUT EVENT ON ACCESS              // Access command
        STOP THIS INPUT TIMER,1                // Stop timer
        CLEAR THIS INPUT FLAG,1
        CLEAR THIS INPUT FLAG,2                // clear flags
ENDIF
IF INPUT EVENT ON SECURE              // Secure command
        START THIS INPUT TIMER,1,60 // Start Exit timer
        SET THIS INPUT FLAG,1                  // Indicate Exit running
ENDIF
IF INPUT EVENT ON UNSEALED
IF ACCESS                        // Input in Access, ignore
        RETURN
ENDIF
IF THIS INPUT FLAG,1                  // Exit timer running
        RETURN                        // ignore
ENDIF
IF THIS INPUT FLAG,2                  // Indicate Entry timer running
        RETURN                        // ignore
ELSE
        START THIS INPUT TIMER,1,30 // start entry timer
        SET THIS INPUT FLAG,2         // indicate, entry running
ENDIF
ENDIF
IF INPUT EVENT ON TIMER EXPIRE        // Timer expired
IF THIS INPUT FLAG,1                  // was is exit timer
        CLEAR THIS INPUT FLAG,1                // Indicate so
        RETURN
ENDIF
IF THIS INPUT FLAG,2                  // was it entry timer
        ALARM ON                      // Generate ALARM
ENDIF
ENDIF
IF INPUT EVENT ON RESET               // Reset command
        ALARM OFF                     // Reset ALARM
ENDIF
```

IF INPUT EVENT ON TIMER EXPIRE

Purpose: Generates Event on expire of input timer

Type: Event

Available for:

☑ Input

Description: This EVENT is generated, when **ANY** timer(s) for this input expires.

Parameters: Nil

See also: START INPUT TIMER, STOP THIS INPUT TIMER,,IF THIS INPUT TIMER EXPIRED, INPUT EVENT SCRIPTS

Example:
Timer is started on unsealed status, and if timer expires, alarm is generated. If input seals prior timer expire, alarm is not generated. This example can be used as delayed input

Example:

```
IF INPUT EVENT ON ACCESS                // Access command
        STOP THIS INPUT TIMER,1                 // Stop timer
        CLEAR THIS INPUT FLAG,1
        CLEAR THIS INPUT FLAG,2         // clear flags
ENDIF
IF INPUT EVENT ON SECURE                // Secure command
        START THIS INPUT TIMER,1,60 // Start Exit timer
        SET THIS INPUT FLAG,1           // Indicate Exit running
ENDIF
IF INPUT EVENT ON UNSEALED
IF ACCESS                       // Input in Access, ignore
        RETURN
ENDIF
IF THIS INPUT FLAG,1                    // Exit timer running
        RETURN                          // ignore
ENDIF
IF THIS INPUT FLAG,2                    // Indicate Entry timer running
        RETURN                          // ignore
ELSE
        START THIS INPUT TIMER,1,30 // start entry timer
        SET THIS INPUT FLAG,2           // indicate, entry running
ENDIF
ENDIF
IF INPUT EVENT ON TIMER EXPIRE  // Any Timer expired, we can add additional test to find out
                                // which timer has expired (see 'IF TIMER EXPIRE, timer'
                                // command
IF THIS INPUT FLAG,1                    // was is exit timer
        CLEAR THIS INPUT FLAG,1         // Indicate so
        RETURN
ENDIF
IF THIS INPUT FLAG,2                    // was it entry timer
        ALARM ON                        // Generate ALARM
ENDIF
ENDIF
IF INPUT EVENT ON RESET                 // Reset command
        ALARM OFF                       // Reset ALARM
```

```
ENDIF
```

IF INPUT EVENT ON TAMPER

Purpose: Generates Event on change of status of the input

Type: Event

Available for:

☑ Input

Description: Generate event when input has changed into TAMPER status.

Parameters: Nil

See also: TAMPER ON, TAMPER OFF, INPUT EVENT SCRIPTS

Example:
When TAMPER status is detected, the TAMPER alarm is generated. Note: see EOL programming.

Example:

```
IF INPUT EVENT ON TAMPER          // Tamper command
        TAMPER ON                 // Latch TAMPER alarm
ENDIF
IF INPUT EVENT ON RESET           // Reset command
        TAMPER OFF                // Reset TAMPER
        ALARM OFF                 // Reset ALARM
ENDIF
IF INPUT EVENT ON UNSEALED
IF SECURE                 // Input in Secure mode
        ALARM ON                  // Set ALARM Latch
ENDIF
ENDIF
```

INPUT FLAG CONTROLS:

Each Input includes number of flags, such as ALARM, TAMPER, ISOLATE and BYPASS. We have included additional 8 optional flags (1-8), which can be used for your own benefit in for your own application.

Each flag can be set, unset or its status tested by the following commands:-

SET THIS INPUT FLAG, *flag*
CLEAR THIS INPUT FLAG, *flag*
IF THIS INPUT FLAG, *flag*

ALARM OFF

Purpose: Removes the alarm latch flag for the input.

Type: Control

Available for:

☑ Input

Description: Removes the alarm latch flag from the input.

Normally this command will be added in the ‘IF INPUT EVENT ON RESET` resetting previously latched alarm condition with the ‘ALARM ON` command.

WARNING: Strong consideration must be if this command is used in any other position to turn the alarm off.

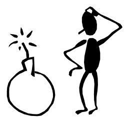
Parameters: Nil

See also: ALARM ON**,** IF INPUT EVENT ON RESET, INPUT FLAG CONTROLS

Example:
The following input script is used to turn the alarm off.

Example:

```
IF INPUT EVENT ON RESET         // did we generated RESET EVENT
   ALARM OFF                    // Remove the Latched ALARM
ENDIF                           // ENDIF
```



Use with care

ALARM ON

Purpose: Turns on the alarm flag for an input.

Type: Control

Available for:

☑ Input

Description: Turns on the alarm flag associated with the input.

Parameters:

See also: ALARM OFF, INPUT FLAG CONTROLS

Please note: When first ALARM ON flag is set on an input in an AREA, system will generate the 'IF AREA EVENT ON ALARM' event. Common action, such as Siren and Strobe outputs, can be auctioned in the AREA SCRIPT.

Example: The following is an extract from the standard input script. When the input is unsealed, it checks to see if the input is secure and, if so, sets the alarm bit.

Example:

```
IF INPUT EVENT ON UNSEAL       // did we generated UNSEAL EVENT
IF SECURE                      // is input in SECURE MODE
   ALARM ON                    // Yes, Latch the ALARM
ENDIF                          // ENDIF 'IF SECURE'
ENDIF                          // ENDIF 'IF INPUT EVENT ON UNSEAL
```

BYPASS OFF

Purpose: Turns off the bypass flag for an input.

Type: Control

Available for:

☑ Input

Description: Turns off the bypass flag associated with the input.

Parameters: NILL,

See also: BYPASS ON, INPUT FLAG CONTROLS

Example: The following is an extract from the input script. When the input is set to ACCESS MODE, we automatically remove the BYPASS Flag, if set.

Please note: Input can ONLY be BYPASS ON and BYPASS OFF by an user, which has an access to SERVICE MENU from the RAS. You do not need to use the BYPASS OFF command within the 'IF INPUT EVENT ON BYPASS OFF'. This flag is cleared automatically by the system, when user REMOVES THE BYPASS input via RAS menu.
If 'BYPASS OFF' command is used within the Script, system automatically generates the 'IF INPUT EVENT BYPASS OFF' event.

Example:

```
IF INPUT EVENT ON ACCESS     // did we received ACCESS EVENT
IF BYPASSED                  // is input in BYPASS MODE
   BYPASS OFF                // Yes, Remove the BYPASS
ENDIF                        // ENDIF 'IF BYPASSED'
ENDIF                        // ENDIF 'IF INPUT EVENT ON ACCESS


Or, when user requests BYPASS OFF

IF INPUT EVENT ON BYPASS OFF // Is this a BYPASS OFF EVENT
   SEND DIALLER,1,9999,3,x   // Yes, Send Dialler
ENDIF                  // ENDIF 'IF INPUT EVENT ON BYPASS OFF
```

BYPASS ON

Purpose: Turns on the bypass flag for an input.

Type: Control

Available for:

☑ Input

Description: Turns on the bypass flag associated with the input.

Parameters: NILL,

See also: BYPASS OFF, INPUT FLAG CONTROLS

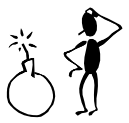
Example: The following is an extract from the input script. When the USER BYPASSES the input, we send a Dialler message.

Please note: Input can ONLY be BYPASS ON and BYPASS OFF by an user, which has an access to SERVICE MENU from the RAS. Do not USE the BYPASS ON command within the 'IF INPUT EVENT ON BYPASS ON'. This flag is set automatically by the system, when user BYPASSES the input.
If 'BYPASS ON' command is used within the Script, system automatically generates the 'IF INPUT EVENT BYPASS ON' event.

Example:

```
IF INPUT EVENT ON BYPASS ON    // did we received BYPASS EVENT
   SEND DIALLER,1,9999,1,x     // Yes, SET the BYPASS Flag
ENDIF                          // ENDIF 'IF INPUT EVENT ON BYPASS
IF INPUT EVENT ON ACCESS
        BYPASS OFF             // Generate the 'IF INPUT EVENT ON BYPASS OFF'
ENDIF
```

Note: When input is in BYPASS MODE, all INPUT EVENTS are suppressed, including UNSEAL, SEAL and TAMPER INPUT EVENTS.



Use with care

ISOLATE OFF

Purpose: Turns off the ISOLATION flag for an input.

Type: Control

Available for:

☑ Input

Description: Turns off the ISOLATE flag associated with the input.

Parameters: NILL,

See also: ISOLATE ON, INPUT FLAG CONTROLS

Please note: If 'ISOLATE OFF' command is used within the Script, system automatically generates the 'IF INPUT EVENT ISOLATE OFF' event on last input deisolated in the area.

Example: The following is an extract from the input script. When an AREA is set to ACCESS mode, we remove the ISOLATE flag. System automatically generates the event, where we send the status to the Dialler.

Example:

```
IF INPUT EVENT ON ACCESS      // did we received ACCESS EVENT
IF ISOLATED                   // is input ISOLATED
   ISOLATE OFF                // Yes, Clear the ISOLATE Flag
ENDIF                         // ENDIF 'IF ISOLATED'
ENDIF                   // ENDIF 'IF INPUT EVENT ON ACCESS
IF INPUT EVENT ON ISOLATE OFF
      SEND DIALLER,1,9999,3,x
ENDIF
```

ISOLATE ON

Purpose: Turns on the ISOLATION flag for an input.

Type: Control

Available for:

☑ Input

Description: Turns on the ISOLATE flag associated with the input.

Parameters: NILL,

See also: ISOLATE OFF, INPUT FLAG CONTROLS

Example: The following is an extract from the input script. When USER ISOLATES the input, we send the ISOLATION status to the Dialler.

Please note: ISOLATE flag is set or cleared automatically by the system, when user ISOLATE or DEISOLATE the input. INPUT CANNOT BE ISOLATE, EITHER BY USER OR COMPUTER CONTROL, WHILE IN SECURE MODE. If 'ISOLATE ON' command is used within the Script, system automatically generates the 'IF INPUT EVENT ISOLATE ON' event on first isolated sector.

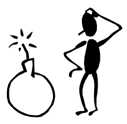
Example:

```
IF INPUT EVENT ON ISOLATE     // did we received ISOLATE EVENT
   SEND DIALLER,1,9999,1,x    // Send Message
ENDIF                   // ENDIF 'IF INPUT EVENT ON ISOLATE OFF
```

Note: When input is in ISOLATE MODE, you MUST test for this condition within scripts, as none of the INPUT EVENTS are suppressed.

Example:

```
IF INPUT EVENT ON UNSEAL      // did we received UNSEAL EVENT
IF ISOLATED                   // is input ISOLATED
   RETURN                     // Ignore UNSEAL STATUS
ELSE                          // Input is not isolated
.                             // DO SOME OTHER COMMANDS
.
ENDIF                         // ENDIF 'IF ISOLATED'
ENDIF                   // ENDIF 'IF INPUT EVENT ON UNSEAL
```



Use with care

TAMPER OFF

Purpose: Turns off the TAMPER flag for an input.

Type: Control

Available for:

☑ Input

Description: Turns off the TAMPER flag associated with the input.

Parameters: NILL,

See also: TAMPER ON, INPUT FLAG CONTROLS

Example: The following is an extract from the input script. When 'IF INPUT EVENT ON RESET' is generated, the TAMPER FLAG is cleared.

Example 1:

```
IF INPUT EVENT ON RESET       // did we received RESET EVENT
   TAMPER OFF                 // Clear the TAMPER Flag
ENDIF                   // ENDIF 'IF INPUT EVENT ON RESET`
```

Example 2: Here is an example how to check, if ALARM and/or TAMPER flag has been set:

```
IF INPUT EVENT ON RESET       // did we received RESET EVENT
IF TAMPER                     // is input in TAMPER
     TAMPER OFF               // Clear Tamper Flag
ENDIF                         //
IF ALARM ON                   // is input in ALARM
     ALARM OFF                // Reset Alarm Flag
ENDIF                         // ENDIF 'IF ISOLATED'
ENDIF                   // ENDIF 'IF INPUT EVENT ON RESET
```

Please note, we are testing for TAMPER as well ALARM Flag, and the ELSE command is not used. We should consider, that ALARM and TAMPER flags could be both set.

Example 3:

If TAMPER and ALARM flags are both set, this script will require two reset commands. On First Reset, the TAMPER flag is cleared, and on Second Reset, the ALARM flag is cleared. This is not recommended.

```
IF TAMPER                     // is input in TAMPER
     TAMPER OFF               // Clear Tamper Flag
ELSE                          //
     ALARM OFF                // Reset Alarm Flag
ENDIF                         // ENDIF 'IF TAMPER'
```

Example 4:

If we do not need to distinguish between the types of reset, the following script can be used:

```
IF INPUT EVENT ON RESET       // did we received RESET EVENT
     TAMPER OFF               // Clear Tamper Flag
     ALARM OFF                // Reset Alarm Flag
ENDIF                   // ENDIF 'IF INPUT EVENT ON RESET'
```

TAMPER ON

Purpose: Turns on the TAMPER flag for an input.

Type: Control

Available for:

☑ Input

Description: Turns on the TAMPER flag associated with the input.

Parameters: NILL,

See also: TAMPER OFF, INPUT FLAG CONTROLS

Please note: When first TAMPER ON flag is set on an input in an AREA, system will generate the 'IF AREA EVENT ON TAMPER' event. Common action, such as Siren and Strobe outputs, can be auctioned in the AREA SCRIPT.

Example: The following is an extract from the input script. When 'IF INPUT EVENT ON TAMPER' is generated, the TAMPER FLAG is set.

Example:

```
IF INPUT EVENT ON TAMPER   // we action the TAMPER EVENT
        TAMPER ON          // Set the TAMPER FLAG
ENDIF
```

CLEAR THIS INPUT FLAG, x

Purpose: Turns off the OPTIONAL flag for this input.

Type: Control

Available for:

☑ Input

Description: Turns off the OPTIONAL flag on this input.

Parameters: Flag number (1-8),

See also: SET THIS INPUT FLAG, IF THIS INPUT FLAG, THIS INPUT FLAG CONTROLS

Example: The following is example how to use the CLEAR THIS INPUT FLAG command.

Example:

```
IF INPUT EVENT ON UNSEAL
        CLEAR THIS INPUT FLAG,1     // Set flag 1 to indicate activity on this input
ENDIF
IF INPUT EVENT ON SECURE
IF INPUT FLAG,1                     // is this input active?
        LOG EVENT,1                 // No, generate report
ENDIF
ENDIF
IF INPUT EVENT ON ACCESS            // Input changed to ACCESS
        SET THIS INPUT FLAG,1       // Set it, indicate no unsealed
ENDIF
```

SET THIS INPUT FLAG, x

Purpose: Turns on the OPTIONAL flag for this input.

Type: Control

Available for:

☑ Input

Description: Turns on the OPTIONAL flag on this input.

Parameters: Flag number (1-8),

See also: CLEAR THIS INPUT FLAG, IF THIS INPUT FLAG, THIS INPUT FLAG CONTROLS

Example: The following is example how to use the SET THIS INPUT FLAG command.

Example:

```
IF INPUT EVENT ON UNSEAL
        CLEAR THIS INPUT FLAG, 1      // Clear flag 1 to indicate activity on this input
ENDIF
IF INPUT EVENT ON SECURE
IF INPUT FLAG, 1                      // is this input active?
        LOG EVENT, 1                  // No, generate report
ENDIF
ENDIF
IF INPUT EVENT ON ACCESS              // Input changed to ACCESS
        SET THIS INPUT FLAG, 1        // Set it, indicate no unsealed
ENDIF
```

IF THIS INPUT FLAG, flag

Purpose: Tests for the status of input flag.

Type: Test

Available for:

☑ Input

Description: Test the condition of flag on local input.

Parameters: Flag number (1-7),

See also: CLEAR THIS INPUT FLAG, SET THIS INPUT FLAG, THIS INPUT FLAG CONTROLS

Example: The following is example how to use the IF INPUT FLAG" command.
If input generate alarm, while in secure mode, flag 1 is set. When user accesses the Area, we test for status of the flag, IF flag has been set, we generate output to indicate sector has generated alarm and flag is cleared.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
IF SECURE
        ALARM ON                        // Generate alarm
ENDIF

ENDIF
IF INPUT EVENT ON RESET
        ALARM OFF                       // Reset ALARM Latch
ENDIF
ENDIF
IF INPUT EVENT ON ACCESS                // Are we changing to Access mode?
        IF THIS INPUT FLAG,1
                OUTPUT CONTROL,1,3,10   // Input alarmed in secure, set output
                                        // indicator
                CLEAR THIS INPUT FLAG,1 // Ensure flag is cleared
        ENDIF
ENDIF
```

STOP THIS INPUT TIMER, timer

Purpose: Stops input timer, if active.

Type: **Control**

Available for:

☑ Input

Description: Stops timer, if active, on this input.

Parameters: Timer number (1-7),

See also: START THIS INPUT TIMER, IF INPUT EVENT ON TIMER EXPIRE, IF INPUT TIMER EXPIRED, IF THIS INPUT TIMER ON, INPUT TIMER CONTROL

Example: The following is example how to use the 'STOP THIS INPUT TIMER' command. If input seals within delay time, the timer is terminated.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
IF ALARM ON
        RETURN                      //If in alarm, ignore
ENDIF

ENDIF
IF INPUT EVENT ON SEAL
        STOP THIS INPUT TIMER, 1            // We sealed, stop timer
ENDIF
IF INPUT EVENT ON RESET
        ALARM OFF                   // Reset ALARM Latch
ENDIF
ENDIF
IF INPUT EVENT ON TIMER EXPIRE      // Any Timer expired
IF UNSEALED                         // Still unsealed
        ALARM ON                    // Set ALARM ON
ENDIF
ENDIF
```

START THIS INPUT TIMER, timer

Purpose: Starts input timer for input.

Type: **Control**

Available for:

☑ Input

Description: Start input timer on this input.

Parameters: Timer number (1-7),

See also: STOP THIS INPUT TIMER, IF INPUT EVENT ON TIMER EXPIRE, IF THIS INPUT TIMER EXPIRED, IF THIS INPUT TIMER ON, INPUT TIMER CONTROL

Example: The following is example how to use the 'START THIS INPUT TIMER' command. If input is unsealed, and reminds unsealed for 4 seconds, alarm condition is latched on. If input seals within 4 seconds, timer is stopped.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
IF ALARM ON
        RETURN                          //If in alarm, ignore
ENDIF
        START THIS INPUT TIMER,1,4              // We unsealed, start timer 4 seconds
ENDIF
IF INPUT EVENT ON SEAL
        STOP THIS INPUT TIMER,1                 // We sealed, stop timer
ENDIF
IF INPUT EVENT ON RESET
        ALARM OFF                       // Reset ALARM Latch
ENDIF
ENDIF
IF INPUT EVENT ON TIMER EXPIRE          // Any Timer expired
        ALARM ON                        // Set ALARM ON
ENDIF
```

IF THIS INPUT TIMER EXPIRED, timer

Purpose: Test, if timer generated expire event.

Type: **Test**

Available for:

☑ Input

Description: Tests, if timer generated expire event.

Parameters: Timer number (1-7),

See also: START THIS INPUT TIMER, STOP THIS INPUT TIMER, IF INPUT EVENT ON TIMER EXPIRE, INPUT TIMER TEST CONTROL

Example: The following is example how to use the 'IF INPUT TIMER EXPIRED' command. When Timer expire event is generated, we test, if the timer 1 generated this event.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
IF ALARM ON
        RETURN                          //If in alarm, ignore
ENDIF
        START THIS INPUT TIMER,1,30     // Start timer for 30 seconds
ENDIF
IF INPUT EVENT ON SEAL
        STOP THIS INPUT TIMER,1                 // We sealed, stop timer
ENDIF
IF INPUT EVENT ON RESET
        ALARM OFF                       // Reset ALARM Latch
ENDIF
ENDIF
IF INPUT EVENT ON TIMER EXPIRE          // Any Timer expired
IF THIS INPUT TIMER EXPIRE,1                    // did timer 1 expired ?
IF UNSEALED                             // Still unsealed
        ALARM ON                        // Set ALARM ON
ENDIF
ENDIF
ENDIF
```

IF THIS INPUT TIMER ON, timer

Purpose: Test, if input timer is active.

Type: **Test**

Available for:

☑ Input

Description: Tests, if input timer is active.

Parameters: Timer number (1-7),

See also: START THIS INPUT TIMER, STOP THIS INPUT TIMER, IF INPUT EVENT ON TIMER EXPIRE, IF INPUT TIMER EXPIRED, INPUT TIMER CONTROL

Example: The following is example how to use the 'IF INPUT TIMER ON' command. If input seals within delay time, the timer is terminated.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
IF INPUT TIMER ON,1
        RETURN
ELSE
IF ALARM ON
        RETURN                          //If in alarm, ignore
ENDIF
        START THIS INPUT TIMER,1,30 // Start Delay Timer
ENDIF
ENDIF
IF INPUT EVENT ON SEAL
        STOP THIS INPUT TIMER,1                 // We sealed, stop timer
ENDIF
IF INPUT EVENT ON RESET
        ALARM OFF                       // Reset ALARM Latch
ENDIF
ENDIF
IF INPUT EVENT ON TIMER EXPIRE          // Any Timer expired
IF UNSEALED                             // Still unsealed
        ALARM ON                        // Set ALARM ON
ENDIF
ENDIF
```

INPUT TEST CONTROL:

Input test control is used to test the status of flag of the input.

Please note: It is recommended; control commands are located within the 'IF INPUT EVENT xxx` functions. Remember this command is a TEST not an EVENT command.

IF ACCESS

Purpose: Tests the ACCESS flag of the input.

Type: Test

Available for:

☑ Input

Description: Tests the ACCESS flags of the input, and if TRUE (in Access), the commands follow the test condition will be executed.
Please note: The Access flag is automatically set or reset by the system.

Parameters: NILL,

See also: IF SECURED, INPUT TEST CONTROL

Example: The following is example how to use the 'IF ACCESS` command.

Example 1:

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEAL EVENT
IF ACCESS                     // is input in ACCESS MODE
     RETURN                   // Yes, terminated
ENDIF
   ALARM ON                   // Secured, Latch the ALARM
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Example 2:

If input is in access mode and the unsealed event is generated, the BUZZER on MASTER panel is activated for duration of 10 seconds.

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEAL EVENT
IF ACCESS                     // is input in ACCESS MODE
     BUZZER CONTROL,1,2,10    // Yes, activate Buzzer
ENDIF
   ALARM ON                   // Secured, Latch the ALARM
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Example 3:

Let assume, this input operates as Duress button, and generates alarm only if in Access mode.

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEAL EVENT
IF ACCESS                     // is input in ACCESS MODE
   ALARM ON                   // Yes, Latch the ALARM
ENDIF
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
```

IF SECURED

Purpose: Tests the SECURE flag of the input.

Type: Test

Available for:

☑ Input

Description: Tests the SECURE flag of the input, and if TRUE (in SECURE), the commands follow the test condition will be executed.
Please note: The Secure Flag is automatically set or reset by the system.

Parameters: NILL,

See also: IF ACCESS, INPUT TEST CONTROL

Example: The following is example how to use the 'IF SECURE` command.

Example 1:

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEAL EVENT
IF SECURED                    // is input in SECURE MODE
   ALARM ON                   // Secured, Latch the ALARM
ENDIF                         // ENDIF 'IF SECURE'
ENDIF                   // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Example 2:

If input is in SECURE mode and the unsealed event is generated, ALARM is LATCHED on, otherwise activates the BUZZER on MASTER.

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEAL EVENT
IF SECURED                    // is input in SECURE MODE
   ALARM ON                   // Secured, Latch the ALARM
ELSE
     BUZZER CONTROL,1,2,10    // Access, activate Buzzer
ENDIF
ENDIF                   // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Example 3:

Let assume, this input operates as Duress button, and generates alarm only if in Access mode.

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEAL EVENT
IF SECURED                    // is input in SECURE MODE
     RETURN                   // Yes, ignore
ELSE
   ALARM ON                   // Access, Latch the ALARM
ENDIF
ENDIF                   // ENDIF 'IF INPUT EVENT ON UNSEAL
```

IF ALARM ON

Purpose: Tests the ALARM flag of the input.

Type: Test

Available for:

☑ Input

Description: Tests the ALARM flags of the input, and if TRUE (in ALARM), the commands follow the test condition will be executed.

Parameters: NILL,

See also: ALARM ON, ALARM OFF, INPUT TEST CONTROL

Example: The following is example how to use the 'IF ALARM ON` command.

Example 1:

```
IF INPUT EVENT ON RESET         // did we generated RESET EVENT
IF ALARM ON                     // is input in ALARM
   ALARM OFF                    // Yes, Reset the ALARM
ENDIF                           // ENDIF 'IF ALARM ON'
ENDIF                     // ENDIF 'IF INPUT EVENT ON RESET
```

Example 2:

If input is already in ALARM, the SENDS DIALLER is ignored; otherwise the alarm is latched and reported via Dialler.

```
IF INPUT EVENT ON UNSEAL        // did we generated UNSEAL EVENT
IF SECURED                      // is input in SECURE MODE
IF ALARM ON                     // Already in ALARM
      RETURN                    // IGNORE
ELSE
      ALARM ON                  // Latch the ALARM
      SEND DIALLER,1,1,140      // Send Dialler
ENDIF
ENDIF                           // ENDIF 'IF SECURED'
ENDIF                     // ENDIF 'IF INPUT EVENT ON UNSEAL
```

IF BYPASSED

Purpose: Tests the BYPASS flag of the input.

Type: Test

Available for:

☑ Input

Description: Tests the BYPASS flags of the input, and if TRUE (BYPASSED), the commands follow the test condition will be executed.

Parameters: NILL,

See also: IF ENABLED, INPUT TEST CONTROL

Example: The following is example how to use the 'IF BYPASSED` command.

Example:

```
IF INPUT EVENT ON ACCESS      // are we changing to ACCESS
IF BYPASSED                   // is input BYPASSED
   BYPASS OFF                 // Yes, Remove BYPASS
ENDIF                         // ENDIF 'IF BYPASSED'
ENDIF                   // ENDIF 'IF INPUT EVENT ON ACCESS
```

IF ENABLED

Purpose: Tests the BYPASS and ISOLATED flag of the input.

Type: Test

Available for:

☑ Input

Description: Tests the BYPASS and ISOLATED flags of the input, and if FALSE (not BYPASSED and NOT ISOLATED), the commands follow the test condition will be executed.

Parameters: NILL,

See also: IF BYPASSED, INPUT TEST CONTROL

Example: The following is example how to use the ‘IF ENABLED` command.

Example:

```
IF INPUT EVENT ON UNSEAL      // Unseal Event
IF ENABLED                    // is input BYPASSED or ISOLATED
   ALARM ON                   // No, Latch the ALARM
ENDIF                         // ENDIF ‘IF ENABLED’
ENDIF                         // ENDIF ‘IF INPUT EVENT ON UNSEAL
```

IF ISOLATED

Purpose: Tests the ISOLATED flag of the input.

Type: Test

Available for:

☑ Input

Description: Tests the ISOLATED flag of the input, and if TRUE (ISOLATED), the commands follow the test condition will be executed.

Parameters: NILL,

See also: ISOLATE ON, ISOLATE OFF, INPUT TEST CONTROL

Example: The following is example how to use the 'IF ISOLATED` command.

Example:

```
IF INPUT EVENT ON UNSEAL      // Unsealed Event
IF ISOLATED                   // is input ISOLATED
     RETURN                   // Yes, Ignore
ELSE
   ALARM ON                   // No, Latch the ALARM
ENDIF                         // ENDIF 'IF ENABLED'
ENDIF                   // ENDIF 'IF INPUT EVENT ON UNSEAL
```

IF SEALED

Purpose: Tests the SEALED status of the input.

Type: Test

Available for:

☑ Input

Description: Tests the SEALED status of the input, and if TRUE (SEALED), the commands follow the test condition will be executed.

Parameters: NILL,

See also: IF UNSEALED, INPUT TEST CONTROL

Example: The following is example how to use the 'IF SEALED` command. We reset the Alarm latch only, if input is in sealed status.

Example:

```
IF INPUT EVENT ON RESET       // Reset Event
IF SEALED                     // is input SEALED
     ALARM OFF                // Reset ALARM Latch
ENDIF                         // ENDIF 'IF SEALED'
ENDIF                   // ENDIF 'IF INPUT EVENT ON RESET
```

IF UNSEALED

Purpose: Tests the UNSEALED status of the input.

Type: Test

Available for:

☑ Input

Description: Tests the UNSEALED status of the input, and if TRUE (UNSEALED), the commands follow the test condition will be executed.

Parameters: NILL,

See also: IF SEALED, INPUT TEST CONTROL

Example: The following is example how to use the 'IF UNSEALED` command. We sound the Buzzer is unsealed; otherwise we reset the Alarm latch.

Example:

```
IF INPUT EVENT ON RESET       // Reset Event
IF UNSEALED                   // is input UNSEALED
     BUZZER CONTROL,1,1,5     // Sound Buzzer, we unsealed
ELSE                          // we must be SEALED
     ALARM OFF                // Reset ALARM Latch
ENDIF                         // ENDIF 'IF UNSEALED'
ENDIF                   // ENDIF 'IF INPUT EVENT ON RESET
```

IF TAMPER

Purpose: Tests the TAMPER flag of the input.

Type: Test

Available for:

☑ Input

Description: Tests the TAMPER flags of the input, and if TRUE (in TAMPER), the commands follow the test condition will be executed.

Parameters: NILL,

See also: TAMPER ON, TAMPER OFF, INPUT TEST CONTROL

Example: The following is example how to use the 'IF TAMPER` command. We reset the TAMPER Flag, if in TAMPER.

Example:

```
IF INPUT EVENT ON RESET       // Reset Event
IF TAMPER                     // is input in TAMPER ALARM
     TAMPER OFF               // Reset TAMPER Latch
ENDIF                         // ENDIF 'IF TAMPER'
     ALARM OFF                // Reset ALARM flag as well
ENDIF                   // ENDIF 'IF INPUT EVENT ON RESET
```

Example 1:

An Instant alarm is generated, when input is in secure mode and the ***'UNSEAL EVENT`*** *is detected. Then the alarm is latched on (require RESET command to remove the latched alarm).*
We assume, input is set to Secure mode (via an Area) and Script is attached to the appropriate Input.
Please note, that Script STARTS with the ***EVENT TYPE SCRIPT*** *and all required commands are within the* ***EVENT*** *and corresponding* ***ENDIF*** *commands.*

The '//` represents comment line ONLY and is not part of the SCRIPT.

Commands are placed in the INPUT TYPE SCRIPT only

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEAL EVENT
IF ACCESS                     // is input in ACCESS MODE
   RETURN                     // we are in ACCESS, then RETURN
ELSE                          // Otherwise
   SEND DIALLER,1,9999,1,140  // Send Dialler event
   ALARM ON                   // Latch the ALARM
ENDIF                         // ENDIF 'IF ACCESS'
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
```

The first thing is to find out what caused the script to run. An event is what ever triggers the script. In this case, the 'IF INPUT EVENT ON UNSEAL` is the event associated with an input.
The action is the next thing to be programmed. In this case the 'IF ACCESS' command was used.

If input was in ACCESS mode, script will 'RETURN' and ignores any further ACTION.
If input is in SECURE mode (opposite to ACCESS), system will action commands between the 'ELSE` and associated 'ENDIF` statement.
In this case, 'SEND DIALLER` and 'ALARM ON' will be generated.

The above SCRIPT function can be written in number of ways, providing same result. Here is another example:

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEALED EVENT
IF ACCESS                     // is input in ACCESS MODE
   RETURN                     // we are in ACCESS, the RETURN
ENDIF                         // Otherwise
   SEND DIALLER,1,9999,1,140  // Send Dialler event
   ALARM ON                   // Latch the ALARM
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Please note, we have changed the 'ELSE` to 'ENDIF` and the 'ENDIF' after the 'ALARM ON` has been removed.

Or

```
IF INPUT EVENT ON UNSEAL      // did we generated UNSEALED EVENT
IF SECURE                     // is input in SECURE MODE
   SEND DIALLER,1,9999,1,140  // Send Dialler event
   ALARM ON                   // Latch the ALARM
ENDIF                         // Otherwise
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Please note, we have changed the ‘IF ACCESS` to ‘IF SECURE` and the ‘ENDIF’ has been moved after the ALARM ON command. This script is more practical and is easier to understand.

Example 2:

Standard Instant Night Alarm SCRIPT. There are number of EVENTS, which should be followed, to ensure proper operation of SCRIPT.
We have used minimum IF INPUT EVENTS required, and assuming the TAMPER event and Dialler is not USED.

INPUT EVENTS.
IF INPUT EVENT ON RESET
IF INPUT EVENT ON UNSEAL

```
IF INPUT EVENT ON RESET        // did we generated RESET EVENT
     ALARM OFF                 // Reset ALARM LATCH
ENDIF                          // ENDIF 'IF INPUT EVENT ON RESET
IF INPUT EVENT ON UNSEAL       // did we generated UNSEALED EVENT
IF SECURE                      // is input in SECURE MODE
   ALARM ON                    // Latch the ALARM
ENDIF                          //
ENDIF                          // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Example 3:

Standard Instant Night Alarm SCRIPT. There are number of EVENTS, which should be followed, to ensure proper operation of SCRIPT.
We have used minimum IF INPUT EVENTS required, and assuming the TAMPER detection is enabled (see EOL programming), therefore is used, and Dialler is not USED.

INPUT EVENTS.
IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER
IF INPUT EVENT ON UNSEAL

```
IF INPUT EVENT ON RESET        // did we generated RESET EVENT
     ALARM OFF                 // Reset ALARM LATCH
     TAMPER OFF                // Reset Tamper Latch
ENDIF                          // ENDIF 'IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER       // did we generated TAMPER EVENT
   TAMPER ON                   // Latch the ALARM, anytime
ENDIF                          // ENDIF 'IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON UNSEAL       // did we generated UNSEALED EVENT
IF SECURE                      // is input in SECURE MODE
   ALARM ON                    // Latch the ALARM
ENDIF                          //
ENDIF                          // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Example 4:

Standard Instant Night Alarm SCRIPT. There are number of EVENTS, which should be followed, to ensure proper operation of SCRIPT.
We have used minimum IF INPUT EVENTS required, and assuming the TAMPER detection is enabled (see EOL programming), therefore is used, and Dialler is USED.

INPUT EVENTS.
IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER
IF INPUT EVENT ON UNSEAL

```
IF INPUT EVENT ON RESET       // did we generated RESET EVENT
IF ALARMED                    // Are we in ALARM
   ALARM OFF                  // Reset ALARM LATCH
   SEND DIALLER,1,9999,3,140  // Send Restore to Dialler
ENDIF
IF TAMPER                     // Are we in TAMPER
   TAMPER OFF                 // Reset Tamper Latch
   SEND DIALLER,1,9999,3,144  // Send Restore to Dialler
ENDIF
ENDIF                         // ENDIF 'IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER      // did we generated TAMPER EVENT
   TAMPER ON                  // Latch the TAMPER ALARM, anytime
   SEND DIALLER,1,9999,1,144  // Send Dialler Tamper Alarm
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON UNSEAL      // did we generated UNSEALED EVENT
IF SECURE                     // is input in SECURE MODE
   ALARM ON                   // Latch the ALARM
   SEND DIALLER,1,9999,1,140  // Send Dialler Alarm
ENDIF                         //
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
```

Example 5:

Standard Instant Night Alarm SCRIPT. There are number of EVENTS, which should be followed, to ensure proper operation of SCRIPT.
We have used minimum IF INPUT EVENTS required, and assuming the TAMPER detection is enabled (see EOL programming), therefore is used, and Dialler is USED.
We have added ISOLATE and BYPASS events, to ensure proper operation.

INPUT EVENTS.
IF INPUT EVENT ON BYPASS OFF
IF INPUT EVENT ON BYPASS ON
IF INPUT EVENT ON ISOLATE
IF INPUT EVENT ON ISOLATE OFF
IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER
IF INPUT EVENT ON UNSEAL

```
IF INPUT EVENT ON RESET       // did we generated RESET EVENT
IF ALARMED                    // Are we in ALARM
   ALARM OFF                  // Reset ALARM LATCH
   SEND DIALLER,1,9999,3,140  // Send Restore to Dialler
ENDIF
IF TAMPER                     // Are we in TAMPER
   TAMPER OFF                 // Reset Tamper Latch
   SEND DIALLER,1,9999,3,144  // Send Restore to Dialler
ENDIF
ENDIF                         // ENDIF ‘IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER      // did we generated TAMPER EVENT
IF TAMPER ON                  // Are we Tamper alarm
   RETURN                     // so ignore it
ENDIF
   TAMPER ON                  // Latch the TAMPER ALARM, anytime
   SEND DIALLER,1,9999,1,144  // Send Dialler Tamper Alarm
ENDIF                         // ENDIF ‘IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON UNSEAL      // did we generated UNSEALED EVENT
IF ISOLATED                   // Input Isolated
   RETURN                     // So Ignore it
ENDIF
IF SECURE                     // is input in SECURE MODE
IF ALARM ON                   // Are we in alarm
   RETURN                     // Ignore it
ENDIF
   ALARM ON                   // Latch the ALARM
   SEND DIALLER,1,9999,1,140  // Send Dialler Alarm
ENDIF                         //
ENDIF                         // ENDIF ‘IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON ISOLATE ON  // Isolation Event
   ISOLATE ON                 // Set Flag
   SEND DIALLER,1,9999,1,570  // Send Isolation to the Dialler
ENDIF
IF INPUT EVENT ON ISOLATE OFF// Remove Isolation Event
   ISOLATE OFF                // remove Flag
   SEND DIALLER,1,9999,3,570  // Send Restore to the Dialler
ENDIF
IF INPUT EVENT ON BYPASS ON   // Bypass Event
   BYPASS ON                  // Set Flag
   SEND DIALLER,1,9999,1,570  // Send Bypass to the Dialler
ENDIF
IF INPUT EVENT ON BYPASS OFF  // Remove Bypass Event
   BYPASS OFF                 // Remove Flag
```

```
      SEND DIALLER,1,9999,3,570 // Send Restore to the Dialler
   ENDIF
```

Example 6:

Now we create a script, which will operate as for 24 Hour alarm. Note, that we have remove test for input in Access and change the alarm code to the Dialler only.

INPUT EVENTS.
IF INPUT EVENT ON BYPASS OFF
IF INPUT EVENT ON BYPASS ON
IF INPUT EVENT ON ISOLATE
IF INPUT EVENT ON ISOLATE OFF
IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER
IF INPUT EVENT ON UNSEAL

```
IF INPUT EVENT ON RESET       // did we generated RESET EVENT
IF ALARMED                    // Are we in ALARM
   ALARM OFF                  // Reset ALARM LATCH
   SEND DIALLER,1,9999,3,133  // Send Restore to Dialler
ENDIF
IF TAMPER                     // Are we in TAMPER
   TAMPER OFF                 // Reset Tamper Latch
   SEND DIALLER,1,9999,3,144  // Send Restore to Dialler
ENDIF
ENDIF                         // ENDIF 'IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER      // did we generated TAMPER EVENT
IF TAMPER ON                  // Are we Tamper alarm
   RETURN                     // so ignore it
ENDIF
   TAMPER ON                  // Latch the TAMPER ALARM, anytime
   SEND DIALLER,1,9999,1,144  // Send Dialler Tamper Alarm
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON UNSEAL      // did we generated UNSEALED EVENT
IF ISOLATED                   // Input Isolated
   RETURN                     // So Ignore it
ENDIF
IF ALARM ON                   // Are we in alarm
   RETURN                     // Ignore it
ENDIF
   ALARM ON                   // Latch the ALARM
   SEND DIALLER,1,9999,1,133  // Send Dialler Alarm
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON ISOLATE ON  // Isolation Event
   ISOLATE ON                 // Set Flag
   SEND DIALLER,1,9999,1,570  // Send Isolation to the Dialler
ENDIF
IF INPUT EVENT ON ISOLATE OFF// Remove Isolation Event
   ISOLATE OFF                // remove Flag
   SEND DIALLER,1,9999,3,570  // Send Restore to the Dialler
ENDIF
IF INPUT EVENT ON BYPASS ON   // Bypass Event
   BYPASS ON                  // Set Flag
   SEND DIALLER,1,9999,1,570  // Send Bypass to the Dialler
ENDIF
IF INPUT EVENT ON BYPASS OFF  // Remove Bypass Event
   BYPASS OFF                 // Remove Flag
   SEND DIALLER,1,9999,3,570  // Send Restore to the Dialler
ENDIF
```

Example 7:

Now we create a script, which will operate as for 24 Hour NON-LACHED alarms for Industrial application, such a fridge. Note, that we have removed all events for isolation (we do not allow Isolation of this input), reset and Tamper alarm. Note, we have included all the events for BYPASS, remember BYPASS can only be done by Technician code.

INPUT EVENTS.
IF INPUT EVENT ON BYPASS OFF
IF INPUT EVENT ON BYPASS ON
IF INPUT EVENT ON UNSEAL

```
IF INPUT EVENT ON UNSEAL       // did we generated UNSEALED EVENT
   SEND DIALLER,1,9999,1,152  // Send Dialler Alarm
ENDIF                          // ENDIF 'IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON SEAL         // did we generated SEAL EVENT
   SEND DIALLER,1,9999,3,152  // Send Dialler Alarm
ENDIF                          // ENDIF 'IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON BYPASS ON    // Bypass Event
   BYPASS ON                   // Set Flag
   SEND DIALLER,1,9999,1,570  // Send Bypass to the Dialler
ENDIF
IF INPUT EVENT ON BYPASS OFF  // Remove Bypass Event
   BYPASS OFF                  // Remove Flag
   SEND DIALLER,1,9999,3,570  // Send Restore to the Dialler
ENDIF
```

Example 8:

This example provides functionality as EXIT/ENTRY input.

INPUT EVENTS.
IF INPUT EVENT ON BYPASS OFF
IF INPUT EVENT ON BYPASS ON
IF INPUT EVENT ON ISOLATE
IF INPUT EVENT ON ISOLATE OFF
IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER
IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON TIMER EXPIRE
IF INPUT EVENT ON SECURE
IF INPUT EVENT ON ACCESS

```
IF INPUT EVENT ON RESET       // did we generated RESET EVENT
IF ALARMED                    // Are we in ALARM
   ALARM OFF                  // Reset ALARM LATCH
   SEND DIALLER,1,9999,3,134  // Send Restore to Dialler
ENDIF
IF TAMPER                     // Are we in TAMPER
   TAMPER OFF                 // Reset Tamper Latch
   SEND DIALLER,1,9999,3,144  // Send Restore to Dialler
ENDIF
ENDIF                         // ENDIF 'IF INPUT EVENT ON RESET
IF INPUT EVENT ON TAMPER      // did we generated TAMPER EVENT
   TAMPER ON                  // Latch the TAMPER ALARM, anytime
   SEND DIALLER,1,9999,1,144  // Send Dialler Tamper Alarm
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON ACCESS
     STOP INPUT TIMER,1       // Stop Exit Timer
     STOP INPUT TIMER,2       // Stop Entry Timer
ENDIF
IF INPUT EVENT ON SECURE      //Input changed to Secure mode
IF ENABLED                    // Don't Start Timer, if isolated
     START THIS INPUT TIMER, 1, 60 // Start Exit Timer #1
 ENDIF
ENDIF
IF INPUT EVENT ON TIMER EXPIRE      // did timer expired
IF THIS INPUT TIMER EXPIRED,1       // EXIT timer EXPIRED ?
IF UNSEALED                   // Is input still unsealed?
     ALARM ON                 // Yes, Latch the ALARM
     SEND DIALLER,1,9999,1,134      // Send LWS Alarm
ENDIF
ENDIF
IF THIS INPUT TIMER EXPIRED, 2      // ENTRY timer EXPIRED?
     ALARM ON                       // Latch the ALARM
     SEND DIALLER,1,9999,1,134      // Send Alarm
ENDIF
ENDIF
IF INPUT EVENT ON UNSEAL      // did we generated UNSEALED EVENT
IF ENABLED                    // Input Isolated or BYPASSED
IF ACCESS                     // In Access
   RETURN                     // Ignored
ENDIF
IF ALARM ON                   // Are we in alarm
   RETURN                     // Ignored it
ENDIF
```

```
IF THIS INPUT TIMER ON, 1     // We are running Exit
     RETURN
ENDIF
IF THIS INPUT TIMER ON,2      //Did we already started Entry ?
     RETURN
ELSE
     START THIS INPUT TIMER, 2, 30 // Start Entry Timer
ENDIF
ENDIF                         // ENDIF 'IF INPUT EVENT ON UNSEAL
IF INPUT EVENT ON ISOLATE ON  // Isolation Event
   SEND DIALLER,1,9999,1,570  // Send Isolation to the Dialler
ENDIF
IF INPUT EVENT ON ISOLATE OFF// Remove Isolation Event
   SEND DIALLER,1,9999,3,570  // Send Restore to the Dialler
ENDIF
IF INPUT EVENT ON BYPASS ON   // Bypass Event
   SEND DIALLER,1,9999,1,570  // Send Bypass to the Dialler
ENDIF
IF INPUT EVENT ON BYPASS OFF  // Remove Bypass Event
   SEND DIALLER,1,9999,3,570  // Send Restore to the Dialler
ENDIF
```

AREA EVENT COMMANDS:

Area events are automatically generated by the system, when status of the area has changed.

IF AREA EVENT ON ACCESS

Purpose: Generate an Event, when Area has changed to Access mode.

Type: Event

Available for:

☑ Area

Description: Area has changed its status to Access mode. This event is generated automatically by system

Parameters: NILL,

See also: IF AREA EVENT ON SECURE, AREA EVENT SCRIPTS

Example: The following is example how to use the 'IF ARE EVENT ON ACCESS` event. Alarm outputs 1 & 2 are activated on Alarm. When Area changes to Access, we reset outputs to normal.

Example:

```
IF ARE EVENT ON ACCESS
     OUTPUT CONTROL,1,1,0          // Stop Output 1
     OUTPUT CONTROL,2,1,0          // Stop Output 2
ENDIF
IF AREA EVENT ON ALARM
     OUTPUT CONTROL,1,2,600        // Activate output 1
     OUTPUT CONTROL,1,2,0          // Activate output 2
ENDIF
```

IF AREA EVENT ON SECURE

Purpose: Generate an Event, when Area has changed to Secure mode.

Type: Event

Available for:

☑ Area

Description: Area has changed its status to Secure mode. This event is generated automatically by system

Parameters: NILL,

See also: IF AREA EVENT ON ACCESS, AREA EVENT SCRIPTS

Example: The following is example how to use the 'IF ARE EVENT ON SECURE` event. Output 1 on Device 12 indicates status of the Area. Output is active, while area in Access mode.

Example:

```
IF ARE EVENT ON SECURE              // Secure event
     OUTPUT CONTROL,12,1,1,0        // Turn output off
ENDIF
IF AREA EVENT ON ACCESS             // Access event
     OUTPUT CONTROL,12,1,2,0        // Activate output 1
ENDIF
```

IF AREA EVENT ON BYPASS OFF

Purpose: Generate an Event, when all last sector BYPASS status has been removed.

Type: Event

Available for:

☑ Area

Description: All inputs are set to normal, all bypass are removed.

Parameters: NILL,

See also: IF AREA EVENT ON BYPASS ON, AREA EVENT SCRIPTS

Example: The following is example how to use the ‘IF ARE EVENT ON BYPASSED OFF` event. Output 2 on Device 12 is activated, when any input in this area is in bypassed status.

Example:

```
IF AREA EVENT ON BYPASS OFF         // Bypass event
     OUTPUT CONTROL,12,1,0          // Turn output off
ENDIF
IF AREA EVENT ON BYPASS ON          // Bypass event
     OUTPUT CONTROL,12,2,0          // Activate output 1
ENDIF
```

IF AREA EVENT ON BYPASS ON

Purpose: Generate Event, when first input in this area has been Bypassed.

Type: Event

Available for:

☑ Area

Description: One or more inputs are in BYPASS status.

Parameters: NILL,

See also: IF AREA EVENT ON BYPASS OFF, AREA EVENT SCRIPTS

Example: The following is example how to use the 'IF ARE EVENT ON BYPASSED ON` event. Output 2 on Device 12 is activated, when any input in this area is in bypassed status.

Example:

```
IF AREA EVENT ON BYPASS OFF          // Bypass event
     OUTPUT CONTROL,12,1,0           // Turn output off
ENDIF
IF AREA EVENT ON BYPASS ON           // Bypass event
     OUTPUT CONTROL,12,2,0           // Activate output 1
ENDIF
```

IF AREA EVENT ON ISOLATE OFF

Purpose: Indicate Event that last input within this Area, has removed the ISOLATION status.

Type: Event

Available for:

☑ Area

Description: All inputs are set to normal, no input area ISOLATED in this Area.

Parameters: NILL,

See also: IF AREA EVENT ON ISOLATE ON, AREA EVENT SCRIPTS

Example: The following is example how to use the ‘IF ARE EVENT ON ISOLATE OFF` event. Output 3 on Device 12 is activated, when any input in this area is in bypassed status.

Example:

```
IF AREA EVENT ON ISOLATE OFF          // Isolation event
     OUTPUT CONTROL,12,1,0            // Turn output off
ENDIF
IF AREA EVENT ON ISOLATE ON           // Isolation event
     OUTPUT CONTROL,12,2,0            // Activate output 1
ENDIF
```

IF AREA EVENT ON ISOLATE ON

Purpose: Indicate Event that some input within this Area, is in ISOLATED status.

Type: Event

Available for:

☑ Area

Description: One or more inputs in this AREA have been ISOLATED.

Parameters: NILL,

See also: IF AREA EVENT ON ISOLATE OFF, AREA EVENT SCRIPTS

Example: The following is example how to use the 'IF ARE EVENT ON ISOLATE ON` event. Output 3 on Device 12 is activated, when any input in this area is in bypassed status.

Example:

```
IF AREA EVENT ON ISOLATE OFF        // Isolation event
     OUTPUT CONTROL,12,1,0          // Turn output off
ENDIF
IF AREA EVENT ON ISOLATE ON         // Isolation event
     OUTPUT CONTROL,12,2,0          // Activate output 1
ENDIF
```

IF AREA EVENT ON RESET

Purpose: Indicate reset has been generated on alarmed Area.

Type: Event

Available for:

☑ Area

Description: Reset has been generated on this AREA.

Parameters: NILL,

See also: IF AREA EVENT ON ALARM, AREA EVENT SCRIPTS

Example: The following is example how to use the 'IF ARE EVENT ON RESET` event. Alarm outputs 1 & 2 are activated on Alarm. When Area is reset, we reset outputs to normal.

Example:

```
IF ARE EVENT ON RESET
     OUTPUT CONTROL,1,1,0           // Stop Output 1
     OUTPUT CONTROL,2,1,0           // Stop Output 2
ENDIF
IF AREA EVENT ON ALARM
     OUTPUT CONTROL,1,2,600         // Activate output 1
     OUTPUT CONTROL,2,2,0           // Activate output 2
ENDIF
```

IF AREA EVENT ON ALARM

Purpose: Indicate new alarm has been generated of this Area.

Type: Event

Available for:

☑ Area

Description: One or more inputs are in alarm status in this Area.

Parameters: NILL,

See also: IF AREA EVENT ON RESET, IF AREA EVENT ON TAMPER, AREA EVENT SCRIPTS

Example: The following is example how to use the ‘IF ARE EVENT ON ALARM` event. Alarm outputs 1 & 2 are activated on Alarm. When Area is reset, we reset outputs to normal.

Example:

```
IF ARE EVENT ON RESET
     OUTPUT CONTROL,1,1,0          // Stop Output 1
     OUTPUT CONTROL,1,1,0          // Stop Output 2
ENDIF
IF AREA EVENT ON ALARM
     OUTPUT CONTROL,1,2,600  // Activate output 1
     OUTPUT CONTROL,2,2,0          // Activate output 2
ENDIF
```

IF AREA EVENT ON TAMPER

Purpose: Indicate new tamper alarm has been generated of this Area.

Type: Event

Available for:

☑ Area

Description: One or more inputs are in tamper status in this Area.

Parameters: NILL,

See also: IF AREA EVENT ON ALARM, AREA EVENT SCRIPTS

Example: The following is example how to use the 'IF ARE EVENT ON TAMPER` event. Alarm outputs 1 & 2 are activated on Alarm. When Area is reset, we reset outputs to normal.

Example:

```
IF ARE EVENT ON RESET
     OUTPUT CONTROL,1,1,0          // Stop Output 1
     OUTPUT CONTROL,2,1,0          // Stop Output 2
ENDIF
IF AREA EVENT ON TAMPER
     OUTPUT CONTROL,1,2,600        // Activate output 1
     OUTPUT CONTROL,2,2,0          // Activate output 2
ENDIF
```

IF AREA EVENT ON SEAL

Purpose: Indicate Event that all inputs in this Area are in sealed status.

Type: Event

Available for:

☑ Area

Description: All inputs in this area are sealed.

Parameters: NILL,

See also: IF AREA EVENT ON UNSEAL, AREA EVENT SCRIPTS

Example: The following is example how to use the 'IF ARE EVENT ON SEAL` event. Output 4 on Device 12 is activated, when any input in this area is unsealed.

Example:

```
IF AREA EVENT ON SEAL              // Seal event
     OUTPUT CONTROL,12,1,0         // Turn output off
ENDIF
IF AREA EVENT ON UNSEAL            // Unsealed event
     OUTPUT CONTROL,12,2,0         // Activate output 1
ENDIF
```

IF AREA EVENT ON UNSEAL

Purpose: Indicate Event that input in this Area is in unsealed status.

Type: Event

Available for:

☑ Area

Description: One or more inputs in this area are unsealed.

Parameters: NILL,

See also: IF AREA EVENT ON SEAL, AREA EVENT SCRIPTS

Example: The following is example how to use the 'IF ARE EVENT ON UNSEAL` event. Output 4 on Device 12 is activated, when any input in this area is unsealed.

Example:

```
IF AREA EVENT ON SEAL                   // Seal event
     OUTPUT CONTROL,12,1,0              // Turn output 12 off
ENDIF
IF AREA EVENT ON UNSEAL                 // Unsealed event
     OUTPUT CONTROL,12,2,0              // Activate output 12 ON
ENDIF
```

AREA CONTROL COMMANDS:

Area controls allow you to control Area status. Normally these commands will be activated via timer control or any other commands.

ACCESS THIS AREA

Purpose: Set Area status to Access mode on this Area.

Type: Command

Available for:

☑ Area

Description: Change the Area status to Access mode. If any alarms are present, the system generates ‘IF INPUT EVENT ON RESET’ on all alarmed inputs.

Parameters: NILL,

See also: SECURE THIS AREA, RESET THIS AREA, AREA CONTROL COMMANDS

Example: The following is example how to use the ‘ACCESS THIS AREA` command. We have started AREA TIMER, which could be done from any script, and on timer expire, we ACCESS THIS AREA.

Example:

NOTE: This is CLOCK SCRIPT

```
IF CLOCK EVENT ON VALID                    // Clock has changed to VALID Mode
          START AREA TIMER DELAY,1,1,600      // Start AREA 1 Timer 1
ENDIF
```

NOTE: This is an AREA 1 SCRIPT

```
IF AREA EVENT ON TIMER EXPIRE         // did timer Expired?
          ACCESS THIS AREA                    // Access AREA
ENDIF
```

RESET THIS AREA

Purpose: Reset Alarms on this Area.

Type: Control

Available for:

☑ Area

Description: Generates Reset on alarmed Area only.

Parameters: NILL,

See also: ACCESS THIS AREA, SECURE THIS AREA, AREA CONTROL COMMANDS

Example: The following is example how to use the 'RESET THIS AREA` command. When we receive Area alarm, we start 10-minute delay timer. When timer expires, the Area alarm is reset.

Example:

```
IF AREA EVENT ON ALARM              // alarm has been generated
        START AREA TIMER,1,600      // Start 10 min delay
ENDIF
IF AREA EVENT ON TIMER EXPIRE       // did timer expired?
        RESET THIS AREA             // Yes, Generate Area Reset Command
ENDIF
```

SECURE THIS AREA

Purpose: Secure this Area.

Type: Control

Available for:

☑ Area

Description: Change the Area to Secure mode.

Parameters: NILL,

See also: ACCESS THIS AREA, RESET THIS AREA, AREA CONTROL COMMANDS

Example: The following is example how to use the 'SECURE THIS AREA` command. When Area is set to ACCESS MODE, either by user or any other commands, we start 10-minute delay timer and when timer expires, the Area is set to Secure mode. ***Please note: All unsealed inputs are ignored, if any, and the AREA is FORCED to SECURE MODE. If any inputs are unsealed at this time, the alarm condition is generated (depends on the input script).***

Example:

```
IF AREA EVENT ON ACCESS             // Area has been set to ACCESS
        START AREA TIMER,1,600      // Start 10 min delay
ENDIF
IF AREA EVENT ON TIMER EXPIRE       // did timer expired?
        SECURE THIS AREA            // Yes, Secure Area again
ENDIF
```

AREA FLAG CONTROLS:

Each Area includes its own 8 Optional flags. Each flag can be used for your own application.

CLEAR THIS AREA FLAG,x

Purpose: Turns off the OPTIONAL flag for this area.

Type: **Control**

Available for:

☑ Area

Description: Turns off the OPTIONAL flag on this area.

Parameters: Flag number (1-8),

See also: SET THIS AREA FLAG, AREA FLAG CONTROL

Example: The following is example how to use the CLEAR THIS AREA FLAG command.

Example:

```
IF AREA EVENT ON ALARM
        SET THIS AREA FLAG,1          // Set flag 1 to indicate Alarm on this AREA
ENDIF
IF AREA EVENT ON SECURE
        CLEAR THIS AREA FLAG,1        // Clear it,
ENDIF
IF AREA EVENT ON ACCESS               // Input changed to ACCESS
IF AREA FLAG,1                        // Is this Area Alarmed?
        SEND RAS MESSAGE,12,1         // Yes, Send Message to RAS Device 12
ENDIF
ENDIF
```

SET THIS AREA FLAG,x

Purpose: Turns on the OPTIONAL flag for this area.

Type: **Control**

Available for:

☑ Area

Description: Turns on the OPTIONAL flag on this area.

Parameters: Flag number (1-8),

See also: CLEAR THIS AREA FLAG, AREA FLAG CONTROL

Example: The following is example how to use the SET THIS AREA FLAG command.

Example:

```
IF AREA EVENT ON ALARM
         SET THIS AREA FLAG,1          // Set flag 1 to indicate Alarm on this AREA
ENDIF
IF AREA EVENT ON SECURE
          CLEAR THIS AREA FLAG,1       // Clear it,
ENDIF
IF AREA EVENT ON ACCESS                // Input changed to ACCESS
IF AREA FLAG,1                         // Is this Area Alarmed?
         SEND RAS MESSAGE,12,1         // Yes, Send Message to RAS Device 12
ENDIF
ENDIF
```

AREA TEST CONTROL:

Area test control allows you to test status of any of the optional flags.

IF THIS AREA FLAG,x

Purpose: Test the Status of optional Area flags.

Type: **Test**

Available for:

☑ Area

Description: Tests status of the OPTIONAL flag on this area.

Parameters: Flag number (1-8),

See also: CLEAR THIS AREA FLAG, SET THIS AREA FLAG, AREA TEST CONTROL

Example: The following is example how to use the IF AREA FLAG command.

Example:

```
IF AREA EVENT ON ALARM
        SET THIS AREA FLAG,1          // Set flag 1 to indicate Alarm on this AREA
ENDIF
IF AREA EVENT ON SECURE
         CLEAR THIS AREA FLAG,1       // Clear it,
ENDIF
IF AREA EVENT ON ACCESS               // Input changed to ACCESS
IF THIS AREA FLAG,1                   // Is this Area Alarmed?
        SEND RAS MESSAGE,12,1         // Yes, Send Message to RAS Device 12
ENDIF
ENDIF
```

AREA TIMER CONTROL:

We have included up 7 Delay timers to each Area. The timer can be used for controls such as, ACESS AREA for Duration of time only, generate RESET after specific time and many more applications.

START THIS AREA TIMER,timer,duration

Purpose: Starts delay time within the Area script.

Type: **Function**

Available for:

☑ Area

Description: Starts Delay timer for this area and when timer expire, the IF TIMER EVENT ON EXPIRE is generated.

Parameters: Timer number (1-7), duration of time

See also: STOP THIS AREA TIMER, START AREA TIMER DELAY, STOP AREA TIMER DELAY, IF AREA TIMER DELAY ON, IF AREA TIMER EXPIRED, IF AREA TIMER ON, AREA TIMER CONTROL

Example: The following is example how to use the START AREA TIMER command.

Example:

```
IF AREA EVENT ON ALARM
        START THIS AREA TIMER,1,600          // Start Delay timer
ENDIF
IF AREA EVENT ON TIMER EXPIRE          //Any Timer has expired
        RESET THIS AREA                // generate Reset to this Area
ENDIF
```

or

```
IF AREA EVENT ON TIMER EXPIRE          //Any Timer has expired
IF THIS AREA TIMER EXPIRED,1           // timer 1 ??
        RESET THIS AREA                // generate Reset to this Area
ENDIF
ENDIF
```

STOP THIS AREA TIMER,timer

Purpose: Stops delay time for the Area.

Type: **Function**

Available for:

☑ Area

Description: Stops Delay timer for this area.

Parameters: Timer number (1-7),

See also: START THIS AREA TIMER, START AREA TIMER DELAY, STOP AREA TIMER DELAY, IF AREA TIMER DELAY ON, IF AREA TIMER ON, AREA TIMER CONTROL

Example: The following is example how to use the STOP AREA TIMER command. This example will start Delay time for duration of 1 minute. When timer expires, Area is automatically secured. If a user secures the Area prior delay expire, timer is stopped.

Example:

```
IF AREA EVENT ON ACCESS
        START THIS AREA TIMER,1,60  // Start Delay timer for 1 minute
ENDIF
IF AREA EVENT ON SECURE
        STOP THIS AREA TIMER,1      // If area secured via code, ensure timer is
                                    // not running
ENDIF
IF AREA EVENT ON TIMER EXPIRE       // Timer has expired
        SECURE THIS AREA            // Secure this Area
ENDIF
```

IF THIS AREA TIMER ON,timer

Purpose: Tests delay time running.

Type: **Test**

Available for:

☑ Area

Description: Tests, if Area timer is active.

Parameters: Timer number (1-7),

See also: START AREA TIMER, START AREA DELAY TIMER, STOP AREA DELAY TIMER, IF AREA TIMER DELAY ON, AREA TIMER CONTROL

Example: The following is example how to use the 'IF AREA TIMER ON' command. This example will start Delay time for duration of 1 minute. When timer expires, Area is automatically secured. If a user secures the Area prior delay expire, timer is stopped. If alarm is generated in this area, while timer active, activates Buzzer only, otherwise activates Output 1.

Example:

```
IF AREA EVENT ON ACCESS
        START AREA TIMER,1,60          // Start Delay timer for 1 minute
ENDIF
IF AREA EVENT ON SECURE
        STOP AREA TIMER,1              // If area secured via code, ensure timer is
                                       // not running
ENDIF
IF AREA EVENT ON TIMER EXPIRE          // Timer has expired
        SECURE THIS AREA               // Secure this Area
ENDIF
IF AREA EVENT ON ALARM                 //Area Alarm
IF AREA TIMER ON,1
        BUZZER CONTROL,2,3,10          // Activate buzzer, if timer active
ELSE
        OUTPUT CONTROL,1,2,600         // timer not running, activate OP1
ENDIF
ENDIF
```

IF AREA TIMER EXPIRED,timer

Purpose: test, if timer generated expire event.

Type: **Test**

Available for:

☑ Area

Description: Test, if timer generated event expire command.

Parameters: Timer number (1-7),

See also: START AREA TIMER, START AREA TIMER DELAY, STOP AREA TIMER DELAY, IF AREA TIMER DELAY ON, AREA TIMER CONTROL, AREA TIMER TEST CONTROL

Example: The following is example how to use the 'IF AREA TIMER EXPIRE' command.
This example will start Delay time for duration of 1 minute. When timer expires, we test for timer 1, and then the Area is automatically secured.

Example:

```
IF AREA EVENT ON ACCESS
        START AREA TIMER,1,60          // Start Delay timer for 1 minute
ENDIF
IF AREA EVENT ON SECURE
        STOP AREA TIMER,1
ENDIF
IF AREA EVENT ON TIMER EXPIRE          // Timer has expired
IF AREA TIMER EXPIRED,1                // was it the Timer 1 ?
        SECURE THIS AREA               // Secure this Area
ENDIF
ENDIF
```

AREA TEST CONTROL:

Area test control is used to test the status of the area.

Please note: It is recommended; control commands are located within the 'IF AREA EVENT xxx` functions. Remember this command is a TEST not an EVENT command.

IF AREA IN ACCESS CONDITION

Purpose: Test status of the Area

Type: **Test**

Available for:

☑ Area

Description: Tests the status of the Area.

Parameters: NILL,

See also: IF AREA IN SECURE CONDITION, AREA TEST CONTROL

Example: The following is example how to use the IF AREA IN ACCESS CONDITION.
This example will test the status of the Area when alarm is generated.
Output 1 is activated while Area is in access and output 2 while in secure mode.

Example:

```
IF AREA EVENT ON ALARM                 // alarm condition
IF AREA IN ACCESS CONDITION            // Are we in Access mode
        OUTPUT CONTROL,1,2,600         // Yes, activate output 1
ELSE
        OUTPUT CONTROL,2,1,600         // No, activate output 2
ENDIF
ENDIF
```

IF AREA IN SECURE CONDITION

Purpose: Test status of the Area

Type: **Test**

Available for:

☑ Area

Description: Tests the status of the Area.

Parameters: NILL,

See also: IF AREA IN ACCESS CONDITION, AREA TEST CONTROL

Example: The following is example how to use the IF AREA IN SECURE CONDITION.
This example will test the status of the Area when alarm is generated.
Output 1 is activated while Area is in access and output 2 while in secure mode.

Example:

```
IF AREA EVENT ON ALARM              // alarm condition
IF AREA IN SECURE CONDITION         // Are we in Secure mode
        OUTPUT CONTROL,2,1,600      // Yes, activate output 2
ELSE
        OUTPUT CONTROL,1,1,600      // No, activate output 1
ENDIF
ENDIF
```

IF AREA IN TAMPER CONDITION

Purpose: Test status of the Area

Type: **Test**

Available for:

☑ Area

Description: Tests the status of the Area.

Parameters: NILL,

See also: IF AREA IN ALARM CONDITION, AREA TEST CONTROL

Example: The following is example how to use the IF AREA IN SECURE CONDITION.
This example will test the status of the Area when alarm is generated.
Output 1 is activated while Area is in access and output 2 while in secure mode.

Example:

```
IF AREA EVENT ON ALARM                  // alarm condition
IF AREA IN TAMPER CONDITION             // Are we in TAMPER
        OUTPUT CONTROL,2,1,600          // Yes, activate output 2
ELSE
        OUTPUT CONTROL,1,1,600          // No, activate output 1
ENDIF
ENDIF
```

IF AREA IN ALARM CONDITION

Purpose: Test status of the Area

Type: Test

Available for:

☑ Area

Description: Tests the status of the Area.

Parameters: NILL,

See also: IF AREA IN TAMPER CONDITION, AREA TEST CONTROL

Example: The following is example how to use the IF AREA IN ALARM CONDITION.
This example will test the status of the Area, and if in alarm, Area is reset.

Example:

```
IF AREA EVENT ON SEAL                    // All Sealed
IF AREA IN ALARM CONDITION               // Do we have any Alarms
        RESET THIS AREA                  // Yes, Generate Reset
ENDIF
ENDIF
```

IF AREA IN BYPASSED CONDITION

Purpose: Test status of the Area

Type: **Test**

Available for:

☑ Area

Description: Tests the status of the Area.

Parameters: NILL,

See also: IF AREA IN ISOLATED CONDITION, AREA TEST CONTROL

Example: The following is example how to use the IF AREA IN BYPASSED CONDITION.
This example will test the status of the Area when bypassed and Area is secured, warning buzzer is activated.

Example:

```
IF AREA EVENT ON SECURE             // Are we securing
IF AREA IN BYPASSED CONDITION       // Are we in BYPASS
        BUZZER CONTROL,1,3,10       // Yes, activate BUZZER
ENDIF
ENDIF
```

IF AREA IN ISOLATED CONDITION

Purpose: Test status of the Area

Type: Test

Available for:

☑ Area

Description: Tests the status of the Area.

Parameters: NILL,

See also: IF AREA IN BYPASSED CONDITION, AREA TEST CONTROL

Example: The following is example how to use the IF AREA IN ISOLATED CONDITION.
This example will test the status of the Area for Isolation, and when securing, if any inputs are isolated then the buzzer is activated.

Example:

```
IF AREA EVENT ON SECURE              // we are securing
IF AREA IN ISOLATED CONDITION        // Do we have any Input Isolated
        BUZZER CONTROL,1,3,10        // Yes, activate Buzzer
ENDIF
ENDIF
```

IF AREA IN SEALED CONDITION

Purpose: Test status of the Area

Type: **Test**

Available for:

☑ Area

Description: Tests the status of the Area.

Parameters: NILL,

See also: IF AREA IN UNSEALED CONDITION, AREA TEST CONTROL

Example: The following is example how to use the IF AREA IN SEALED CONDITION.
This example, Timer is started on Area alarm condition. On Timer Expiry, we test the UNSEALED status of the AREA, and if SEALED, we reset the ALARM. We can generate reset without the test, but if input(s) are still unsealed, new ALARM will be generated and the siren will be restarted. This way, we do not restart the SIREN, if any area inputs are still unsealed.

Example:

```
IF AREA EVENT ON ALARM              // alarm condition
        START AREA TIMER,1,600      // Start 10 minute timer
        OUTPUT CONTROL,1,2,600      // ACTIVATE the SIREN
ENDIF
IF AREA EVENT ON TIMER EXPIRE
IF AREA IN SEALED CONDITION
        RESET THIS AREA             // Reset Area Alarm
ELSE
        START AREA TIMER,1,60       // Restart the Timer again
ENDIF
ENDIF
IF AREA EVENT ON ACCESS
        STOP AREA TIMER,1           // Stop timer, if running
ENDIF
IF AREA EVENT ON RESET              // Reset condition
        OUTPUT CONTROL,1,1,600      // Stop SIREN
ENDIF
```

IF AREA IN UNSEALED CONDITION

Purpose: Test status of the Area

Type: **Test**

Available for:

☑ Area

Description: Tests the status of the Area.

Parameters: NILL,

See also: IF AREA IN SEALED CONDITION, AREA TEST CONTROL

Example: The following is example how to use the IF AREA IN UNSEALED CONDITION.
This example, Timer is started on Area alarm condition. On Timer Expiry, we test the UNSEALED status of the AREA, and if SEALED, we reset the ALARM. We can generate reset without the test, but if input(s) are still unsealed, new ALARM will be generated and the siren will be restarted. This way, we do not restart the SIREN, if any area inputs are still unsealed.

Example: AREA SCRIPT

```
IF AREA EVENT ON ALARM              // alarm condition
        START AREA TIMER,1,600      // Start 10 minute timer
        OUTPUT CONTROL,1,2,600      // ACTIVATE SIREN
ENDIF
IF AREA EVENT ON TIMER EXPIRE
IF AREA IN UNSEALED CONDITION
        START AREA TIMER,1,60       // Restart the timer again
ELSE
        RESET THIS AREA             // Reset Area Alarm
ENDIF
ENDIF
IF AREA EVENT ON ACCESS
        STOP AREA TIMER,1           // Stop Timer, if is running
ENDIF
IF AREA EVENT ON RESET              // Reset condition
        OUTPUT CONTROL,1,1,600      // Stop SIREN
ENDIF
```

CLOCK EVENTS:

Clock events are always allocated to Clock Script. These Events are generated, when Clock changes to VALID or VOID time zone.

IF CLOCK EVENT TO VOID

Purpose: To allow clock function events, when clock has reached the end time.

Type: Event

Available for:

☑ Clock

Description: Event is automatically generated by the system, when clock has exited the time, as programmed between the Start and End window of the corresponding day of the week. ***Please note: when the Holiday is active, within the Holiday list, the 8th day entry is used.***

Parameters: NILL,

See also: IF CLOCK EVENT TO VALID, CLOCK EVENT SCRIPTS

Example: The following is example how to use the IF CLOCK EVENT TO VOID. This example, Door 1 is unlocked, when clock valid and is relocked when clock is void.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VALID                 // Clock has changed to VALID Mode
          UNLOCK DOOR,1                 // Unlock the door
ENDIF

IF CLOCK EVENT ON VOID                  // Clock has changed to VOID Mode
          SECURE DOOR,1                 // Secure door again
ENDIF
```

IF CLOCK EVENT TO VALID

Purpose: To allow clock function events, when clock has reached the start time.

Type: **Event**

Available for:

☑ Clock

Description: Event is automatically generated by the system, when clock has reached the start time, as programmed between in the Start window of the corresponding day of the week. ***Please note: when the Holiday is active, within the Holiday list, the 8th day entry is used.***

Parameters: NILL,

See also: IF CLOCK EVENT TO VOID, CLOCK EVENT SCRIPTS

Example: The following is example how to use the IF CLOCK EVENT TO VOID. This example, Door 1 is unlocked, when clock valid and is relocked when clock is void.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VALID           // Clock has changed to VALID Mode
        UNLOCK DOOR,1             // Unlock the door
ENDIF

IF CLOCK EVENT ON VOID            // Clock has changed to VOID Mode
        SECURE DOOR,1             // Secure door again
ENDIF
```

DOOR EVENTS:

Door events are always allocated to Door Script. These Events are generated, when Door generated changes in its alarm status.

IF DOOR EVENT ON DOTL

Purpose: The Door Open Too Long (DOTL) alarm has triggered.

Type: Function

Available for:

☑ Door

Description: The Door Open Too Long alarm occurs after a valid card or user has entered but the door has not been closed within the DOTL time.

The DOTL event provides a very convenient method to turn on a local buzzer or to notify the monitoring station. In many applications a timer can also be used to switch between the two responses.

If the door is forced open without a valid card or PIN then this is a Forced Door Alarm (FDA) and not a DOTL.

Parameters:

See also: IF DOOR EVENT ON CLOSED, IF DOOR EVENT ON TAMPER, DOOR EVENT SCRIPTS

Example:
The following script notifies the monitoring company if a DOTL has occurred after hours, otherwise turns on a local buzzer.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON DOTL
IF TIMER ACTIVE,2              // is timer active
   OUTPUT CONTROL,1,2,0        // Activate Output 1
ELSE
   SEND DIALLER,1,123,1,132    // Send Alarm to Dialler
ENDIF
ENDIF
```

IF DOOR EVENT ON CLOSED

Purpose: The Door has been closed, after DOTL or FORCED Door alarm.

Type: Function

Available for:

☑ Door

Description: The Door Closed occurs after a DOTL or FORCED door alarm has been activated.

Parameters: NILL,

See also: IF DOOR EVENT ON DOTL, IF DOOR EVENT ON TAMPER, DOOR EVENT SCRIPTS

Example:
The following script notifies the monitoring company if a DOTL has occurred after hours, otherwise turns on a local buzzer. When door is closed, the buzzer is de-activated.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON DOTL
IF CLOCK VALID,2               // Is Clock VALID
   OUTPUT CONTROL,12,2,0       // Activate Output
ELSE
   SET THIS DOOR FLAG,1        // Indicate, we send Dialler
   SEND DIALLER,1,123,1,132    // Send Alarm to Dialler
ENDIF
ENDIF
IF DOOR EVENT ON CLOSED
IF THIS DOOR FLAG,1
   SEND DIALLER,1,123,3,132    // Send Reset to Dialler
   CLEAR THIS DOOR FLAG,1      // Clear Indication
ENDIF
   OUTPUT CONTROL,12,1,0       // Reset Output anyway
ENDIF
```

IF DOOR EVENT ON FORCED

Purpose: The Door has been opened without any authorization.

Type: Function

Available for:

☑ Door

Description: The Door Forced occurs alarm has been activated.

Parameters: NILL,

See also: IF DOOR EVENT ON DOTL, IF DOOR EVENT ON CLOSED, DOOR EVENT SCRIPTS

Example:
The following script activates alarm, if FORCED DOOR alarm has occurred. Turn on a local SIREN ON and send alarm to Dialler. When door is restored, the Siren is deactivated.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON FORCED
    OUTPUT CONTROL,12,2,0      // Activate Output
    SEND DIALLER,1,123,1,132   // Send Alarm to Dialler
ENDIF
IF DOOR EVENT ON CLOSED
    OUTPUT CONTROL,12,1,0      // Stop Output
    SEND DIALLER,1,123,3,132   // Send Restore to Dialler
ENDIF
```

IF DOOR EVENT ON TAMPER

Purpose: The Door has been TAMPERED with.

Type: Function

Available for:

☑ Door

Description: The Door TAMPER alarm has been activated, when door switch values has changed into tamper (see EOL programming).

Parameters: NILL,

See also: IF DOOR EVENT ON DOTL, IF DOOR EVENT ON CLOSED, DOOR EVENT SCRIPTS

Example:
The following script activates alarm if a TAMPER DOOR alarm has occurred, turns on a local SIREN ON ands sends alarm to the Dialler.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TAMPER
    OUTPUT CONTROL,12,2,0       // Activate Output
    SEND DIALLER,1,123,1,137    // Send Alarm to Dialler
ENDIF
IF DOOR EVENT ON CLOSED
    OUTPUT CONTROL,12,1,0       // Stop Output
    SEND DIALLER,1,123,3,137    // Send Restore to Dialler
ENDIF
```

IF DOOR EQUAL,door

Purpose: Test for Door numbers.

Type: Test

Available for:

☑ Door

Description: The system test for the door number, and if equal, will continue with commands under the command.
Door TIMER EXPIRE is generated on timer expire. When timer expires, the Buzzer is activated on the appropriate device.

Parameters: Door number

See also: IF DOOR RANGE, IF DOOR FLAG, DOOR TEST CONTROL

Example:
The following script shows how to use the 'IF DOOR EQUAL' command.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
IF DOOR EQUAL,1                // Door 1?
    BUZZER CONTROL,12,2,10     // Activate buzzer
ENDIF
IF DOOR EQUAL,2                // Door 2 ?
    BUZZER CONTROL,13,2,10     // Activate buzzer
ENDIF
IF DOOR EQUAL,3                // Door 3 ?
    BUZZER CONTROL,14,2,10     // Activate buzzer
ENDIF
ENDIF
IF DOOR EVENT ON DOTL          // Start timer on DOTL
    START DOOR TIMER,1,5
ENDIF
```

IF DOOR RANGE,door1,door2

Purpose: Test for range of Door numbers.

Type: Test

Available for:

☑ Door

Description: The system test for the range of door numbers, and if equal, will continue with commands under the command.
Door TIMER EXPIRE is generated on timer expire. When timer expires, the Buzzer is activated on common output.

Parameters: Door number

See also: IF DOOR EQUAL, IF DOOR FLAG, DOOR TEST CONTROL

Example:
The following script shows how to use the 'IF DOOR RANGE' command.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
IF DOOR RANGE,1,3            // Door 1,2 and 3 ?
   BUZZER CONTROL,50,2,10     // Activate buzzer
ENDIF
ENDIF
IF DOOR EVENT ON DOTL         // Start timer on DOTL
   START DOOR TIMER,1,5
ENDIF
```

IF DOOR EVENT ON TIMER EXPIRE

Purpose: Time control to the door scripts.

Type: Function

Available for:

☑ Door

Description: The Door TIMER EXPIRE is generated on timer expire. When timer expires, the Buzzer on Device 12 is activated. Timer has been started in different script type.
Please note: Each door can starts up to 7 different timers.

Parameters: NILL,

See also: START DOOR TIMER, STOP DOOR TIMER, IF DOOR TIMER EXPIRED, DOOR EVENT SCRIPTS

Example:
The following script detects the DOOR TIMER EXPIRE and activates Buzzer for 10 seconds.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
   BUZZER CONTROL,12,2,10     // Activate buzzer
ENDIF
```

SET THIS DOOR FLAG,flag

Purpose: Sets flag on this door.

Type: Control

Available for:

☑ Door

Description: The Door flag is set on.

Please note: Each door includes 8 flags.

Parameters: Flag number

See also: CLEAR THIS DOOR FLAG, IF THIS DOOR FLAG, DOOR FLAG CONTROL

Example:
The following script shows how to use the 'SET THIS DOOR FLAG' command.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON DOTL
    SET THIS DOOR FLAG,1        // Set Flag 1
ENDIF
IF DOOR EVENT ON CLOSED
    CLEAR THIS DOOR FLAG,1      // Clear Flag 1
ENDIF
```

CLEAR THIS DOOR FLAG,flag

Purpose: Clear flag on this door.

Type: Control

Available for:

☑ Door

Description: The Door flag is cleared.

Please note: Each door includes 8 flags.

Parameters: Flag number

See also: SET THIS DOOR FLAG, IF THIS DOOR FLAG, DOOR FLAG CONTROL

Example:
The following script shows how to use the 'CLEAR THIS DOOR FLAG' command.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON DOTL
   SET THIS DOOR FLAG,1        // Set Flag 1
ENDIF
IF DOOR EVENT ON CLOSED
   CLEAR THIS DOOR FLAG,1      // Clear Flag 1
ENDIF
```

IF THIS DOOR FLAG,flag

Purpose: Test status flag on this door.

Type: Test

Available for:

☑ Door

Description: The Door flag is tested.

Please note: Each door includes 8 flags.

Parameters: Flag number

See also: SET THIS DOOR FLAG, CLEAR THIS DOOR FLAG, DOOR FLAG CONTROL

Example:
The following script shows how to use the 'IF THIS THIS DOOR FLAG' command.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON DOTL
    SET THIS DOOR FLAG,1        // Set Flag 1
ENDIF
IF DOOR EVENT ON CLOSED
    IF THIS DOOR FLAG,1         // Is flag on
       CLEAR THIS DOOR FLAG,1   // Clear Flag 1
    ENDIF
ENDIF
```

START THIS DOOR TIMER,timer,time

Purpose: Starts door timer for duration of time.

Type: Control

Available for:

☑ Door

Description: The Door TIMER is started and DOOR TIMER EXPIRE is generated on timer expire.

Please note: Each door can starts up to 7 different timers.

Parameters: Timer number and time duration,

See also: IF DOOR EVENT ON TIMER EXPIRE, STOP DOOR TIMER, IF DOOR TIMER EXPIRED, IF THIS DOOR TIMER ON, DOOR TIMER CONTROL

Example:
The following script shows, how to use the START DOOR TIMER' command.
On timer expire; the 'IF DOOR EVENT ON TIMER EXPIRE' event is generated.

Please note: If you use a command 'IF DOOR EQUAL' or 'IF DOOR RANGE` within the script, it may be activated on 'IF DOOR EVENT ON TIMER EXPIRE'.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
   BUZZER CONTROL,12,2,10     // Activate buzzer
ENDIF
IF DOOR EVENT ON DOTL
   START THIS DOOR TIMER,1,30 // Start Timer1,30 seconds
ENDIF
```

STOP THIS DOOR TIMER,timer

Purpose: Stops door timer, if active.

Type: Control

Available for:

☑ Door

Description: The Door TIMER is stopped, if active.

Please note: Each door can starts up to 7 different timers.

Parameters: Timer number,

See also: START THIS DOOR TIMER, IF DOOR EVENT ON TIMER EXPIRE, IF THIS DOOR TIMER EXPIRED, IF THIS DOOR TIMER ON, DOOR TIMER CONTROL

Example:
The following script shows, how to use the STOP DOOR TIMER' command.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
   BUZZER CONTROL,12,2,10     // Activate buzzer
ENDIF
IF DOOR EVENT ON DOTL
   START THIS DOOR TIMER,1,30 // Start Timer1,30 seconds
ENDIF
IF DOOR EVENT ON CLOSED
   STOP THIS DOOR TIMER,1           // Stop timer
ENDIF
```

IF THIS DOOR TIMER ON,timer

Purpose: Test, if door timer is active.

Type: Control

Available for:

☑ Door

Description: Tests if Door TIMER is active.

Please note: Each door can starts up to 7 different timers.

Parameters: Timer number,

See also: START THIS DOOR TIMER, STOP THIS DOOR TIMER, IF DOOR EVENT ON TIMER EXPIRE, IF THIS DOOR TIMER EXPIRED, DOOR TIMER CONTROL

Example:
The following script shows, how to use the ';IF DOOR TIMER ON' command.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
   BUZZER CONTROL,12,2,10     // Activate buzzer
ENDIF
IF DOOR EVENT ON DOTL
IF THIS DOOR TIMER ON,1
   RETURN
ELSE
   START THIS DOOR TIMER,1,30 // Start Timer1,30 seconds
ENDIF
ENDIF
IF DOOR TIMER EVENT ON CLOSED
   STOP THIS DOOR TIMER,1            // Stop timer
ENDIF
```

LOCK ALL ON THIS DOOR

Purpose: Disable all controls to this door.

Type: Control

Available for:

☑ Door

Description: Door controls are disabled. The Egress and Proximity reader control are not able to open the door.

Parameters: NILL,

See also: SECURE THIS DOOR, SECURE DOOR, LOCK ALL ON DOOR, UNLOCK DOOR, LOCK THIS DOOR ENTRY, RELEASE THIS DOOR, UNLOCK THIS DOOR, DOOR CONTROL COMMANDS

Example:
The following script shows, how to use the 'LOCK ALL ON THIS DOOR' command. Door is locked, while CLOCK is VOID and is set to normal operation, when CLOCK is VALID.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VOID
   LOCK ALL ON THIS DOOR       // Lock the DOOR
ENDIF
IF CLOCK EVENT ON VALID
   SECURE THIS DOOR            // Allow normal access
ENDIF
```

LOCK THIS DOOR ENTRY

Purpose: Disable entry to this door.

Type: Control

Available for:

☑ Door

Description: Door controls on the entry are disabled. The Egress control remains operative (if programmed).

Parameters: NILL,

See also: LOCK ALL ON THIS DOOR, SECURE THIS DOOR, SECURE DOOR, RELEASE THIS DOOR, UNLOCK THIS DOOR, UNLOCK DOOR, DOOR CONTROL COMMANDS,

Example:
The following script shows, how to use the 'LOCK THIS DOOR ENTRY' command. Door entry is locked, while CLOCK is VOID and is set to normal operation, when CLOCK is VALID.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VOID
   LOCK THIS DOOR ENTRY        // Lock entry on this DOOR
ENDIF
IF CLOCK EVENT ON VALID
   SECURE THIS DOOR            // Allow normal access
ENDIF
```

RELEASE THIS DOOR

Purpose: Release door for duration of Relay time.

Type: Control

Available for:

☑ Door

Description: Release door.

Parameters: NILL,

See also: LOCK ALL ON THIS DOOR, UNLOCK DOOR, SECURE THIS DOOR, SECURE DOOR, DOOR CONTROL COMMANDS

Please note: Door cannot be RELEASED, if UNLOCKED or LOCKED.

Example:
The following script shows, how to use the 'RELEASE THIS DOOR' command. Door is released, when timer expire. Please note: Timer has been started from different script.

Example: DOOR SCRIPT

```
IF DOOR TIMER EXPIRED
IF DOOR TIMER EXPIRED,1
   RELEASE THIS DOOR         // Open the DOOR
ENDIF
ENDIF
```

SECURE THIS DOOR

Purpose: Return door to normal operation on this door.

Type: Control

Available for:

☑ Door

Description: Door controls are returned to normal operation. The Egress control and Proximity reader operate as normal.

Parameters: NILL,

See also: LOCK ALL ON THIS DOOR, RELEASE THIS DOOR, UNLOCK DOOR, DOOR CONTROL COMMANDS

Example:
The following script shows, how to use the 'LOCK THIS DOOR ENTRY' command. Door entry is locked, while CLOCK is VOID and is set to normal operation, when CLOCK is VALID.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VOID
   LOCK THIS DOOR ENTRY       // Lock entry on this DOOR
ENDIF
IF CLOCK EVENT ON VALID
   SECURE THIS DOOR           // Allow normal access
ENDIF
```

SECURE THIS DOOR

Purpose: Disable all controls to this door.

Type: Control

Available for:

☑ Door

Description: Door controls are disabled. The Egress and Proximity reader control are not able to open the door.

Parameters: NILL,

See also: LOCK ALL ON THIS DOOR, UNLOCK DOOR, LOCK THIS DOOR ENTRY, DOOR CONTROL COMMANDS, GLOBAL DOOR SCRIPTS

Example:
The following script shows, how to use the 'LOCK ALL ON THIS DOOR' command. Door is locked, while CLOCK is VOID and is set to normal operation, when CLOCK is VALID.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VOID
   LOCK ALL ON THIS DOOR      // Lock the DOOR
ENDIF
IF CLOCK EVENT ON VALID
   SECURE THIS DOOR           // Allow normal access
ENDIF
```

UNLOCK THIS DOOR

Purpose: Unlocks to this door.

Type: Control

Available for:

☑ Door

Description: Door is unlocked and all controls are disabled. Door must be return to SECURE DOOR to enable normal operation.

Parameters: NILL,

See also: LOCK ALL ON THIS DOOR, UNLOCK DOOR, LOCK THIS DOOR ENTRY, SECURE THIS DOOR, SECURE DOOR, DOOR CONTROL COMMANDS

Example:
The following script shows, how to use the 'UNLOCK THIS DOOR' command. Door is unlocked, while DOOR TIMER EXPIRE. Please note: Door timer has been started from different script.

Example: DOOR 1 SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
IF DOOR TIMER EXPIRED,1
   UNLOCK THIS DOOR           // Unlock this DOOR
ENDIF
ENDIF
```

Some other script:

```
       START DOOR DELAY TIMER,1,1,30    // start Timer1,Door 1
```

IF THIS DOOR TIMER EXPIRED,timer

Purpose: Test, if door timer generated the expire event.

Type: Test

Available for:

☑ Door

Description: Test, if Door TIMER generated expire event.

Parameters: Timer number,

See also: START THIS DOOR TIMER, STOP THIS DOOR TIMER, IF DOOR EVENT ON TIMER EXPIRE, IF THIS DOOR TIMER ON, DOOR TIMER TEST CONTROL,

Example:

The following script shows, how to use the ‘IF DOOR TIMER EXPIRED’ command.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
IF THIS DOOR TIMER EXPIRED,1   // timer 1 ?
   BUZZER CONTROL,12,2,10      // Activate buzzer
ENDIF
ENDIF
IF DOOR EVENT ON DOTL
   START THIS DOOR TIMER,1,30  // Start Timer1,30 seconds
ENDIF
IF DOOR TIMER EVENT ON CLOSED
   STOP THIS DOOR TIMER,1      // Stop timer
ENDIF
```

LOCAL ACCESS SCRIPT COMMANDS:

Access script commands are always used in the Access Scripts. The Access script is activated n valid entry of the user.

IF ACCESS DOOR EQUAL,door

Purpose: Tests door number in the Access script.

Type: Test

Available for:

☑ Access

Description: The Door is tested on valid entry of the User.

Parameters: Door,

See also: IF ACCESS DOOR RANGE, ACCESS TEST CONTROL

Example:
The following script indicates how to use the 'IF ACCESS DOOR EQUAL' command.

Example: ACCESS SCRIPT

```
IF ACCESS DOOR EQUAL,1
   BUZZER CONTROL,2,2,10      // Activate buzzer on
                              // Device 2
ENDIF
```

IF ACCESS DOOR RANGE,door1,door2

Purpose: Tests range of doors in the Access script.

Type: Test

Available for:

☑ Access

Description: Range of Doors is tested on valid user entry.

Parameters: Door,

See also: IF ACCESS DOOR EQUAL, ACCESS TEST CONTROL

Example:
The following script indicates how to use the 'IF ACCESS DOOR RANGE' command. If valid user code/card is entered on doors 1,2,3 or 4, the buzzer is activated.

Please note: The Door 1 value must be smaller that the Door 2.

Example: ACCESS SCRIPT

```
IF ACCESS DOOR RANGE,1,4
   BUZZER CONTROL,2,2,10       // Activate buzzer on
                               // Device 2
ENDIF
```

IF GROUP EQUAL, group

Purpose: Tests the group number of the user in the Access script.

Type: Test

Available for:

☑ Access

Description: The user group is tested on valid entry of the User.

Parameters: Group,

See also: IF GROUP RANGE, ACCESS TEST CONTROL

Example:
The following script indicates how to use the 'IF GROUP EQUAL' command. If user belongs to Group 2, buzzer is activated.

Example: ACCESS SCRIPT

```
IF GROUP EQUAL, 2
   BUZZER CONTROL, 2,2,10     // Activate buzzer on
                              // Device 2
ENDIF
```

IF GROUP RANGE, group1, group2

Purpose: Tests the range of groups of the user in the Access script.

Type: Test

Available for:

☑ Access

Description: The user group is tested on valid entry of the User.

Parameters: Group,

See also: IF GROUP EQUAL, ACCESS TEST CONTROL

Example:
The following script indicates how to use the ‘IF GROUP RANGE’ command. If user belongs to Group 2,3 OR 4, buzzer is activated.

Please note: The Group 1 value must be smaller that the Group 2.

Example: ACCESS SCRIPT

```
IF GROUP RANGE, 2,4
   BUZZER CONTROL, 2,2,10     // Activate buzzer on
                              // Device 2
ENDIF
```

IF USER EQUAL, user

Purpose: Tests the user number in the Access script.

Type: Test

Available for:

☑ Access

Description: The user group is tested on valid entry of the User.

Parameters: User number,

See also: IF USER RANGE, ACCESS TEST CONTROL

Example:
The following script indicates how to use the 'IF USER EQUAL' command. If user 100 has been used, the buzzer is activated.

Please note: Ensure User 100 is allocated to correct Group and Group uses this Access script.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 100
   BUZZER CONTROL, 2,2,10     // Activate buzzer on
                              // Device 2
ENDIF
```

IF USER RANGE, user1, user2

Purpose: Tests range of user numbers in the Access script.

Type: Test

Available for:

☑ Access

Description: The user group is tested on valid entry of the User.

Parameters: User1, User2

See also: IF USER EQUAL, ACCESS TEST CONTROL

Example:
The following script indicates how to use the 'IF USER RANGE' command. If user 100 or 101 has been entered, the buzzer is activated.

Please note: The User 1 value must be smaller that the User 2. Ensure User 100 and 101 are allocated to correct Group and Group uses this Access script.

Example: ACCESS SCRIPT

```
IF USER RANGE, 100,101
   BUZZER CONTROL, 2,2,10     // Activate buzzer on
                              // Device 2
ENDIF
```

IF THIS GROUP FLAG, flag

Purpose: Tests flag of group number in the Access script.

Type: Test

Available for:

☑ Access

Description: The user group is tested on valid entry of the User.

Parameters: Group flag,

See also:RESET THIS GROUP FLAG, SET THIS GROUP FLAG, ACCESS TEST CONTROL

Example:
The following script indicates how to use the 'IF THIS GROUP FLAG' command. If user 100 or 101 has been entered and Flag 1 is set, then the buzzer is activated. The Group flag could be set in different script. For example, flag has been previously set or reset in the CLOCK script.

Please note: The Group flag is used on Group, where this script is allocated. You cannot test for flag 9, suspended flag; script will not be accessed, as suspended error will be generated.

Example: ACCESS SCRIPT

```
IF USER RANGE, 100,101
IF THIS GROUP FLAG, 1
    BUZZER CONTROL, 2,2,10     // Activate buzzer on
                               // Device 2
ENDIF
ENDIF
```

IF THIS USER FLAG, flag

Purpose: Tests flag of the user number in the Access script.

Type: Test

Available for:

☑ Access

Description: The user group is tested on valid entry of the User.

Parameters: User flag,

See also: IF THIS GROUP FLAG, RESET THIS USER FLAG, SET THIS USER FLAG, ACCESS FLAGS CONTROL

Example:
The following script indicates how to use the 'IF USER FLAG' command. If user 10 or 11 has been entered and Flag 1 is set, then the buzzer is activated. The User flag could be set in different script. For example, flag has been previously set and reset in the CLOCK script.

Please note: The User flag is used on User generated this script. This script must be allocated to Group, where this user is allocated. You cannot test for flag 9, suspended flag; script will not be accessed, as suspended error will be generated.

Example: ACCESS SCRIPT

```
IF USER RANGE, 10,11
IF THIS USER FLAG, 1
    BUZZER CONTROL, 2,2,10      // Activate buzzer on
                                // Device 2
ENDIF
ENDIF
```

RESET THIS GROUP FLAG, flag

Purpose: Clears flag of the group number in the Access script.

Type: Control

Available for:

☑ Access

Description: The group flag is reset.

Parameters: Group flag,

See also: SET THIS GROUP FLAG, IF THIS GROUP FLAG, RESET THIS GROUP FLAG, ACCESS FLAGS CONTROL

Example:
The following script indicates how to use the 'RESET THIS GROUP FLAG' and 'SET THIS GROUP FLAG' commands. If user 10 enters Door 1, Flag 1 of the Group is reset. When user 10 enters Door 2, the Group flag is set.

Please note: The Group flag is used on Group generated this script. This script must be allocated to group, where this user is allocated. You cannot set or reset flag 9, suspended flag; script will not be accessed, as suspended error will be generated.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 10
IF DOOR EQUAL, 1
   RESET THIS GROUP FLAG, 1   // Reset Flag, if Door 1
used
ENDIF
IF DOOR EQUAL, 2
   SET THIS GROUP FLAG, 1     // Set Flag, if Door 2 used
ENDIF
ELSE                        // Any other USERS?,
                            // so sound buzzer, if flag set
IF THIS GROUP FLAG, 1
   BUZZER CONTROL, 2,2,10     // Activate buzzer on
                              // Device 2, if flag set
ENDIF
ENDIF
```

SET THIS GROUP FLAG, flag

Purpose: Sets flag of the group number in the Access script.

Type: Control

Available for:

☑ Access

Description: The group flag is set.

Parameters: Group flag,

See also: RESET GROUP FLAG, IF GROUP FLAG, CLEAR GROUP FLAG, ACCESS FLAGS CONTROL

Example:
The following script indicates how to use the 'RESET THIS GROUP FLAG' and 'SET THIS GROUP FLAG' commands. If user 10 enters Door 1, Flag 1 of the Group is reset. When user 10 enters Door 2, the Group flag is set.

Please note: The Group flag is used on Group generated this script. This script must be allocated to group, where this user is allocated. You cannot set or reset flag 9, suspended flag; script will not be accessed, as suspended error will be generated.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 10
IF DOOR EQUAL, 1
    RESET THIS GROUP FLAG, 1   // Reset Flag, if Door 1
used
ENDIF
IF DOOR EQUAL, 2
    SET THIS GROUP FLAG, 1     // Set Flag, if Door 2 used
ENDIF
ELSE
IF THIS GROUP FLAG, 1
    BUZZER CONTROL, 2,2,10     // Activate buzzer on
                               // Device 2, if flag set
ENDIF
ENDIF
```

RESET THIS USER FLAG, flag

Purpose: Clears flag of the user in the Access script.

Type: Control

Available for:

☑ Access

Description: The User flag is cleared.

Parameters: User flag,

See also: IF THIS USER FLAG, SET THIS USER FLAG, ACCESS FLAGS CONTROL

Example:
The following script indicates how to use the 'RESET USER FLAG' and 'SET USER FLAG' commands. If user 10 enters Door 1, Flag 1 of the User is reset. When user 10 enters Door 2, the User flag is set.

Please note: The User flag is used on User generated this script. You cannot set or reset flag 9, suspended flag; script will not be accessed, as suspended error will be generated.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 10
IF DOOR EQUAL, 1
   RESET THIS USER FLAG, 1     // Reset Flag, if Door 1
used
ENDIF
IF DOOR EQUAL, 2
   SET THIS USER FLAG, 1       // Set Flag, if Door 2 used
ENDIF
ENDIF
```

SET THIS USER FLAG, flag

Purpose: Sets flag of the user in the Access script.

Type: Control

Available for:

☑ Access

Description: The User flag is set.

Parameters: User flag,

See also: RESET THIS USER FLAG, IF THIS USER FLAG, ACCESS FLAGS CONTROL

Example:
The following script indicates how to use the ‘RESET THIS USER FLAG’ and ‘SET THIS USER FLAG’ commands. If user 10 enters Door 1, Flag 1 of the User is reset. When user 10 enters Door 2, the User flag is set.

Please note: The User flag is used on User generated this script. You cannot set or reset flag 9, suspended flag; script will not be accessed, as suspended error will be generated.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 10
IF DOOR EQUAL, 1
    RESET THIS USER FLAG, 1    // Reset Flag, if Door 1
used
ENDIF
IF DOOR EQUAL, 2
    SET THIS USER FLAG, 1      // Set Flag, if Door 2 used
ENDIF
ENDIF
```

ACCESS AREA, area

Purpose: Disarms the area.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Provides the ability to disarm any nominated area.

Parameters: Area number.

See also: SECURE AREA, RESET AREA, AREA GLOBAL SCRIPTS

Example:
The following CLOCK script will automatically arm and disarm area 2. It would be associated with a clock that would automatically arm and disarm the area at the specific times. Changing the clock times would change when the area is armed and disarmed.

This script would be useful for allowing public access during the day, or disarming automatically for cleaners at certain times. Note that a user still can operate the area from RAS or TDC.

Note: When script includes the command SECURE AREA, system will ignore any unsealed inputs, if any, and will force the area to requested mode. If any inputs are unsealed, the appropriate script event will be activated.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT TO VALID
   ACCESS AREA, 2
ENDIF
IF CLOCK EVENT TO VOID
   SECURE AREA, 2
ENDIF
```

The script could be modified so as to only disarm the area by removing the other timer condition. For example, a customer may require area 2 to automatically disarm in the morning but they must be manually armed in the evening.

```
IF CLOCK EVENT TO VALID
   ACCESS AREA, 2
ENDIF
```

SECURE AREA, area

Purpose: Secures (arms) an area.

Type: Control

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: Arms the nominated area. If the area is unsealed then the area will be secure.

Parameters: Area number.

See also: ACCESS AREA, RESET AREA, AREA GLOBAL SCRIPTS

Example:

The following clock script will automatically arm and disarm area 2. It would be associated with a clock that would automatically arm and disarm the area. Changing the clock time would change when the area is armed and disarmed.

This script would be useful for allowing public access during the day, or disarming automatically for cleaners at certain times. Note that a user still can operate the area from RAS or TDC.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT TO VALID
        ACCESS AREA, 2
ENDIF
IF CLOCK EVENT TO VOID
        SECURE AREA, 2
ENDIF
```

The script could be modified so as to only arm the area by removing the other timer condition. For example a customer may require area 2 to be automatically armed at night if they have not already been armed.

```
IF CLOCK EVENT TO VOID
        SECURE AREA, 2
ENDIF
```

RESET AREA, area

Purpose: Resets the alarmed area.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: If an area is in alarm, it will generate reset command to all inputs associated with this area. If all inputs within the area are sealed then the alarm will be removed from the system. If any inputs are unsealed, and the ALARM ON flag is set, and then it will return to the alarm state.

Parameters: Area number.

See also: SECURE AREA, ACCESS AREA, AREA GLOBAL SCRIPTS

Example:
The following INPUT SCRIPT will reset three areas when the input is unsealed. This input could be connected to external key switch.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEALED
IF AREA IN ALARM, 3
        RESET AREA, 3
ENDIF
IF AREA IN ALARM, 4
        RESET AREA, 4
ENDIF
IF AREA IN ALARM, 5
        RESET AREA, 5
ENDIF
ENDIF
```

Or, we could ignore the test for area alarm condition, as reset will not be generated, if area is not in alarm. This command structure uses less Script memory.

```
IF INPUT EVENT ON UNSEALED
        RESET AREA, 3
        RESET AREA, 4
        RESET AREA, 5
ENDIF
```

CLEAR AREA FLAG, area, flag

Purpose: Resets the area flag.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: If an area flag is set, it will be reset.

Parameters: Area and flag number.

See also: SET AREA FLAG ON, AREA GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will reset the area 1 flag, if set, and generate log event. Flag is set, when area generated an alarm, while in access mode.

Example: AREA SCRIPT

```
IF AREA EVENT ON SECURE
IF AREA FLAG ON, 1,1
        LOG EVENT 2
        CLEAR AREA FLAG, 1,1
ENDIF
ENDIF
IF AREA EVENT ON ALARM
IF AREA IN ACCESS, 1
        SET AREA FLAG, 1,1
ENDIF
ENDIF
```

SET AREA FLAG ON, area, flag

Purpose: Sets the area flag.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Sets the area flag.

Parameters: Area and flag number.

See also: CLEAR AREA FLAG, AREA GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will reset the area 1 flag, if set, and generate log event. Flag is set, when area generated an alarm, while in access mode. The Log event will include the additional value of 2, which can indicate, that the Area been in alarm condition.

Example: AREA SCRIPT

```
IF AREA EVENT ON SECURE
IF AREA FLAG ON, 1,1
        LOG EVENT 2
        CLEAR AREA FLAG, 1,1
ENDIF
ENDIF
IF AREA EVENT ON ALARM
IF AREA IN ALARM, 1
        SET AREA FLAG ON, 1,1
ENDIF
ENDIF
```

IF AREA DELAY TIMER ON, area, timer

Purpose: Test the area timer.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if the area timer is active.

Parameters: Area and timer number.

See also: START AREA TIMER DELAY, STOP AREA TIMER DELAY, START AREA TIMER, STOP AREA TIMER, IF AREA TIMER EXPIRED, AREA GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will start the area timer. When timer expires, we set area to secure. If area is secured, prior it's timer expire, the timer is stopped.
Please note: we assume, this script is associated with area 1 only.

Example: AREA SCRIPT

```
IF AREA EVENT ON TIMER EXPIRE
        SECURE AREA, 1
ENDIF
IF AREA EVENT ON SECURE
IF AREA TIMER DELAY ON, 1,1
        STOP AREA TIMER DELAY, 1,1
ENDIF
ENDIF
IF AREA EVENT ON ACCESS
        START AREA TIMER DELAY, 1,1
ENDIF
```

IF AREA IN ACCESS, area

Purpose: Test the area status.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if the area is in access.

Parameters: Area number.

See also: IF AREA IN ALARM, IF AREA IN TAMPER, IF AREA IN TAMPER, IF AREA IS BYPASSED, IF AREA IS ISOLATED, IF AREA IS SEALED, IF AREA IS UNSEALED, AREA GLOBAL SCRIPTS

Example:
The following INPUT SCRIPT will secure area 1, if in access.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
        IF AREA IN ACCESS,1              // Is area in access ?
                SECURE AREA, 1
        ENDIF
ENDIF
```

IF AREA IN ALARM, area

Purpose: Test the area status.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if the area is in alarm.

Parameters: Area number.

See also: IF AREA IN ACCESS, IF AREA IN TAMPER, IF AREA IN TAMPER, IF AREA IS BYPASSED, IF AREA IS ISOLATED, IF AREA IS SEALED, IF AREA IS UNSEALED, AREA GLOBAL SCRIPTS

Example:
The following INPUT SCRIPT will reset area 1, if in alarm.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
        IF AREA IN ALARM,1              // Is area in ALARM ?
                RESET AREA, 1
        ENDIF
ENDIF
```

IF AREA IN SECURE, area

Purpose: Test the area status.

Type: Test

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: Tests if the area is in secure.

Parameters: Area number.

See also: IF AREA IN ACCESS, IF AREA IN ALARM, IF AREA IN TAMPER, IF AREA IS BYPASSED, IF AREA IS SEALED, IF AREA IS UNSEALED, AREA GLOBAL SCRIPTS

Example:
The following INPUT SCRIPT will access area 1, if in secure.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
        IF AREA IN SECURE,1          // Is area in secure ?
                ACCESS AREA, 1
        ENDIF
ENDIF
```

IF AREA IN TAMPER, area

Purpose: Test the area status.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if the area is in tamper alarm.

Parameters: Area number.

See also: IF AREA IN ACCESS, IF AREA IN ALARM, IF AREA IN SECURE, IF AREA IS BYPASSED, IF AREA IS ISOLATED, IF AREA IS SEALED, IF AREA IS UNSEALED, AREA GLOBAL SCRIPTS

Example:

The following INPUT SCRIPT will reset area 1, if in tamper alarm.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
        IF AREA IN TAMPER,1             // Is area in tamper ?
                RESET AREA, 1
        ENDIF
ENDIF
```

IF AREA IS BYPASSED, area

Purpose: Test the area status.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if the area is in bypassed mode.

Parameters: Area number.

See also: IF AREA IN ACCESS, IF AREA IN ALARM, IF AREA IN SECURE, IF AREA IN TAMPER, IF AREA IS ISOLATED, IF AREA IS SEALED, IF AREA IS UNSEALED, AREA GLOBAL SCRIPTS

Example:

The following AREA SCRIPT will sound buzzer, if area is set to access mode and area includes inputs in bypassed mode.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS
        IF AREA IS BYPASSED,1                   // Is area in tamper ?
                BUZZER CONTROL,1,3,10           // Sound Buzzer
        ENDIF
ENDIF
```

IF AREA IS ISOLATED, area

Purpose: Test the area status.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if the area is in isolated mode.

Parameters: Area number.

See also: IF AREA IN ACCESS, IF AREA IN ALARM, IF AREA IN SECURE, IF AREA IN TAMPER, IF AREA IS SEALED, IF AREA IS UNSEALED, AREA GLOBAL SCRIPTS

Example:

The following AREA SCRIPT will sound buzzer, if area is set to access mode and area includes inputs in isolated mode.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS
        IF AREA IS ISOLATED,1                   // Has area isolated inputs ?
                BUZZER CONTROL,1,3,10           // Sound Buzzer
        ENDIF
ENDIF
```

IF AREA IS SEALED, area

Purpose: Test the area status.

Type: Test

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: Tests if the area is in sealed condition.

Parameters: Area number.

See also: IF AREA IN ACCESS, IF AREA IN ALARM, IF AREA IN SECURE, IF AREA IN TAMPER, IF AREA IS UNSEALED, AREA GLOBAL SCRIPTS

Example:

The following AREA SCRIPT will sound buzzer, if area is set to access mode and all inputs in the area area sealed.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS
        IF AREA IS SEALED,1                 // Area all inputs sealed ?
                BUZZER CONTROL,1,3,10       // Sound Buzzer
        ENDIF
ENDIF
```

IF AREA IS UNSEALED, area

Purpose: Test the area status.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if the area is in unsealed condition.

Parameters: Area number.

See also: IF AREA IN ACCESS, IF AREA IN ALARM, IF AREA IN SECURE, IF AREA IN TAMPER, IF AREA IS SEALED, AREA GLOBAL SCRIPTS

Example:

The following AREA SCRIPT will sound buzzer, if area is set to access mode and any inputs in the area area unsealed.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS
        IF AREA IS UNSEALED,1                // Area any inputs unsealed ?
                BUZZER CONTROL,1,3,10        // Sound Buzzer
        ENDIF
ENDIF
```

STOP AREA DELAY TIMER, area, timer

Purpose: Stops area timer.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Stops the area timer if active.

Parameters: Area and timer number.

See also: START AREA DELAY TIMER, , START AREA TIMER, STOP AREA TIMER, AREA CONTROL, IF AREA TIMER EXPIRED, AREA GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will start the area timer. When timer expires, we set area to secure. If area is secured, prior it's timer expire, the timer is stopped.
Please note: we assume, this script is associated with area 1 only.

Example: AREA SCRIPT

```
IF AREA EVENT ON TIMER EXPIRE
        SECURE AREA,1
ENDIF
IF AREA EVENT ON SECURE
IF AREA DELAY TIMER ON,1,1
        STOP AREA DELAY TIMER,1,1
ENDIF
ENDIF
IF AREA EVENT ON ACCESS
        START AREA DELAY TIMER,1,1
ENDIF
```

START AREA DELAY TIMER, area, timer, duration

Purpose: Starts area timer.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Starts the area timer.

Parameters: Area , area timer number and duration time.

See also: STOP AREA DELAY TIMER, START AREA TIMER, STOP AREA TIMER, IF AREA TIMER EXPIRED, AREA CONTROL, AREA GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will start the area timer. When timer expires, we set area to secure. If area is secured, prior it's timer expire, the timer is stopped.
Please note: we assume, this script is associated with area 1 only.

Example: AREA SCRIPT

```
IF AREA EVENT ON TIMER EXPIRE
        SECURE AREA, 1
ENDIF
IF AREA EVENT ON SECURE
IF AREA DELAY TIMER, 1,1
        STOP AREA TIMER DELAY, 1,1
ENDIF
ENDIF
IF AREA EVENT ON ACCESS
        START AREA DELAY TIMER, 1,1, 50
ENDIF
```

BUZZER CONTROL, device, action, time

Purpose: Activates or deactivates buzzer on requested device.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Activates or deactivates buzzer for selected action and duration of time.

Parameters: Device, action and duration of time.

See also: OUTPUT CONTROL, OLIST CONTROL, IF OUTPUT ON, OUTPUT GLOBAL SCRIPTS

Example:
The following DOOR SCRIPT will activate buzzer for 10 seconds, when DOTL alarm is activated.
Please note: we assume, this script is associated with device 5 (TDC) only. If buzzer is set to off action and duration timer is used, after timer expiry, buzzer is set to on status.

Action codes:

0 – not used
1 – off
2 – Constant ON
3 – slow pulse
4 – fast pulse

Example: DOOR SCRIPT

```
IF DOOR EVENT ON DOTL
        BUZZER CONTROL, 5,3,10
ENDIF
```

OUTPUT CONTROL, output, action, time

Purpose: Activates or deactivates selected output.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Activates or deactivates output for selected action and duration of time.

Parameters: Output, action and duration of time.

See also: BUZZER CONTROL, OLIST CONTROL, IF OUTPUT ON, OUTPUT GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will activate output 1 for duration of 10 minutes, when an area alarm is generated. If area alarm is reset, the output is turned off.

Please note: If output is set to off action and duration timer is used, after timer has expired, the output is set to on status. Door device cannot use this command.

Action codes:

0 – not used
1 – off
2 – slow pulse
3 – fast pulse

Example: AREA SCRIPT

```
IF AREA EVENT ON ALARM
        OUTPUT CONTROL,1,2,600      // Activate output 1, constant, 600 secs
ENDIF
IF AREA EVENT ON RESET
        OUTPUT CONTROL,1,1,0        // Stop Output
ENDIF
```

IF OUTPUT ON, output

Purpose: Tests the status of the output.

Type: Test

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: Output is tested for present condition.

Parameters: Output number

See also: BUZZER CONTROL, OLIST CONTROL, OUTPUT CONTROL, OUTPUT GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will activate output 1 for duration of 10 minutes, when an area alarm is generated. If area alarm is reset, the output is turned off. When area is set to secure, we test status of the output, and if still active, we turn it off..

Example: AREA SCRIPT

```
IF AREA EVENT ON ALARM
        OUTPUT CONTROL,1,2,600      // Activate output 1, constant, 600 secs
ENDIF
IF AREA EVENT ON RESET
        OUTPUT CONTROL,1,1,0        // Stop Output
ENDIF
IF AREA EVENT ON SECURE
        IF OUTPUT ON,1              // Is output ON
                OUTPUT CONTROL,1,1,0        // Turn it off
        ENDIF
ENDIF
```

OLIST CONTROL, olist, action, time

Purpose: Activates or deactivates group of outputs.

Type: Control

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: Activates or deactivates LIST OF OUTPUTS, AS PROGRAMMED WITHIN THE OLIST, for selected action and duration of time.

Parameters: Olist, action and duration of time.

See also: OUTPUT CONTROL, BUZZER CONTROL, IF OUTPUT ON, OUTPUT GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will activate OLIST 1 for duration of 10 minutes, when an area alarm is generated. If area alarm is reset, the outputs are turned off.

Please note: If outputs are set to off action and duration timer is used, after timer has expired, the output is set to on status.

Action codes:

0 – not used
1 – off
2 – Constant ON
3 – slow pulse
4 – fast pulse

Example: AREA SCRIPT

```
IF AREA EVENT ON ALARM
        OLIST CONTROL, 1,2,600        // Activate ALL outputs in OLIST 1
ENDIF
IF AREA EVENT ON RESET
        OLIST CONTROL, 1,1,0                // Deactivate all Outputs in
OLIST 1
ENDIF
```

BYPASS INPUT, input

Purpose: Set selected input into bypassed mode.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Select input into requested status.

Parameters: Input number**,**

See also: REMOVE BYPASS, INPUT GLOBAL SCRIPTS

Example:
The following CONTROL SCRIPT will test for command 1. If true, input 1 will be set to bypassed mode.

Example: CONTROL SCRIPT

```
IF CONTROL EQUAL, 1
        BYPASS INPUT, 1                // Bypass input 1
ENDIF
IF CONTROL EQUAL, 2
        REMOVE BYPASS, 1               // Remove Bypass for input 1
ENDIF
```

REMOVE BYPASS, input

Purpose: Removes selected input from bypassed mode.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Removes input from requested status.

Parameters: Input number**,**

See also: BYPASS INPUT, INPUT GLOBAL SCRIPTS

Example:
The following CONTROL SCRIPT will test for command 2. If true, bypass EVENT will be generated on this input.

Please note: DO not use SEND DIALLER COMMAND within CONTROL SCRIPT. Once the BYPASS INPUT or other global commands are used, system will generate the appropriate EVENT command, therefore the SEND DIALLER should be located in the corresponding script (e.g., INPUT SCRIPT in this example).

Example: CONTROL SCRIPT

```
IF CONTROL EQUAL, 1
        BYPASS INPUT, 1
ENDIF
IF CONTROL EQUAL, 2
        REMOVE BYPASS, 1
ENDIF
```

ISOLATE INPUT, input

Purpose: Allows isolation of an input from any scripts.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Generate ISOLATE ON EVENT on selected input.

Parameters: Input number**,**

See also: REMOVE ISOLATION, INPUT GLOBAL SCRIPTS

Example:
The following CONTROL SCRIPT will test for command 3. If true, input will generate the ISOLATION ON EVENT.

Please note: DO not use SEND DIALLER COMMAND within CONTROL SCRIPT. Once the BYPASS INPUT or other global commands are used, system will generate the appropriate EVENT command, therefore the SEND DIALLER should be located in the corresponding script (e.g., INPUT SCRIPT in this example).

Example: CONTROL SCRIPT

```
IF CONTROL EQUAL, 3
        ISOLATE INPUT, 1
ENDIF
```

REMOVE ISOLATION, input

Purpose: Removes isolation of an input from any scripts.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Generate ISOLATE OFF on selected input.

Parameters: Input number**,**

See also: ISOLATE INPUT, INPUT GLOBAL SCRIPTS

Example:
The following CONTROL SCRIPT will test for command 4. If true, input will generate the ISOLATION OFF EVENT.

Please note: DO not use SEND DIALLER COMMAND within CONTROL SCRIPT. Once the BYPASS INPUT or other global commands are used, system will generate the appropriate EVENT command, therefore the SEND DIALLER should be located in the corresponding script (e.g., INPUT SCRIPT in this example).

Example: CONTROL SCRIPT

```
IF CONTROL EQUAL, 4
        REMOVE ISOLATION, 1
ENDIF
```

CLEAR INPUT FLAG, input, flag

Purpose: Clears input flag on selected input from any scripts.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Clears flag on selected input.

Parameters: Input and Flag number**,**

See also: SET INPUT FLAG, IF INPUT FLAG ON, INPUT GLOBAL SCRIPTS

Example:
The following AREA SCRIPT clears input 1 and 2 flag 1.

Example: AREA SCRIPT

```
IF AREA EVENT ON SECURE
        CLEAR INPUT FLAG, 1,1
        CLEAR INPUT FLAG, 2,1
ENDIF
```

SET INPUT FLAG, input, flag

Purpose: Sets input flag on selected input from any scripts.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Sets flag on selected input.

Parameters: Input and Flag number**,**

See also: CLEAR INPUT FLAG, IF INPUT FLAG ON, INPUT GLOBAL SCRIPTS

Example:
The following ACESS SCRIPT Sets flag 1 for input 1, when user 1 code has been entered at Door 2.

Example: ACCESS SCRIPT

```
IF USER EQ, 1
IF DOOR EQ, 2
        SET INPUT FLAG, 1,1
ENDIF
ENDIF
```

IF INPUT FLAG ON, input, flag

Purpose: Test input flag on selected input from any scripts.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests flag on selected input.

Parameters: Input and Flag number,

See also: CLEAR INPUT FLAG, SET THIS INPUT FLAG, INPUT GLOBAL SCRIPTS

Example:
The following ACESS SCRIPT Clears flag 1 for input 1, when user 1 code has been entered at Door 2 and flag has been previously set.

Example: ACCESS SCRIPT

```
IF USER EQ,1
IF DOOR EQ,2
IF INPUT FLAG ON,1,1
        CLEAR INPUT FLAG,1,1
ENDIF
ENDIF
ENDIF
```

INPUT ALARM ON, input

Purpose: Activates selected input **alarm on** from any scripts.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Sets the input ALARM ON flag for selected input.

Parameters: Input number,

See also: INPUT ALARM OFF, INPUT GLOBAL SCRIPTS

Example:
The following AREA SCRIPT starts area timer on ACCESS and resets input 1 alarm.
If area is not set into secure mode, prior timer expire, input 1 alarm is generated.
This example can be used, to generate alarm, if area is in access mode longer than 5 minutes. (e.g. Strong room, consider as high security area).

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS
        START AREA TIMER,1,300
        INPUT ALARM OFF,1
ENDIF
IF AREA EVENT ON SECURE
        STOP AREA TIMER,1
ENDIF
IF AREA EVENT ON TIMER EXPIRE
        INPUT ALARM ON,1
ENDIF
```

IF INPUT DELAY TIMER ON, input, timer

Purpose: Test, if input timer is active.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if input timer is presently active.

Parameters: Input number,

See also: INPUT GLOBAL SCRIPTS

Example:
The following INPUT SCRIPT shows, how to use the 'IF INPUT TIMER ON' command. Here we are testing, if input delay timer is active, and if not, we generate alarm condition.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
IF ACCESSED
        RETURN
ENDIF
IF INPUT DELAY TIMER ON,1,1
        RETURN
ELSE
        ALARM ON
ENDIF
ENDIF
```

START INPUT DELAY TIMER, input, timer, duration

Purpose: Starts input timer.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Starts the input timer.

Parameters: Input, timer number and duration time.

See also: STOP INPUT DELAY TIMER, INPUT GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will start input 1 timer 2 for 10 minutes (600 seconds), when alarm is generated. When timer expire, the area alarm is reset..

Example: AREA SCRIPT

```
IF AREA EVENT ON SECURE
        SEND DIALLER,1,999,3,401        // Send Dialler
ENDIF
ENDIF
IF AREA EVENT ON ACCESS
        SEND DIALLER,1,999,1,401        // Send Dialler
ENDIF
IF AREA EVENT ON ALARM
        START INPUT DELAY TIMER,1,2,600   // Start Input 1 Timer
ENDIF
```

Inut Script: (Note this is a Script allocated to Input 1)

```
IF INPUT EVENT ON TIMER EXPIRE
        RESET AREA, 1
ENDIF
```

STOP INPUT DELAY TIMER, input, timer

Purpose: Stopss input timer.

Type: Control

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: Stops input timer.

Parameters: Input, timer number and duration time.

See also: START INPUT DELAY TIMER, INPUT GLOBAL SCRIPTS

Example:
The following AREA SCRIPT will start input 1 timer 2 for 10 minutes (600 seconds), when alarm is generated. When timer expire, the area alarm is reset. If Area is reset (which may be by user), the timer is stopped.

Example: AREA SCRIPT

```
IF AREA EVENT ON SECURE
        SEND DIALLER,1,999,3,401        // Send Dialler
ENDIF
ENDIF
IF AREA EVENT ON ACCESS
        SEND DIALLER,1,999,1,401        // Send Dialler
ENDIF
IF AREA EVENT ON ALARM
        START INPUT DELAY TIMER,1,2,600     // Start Input 1 Timer 2
ENDIF
IF AREA EVENT ON RESET
        STOP INPUT DELAY TIMER,1,2          // Stop Input 1 Timer 2
ENDIF
```

Inut Script: (Note this is a Script allocated to Input 1)

```
IF INPUT EVENT ON TIMER EXPIRE
        RESET AREA, 1
ENDIF
```

INPUT ALARM OFF, input

Purpose: Reset selected input alarm condition.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Resets the input ALARM ON flag for selected input.

Parameters: Input number,

See also: INPUT ALARM ON, INPUT GLOBAL SCRIPTS

Example:
The following AREA SCRIPT starts area timer on ACCESS and resets input 1 alarm. If area is not set into secure mode, prior timer expire, input 1 alarm is generated.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS
        START AREA TIMER, 1,300
        INPUT ALARM OFF, 1
ENDIF
IF AREA EVENT ON SECURE
        STOP AREA TIMER, 1
ENDIF
IF AREA EVENT ON TIMER EXPIRE
        INPUT ALARM ON, 1
ENDIF
```

CALL SCRIPT, script, param

Purpose: Calls control script, where common functions to multiple scripts can be reused. For example turning on or off a group of outputs or buzzers or locking a series of doors.

Type: Function

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Calls control script with specified parameter value.

Parameters: Control Script number and passing value (0 – 255),

See also: CHAIN, IF CONTROL VALUE EQUAL, GLOBAL SCRIPTS

Example:
The following AREA SCRIPT calls control script, where common outputs are activated or de-activated. ***Please note: CALL SCRIPT can ONLY call CONTROL SCRIPT TYPES. When control script is called, system performs functions within the control script and returns to calling script. Then it will continue functions within the calling script until reaches the end of the script.***
System does not pass any other variables, except 'passing value', into the called script. Therefore ONLY global commands can be used in the control script.

```
Example: AREA SCRIPT
        IF AREA EVENT ON ALARM
                CALL SCRIPT, 5,2                // Call Script 5 with parameter 2
                BUZZER CONTROL, 1,2,5           // Activate buzzer on device 1
        ENDIF
        IF AREA EVENT ON TAMPER
                CALL SCRIPT, 5,2                // Call Script 5 with parameter 2
        ENDIF
        IF AREA EVENT ON RESET
                CALL SCRIPT, 5,3                // Call script 5 with parameter 3
        ENDIF
CONTROL SCRIPT 5:
        IF CONTROL VALUE EQUAL, 2
                OUTPUT CONTROL, 1,2,0                   // Activates Strobe light
                OUTPUT CONTROL, 1,2,600                 // Activates Siren for 10 minutes
        ENDIF
        IF CONTROL VALUE EQUAL, 3
                OUTPUT CONTROL, 1,1,0                   // De-Activates Strobe light
                OUTPUT CONTROL, 1,1,0                   // De-Activates Siren
        ENDIF
```

IF CONTROL VALUE EQUAL, value

Purpose: Test value passed from call script.

Type: Test

Available for:

☑ Control

Description: Call script or PC computer can pass value to Control script, which can be tested for. Then different action can be generated, depending on the passed value to the script.

Parameters: Passing value (0 – 255),

See also: CALL SCRIPT, GLOBAL SCRIPTS

Example:
The following AREA SCRIPT calls control script, where common outputs are activated or de-activated. ***Please note: CALL SCRIPT can ONLY call CONTROL SCRIPT TYPES. When control script is called, system performs functions within the control script and returns to calling script. Then it will continue functions within the calling script until reaches the end of the script.***
System does not pass any other variables, except 'passing value', into the called script. Therefore ONLY global commands can be used in the control script.

Example: AREA SCRIPT

```
IF AREA EVENT ON ALARM
        CALL SCRIPT, 5,2                // Call Script 5 with parameter 2
        BUZZER CONTROL, 1,2,5           // Activate buzzer on device 1
ENDIF
IF AREA EVENT ON TAMPER
        CALL SCRIPT, 5,2                // Call Script 5 with parameter 2
ENDIF
IF AREA EVENT ON RESET
        CALL SCRIPT, 5,3                // Call script 5 with parameter 3
ENDIF
```

CONTROL SCRIPT 5:

```
IF CONTROL VALUE EQUAL, 2
        OUTPUT CONTROL, 1,2,0                   // Activates Strobe light
        OUTPUT CONTROL, 1,2,600                 // Activates Siren for 10 minutes
ENDIF
IF CONTROL VALUE EQUAL, 3
        OUTPUT CONTROL, 1,1,0                   // De-Activates Strobe light
        OUTPUT CONTROL, 1,1,0                   // De-Activates Siren
ENDIF
```

CHAIN, script

Purpose: Transfer control to another script.

Type: Function

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Jumps to another script, must be same script type, and transfer all hidden values.

Parameters: CHAIN Script number,

See also: CALL SCRIPT, GLOBAL SCRIPTS

Example:
The following AREA SCRIPT CHAIN to another area script, where common outputs are activated or de-activated. ***Please note: CHAIN can call ONLY SAME SCRIPT TYPE. When script is called, system performs functions within the control script and DOES NOT returns to the calling script.***

Example: AREA SCRIPT

```
IF AREA EVENT ON ALARM
        BUZZER CONTROL, 1,2,5          // Activate buzzer on device 1
ENDIF
CHAIN, 20                              // CHAIN to Script 20
// NO OTHER COMMANDS WILL BE EXECUTED HERE
```

Example:AREA SCRIPT 20:

```
IF AREA EVENT ON TAMPER
        OUTPUT CONTROL,1,2,0           // Activate Output
ENDIF
IF AREA EVENT ON RESET
        OUTPUT CONTROL,1,1,0           // Stop Output
ENDIF
```

SET CLOCK FLAG, clock, flag

Purpose: Set clock flag, when required.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each clock includes flags 1- 9, which can be set or reset under script control. Flag 9 is used as DISABLE CLOCK.

Parameters: Clock and flag number,

See also: CLEAR CLOCK FLAG, GLOBAL CLOCK SCRIPTS

Example:
The following CLOCK SCRIPT indicates, how SET CLOCK FLAG function can be used.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VALID
IF AREA IN ACCESS, 1
        SET CLOCK FLAG, 1,2          // Set Clock 1 flag 2
ENDIF
ENDIF
IF CLOCK EVENT ON VOID
IF CLOCK FLAG, 1,2
        CLEAR CLOCK FLAG, 1,2        // Clear Clock 1 flag 2
        LOG EVENT, 5                 // Log event
ENDIF
ENDIF
```

CLEAR CLOCK FLAG, clock, flag

Purpose: Set clock flag, when required.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each clock includes flags 1- 9, which can be set or reset under script control. Flag 9 is used as DISABLE CLOCK.

Parameters: Clock and flag number,

See also: SET CLOCK FLAG, GLOBAL CLOCK SCRIPTS

Example:
The following CLOCK SCRIPT indicates, how CLEAR CLOCK FLAG function can be used.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VALID
IF AREA IN ACCESS, 1
        SET CLOCK FLAG, 1,2             // Set Clock 1 flag 2
ENDIF
ENDIF
IF CLOCK EVENT ON VOID
IF CLOCK FLAG, 1,2
        CLEAR CLOCK FLAG, 1,2           // Clear Clock 1 flag 2
        LOG EVENT, 5                    // Log event
ENDIF
ENDIF
```

DISABLE CLOCK, clock

Purpose: Temporary disable clock under script control.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each clock can be disabled and enabled, even if clock is valid via clock control.
Please note: Clock to be in VALID mode, must be valid under timer control as well ENABLE via clock control. 'IF CLOCK EVENT TO VOID' will trigger if clock time was valid when CLOCK DISABLE command been issued.

Parameters: Clock number,

See also: ENABLE CLOCK, GLOBAL CLOCK SCRIPTS

Example:
The following INPUT SCRIPT indicates, how use the DISABLE and ENABLE control command.
When input is unsealed, the Clock is disabled, and when sealed, clock is enabled.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
        DISABLE CLOCK, 2            // Disable Clock 2
ENDIF
IF INPUT EVENT ON SEAL
        ENABLE CLOCK, 2             // Enable Clock 2
ENDIF
```

ENABLE CLOCK, clock

Purpose: Enable clock under script control.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each clock can be disabled and enabled, even if clock is valid via clock control.
Please note: Clock to be in VALID mode, must be valid under timer control as well ENABLE via clock control. 'IF CLOCK EVENT TO VALID' will trigger, if clock time was valid, when CLOCK ENABLE command been issued

Parameters: Clock number,

See also: DISABLE CLOCK, GLOBAL CLOCK SCRIPTS

The following INPUT SCRIPT indicates, how use the DISABLE and ENABLE control command.
When input is unsealed, the Clock is disabled, and when sealed, clock is enabled.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
        DISABLE CLOCK, 2                // Disable Clock 2
ENDIF
IF INPUT EVENT ON SEAL
        ENABLE CLOCK, 2                 // Enable Clock 2
ENDIF
```

IF CLOCK VALID, clock

Purpose: Allows test to the status of the Clock.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Status of each clock can be tested, allowing different response on the status of the clock.
The clock can be disabled and enabled, while clock is valid via clock time control.

Parameters: Clock number,

See also: IF CLOCK VOID, CLOCK TEST CONTROL, GLOBAL CLOCK SCRIPTS

Example:
The following INPUT SCRIPT shows the example of usage of the 'IF CLOCK VALID' command.
Please note: Each clock can be set to enable or disable status. The valid status will be true only, if clock is enable and valid via clock control as well.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
IF CLOCK VALID, 5
        ALARM ON                    // Latch the alarm
ENDIF
ENDIF
IF INPUT EVENT ON RESET
        ALARM OFF                   // Reset alarm latch
ENDIF
```

IF CLOCK VOID, clock

Purpose: Allows test to the status of the Clock.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Status of each clock can be tested, allowing different response on the status of the clock. Each clock can be disabled and enabled, even if clock is valid via clock control.

Parameters: Clock number,

See also: IF CLOCK VALID, CLOCK TEST CONTROL, GLOBAL CLOCK SCRIPTS

Example:
The following INPUT SCRIPT shows the example of usage of the 'IF CLOCK VOID' command.
Please note: Each clock can be set to enable or disable status. The valid status will be true only, if clock is enable and valid via clock control as well.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
IF CLOCK VOID, 5
          ALARM ON                      // Latch the alarm
ENDIF
ENDIF
IF INPUT EVENT ON RESET
          ALARM OFF                     // Reset alarm latch
ENDIF
```

CLEAR DOOR FLAG, door, flag

Purpose: Clears flag of selected door.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each door includes flags 1-8, which can be set, cleared or tested on. Within your scripts, you can use flags 1-8 for your own functionality

Parameters: Door and flag number,

See also: SET DOOR FLAG, IF DOOR FLAG ON, GLOBAL DOOR SCRIPTS

Example:
The following DOOR SCRIPT shows the example of usage of the 'CLEAR DOOR FLAG`, 'SET DOOR FAG' and 'IF DOOR FLAG ON` commands.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON FORCED
        SET DOOR FLAG, 1,1          // Set Flag
ENDIF
IF DOOR EVENT ON NORMAL
        CLEAR DOOR FLAG, 1,1        // Reset Flag
ENDIF
```

Example ACCESS SCRIPT:

```
IF DOOR EQUAL, 1
IF DOOR FLAG ON, 1,1
        BUZZER CONTROL, 12,3,5
ENDIF
ENDIF
```

SET DOOR FLAG, door, flag

Purpose: Set flag of selected door.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each door includes flags 1-8, which can be set, cleared or tested on. Within your scripts, you can use flags 1-8 for your own functionality

Parameters: Door and flag number,

See also: CLEAR DOOR FLAG, IF DOOR FLAG ON, GLOBAL DOOR SCRIPTS

Example:
The following DOOR SCRIPT shows the example of usage of the 'CLEAR DOOR FLAG`, 'SET DOOR FAG' and 'IF DOOR FLAG ON` commands.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON FORCED
        SET DOOR FLAG, 1,1          // Set Flag
ENDIF
IF DOOR EVENT ON NORMAL
        CLEAR DOOR FLAG, 1,1        // Reset Flag
ENDIF
```

Example ACCESS SCRIPT:

```
IF DOOR EQUAL, 1
IF DOOR FLAG ON, 1,1
        BUZZER CONTROL, 12,3,5
ENDIF
ENDIF
```

IF DOOR FLAG ON, door, flag

Purpose: Test flag status of selected door.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each door includes flags 1-8, which can be set, cleared or tested on. Within your scripts, you can use flags 1-8 for your own functionality

Parameters: Door and flag number,

See also: SET DOOR FLAG, CLEAR DOOR FLAG, GLOBAL DOOR SCRIPTS

Example:
The following DOOR SCRIPT shows the example of usage of the 'CLEAR DOOR FLAG`, 'SET DOOR FAG' and 'IF DOOR FLAG ON` commands.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON FORCED
        SET DOOR FLAG, 1,1             // Set Flag
ENDIF
IF DOOR EVENT ON NORMAL
        CLEAR DOOR FLAG, 1,1           // Reset Flag
ENDIF
```

Example ACCESS SCRIPT:

```
IF DOOR EQUAL, 1
IF DOOR FLAG ON, 1,1
        BUZZER CONTROL, 12,3,5
ENDIF
ENDIF
```

LOCK ALL ON DOOR, door

Purpose: Allows control of selected door.

Type: Control

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: Each door can be selected to required status. LOCK ALL ON DOOR will suspend any operation on this door using Proximity reader or egress button. SECURE DOOR will return door to normal operation.

Parameters: Door number,

See also: SECURE DOOR, SECURE THIS DOOR, LOCK DOOR ENTRY, LOCK THIS DOOR ENTRY, RELEASE DOOR, UNLOCK DOOR, LOCK ALL ON THIS DOOR, GLOBAL DOOR SCRIPTS

Example:
The following AREA SCRIPT shows the example of how to use the 'LOCK ALL ON DOOR` as well 'SECURE DOOR` commands. When area is set to SECURE mode, the Door 1 is locked, and when area is set to access, normal operation is resume.

Example: AREA SCRIPT

```
IF AREA EVENT ON SECURE
        LOCK ALL ON DOOR, 1         // Lock DOOR, while in secure
ENDIF
IF AREA EVENT ON ACCESS
        SECURE DOOR, 1              // Allow normal access
ENDIF
```

SECURE DOOR, door

Purpose: Allows control of selected door.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each door can be selected to required status. UNLOCK DOOR will unlock the door, suspend any user operation and DOTL and FORCED door alarms are ignored. SECURE DOOR will return door back to normal operation.

Parameters: Door number,

See also: LOCK ALL ON DOOR, LOCK THIS DOOR ENTRY, RELEASE DOOR, UNLOCK DOOR, LOCK ALL ON THIS DOOR, GLOBAL DOOR SCRIPTS

Example:
The following TIMER SCRIPT shows the example of how to use the ‘UNLOCK DOOR` as well ‘SECURE DOOR` commands. When clock is validated, the door is unlocked allowing free access. When the clock is void, the door is returned back to normal operation, allowing access using pin or card access.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VALID
        UNLOCK DOOR, 1                // Allow free access to door 1
ENDIF
IF CLOCK EVENT ON VOID
        SECURE DOOR, 1                // Set normal security
ENDIF
```

LOCK DOOR ENTRY, door

Purpose: Allows control of selected door.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each door can be selected to required status. LOCK DOOR ENTRY will suspend any operation on this door using Proximity reader, but allows operation via egress button. SECURE DOOR will remove previously locked door.

Parameters: Door number,

See also: LOCK ALL ON DOOR, LOCK ALL ON THIS DOOR, RELEASE DOOR, SECURE DOOR, SECURE THIS DOOR, UNLOCK DOOR, LOCK THIS DOOR ENTRY, GLOBAL DOOR SCRIPTS

Example:

The following AREA SCRIPT shows the example of how to use the ‘LOCK DOOR ENTRY` as well ‘UNLOCK DOOR` commands. When area is set to SECURE mode, the Door 1 user access is locked, and when area is set to access, the door is opened.

Example: AREA SCRIPT

```
IF AREA EVENT ON SECURE
        LOCK DOOR ENTRY, 1                // Lock user access, while in
secure
ENDIF
IF AREA EVENT ON ACCESS
        SECURE DOOR, 1              // Set door to normal operation
ENDIF
```

RELEASE DOOR, door

Purpose: Allows control of selected door.

Type: Control

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: Door can be released using external control. Release command will open the door for duration of RELAY TIME.

Please note: While door is locked it cannot be released.

Parameters: Door number,

See also: LOCK ALL ON DOOR, LOCK ALL ON THIS DOOR, LOCK DOOR ENTRY, LOCK THIS DOOR ENTRY, SECURE DOOR, UNLOCK DOOR, GLOBAL DOOR SCRIPTS

Example:

The following INPUT SCRIPT shows the example of how to use the ‘RELEASE DOOR` command.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL
        RELEASE DOOR, 1                  // Release door
ENDIF
```

UNLOCK DOOR, door

Purpose: Allows control of selected door.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each door can be selected to required status. UNLOCK DOOR will unlock the door, suspend any user operation and DOTL and FORCED door alarms are ignored. SECURE DOOR will return door back to normal operation.

Parameters: Door number,

See also: SECURE DOOR, SECURE THIS DOOR, LOCK ALL ON DOOR, LOCK THIS DOOR ENTRY, UNLOCK THIS DOOR, LOCK ALL ON THIS DOOR, RELEASE THIS DOOR, LOCK DOOR ENTRY, GLOBAL DOOR SCRIPTS

Example:
The following TIMER SCRIPT shows the example of how to use the 'UNLOCK DOOR` as well 'SECURE DOOR` commands. When clock is validated, the door is unlocked allowing free access. When the clock is void, the door is returned back to normal operation, allowing access using pin or card access.

Example: CLOCK SCRIPT

```
IF CLOCK EVENT ON VALID
        UNLOCK DOOR, 1              // Allow free access to door 1
ENDIF
IF CLOCK EVENT ON VOID
        SECURE DOOR, 1              // Set normal security
ENDIF
```

IF DOOR DELAY TIMER ON, area, timer

Purpose: Test the door timer.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Tests if the door timer is active.

Parameters: Area and timer number.

See also: GLOBAL DOOR SCRIPTS

Example:
The following DOOR SCRIPT will start the door timer, when force door alarm is genrated. If door timer is active, and doro has been set to normal, we stop the timer. If the timer expires, we send alarm via Dialler.
Please note: we assume, this script is associated with door1.

Example: DOOR SCRIPT

```
IF DOOR EVENT ON TIMER EXPIRE
        SEND DIALLER,1,999,1,136
        SET DOOR FLAG,1                         // set Flag, we TX Alar
ENDIF
IF DOOR EVENT ON FORCED
        START DOOR DELAY,1,1,600                // Start Door Timer
ENDIF
IF DOOR EVENT ON CLOSE
        IF DOOR DELAY TIMER ON,1,1
                STOP DOOR TIMER,1,1
        ENDIF
        IF THIS DOOR FLAG,1                     // Is Flag set,
                SEND DIALLER,1,999,3,136        // Yes, send restoral
        ENDIF
        CLEAR THIS DOOR FLAG,1                  // Clear anyway
ENDIF
```

CLEAR GLOBAL FLAG, flag

Purpose: Clears Global Flag.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Global flags can be set, clear or tested on. Each global flag can be used as indicator, latch or any another means to provide further flexibility to your scripts.

Parameters: Global Flag number (max 128 flags),

See also: IF GLOBAL FLAG ON, SET GLOBAL FLAG,

Example:
The following ACCESS SCRIPT shows the example of how to use the ‘CLEAR GLOBAL FLAG` and ‘SET GLOBAL FLAG`. If Technician code has been inserted, suspend cabinet tamper detection.

Example: ACCESS SCRIPT

```
IF USER EQUAL,50                    // is this a Technician code ?
IF GLOBAL FLAG ON,1
        CLEAR GLOBAL FLAG,1
        OUTPUT CONTROL,3,2,0        // Make some indication
ELSE
        SET GLOBAL FLAG,1
        OUTPUT CONTROL,3,1,0        // Reset some indication
ENDIF
ENDIF
```

Example: ALARM SCRIPT:

```
IF ALARM EVENT ON,3 [[003] TAMPER ALARM]
IF GLOBAL FLAG ON,1
        RETURN
ELSE
        SEND DIALLER,1,9999,1,370
ENDIF
ENDIF
IF ALARM EVENT OFF,3 [[003] TAMPER ALARM]
IF GLOBAL FLAG ON,1
        RETURN
ELSE
        SEND DIALLER,1,9999,3,370
ENDIF
ENDIF
```

IF GLOBAL FLAG ON, flag

Purpose: Test status of Global Flag.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Test status of Global flag. If Global flag is set, code within IF and ENDIF will be executed.

Parameters: Global Flag number (max 128 flags),

See also: CLEAR GLOBAL FLAG, SET GLOBAL FLAG

Example:
The following ACCESS SCRIPT shows the example of how to use the 'CLEAR GLOBAL FLAG` and 'SET GLOBAL FLAG`. If Technician code has been inserted, suspend cabinet tamper detection.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 50                    // is this a Technician code?
IF GLOBAL FLAG ON, 1
        CLEAR GLOBAL FLAG, 1
        OUTPUT CONTROL, 3,2,0        // Make some indication
ELSE
        SET GLOBAL FLAG, 1
        OUTPUT CONTROL, 3,1,0        // Reset some indication
ENDIF
ENDIF
```

Example: ALARM SCRIPT:

```
IF ALARM EVENT ON, 3 [[003] TAMPER ALARM]
IF GLOBAL FLAG ON, 1
        RETURN
ELSE
        SEND DIALLER, 1,9999,1,370
ENDIF
ENDIF
IF ALARM EVENT OFF, 3 [[003] TAMPER ALARM]
IF GLOBAL FLAG ON, 1
        RETURN
ELSE
        SEND DIALLER, 1,9999,3,370
ENDIF
ENDIF
```

SET GLOBAL FLAG, flag

Purpose: Sets Global Flag.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Global flags can be set, clear or tested on. Each global flag can be used as indicator, latch or any another means to provide further flexibility to your scripts.

Parameters: Global Flag number (max 128 flags),

See also: CLEAR GLOBAL FLAG, IF GLOBAL FLAG ON

Example:
The following ACCESS SCRIPT shows the example of how to use the 'CLEAR GLOBAL FLAG` and 'SET GLOBAL FLAG`. If Technician code has been inserted, suspend cabinet tamper detection.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 50                   // is this a Technician code?
IF GLOBAL FLAG ON, 1
        CLEAR GLOBAL FLAG, 1
        OUTPUT CONTROL, 3,2,0       // Make some indication
ELSE
        SET GLOBAL FLAG, 1
        OUTPUT CONTROL, 3,1,0       // Reset some indication
ENDIF
ENDIF
```

Example: ALARM SCRIPT:

```
IF ALARM EVENT ON, 3 [[003] TAMPER ALARM]
IF GLOBAL FLAG ON, 1
        RETURN
ELSE
        SEND DIALLER, 1,9999,1,370
ENDIF
ENDIF
IF ALARM EVENT OFF, 3 [[003] TAMPER ALARM]
IF GLOBAL FLAG ON, 1
        RETURN
ELSE
        SEND DIALLER, 1,9999,3,370
ENDIF
ENDIF
```

SET GROUP FLAG ON, group, flag

Purpose: Sets associated Group Flag.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each Group has 10 flags. Each Flag can be set, cleared or tested on.
Please note: Flag 9 is used as Group Suspend Flag. When flag 9 is set, the group is suspended.
Flag 10 is used for Technician debugging only. If this Flag is set, system will generate special ACCESS EVENT Codes. This event will include Rejection code errors for Area, Door and Menu access, when valid user code has been presented at the RAS.

Parameters: Group Flag number,

See also: CLEAR GROUP FLAG, IF GROUP FLAG ON

Example:
The following ACCESS SCRIPT shows the example of how to use the 'CLEAR GROUP FLAG` and 'SET GLOBAL FLAG ON`. If Technician code has been inserted, the flag 1 is set.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 50                       // is this a Technician code?
IF GROUP FLAG ON, 1,2
        CLEAR GROUP FLAG, 1,2
ELSE
        SET GROUP FLAG ON, 1,2          // Group 1, Flag 2
ENDIF
ENDIF
```

CLEAR GROUP FLAG, group, flag

Purpose: Clears associated Group Flag.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each Group has 10 flags. Each Flag can be set, cleared or tested on.
Please note: Flag 9 is used as Group Suspend Flag. When flag 9 is set, the group is suspended.
Flag 10 is used for Technician debugging only. If this Flag is set, system will generate special ACCESS EVENT Codes. This event will include Rejection code errors for Area, Door and Menu access, when valid user code has been presented at the RAS.

Parameters: Group and Flag number,

See also: SET GROUP FLAG ON, IF GROUP FLAG ON

Example:
The following ACCESS SCRIPT shows the example of how to use the 'CLEAR GROUP FLAG` and 'SET GROUP FLAG ON`. If Technician code has been inserted, the flag 1 is set.

Example: ACCESS SCRIPT

```
IF USER EQUAL, 50                    // is this a Technician code?
IF GROUP FLAG ON, 1,1
        CLEAR GROUP FLAG, 1,1
ELSE
        SET GROUP FLAG ON, 1,1
ENDIF
ENDIF
```

SET USER FLAG ON, user, flag

Purpose: Sets associated USER Flag.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each User has 9 flags. Each Flag can be set, cleared or tested on.
Please note: Flag 9 is used as User Suspend Flag. When flag 9 is set, the user is suspended.

Parameters: User and Flag number,

See also: CLEAR USER FLAG, IF USER FLAG ON

Example:
The following INPUT SCRIPT shows the example of how to use the 'SET USER FLAG ON` and 'CLEAR USER FLAG`. If input is unsealed, the User 2 is suspended, and when sealed, User suspend flag is cleared.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL          // Input has unsealed
        SET USER FLAG ON, 2,9
ENDIF
IF INPUT EVENT ON SEAL            // Input has sealed
        CLEAR USER FLAG, 2,9
ENDIF
```

CLEAR USER FLAG, user, flag

Purpose: Clears associated USER Flag.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each User has 9 flags. Each Flag can be set, cleared or tested on. ***Please note: Flag 9 is used as User Suspend Flag. When flag 9 is set, the user is suspended.***

Parameters: User and Flag number,

See also: SET USER FLAG ON, IF USER FLAG ON

Example:
The following INPUT SCRIPT shows the example of how to use the ‘SET USER FLAG ON` and ‘CLEAR USER FLAG`. If input is unsealed, the User 2 is suspended, and when sealed, User suspend flag is cleared.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON UNSEAL            // Input has unsealed
        SET USER FLAG ON, 2,9
ENDIF
IF INPUT EVENT ON SEAL              // Input has sealed
        CLEAR USER FLAG, 2,9
ENDIF
```

IF USER FLAG ON, user, flag

Purpose: Tests associated USER Flag.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each User has 9 flags. Each Flag can be set, cleared or tested on. ***Please note: Flag 9 is used as User Suspend Flag. When flag 9 is set, the user is suspended.***

Parameters: User and Flag number,

See also: SET USER FLAG ON, CLEAR USER FLAG,

Example:
The following AREA SCRIPT shows the example of how to use the ‘IF USER FLAG ON`. If area is set to Access mode, user 2 flag is tested.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS           // Area changed to access
IF USER FLAG ON, 2,9
        CLEAR USER FLAG, 2,9      // Clear Flag
ENDIF
ENDIF
```

IF GROUP FLAG ON, group, flag

Purpose: Tests associated GROUP Flag.

Type: Test

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Each Group has 10 flags. Each Flag can be set, cleared or tested on. ***Please note: Flag 9 is used as Group Suspend Flag. When flag 9 is set, the Group is suspended.***

Parameters: Group and Flag number,

See also: SET GROUP FLAG ON, CLEAR GROUP FLAG

Example:
The following AREA SCRIPT shows the example of how to use the 'IF GROUP FLAG ON`. If area is set to Access mode, group 2 flag is tested.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS            // Area changed to access
IF GROUP FLAG ON, 2,9
        CLEAR GROUP FLAG, 2,9      // Clear Flag
ENDIF
ENDIF
```

CLEAR VARIABLE, variable

Purpose: Clears specified Variable to '0'.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: System includes 128 variables. Each variable can be cleared, set to specific value and incremented. You can test each variable to a value or compare to other variables.

Parameters: Variable number (max 128 variables),

See also: SET VARIABLE, INCREMENT VARIABLE, DECREMENT VARIABLE, IF VARIABLE COMPARE, IF VARIABLE

Example:
The following AREA SCRIPT shows the example of how to use the 'CLEAR VARIABLE' commands. The ACCESS SCRIPT shows, how to use the 'INCREMENT VARIABLE' and 'DECREMENT VARIABLE' commands.
We allow 10 people to enter on DOOR 1. When people leave via Door 2, we decrement the counter. Once the count reaches 10, we lock the door and when counter is decremented to 5, we put entry door back to normal operation.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS         // Area changed to access
    CLEAR VARIABLE, 1           // Clear Variable 1
    SET DOOR NORMAL, 1          // Set DOOR to normal entry
ENDIF
```

Example ACCESS SCRIPT:

```
IF DOOR EQUAL, 1                // Door 1 Entry
    INCREMENT VARIABLE, 1       // More entries
    LOCK DOOR,1                 // Lock the DOOR
    OUTPUT CONTROL, 4,2,0       // give an Indication
ENDIF
IF DOOR EQUAL, 2                // Door 2 Exit
    DECREMENT VARIABLE, 1       // Less entries
    SECURE DOOR,1               // Set DOOR to Normal entry
    OUTPUT CONTROL, 4,1,0       // Clear Indication
ENDIF
```

INCREMENT VARIABLE, variable

Purpose: Adds 1 to specified Variable.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: System includes 128 variables. Each variable can be cleared, set to specific value and incremented. You can test each variable to a value or compare to other variables.

Parameters: Variable number (max 128 variables),

See also: SET VARIABLE, CLEAR VARIABLE, DECREMENT VARIABLE, IF VARIABLE COMPARE, IF VARIABLE

Example: The following AREA SCRIPT shows the example of how to use the 'CLEAR VARIABLE' commands. The ACCESS SCRIPT shows, how to use the 'INCREMENT VARIABLE' and 'DECREMENT VARIABLE' commands.
We allow 10 people to enter on DOOR 1. When people leave via Door 2, we decrement the counter. Once the count reaches 10, we lock the door and when counter is decremented to 5, we put entry door back to normal operation.

Example: AREA SCRIPT
```
IF AREA EVENT ON ACCESS          // Area changed to access
        CLEAR VARIABLE, 1        // Clear Variable 1
        SET DOOR NORMAL, 1       // Set DOOR to normal entry
ENDIF
```
Example ACCESS SCRIPT:
```
IF DOOR EQUAL, 1                 // Door 1 Entry
        INCREMENT VARIABLE, 1    // More entries

        LOCK DOOR,1              // Lock the DOOR
        OUTPUT CONTROL, 4,2,0    // give an Indication
ENDIF
IF DOOR EQUAL, 2                 // Door 2 Exit
        DECREMENT VARIABLE, 1    // Less entries
        SECURE DOOR,1            // Set DOOR to Normal entry
        OUTPUT CONTROL, 4,1,0    // Clear Indication
ENDIF
```

DECREMENT VARIABLE, variable

Purpose: Decrements specified Variable by 1.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: System includes 128 variables. Each variable can be cleared, set to specific value and incremented. You can test each variable to a value or compare to other variables.

Parameters: Variable number (max 128 variables),

See also: SET VARIABLE, INCREMENT VARIABLE, CLEAR VARIABLE, IF VARIABLE COMPARE, IF VARIABLE

Example:
The following AREA SCRIPT shows the example of how to use the 'CLEAR VARIABLE' commands. The ACCESS SCRIPT shows, how to use the 'INCREMENT VARIABLE' and 'DECREMENT VARIABLE' commands.
We allow 10 people to enter on DOOR 1. When people leave via Door 2, we decrement the counter. Once the count reaches 10, we lock the door and when counter is decremented to 5, we put entry door back to normal operation.

Example: AREA SCRIPT
```
IF AREA EVENT ON ACCESS             // Area changed to access
        CLEAR VARIABLE, 1           // Clear Variable 1
        SET DOOR NORMAL, 1          // Set DOOR to normal entry
ENDIF
```
Example ACCESS SCRIPT:
```
IF DOOR EQUAL, 1                    // Door 1 Entry
        INCREMENT VARIABLE, 1       // More entries
        LOCK DOOR,1                 // Lock the DOOR
        OUTPUT CONTROL, 4,2,0       // give an Indication
ENDIF
IF DOOR EQUAL, 2                    // Door 2 Exit
        DECREMENT VARIABLE, 1       // Less entries
        SECURE DOOR, 1              // Set DOOR to Normal entry
        OUTPUT CONTROL, 4,1,0       // Clear Indication
ENDIF
```

SET VARIABLE, variable, value

Purpose: Set specified Variable to required value.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: System includes 128 variables. Each variable can be cleared, set to specific value and incremented. You can test each variable to a value or compare to other variables.

Parameters: Variable number (max 128 variables), and Value (0-65534),

See also: INCREMENT VARIABLE, CLEAR VARIABLE, DECREMENT VARIABLE, IF VARIABLE COMPARE,

Example:
The following AREA SCRIPT shows the example of how to use the 'SET VARIABLE' commands. The ACCESS SCRIPT shows, how to use the 'INCREMENT VARIABLE' and 'DECREMENT VARIABLE' commands.
We allow 10 people to enter on DOOR 1. When people leave via Door 2, we decrement the counter. Once the count reaches 10, we lock the door and when counter is decremented to 5, we put entry door back to normal operation.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS              // Area changed to access
        SET VARIABLE, 1,0            // Clear Variable 1
        SET DOOR NORMAL, 1           // Set DOOR to normal entry
ENDIF
```

Example ACCESS SCRIPT:

```
IF DOOR EQUAL, 1                     // Door 1 Entry
        INCREMENT VARIABLE, 1        // More entries
        LOCK DOOR,1                  // Lock the DOOR
        OUTPUT CONTROL, 4,2,0        // give an Indication
ENDIF
IF DOOR EQUAL, 2                     // Door 2 Exit
        DECREMENT VARIABLE, 1        // Less entries
        SECURE DOOR,1                // Set DOOR to Normal entry
        OUTPUT CONTROL, 4,1,0        // Clear Indication
ENDIF
```

IF VARIABLE COMPARE, variable1, variable2

Purpose: Compare two variables.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: System includes 128 variables. Each variable can be cleared, set to specific value and incremented. You can test each variable to a value or compare to other variables.

Parameters: Variable 1 and 2 number (max 128 variables),

See also: INCREMENT VARIABLE, SET VARIABLE, CLEAR VARIABLE, DECREMENT VARIABLE, IF VARIABLE,

Example:
The following AREA SCRIPT shows the example of how to use the 'SET VARIABLE' commands. The ACCESS SCRIPT shows, how to use the 'INCREMENT VARIABLE' and 'IF VARIABLE COMPARE' commands.
We count people in and out on DOORS 1 and 2. When people come or leave we test if counter values are the same. If same, we sound a buzzer and clears the Variables.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS             // Area changed to access
        SET VARIABLE, 1,0           // Clear Variable 1
        SET VARIABLE, 2,0           // Clear Variable 2
        SET DOOR NORMAL, 1          // Set DOOR to normal entry
        SET DOOR NORMAL, 2          // Set DOOR to normal entry
ENDIF
```

Example ACCESS SCRIPT:

```
IF DOOR EQUAL, 1                    // Door 1 Entry
        INCREMENT VARIABLE, 1       // More entries
ENDIF
IF DOOR EQUAL, 2                    // Door 2 Exit
        INCREMENT VARIABLE, 2       // More exits
ENDIF
IF VARIABLE COMPARE,1,2             // Same number in as out ?
        BUZZER CONTROL,1,3,5        // Indicate no person in
        SET VARIABLE, 1,0           // Clear Variable 1
        SET VARIABLE, 2,0           // Clear Variable 2
ENDIF
```

IF VARIABLE, condition, value

Purpose: Compare conditionaly variable to a value.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: System includes 128 variables. Each variable can be cleared, set to specific value and incremented. You can test each variable to a value or compare to other variables.

Parameters: Variable 1 and 2 number (max 128 variables),

See also: INCREMENT VARIABLE, SET VARIABLE, CLEAR VARIABLE, DECREMENT VARIABLE,

Value used at the condition represent the following:

1= Less
2 – Less or Equal
3 – Equal
4 – Greater or Equal
5 – Greater

Example:
The following AREA SCRIPT shows the example of how to use the ‘SET VARIABLE’ commands. The ACCESS SCRIPT shows, how to use the ‘INCREMENT VARIABLE’, ‘DECREMENT VARIABLE’ and ‘IF VARIABLE’ commands.
We count people in and out on DOORS 1 and 2, then we check, if entry exeeds 50 peaple, and if yes, we lock the door. Door is unlocked, when exit event is generated and counter is decremented. This example can be used in car park, when limited space is allocated to tenant.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS           // Area changed to access
        SET VARIABLE, 1,0         // Clear Variable 1
        SET VARIABLE, 2,0         // Clear Variable 2
        SET DOOR NORMAL, 1        // Set DOOR to normal entry
        SET DOOR NORMAL, 2        // Set DOOR to normal entry
ENDIF
```

Example ACCESS SCRIPT:

```
IF DOOR EQUAL, 1                  // Door 1 Entry
        INCREMENT VARIABLE, 1     // More entries
ENDIF
IF DOOR EQUAL, 2                  // Door 2 Exit
```

```
        DECREMENT VARIABLE, 1       // More exits
ENDIF
IF VARIABLE, 4, 51                  // Is Equal or Greater to 50
        LOCK DOOR,1                 // Lock Entry Door
ELSE
        UNLOCK DOOR,1               // Unlock Door again
ENDIF
```

SEND RAS MESSAGE, ras, message

Purpose: Send selected message to RAS Display.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: System includes 100 programmable messages. System allows messages to be transmitted to RAS on any event within the system.

Parameters: RAS device number and message number,

See also: SEND COMM MESSAGE, GLOBAL SCRIPTS

Example:
The following AREA SCRIPT shows the example of how to use the 'SEND RAS MESSAGE' commands. We assume RAS is programmed at the device 2.

Example: AREA SCRIPT

```
IF AREA EVENT ON ACCESS             // Area changed to access
        SEND RAS MESSAGE, 2,1       // Send Message 1
ENDIF
IF AREA EVENT ON SECURE             // Area changed to secure
        SEND RAS MESSAGE, 2,2       // Send Message 2
ENDIF
```

SEND COMM MESSAGE, unit, port, message

Purpose: Send selected message to RS232 located on external device.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: System includes 100 programmable messages. System allows messages to be transmitted to RAS or TEXT DEVICE (RS232 PORT) in any event within the system.

Parameters: Unit, communication port and message number,

See also: SEND RAS MESSAGE, GLOBAL SCRIPTS

Example:
The following INPUT SCRIPT shows the example of how to use the 'SEND COMM MESSAGE' commands. We assume TDC is programmed at the device 3. Message will be send to port 1.

Please note: Device 2 communication port must be programmed with driver for TEXT DEVICE.

Example: INPUT SCRIPT

```
IF INPUT EVENT ON ACCESS              // Input changed to access
        SEND COMM MESSAGE, 2,1,2      // Send Message 2
ENDIF
IF INPUT EVENT ON SECURE              // Input changed to secure
        SEND COMM MESSAGE, 2,1,3      // Send Message 3
ENDIF
```

IF ALARM EVENT ON, 1 [UNIT OFFLINE]

Purpose: The master cannot communicate with a unit

Type: Event

Available for:

☑ Alarm

Description: The master is unable to communicate with a unit. All units are able to continue a certain amount of operation, but full capabilities are not possible. For example, the two door controller will not continue to provide normal access, but any unlock command are remembered.

A unit-off line should be investigated immediately. It should be initially treated as an alarm with the possibility of an intruder. It should be fixed as soon as possible.

Parameters: Alarm Event Number,

See also: IF ALARM EVENT OFF, 1

Example:
The following example will dial out if a unit goes off line.

```
IF ALARM EVENT ON, 1
   SEND DIALLER, 1,1234,1,143
ENDIF
```

IF ALARM EVENT OFF, 1 [UNIT ONLINE]

Purpose: The master restored communicate with a unit

Type: Event

Available for:

☑ Alarm

Description: The master restored communication with a unit.

Parameters: Alarm Event Number,

See also: IF ALARM EVENT OFF, 1

Example:
The following example will dial out if a unit goes on line.

```
IF ALARM EVENT OFF, 1
   SEND DIALLER, 1,1234,3,143
ENDIF
```

IF ALARM EVENT OFF, 2 [DURESS ALARM RESET]

Purpose: Triggered, when the duress alarm is cancelled by the user.

Type: Event

Available for:

☑ Alert

Description: Triggered when the user cancels the duress alarm on the RAS. This should be used to notify the monitoring company as well as to turn off any alarm outputs associated with the duress alarm.

Parameters:

See also: IF ALARM EVENT ON, 2

Example:
The following is from the standard alert script. It will dial out when the duress alarm is triggered and reset.

```
IF ALARM EVENT OFF, 2
   SEND DIALLER, 1,1234,3,120
ENDIF
IF ALARM EVENT ON, 2
   SEND DIALLER, 1,1234,1,120
ENDIF
```

IF ALARM EVENT ON, 2 [DURESS ALARM]

Purpose: Triggered by a duress (PIN + 1) code entered at the RAS.

Type: Event

Available for:

☒ Alarm

Description: When a user has ***duress enabled***, entering the number one more than their PIN will provide them with their normal access as well as triggering the duress alarm.

Typically the duress alarm should NOT produce any alarms in the immediate vicinity. This would put the user's life at risk. It should, however, notify the monitoring companies immediately as well as possibly increase the level of security within the building.

Parameters:

See also: IF ALARM EVENT OFF, 2

Example:
The following script will dial out when the duress alarm occurs.

```
IF ALARM EVENT OFF, 2
   SEND DIALLER, 1,1234,3,120
ENDIF
IF ALARM EVENT ON, 2
   SEND DIALLER, 1,1234,1,120
ENDIF
```

IF ALARM EVENT OFF, 3 [CABINET TAMPER OFF]

Purpose: This event is generated, when cabinet tamper been closed.

Type: Event

Available for:

☑ Alarm

Description: The unit tamper has been closed. This generally indicates that the technician has replaced the lid, but not necessarily. It could also indicate that someone tampering with the panel is now holding down the tamper switch.

Parameters:

See also: IF ALARM EVENT ON, 3

Please note: The cabinet Tamper is non-latching and will follow the status of the Tamper switch.

Example:

```
IF ALARM EVENT OFF, 3
   SEND DIALLER 1,123,3,145
ENDIF
IF ALARM EVENT ON, 3
   SEND DIALLER 1,123,1,145
ENDIF
```

IF ALARM EVENT ON, 3 –[CABINET TAMPER ON]

Purpose: Triggers when the cabinet tamper is activated.

Type: Function

Available for:

☑ Alarm

Description: The tamper input on the board detects an open circuit. Either a technician, or someone else, while is opening the unit, or unit could be removed from the wall.

This event is typically used to notify the monitoring company. It could also be used to make the SIREN activated, indicating that something could be wrong.

Parameters: Alarm Event Number

See also: IF ALARM EVENT OFF, 3

Please note: The cabinet Tamper is non-latching and will follow the status of the Tamper switch.

Example:

```
IF ALARM EVENT OFF, 3
   SEND DIALLER 1,123,3,145
ENDIF
IF ALARM EVENT ON, 3
   SEND DIALLER 1,123,1,145
ENDIF
```

IF ALARM EVENT ON, 4 –[MAINS FAIL DETECTED]

Purpose: The mains power has failed.

Type: Event

Available for:

☑ Alarm

Description: The mains power has failed. Technically this is the 16V AC input. It could also indicate that the plug pack has been disconnected, or the switch turned off. Whatever the cause, the system is now operating on batteries.

Thought can be given as to how to ensure the battery life is maximized. For example, any power to unlock strikes may be automatically locked.

It may be appropriate to give some indication to the end user, as well notify the monitoring station of this event.

Parameters: Alarm Event Number

See also: IF ALARM EVENT OFF, 4

Example:

The following will notify the monitoring company if the mains fails or is restored. It will also activate the sounder in the RAS (assuming device 2) if the power fails when the area is occupied.

```
IF ALARM EVENT ON, 4
   SEND DIALLER, 1,1234,1,301
IF AREA IN ACCESS, 1
   BUZZER CONTROL, 2,3,10
ENDIF
ENDIF
IF ALARM EVENT OFF, 4
   SEND DIALLER, 1,1234,3,301
ENDIF
```

IF ALARM EVENT OFF, 4 –[MAINS FAIL RESTORED]

Purpose: The mains power has been restored.

Type: Event

Available for:

☑ Alarm

Description: The mains power has restored as previously failed.

It may be appropriate to notify the monitoring station of this event.

Parameters: Alarm Event Number

See also: IF ALARM EVENT ON, 4

Example:

The following will notify the monitoring company if the mains fails or is restored. It will also activate the sounder in the RAS (assuming device 2) if the power fails when the area is occupied.

```
IF ALARM EVENT ON, 4
   SEND DIALLER, 1,1234,1,301
IF AREA IN ACCESS, 1
   BUZZER CONTROL, 2,3,10
ENDIF
ENDIF
IF ALARM EVENT OFF, 4
   SEND DIALLER, 1,1234,3,301
ENDIF
```

IF ALARM EVENT ON, 5 –[BATTERY LOW ALARM]

Purpose: The battery's voltage is too low.

Type: Event

Available for:

☑ Alarm

Description: The battery's voltage has fallen below 11.2 volts.

Parameters: Alarm Event number,

See also: IF ALARM EVENT OFF, 5

Please note: Normally detection devices, such as PIR, will generate alarm condition. It is very important; Low Battery alarm is treated with high priority, ensuring system proper operation. Please check battery to ensure its proper condition.

Example:

The following will send Battery low and Battery normal events to the monitoring station.

```
IF ALARM EVENT ON, 5
   SEND DIALLER 1,123,1,302
ENDIF
IF ALARM EVENT OFF, 5
   SEND DIALLER 1,123,3,302
ENDIF
```

IF ALARM EVENT OFF, 5 –[BATTERY NORMAL]

Purpose: The battery voltage has returned to within normal limits.

Type: Event

Available for:

☑ Alarm

Description: The battery voltage has returned to normal limits. The Battery Normal is reported, when Battery voltage has reach 13 volts.

Parameters: Alarm Event Number

See also: IF ALARM EVENT ON, 5

Please note: Normally detection devices, such as PIR, will generate alarm condition. It is very important; Low Battery alarm is treated with high priority, ensuring system proper operation. When mains are restored, the Battery normal may be generated. Please check battery to ensure its proper condition.

Example:

The following will send Battery low and Battery normal events to the monitoring station.

```
IF ALARM EVENT ON, 5
   SEND DIALLER 1,123,1,302
ENDIF
IF ALARM EVENT OFF, 5
   SEND DIALLER 1,123,3,302
ENDIF
```

IF ALARM EVENT ON, 6 –[POWER FUSE 1 or 2 FAIL]

Purpose: Triggered when fuse 1 or 2 fails (F1, F2).

Type: Event

Available for:

☑ Alarm

Description: Triggered when Power fuse 1 or 2 (fuse F1 and F2) fails. This indicates that the 12v outputs for detectors have failed. Consequently all detectors will not operate. All inputs will be unsealed, which may result in alarm condition, or prevent user to secure they Area.

Parameters: Alarm Event Number

See also: IF ALARM EVENT OFF, 6

Example:

When Fuse Failure or restore is detected, system report event to the Dialler.

```
IF ALARM EVENT ON, 6
        SEND DIALLER, 1,1234,1,300
ENDIF
IF ALARM EVENT OFF, 6
        SEND DIALLER, 1,1234,3,300
ENDIF
```

IF ALARM EVENT OFF, 6 –[POWER FUSE 1 or 2 RESTORED]

Purpose: Triggered when fuse 1 or 2 restored (F1, F2).

Type: Event

Available for:

☑ Alarm

Description: Triggered when Power fuse 1 or 2 (fuse F1 and F2) has return to normal condition.

Parameters: Alarm Event Number

See also: IF ALARM EVENT ON, 6

Example:

When Fuse Failure or restore is detected, system report event to the Dialler.

```
IF ALARM EVENT ON, 6
        SEND DIALLER, 1,1234,1,300
ENDIF
IF ALARM EVENT OFF, 6
        SEND DIALLER, 1,1234,3,300
ENDIF
```

IF ALARM EVENT ON, 7 –[SIREN FUSE 1 or 2 FAIL]

Purpose: Triggered when Siren fuse 1 or 2 fails (F3, F5).

Type: Event

Available for:

☑ Alarm

Description: Triggered when Siren fuse 1 or 2 (fuse F3 and F5) fails. This indicates that the outputs for Sirens have failed. This could be due to faulty Siren, or cable has been tampered with.

Parameters: Alarm Event Number

See also: IF ALARM EVENT OFF, 7

Example:

When Fuse Failure or restore is detected, system reports events to the Dialler.

```
IF ALARM EVENT ON, 7
        SEND DIALLER, 1,1234,1,320
ENDIF
IF ALARM EVENT OFF, 7
        SEND DIALLER, 1,1234,3,320
ENDIF
```

IF ALARM EVENT OFF, 7 –[SIREN FUSE 1 or 2 RESTORED]

Purpose: Triggered when Siren fuse 1 or 2 restored (F3, F5).

Type: Event

Available for:

☑ Alarm

Description: Triggered when Siren fuse 1 or 2 (fuse F3 and F5) has return to normal condition.

Parameters: Alarm Event Number

See also: IF ALARM EVENT ON, 7

Example:

When Fuse Failure or restore is detected, system reports events to the Dialler.

```
IF ALARM EVENT ON, 7
        SEND DIALLER, 1,1234,1,320
ENDIF
IF ALARM EVENT OFF, 7
        SEND DIALLER, 1,1234,3,320
ENDIF
```

IF ALARM EVENT ON, 8 –[STROBE FUSE 1 or 2 FAIL]

Purpose: Triggered when Strobe fuse 1 or 2 fails (F4, F6).

Type: Event

Available for:

☑ Alarm

Description: Triggered when Strobe fuse 1 or 2 (fuse F4 and F6) fails. This indicates that the outputs for Strobes have failed. This could be due to faulty Strobe, or cable has been tampered with.

Parameters: Alarm Event Number

See also: IF ALARM EVENT OFF, 8

Example:

When Fuse Failure or restore is detected, system reports events to the Dialler.

```
IF ALARM EVENT ON, 8
        SEND DIALLER, 1,1234,1,321
ENDIF
IF ALARM EVENT OFF, 8
        SEND DIALLER, 1,1234,3,321
ENDIF
```

IF ALARM EVENT OFF, 8 –[STROBE FUSE 1 or 2 RESTORED]

Purpose: Triggered when Strobe fuse 1 or 2 restored (F4, F6).

Type: Event

Available for:

☑ Alarm

Description: Triggered when Strobe fuse 1 or 2 (fuse F4 and F6) has return to normal condition.

Parameters: Alarm Event Number

See also: IF ALARM EVENT ON, 8

Example:

When Fuse Failure or restore is detected, system reports events to the Dialler.

```
IF ALARM EVENT ON, 8
        SEND DIALLER, 1,1234,1,321
ENDIF
IF ALARM EVENT OFF, 8
        SEND DIALLER, 1,1234,3,321
ENDIF
```

IF ALARM EVENT ON, 9 [DIALLER ALARM]

Purpose: Triggers if there is a problem sending a message to a monitoring station.

Type: Event

Available for:

☑ Alarm

Description: Triggers when the system is unable to send a message to the monitoring station. This could be due to cut lines, disconnected leads, or repeated busy signals. ***This error code is generated, when Dialler reaches maximum number of retries***, as programmed in the Dialler box (see Dialler programming data).

The DIALLER ERROR event is a good opportunity to start a local response to the problem as well as attempting alternative dial out procedures. For example, local sirens could be initiated to try and attract local attention. Switch to second line for Dialler could be used to dial out.

Parameters: Alarm Event number

See also: IF ALARM EVENT OFF, 9

Please note: Dialler error could be generated due to number of faults:

1/ Line is missing or cut
2/ Phone number is engaged
3/ Wrong phone number

Example:

The following script will turn OUTPUT 10 ON, when Dialler Error is detected. This output is pulsed, to ensure better attention on this error.

```
IF ALERT EVENT ON, 9
        OUTPUT CONTROL, 10,4, 0             // Fast Pulse
ENDIF
IF ALERT EVENT OFF, 9
        OUTPUT CONTROL, 10,1, 0             // Switch it off
ENDIF
```

NOTE: Do not SEND DIALLER MESSAGES, when Dialler Error has been detected.

IF ALARM EVENT OFF, 9 [DIALLER ALARM RESTORED]

Purpose: Triggers if Dialler has restored communicated with monitoring station.

Type: Event

Available for:

☑ Alarm

Description: Triggers when the system has restored communication to the monitoring station.

Parameters: Alarm Event number

See also: IF ALARM EVENT ON, 9

Example:

The following script will turn LED C (programmed as output 10) on so that the user knows that there has been a problem. It will remain on until the error is restored.

```
IF ALERT EVENT ON, 9
        OUTPUT CONTROL, 10,3, 0
ENDIF
IF ALERT EVENT OFF, 9
        OUTPUT CONTROL, 10,1, 0
ENDIF
```

NOTE: Do not SEND DIALLER MESSAGES, when Dialler Error has been detected.

IF ALARM EVENT ON, 10 [SIU ALARM]

Purpose: Triggers if Securitel has detected an error and lost communication.

Type: Event

Available for:

☑ Alarm

Description: Triggers when the system lost communication with Securitel interface unit.

Parameters: Alarm Event number

See also: IF ALARM EVENT OFF, 10

Example:

The following script will turn LED B (programmed as output 9) on so that the user knows that there has been a problem. It will remain on until the error is restored.

```
IF ALERT EVENT ON, 10
        OUTPUT CONTROL, 9,3, 0
ENDIF
IF ALERT EVENT OFF, 10
        OUTPUT CONTROL, 9,1, 0
ENDIF
```

NOTE: Do not SEND SECURITEL MESSAGES, when Securitel Error has been detected.

IF ALARM EVENT OFF, 10 [SIU ALARM RESTORED]

Purpose: Triggers if Securitel has restored communication to the monitoring station.

Type: Event

Available for:

☑ Alarm

Description: Triggers when the system has restored communication with Securitel interface unit.

Parameters: Alarm Event number

See also: IF ALARM EVENT ON, 10

Example:

The following script will turn LED B (programmed as output 9) on so that the user knows that there has been a problem. It will remain on until the error is restored.

```
IF ALERT EVENT ON, 10
        OUTPUT CONTROL, 9,3, 0
ENDIF
IF ALERT EVENT OFF, 10
        OUTPUT CONTROL, 9,1, 0
ENDIF
```

NOTE: Do not SEND SECURITEL MESSAGES, when Securitel Error has been detected.

IF ALERT EVENT ON, 1 [DEVICE RESET]

Purpose: The device unit has been reset.

Type: Event

Available for:

☑ Alert

Description: The device has generated reset by the reset switch, power has been applied or software command.

The RESET event should be used to do any initial processing that may be required.

Parameters: Alert Event number;

See also:

Please note: Alert event are generated, when the error is generated, and does not generates the opposite conditions. There are no restores, or off event status for alert events. When Dialler command is used, system will automatically add device number into Dialler message.

Example:

The following alert script send message to Dialler, when device has generated reset.

```
IF ALERT EVENT ON, 1
   SEND DIALLER 1,123,1,305
ENDIF
```

IF ALERT EVENT ON, 2 [TIME CHANGED]

Purpose: The PC or RAS has changed the system time.

Type: Event

Available for:

☑ Alert

Description: The PC or RAS has changed the system time. This means that there may have been a significant change in the time.

Parameters: Alert Event number,

See also:

Example:

The following code will make a RAS (device 2) beeper sound for 10 seconds. This would be an indication that a significant change is being made to the system.

```
IF ALERT EVENT ON, 2
   BUZZER CONTROL, 2,2,10
ENDIF
```

IF ALERT EVENT ON, 3 [SERVICE REQUEST]

Purpose: A service request has been made from a RAS.

Type: Event

Available for:

☑ Alert

Description: A service request has been made from a RAS.

Parameters: Alert Event Number,

See also:

Example:

The following script will send Dialler message to the monitoring station, requesting service call.

```
IF ALERT EVENT ON, 3
        SEND DIALLER 1,123,1,411
ENDIF
```

IF ALERT EVENT ON, 4 [BATTERY TEST STARTED]

Purpose: Triggered when the battery test has started.

Type: Event

Available for:

☑ Alert

Description: System indicates battery test has been activated.

Parameters:

See also:

Please note: This function is not available on present software.

Example:

```
IF ALERT EVENT ON, 4
        OUTPUT CONTROL, 1,9,3,20
ENDIF
```

IF ALERT EVENT ON, 5 [REMOTE ACCESS]

Purpose: Indicates, remote access has been enabled.

Type: Event

Available for:

☑ Alert

Description: Remote access has been granted.

Parameters:

See also:

Example:

Activate output, to indicate this event.

```
IF ALERT EVENT ON, 5
        OUTPUT CONTROL, 9,3,20
ENDIF
```

IF ALERT EVENT ON, 6 [LOG RESET]

Purpose: Indicates, Log file has been cleared.

Type: Event

Available for:

☑ Alert

Description: All events in the log file have been cleared.

Parameters:

See also:

Example:

Activate output, to indicate this event.

```
IF ALERT EVENT ON, 6
        OUTPUT CONTROL, 9,3,20
ENDIF
```

IF ALERT EVENT ON, 7 [POWER DOWN REQUEST]

Purpose: Indicates, Power down has been requested.

Type: Event

Available for:

☑ Alert

Description: System is shutting down.

Parameters:

See also:

Example:

Activate Buzzer, to indicate this event.

```
IF ALERT EVENT ON, 7
        BUZZER CONTROL, 4,3,20
ENDIF
```

SEND DIALLER, dialler, subscriber, type, code

Purpose: Send a Contact ID message to the monitoring station.

Type: Control

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Door

Description: Sends the Contact ID message to the monitoring station using pre-defined values as well as the dynamic components. That is, the event type and code are predefined, but the actual point is dynamically defined.

Parameters: Dialler is the physical Dialler that will be used. Subscriber is the customer identification for the monitoring station. Both of these settings must be configured in the database first.

The event type provides information on the status of the information. The three event types available are:

New alarm: The alarm has just occurred and is active. This event type may require an immediate response.
New restore: The alarm has returned to its normal state. Typically this means that there has been a response to the alarm.
Previously reported: The alarm has been reported is previous report.

For any alarm, there are typically two messages that must be sent to the monitoring company. One indicates the start of the problem, and the other indicates the end. All the information is identical except for the type.

The codes are broken up into a number of categories. The category, code and meaning are shown on the following page.

See also: SEND SECURITEL, GLOBAL SCRIPTS

Example:

PIR detects movement:
SEND DIALLER 1,12,**1**,130
The alarm is then reset
SEND DIALLER 1,12,**3**,130

Duress code is entered:
SEND DIALLER 1,12,**1**,121
Duress alarm is reset:
SEND DIALLER 1,12,**3**,121

Unit goes offline:
SEND DIALLER 1,12,**1**,354
Unit returns online:
SEND DIALLER 1,12,**3**,354

CONTACT ID CODES

Category	Code	Meaning
Medical (10x)	100	Personal
	101	Pendant
	102	Fail to check in
Fire (11x)	110	Alarm
	111	Smoke detector
	112	Combustion
	113	Water flow
	114	Heat sensor
	115	Pull station
	116	Duct sensor
	117	Flame sensor
	118	Near alarm
Panic (12x)	120	Alarm
	121	Duress
	122	Silent
	123	Audible
Burglar (13x)	130	Burglary
	131	Perimeter
	132	Interior
	133	24 Hour
	134	Entry/exit
	135	Day/night
	136	Outdoor
	137	Tamper
	138	Near alarm
Alarm (14x)	140	General
	141	Poll loop open
	142	Poll loop short
	143	LAN device fail
	144	Input tamper
	145	Device tamper
24 Hour non-burglary (15x)	150	Alarm
	151	Gas detected
	152	Refrigeration
	153	Heating system
	154	Water leakage
	155	Foil break
	156	Day zone
	157	Low gas level
	158	High temperature
	159	Low temperature
	161	Air flow
Supervisor (20x)	200	Fire supervisor
	201	Low water pressure
	202	Low CO_2
	203	Gate valve
	204	Low water level
	205	Pump activation
	206	Pump failure
Trouble (30x)	300	System
	301	Mains fail
	302	Low battery
	303	Bad RAM checksum
	304	Bad ROM checksum
	305	System reset
	306	Program altered
	307	Self test fail
	308	System shutdown
	309	Battery test fail

Category	Code	Meaning
24 Hour	310	Ground fault
Trouble	320	Siren/output
	321	Bell/siren #1
	322	Bell/siren #2
	323	Alarm relay
	324	Trouble relay
	325	Reversing relay
	330	Peripheral
	331	Poll loop open
	332	Poll loop short
	333	LAN device fail
	334	Repeater fail
	335	Printer paper
	336	Local printer
	350	Communications
	351	Phone line #1
	352	Phone line #2
	353	Radio transmitter
	354	Fail to communicate
	355	Radio supervisor
	356	Radio polling
	370	Protection loop
	371	Protection loop open
	372	Protection loop short
	373	Fire loop
	380	Sensor
	381	RF Sensor
	382	RPM Sensor
	383	Sensor tamper
	384	RF Sensor battery
Closing (40x)	400	Closing
	401	By user
	402	Group/user
	403	Automatic
	404	Late
	405	Deferred
	406	Cancel
	407	Remote
	408	Quick-arm
	409	Key switch
Remote (41x)	411	Callback request
	412	Access OK
	413	No access
	414	System shutdown
	415	Dialler disable
Access (42x)	421	No access, user
	422	Access, user
Disable (52x)	520	Siren output
	521	Bell/siren #1
	522	Bell/siren #2
	523	Alarm relay
	524	Trouble relay
	525	Reversing relay
	551	Dialler
	552	Radio
Test (60x)	601	Manual trigger
	602	Periodic
	603	Periodic radio
	604	Fire
	605	Status follows
	606	Listen-in on
	607	Walk test

SEND SECURITEL,priority, event, value

Purpose: FUNCTION IMPLEMENTED ONLY IN BUILD 91 or HIGHER

Type: Control

Available for:

☑ Input
☑ Area

Description: Sends message to SECURITEL Device. ***Please note: Optional Chip set must be installed on the Master Device.***

Parameters: ,priority,event,value

See also: SEND DIALLER, GLOBAL SCRIPTS

Example:

Area Script:

```
IF AREA EVENT ON ACCESS
        SEND SECURITEL,1,49,0
ENDIF
IF AREA EVENT ON SECURE
        SEND SECURITEL,1,81,0
ENDIF
```

Input Script:

```
IF INPUT EVENT ON RESET
        SEND SECURITEL,1,1,0
        ALARM OFF
ENDIF
IF INPUT EVENT ON UNSEAL
        SEND SECURITEL,1,2,0
        ALARM ON
ENDIF
```

Please note: When value is set to 0, system automatically adds the Area number to the Event code and value is replaced with the Areaor Sector number. If value is set to value between 1-255, system will send the Event code, unchanged, and the specified value.

Genesis is using channels, ***which are available on poll command from monitoring station only***, and are assigned as follows:

Channel 01: Any user duress alarm is active in the system
Channel 02: Any alarm is active in the system
Channel 03: Any assigned devices are offline
Channel 04: Any tamper alarm is active in the system
Channel 05: Any mains failures are present in the system
Channel 06: Any lowbat battery alarm in active the system
Channel 07: Any input is isolated
Channel 08: Any area in access mode
Channel 09: Master panel generated system reset

All other Channels are not used by the Genesis system.

ELSE

Purpose: Provides an alternative action for the associated IF statement.

Type: Function

Available for:

☑ Input
☑ Area
☑ Clock
☑ Alert
☑ Alarm
☑ Access
☑ Door
☑ Control

Description: An IF statement provides either a true or false result. The code immediately after it is only executed (run) if the result is true. The ELSE statement provides the ability to execute some code if the result is false.

Parameters: None.

See also: ENDIF, GLOBAL SCRIPTS, THIS INPUT FLAG CONTROLS, INPUT TEST CONTROL,

The else statement is optional. An' IF` statement must always have a matching 'ENDIF` statement. For nested 'IF` statements (IF statements within other IF statements), the 'ELSE` is always linked with the last 'IF` statement, which has not been terminated by the 'ENDIF' statement.
Care should be taken to ensure that the 'ENDIF` is matched with the correct 'IF` statement.
We do not recommend, that ELSE command be USED on EVENT commands, unless you have lots of experience in Scripting.

Example: AREA SCRIPT

For example, in the following code, there is multiple IF statements and else statements. Line numbers are included to simplify the explanation.

```
1. IF AREA EVENT ON ACCESS
2. IF CLOCK VALID, 1
3. IF CLOCK VALID, 2
4. <<a>>
5. ENDIF
6. ELSE
7. <<b>>
8. ENDIF
9. ELSE
10. <<c>>
11. ENDIF
```

The only way that the script will get to <<a>> on line 4 is if the ON ACCESS event occurred (line 1), and clock 1 is valid (line 2), and clock 2 is valid (line 3). <<b>> will be executed if it was an ON ACCESS event (line 1) and clock 1 was not valid (line 2 and 6). The ELSE statement on line 6 was linked to the IF statement on line 2 because the ENDIF statement on line 5 "closed" the IF statement on line 3.

The <<c>> code on line 10 is executed if it was not an ON ACCESS event.

RETURN

Purpose: Terminates script.

Type: Function

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: If RETURN is used in a script, system will terminate and returns without actioning any futher commands.

Parameters: None.

See also: ENDIF, ELSE**,** GLOBAL SCRIPTS, THIS INPUT FLAG CONTROLS, INPUT TEST CONTROL,

The else statement is optional. An' IF` statement must always have a matching

Example: INPUT SCRIPT

For example, in the following code, return is used to ignore any futher commands.

```
IF INPUT EVENT ON UNSEAL            // Unsealed event
        IF ACCESS                   // Are we in Access
                RETURN              // Yes, return
        ENDIF
        ALARM ON
ENDIF
```

ENDIF

Purpose: End to IF/ELSE statement.

Type: Function

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Every IF statement must have a matching END statement. The ELSE statement is optional.

Parameters:

See also: ELSE, GLOBAL SCRIPTS, THIS INPUT FLAG CONTROLS, INPUT TEST CONTROL,

All the code between the IF statement and the ELSE will be executed if the IF statement is true. All the code between the ELSE and END statement will be executed if the IF statement was false. If there is no ELSE statement, all the code between the IF and END statement will be executed if the IF statement was true.

Example: CLOCK SCRIPT

It is possible to include additional IF statements within the IF ELSE END statements. This is called nesting. For example:

```
IF CLOCK EVENT TO VALID
IF AREA IN ALARM,1
<<a>>
ENDIF
ENDIF
```

The code <<a>> will only be executed if both the clock event was TO VALID **and** area 1 is IN ALARM.

TRACE OFF

Purpose: Turns the diagnostic tracing off within a Script.

Type: Debug

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Script commands will not be recorded in the trace log.

Parameters:

See also:

Example: TRACE ON, GLOBAL SCRIPTS

```
TRACE ON
IF INPUT EVENT ON UNSEAL
ENDIF
TRACE OFF
```

TRACE ON

Purpose: Turns the diagnostic tracing on within a Script.

Type: Debug

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description: Script commands will be recorded in the trace log and should only be used when requested by the Genesis Trained personnel.

Parameters:

See also: TRACE OFF, GLOBAL SCRIPTS

Example:

```
TRACE ON
IF INPUT EVENT ON UNSEAL


ENDIF
TRACE OFF
```

IF TIME IS, condition, hour, minute

Purpose: Allows conditional test to system clock.

Type: Debug

Available for:

- ☑ Input
- ☑ Area
- ☑ Clock
- ☑ Alert
- ☑ Alarm
- ☑ Access
- ☑ Door
- ☑ Control

Description:.

Parameters:

See also: GLOBAL SCRIPTS

Value of condition will represent test as bellow:

1 – LESS
2 - LESS OR EQUAL
3 – EQUAL
4 - GREATER OR EQUAL
5 - GREATER

Example: System clock is tested, if time is less that 12:30, and if so, system activates buzzer on device 1.

Input Script:

```
IF INPUT EVENT ON UNSEAL
        IF TIME IS,1,12,30
                BUZZER CONTROL,1,2,5            // Buzzer
        ENDIF
ENDIF
```

Sample scripts

Alarm Input Script

// Are used to give explanation of the scripts and are never inserted in real script. Color is used to indicate start and stop of each 'IF' statement. Please note, that each section starts with 'IF INPUT EVENT' command and individual Control or Test commands are within the start and end of the 'EVENT' command. Sections of the script are separated by <line> and are only used in example for easier explanations.

Input Script: Standard Secure Script, using Dialler to Report all changes

TAMPER alarm is generated at any time
ALARM is generated ONLY, when INPUT is in SECURE MODE and system detected UNSEAL status,
ISOLATE Message is generated, when USER Isolate input
REMOVE ISOLATION is generated, when Input is changed to ACCESS mode, or USER has remove isolation.
BYPASS Message is generated, when USER/TECHNICAN Bypass the input.
REMOVE BYPASS Message is generated ONLY, when USER/TECHNICIAN removes Bypass.
ALARM and TAMPER Restore is generated, when USER or SYSTEM generate Reset to ALARMED/TAMPER input.

```
IF INPUT EVENT ON RESET                 // TAMPER EVENT
IF ALARM ON                             // IS INPUT IN ALARM
      SEND DIALLER,1,1234,3,140         // SEND ALARM RESTORAL
      ALARM OFF                         // CLEAR ALARM LATCH
ENDIF
IF TAMPER                               // IS INPUT IN TAMPER ALARM
      SEND DIALLER,1,1234,3,137         // SEND TAMPER RESORAL
      TAMPER OFF                        // CLEAR TAMPER LATCH
ENDIF
ENDIF

IF INPUT EVENT ON ACCESS                // Access EVENT
      ISOLATE OFF                       // Remove Isolation
ENDIF

IF INPUT EVENT ON ISOLATE               // Isolation EVENT
      SEND DIALLER,1,1234,1,570         // Send Dialler Isolation
ENDIF

IF INPUT EVENT ON ISOLATE OFF           // Isolation off EVENT
      SEND DIALLER,1,1234,3,570         // Send Restore
ENDIF

IF INPUT EVENT ON BYPASS                // Bypass EVENT
      SEND DIALLER,1,1234,1,572         // Send Dialler Bypass
ENDIF

IF INPUT EVENT ON BYPASS OFF            // Bypass off EVENT
      SEND DIALLER,1,1234,3,572         // Send Restore
ENDIF

IF INPUT EVENT ON TAMPER                // TAMPER EVENT
IF TAMPER
        RETURN                              // ALREADY IN TAMPER, IGNORE
ENDIF
        TAMPER ON                           // LATCH TAMPER ON
        SEND DIALLER,1,1234,1,137           // SEND TAMPER TO DIALLER
ENDIF

IF INPUT EVENT ON UNSEAL                    // UNSEAL EVENT
IF ENABLED                                  // Input Normal
IF SECURE                                   // IS input in SECURE mode
IF ALARM ON                                 // Already in ALARM
        RETURN                              // Ignore
ENDIF
        ALARM ON                            // Latch ALARM ON
        SEND DIALLER,1,1234,1,140           // Send Dialler ALARM
ENDIF
ENDIF
ENDIF
```

Input Script: Standard LOCAL Alarm Secure Script, NOT using Dialler

TAMPER alarm is generated at any time
ALARM is generated ONLY, when INPUT is in SECURE MODE and system detected UNSEAL status,
ISOLATE Message is IGNORED.
REMOVE ISOLATION is IGNORED.
BYPASS Message is IGNORED.
REMOVE BYPASS Message is IGNORED.
ALARM and TAMPER Restore is generated, when USER or SYSTEM generate Reset to ALARMED/TAMPER input.

```
IF INPUT EVENT ON RESET                 // TAMPER EVENT
IF ALARM ON                             // IS INPUT IN ALARM
      ALARM OFF                         // CLEAR ALARM LATCH
ENDIF
IF TAMPER                               // IS INPUT IN TAMPER ALARM
      TAMPER OFF                        // CLEAR TAMPER LATCH
ENDIF
ENDIF

IF INPUT EVENT ON ACCESS                // Access EVENT
      ISOLATE OFF                       // Remove Isolation
ENDIF

IF INPUT EVENT ON TAMPER                // TAMPER EVENT
IF TAMPER
      RETURN                            // ALREADY IN TAMPER, IGNORE
ENDIF
      TAMPER ON                         // LATCH TAMPER ON
      ENDIF

IF INPUT EVENT ON UNSEAL                // UNSEAL EVENT
IF ENABLED                              // Input Normal
0IF SECURE                              // IS input in SECURE mode
IF ALARM ON                             // Already in ALARM
      RETURN                            // Ignore
ENDIF
      ALARM ON                          // Latch ALARM ON

ENDIF
ENDIF
ENDIF
```

Input Script: Standard EXIT/ENTRY Secure Script, using Dialler to Report all changes

TAMPER alarm is generated at any time
ALARM is generated ONLY, when INPUT is in SECURE MODE and system detected UNSEAL status,
ISOLATE Message is generated, when USER Isolate input
REMOVE ISOLATION is generated, when Input is changed to ACCESS mode, or USER has remove isolation.
BYPASS Message is generated, when USER/TECHNICAN Bypass the input.
REMOVE BYPASS Message is generated ONLY, when USER/TECHNICIAN removes Bypass.
ALARM and TAMPER Restore is generated, when USER or SYSTEM generate Reset to ALARMED/TAMPER input.
EXIT TIMER is started, when Input has been changed into SECURE mode.
When EXIT timer Expire and Input is unsealed, generate alarm, don't wait for Entry timer to run. (LFWS code could be send).
When ENTRY timer is started, BUZZER on DEVICE 2 is activated. If Input is not set to ACCESS mode within ENTRY time, an Alarm is generated.

```
IF INPUT EVENT ON RESET                 // TAMPER EVENT
IF ALARM ON                             // IS INPUT IN ALARM
      SEND DIALLER,1,1234,3,140         // SEND ALARM RESTORAL
      ALARM OFF                         // CLEAR ALARM LATCH
ENDIF
IF TAMPER                               // IS INPUT IN TAMPER ALARM
      SEND DIALLER,1,1234,3,137         // SEND TAMPER RESORAL
      TAMPER OFF                        // CLEAR TAMPER LATCH
ENDIF
ENDIF

IF INPUT EVENT ON ACCESS                // Access EVENT
      ISOLATE OFF                       // Remove Isolation
      STOP INPUT TIMER,1                // Must STOP EXIT timer
      STOP INPUT TIMER,2                // Must STOP ENTRY timer
ENDIF

IF INPUT EVENT ON SECURE                // Secure EVENT
IF ENABLED
      START THIS INPUT TIMER,1,60             // Start 60 Secs EXIT timer
ENDIF
ENDIF

IF INPUT EVENT ON TIMER EXPIRE          // Timer Expired EVENT
IF THIS INPUT TIMER EXPIRED,2
      ALARM ON                          // Generate ALARM
      SEND DIALLER,1,1234,1,140         // Send Dialler ALARM
ENDIF
IF THIS INPUT TIMER EXPIRED,1
      IF UNSEALED                       // is input UNSEALED
            SEND DIALLER,1,1234,1,140   // Can send Left Without Seal, if you wish
            ALARM ON                    // Set ALARM
      ENDIF
ENDIF
ENDIF

IF INPUT EVENT ON ISOLATE               // Isolation EVENT
```

```
      SEND DIALLER,1,1234,1,570     // Send Dialler Isolation
ENDIF

IF INPUT EVENT ON ISOLATE OFF       // Isolation off EVENT
      SEND DIALLER,1,1234,3,570     // Send Restore
ENDIF

IF INPUT EVENT ON BYPASS            // Bypass EVENT
      SEND DIALLER,1,1234,1,572     // Send Dialler Bypass
ENDIF

IF INPUT EVENT ON BYPASS OFF        // Bypass off EVENT
      SEND DIALLER,1,1234,3,572     // Send Restore
ENDIF

IF INPUT EVENT ON TAMPER            // TAMPER EVENT
IF TAMPER
      RETURN                        // ALREADY IN TAMPER, IGNORE
ENDIF
      TAMPER ON                     // LATCH TAMPER ON
      SEND DIALLER,1,1234,1,137     // SEND TAMPER TO DIALLER
ENDIF

IF INPUT EVENT ON UNSEAL            // UNSEAL EVENT
IF SECURE                           // IS input in SECURE mode
IF ENABLED
IF ALARM ON                         // Already in ALARM
      RETURN                        // Ignore
ENDIF
IF TNIS INPUT TIMER ON, 1           // Are we in EXIT?
      RETURN                        // Ignore
ENDIF
IF THIS INPUT TIMER ON,2            // Entry Timer Running ??
      RETURN
ELSE
      START THIS INPUT TIMER,2,30         // Start ENTRY timer
      BUZZER CONTROL,2,3,30         // Activate ENTRY BUZZER
ENDIF
ENDIF
ENDIF
ENDIF
```

Input Script: Standard EXIT/ENTRY LOCAL Secure Script, NOT using Dialler.

TAMPER alarm is generated at any time
ALARM is generated ONLY, when INPUT is in SECURE MODE and system detected UNSEAL status,
ALARM and TAMPER Restore is generated, when USER or SYSTEM generate Reset to ALARMED/TAMPER input.
EXIT TIMER is started, when Input has been changed into SECURE mode.
When EXIT timer Expire and Input is unsealed, generate alarm, don't wait for Entry timer to run.
When ENTRY timer is started, BUZZER on DEVICE 2 is activated. If Input is not set to ACCESS mode within ENTRY time, an Alarm is generated.

```
IF INPUT EVENT ON RESET                 // TAMPER EVENT
IF ALARM ON                             // IS INPUT IN ALARM
      ALARM OFF                         // CLEAR ALARM LATCH
ENDIF
IF TAMPER                               // IS INPUT IN TAMPER ALARM
      TAMPER OFF                        // CLEAR TAMPER LATCH
ENDIF
ENDIF

IF INPUT EVENT ON ACCESS                // Access EVENT
      ISOLATE OFF                       // Remove Isolation
      STOP THIS INPUT TIMER,1           // Must STOP EXIT timer
      STOP THIS INPUT TIMER,2           // Must STOP ENTRY timer
ENDIF

IF INPUT EVENT ON SECURE                // Secure EVENT
IF ENABLED
      START THIS INPUT TIMER,1,60             // Start 60 Secs EXIT timer
ENDIF
ENDIF

IF INPUT EVENT ON TIMER EXPIRE          // Timer Expired EVENT
IF THIS INPUT TIMER EXPIRED,2
      ALARM ON                          // Generate ALARM
ENDIF
IF THIS INPUT TIMER EXPIRED,1
      IF UNSEALED                       // is input UNSEALED
            ALARM ON                    // Set ALARM
      ENDIF
ENDIF
ENDIF

IF INPUT EVENT ON TAMPER                // TAMPER EVENT
IF TAMPER
      RETURN                            // ALREADY IN TAMPER, IGNORE
ENDIF
      TAMPER ON                         // LATCH TAMPER ON
      ENDIF

IF INPUT EVENT ON UNSEAL                // UNSEAL EVENT
IF SECURE                               // IS input in SECURE mode
IF ENABLED
IF ALARM ON                             // Already in ALARM
      RETURN                            // Ignore
```

```
ENDIF
IF INPUT TIMER ONFLAG, 1                    // Are we in EXIT?
      RETURN                                // Ignore
ENDIF
IF INPUT TIMER ON,2                         // Entry Timer Running ??
      RETURN
ELSE
      START THIS INPUT TIMER,2,30                 // Start ENTRY timer
      BUZZER CONTROL,2,3,30                 // Activate ENTRY BUZZER
ENDIF
ENDIF
ENDIF
ENDIF
```

Input Script: Standard ACCESS Script, using Dialler to Report all changes

TAMPER alarm is generated at any time
ALARM is generated ONLY, when INPUT is in ACCESS MODE and system detected UNSEAL status,
ISOLATE Message is generated, when USER Isolate input
REMOVE ISOLATION is generated, when Input is changed to ACCESS mode, or USER has remove isolation.
BYPASS Message is generated, when USER/TECHNICAN Bypass the input.
REMOVE BYPASS Message is generated ONLY, when USER/TECHNICIAN removes Bypass.
ALARM and TAMPER Restore is generated, when USER or SYSTEM generate Reset to ALARMED/TAMPER input.

```
IF INPUT EVENT ON RESET              // TAMPER EVENT
IF ALARM ON                          // IS INPUT IN ALARM
      SEND DIALLER,1,1234,3,120      // SEND ALARM RESTORAL
      ALARM OFF                      // CLEAR ALARM LATCH
ENDIF
IF TAMPER                            // IS INPUT IN TAMPER ALARM
      SEND DIALLER,1,1234,3,137      // SEND TAMPER RESORAL
      TAMPER OFF                     // CLEAR TAMPER LATCH
ENDIF
ENDIF

IF INPUT EVENT ON ACCESS             // Access EVENT
      ISOLATE OFF                    // Remove Isolation
ENDIF

IF INPUT EVENT ON ISOLATE            // Isolation EVENT
      SEND DIALLER,1,1234,1,570      // Send Dialler Isolation
ENDIF

IF INPUT EVENT ON ISOLATE OFF        // Isolation off EVENT
      SEND DIALLER,1,1234,3,570      // Send Restore
ENDIF

IF INPUT EVENT ON BYPASS             // Bypass EVENT
      SEND DIALLER,1,1234,1,572      // Send Dialler Bypass
ENDIF

IF INPUT EVENT ON BYPASS OFF         // Bypass off EVENT
      SEND DIALLER,1,1234,3,572      // Send Restore
ENDIF

IF INPUT EVENT ON TAMPER             // TAMPER EVENT
IF TAMPER
      RETURN                         // ALREADY IN TAMPER, IGNORE
ENDIF
      TAMPER ON                      // LATCH TAMPER ON
      SEND DIALLER,1,1234,1,137      // SEND TAMPER TO DIALLER
ENDIF

IF INPUT EVENT ON UNSEAL             // UNSEAL EVENT
IF ACCESS                            // is input in ACCESS mode
IF ALARM ON                          // Already in ALARM
      RETURN                         // Ignore
ENDIF
```

```
          ALARM ON                                // Latch ALARM ON
          SEND DIALLER,1,1234,1,120               // Send Dialler ALARM
ENDIF
ENDIF
```

Input Script: Standard 24 HOUR Script, using Dialler to Report all changes

TAMPER alarm is generated at any time
ALARM is generated any time, independent of input status, and system detected UNSEAL status,
ISOLATE Message is generated, when USER Isolate input
REMOVE ISOLATION is generated, when Input is changed to ACCESS mode, or USER has remove isolation.
BYPASS Message is generated, when USER/TECHNICAN Bypass the input.
REMOVE BYPASS Message is generated ONLY, when USER/TECHNICIAN removes Bypass.
ALARM and TAMPER Restore is generated, when USER or SYSTEM generate Reset to ALARMED/TAMPER input.

```
IF INPUT EVENT ON RESET                 // TAMPER EVENT
IF ALARM ON                             // IS INPUT IN ALARM
      SEND DIALLER,1,1234,3,133         // SEND ALARM RESTORAL
      ALARM OFF                         // CLEAR ALARM LATCH
ENDIF
IF TAMPER                               // IS INPUT IN TAMPER ALARM
      SEND DIALLER,1,1234,3,137         // SEND TAMPER RESORAL
      TAMPER OFF                        // CLEAR TAMPER LATCH
ENDIF
ENDIF

IF INPUT EVENT ON ACCESS                // Access EVENT
      ISOLATE OFF                       // Remove Isolation
ENDIF

IF INPUT EVENT ON ISOLATE               // Isolation EVENT
      SEND DIALLER,1,1234,1,570         // Send Dialler Isolation
ENDIF

IF INPUT EVENT ON ISOLATE OFF           // Isolation off EVENT
      SEND DIALLER,1,1234,3,570         // Send Restore
ENDIF

IF INPUT EVENT ON BYPASS                // Bypass EVENT
      SEND DIALLER,1,1234,1,572         // Send Dialler Bypass
ENDIF

IF INPUT EVENT ON BYPASS OFF            // Bypass off EVENT
      SEND DIALLER,1,1234,3,572         // Send Restore
ENDIF

IF INPUT EVENT ON TAMPER                // TAMPER EVENT
IF TAMPER
      RETURN                            // ALREADY IN TAMPER, IGNORE
ENDIF
      TAMPER ON                         // LATCH TAMPER ON
      SEND DIALLER,1,1234,1,137         // SEND TAMPER TO DIALLER
ENDIF

IF INPUT EVENT ON UNSEAL                // UNSEAL EVENT
IF ALARM ON                             // Already in ALARM
      RETURN                            // Ignore
ENDIF
```

```
        ALARM ON                              // Latch ALARM ON
        SEND DIALLER,1,1234,1,133             // Send Dialler ALARM
ENDIF
```

Input Script: Standard ACCESS DELAY/ INSTANT Secure Script, using Dialler to Report all changes

TAMPER alarm is generated at any time
ALARM is generated ONLY, when system detects UNSEALS status, and then, if in Access the delay TIMER is started, and instant Alarm is Generated in Secure mode.
ISOLATE Message is generated, when USER Isolate input
REMOVE ISOLATION is generated, when Input is changed to ACCESS mode, or USER has remove isolation.
BYPASS Message is generated, when USER/TECHNICAN Bypass the input.
REMOVE BYPASS Message is generated ONLY, when USER/TECHNICIAN removes Bypass.
ALARM and TAMPER Restore is generated, when USER or SYSTEM generate Reset to ALARMED/TAMPER input.
If input is unsealed, the DELAY timer is started; BUZZER on DEVICE 2 is activated. If Timer expires, an Alarm is send to Dialler. If Input is reset, the Delay timer is stopped.
Please note: Ensure ALARM ON flag is used, otherwise RESET will not be generated, and allowing stop to delay timer, if active.

```
IF INPUT EVENT ON RESET                 // TAMPER EVENT
IF ALARM ON                             // IS INPUT IN ALARM
IF INPUT FLAG,1
      SEND DIALLER,1,1234,3,122         // SEND ALARM RESTORAL
ENDIF
      CLEAR THIS INPUT FLAG,1           // Clear flag
      ALARM OFF                         // CLEAR ALARM LATCH
      STOP INPUT TIMER,1                // Ensure, we stop delay
ENDIF
IF TAMPER                               // IS INPUT IN TAMPER ALARM
      SEND DIALLER,1,1234,3,137         // SEND TAMPER RESORAL
      TAMPER OFF                        // CLEAR TAMPER LATCH
ENDIF
ENDIF

IF INPUT EVENT ON ACCESS                // Access EVENT
      ISOLATE OFF                       // Remove Isolation
      STOP INPUT TIMER,1                // Must STOP EXIT timer
ENDIF

IF INPUT EVENT ON TIMER EXPIRE          // Timer Expired EVENT
      SET THIS INPUT FLAG,1             //Indicate alarm has been send
      SEND DIALLER,1,1234,1,122         // Send Dialler ALARM
ENDIF

IF INPUT EVENT ON ISOLATE               // Isolation EVENT
      SEND DIALLER,1,1234,1,570         // Send Dialler Isolation
ENDIF

IF INPUT EVENT ON ISOLATE OFF           // Isolation off EVENT
      SEND DIALLER,1,1234,3,570         // Send Restore
ENDIF

IF INPUT EVENT ON BYPASS                // Bypass EVENT
      SEND DIALLER,1,1234,1,572         // Send Dialler Bypass
ENDIF

IF INPUT EVENT ON BYPASS OFF            // Bypass off EVENT
      SEND DIALLER,1,1234,3,572         // Send Restore
```

```
ENDIF

IF INPUT EVENT ON TAMPER                    // TAMPER EVENT
IF TAMPER
        RETURN                              // ALREADY IN TAMPER, IGNORE
ENDIF
        TAMPER ON                           // LATCH TAMPER ON
        SEND DIALLER,1,1234,1,137           // SEND TAMPER TO DIALLER
ENDIF

IF INPUT EVENT ON UNSEAL                    // UNSEAL EVENT
IF ACCESS                                   // IS input in ACCESS mode
IF ALARM ON                                 // Already in ALARM
        RETURN                              // Ignore
ENDIF
        START THIS INPUT TIMER,1,30              // Start 30 Secs DELAY timer
        ALARM ON                            // LATCH ALARM
        BUZZER CONTROL,2,3,30               // Activate ENTRY BUZZER
ELSE
        ALARM ON                            // Instant alarm in Secure
        SEND DIALLER,1,1234,1,122           // SEND ALARM
ENDIF
ENDIF
```

Area script

The purpose of the standard area script is to automatically dial out on changes. Normally the ALARM is GENERATED within the AREA SCRIPT. Here we have the option to activate different outputs, when ALARM or TAMPER ALARM has been generated.
Normally, if we require RE-TRIGABLE alarms, the alarm outputs are activated in the INPUT SCRIPTS and not in AREA Script.

```
IF AREA EVENT ON ACCESS
   OUTPUT CONTROL,20,2,0        // Give some indication
   SEND DIALLER,1,1234,1,402
ENDIF

IF AREA EVENT ON SECURE
   OUTPUT CONTROL,20,1,0        // Give some indication
   SEND DIALLER,1,1234,3,402
ENDIF

IF AREA EVENT ON ALARM
   OUTPUT CONTROL,1,2,600       // Activate Siren
   OUTPUT CONTROL,2,2,0         // Activate Strobe
ENDIF

IF AREA EVENT ON TAMPER
   OUTPUT CONTROL,3,2,600       // Activate Siren 2
   OUTPUT CONTROL,2,2,0         // Activate Strobe
ENDIF

IF AREA EVENT ON RESET
   OUTPUT CONTROL,1,1,0         // Stop Siren
   OUTPUT CONTROL,3,1,0         // Stop Siren 2
   OUTPUT CONTROL,2,1,0         // Stop Strobe
ENDIF
```

SYSTEM REPORT:

We have taken special consideration, to supply you with maximum events, which should to help you. Number of events can be enabled or disabled via programming menu. Additional command 'LOG EVENT,x' can be added into each script, providing selective reporting. Error events generated due to programming errors are always reported.

Example of logs Events.
Please note: Date and Event numbers are not sequance. As they are example only:

```
[17:59:41][25,01,02][000119] AUDIT Event=Dialler Report [4],Inf=000,Parm [1,0,0],Ecode=731 [731-Dialler not programmed,]
[23:52:38][25,01,02][000264] CLOCK 1,Event= Clock Validated [3],Inf=000,Src=0,Wait=0
[23:52:38][25,01,02][000265] AUDIT Event=Error Event [1],Inf=000,Parm [1011,2,0],Ecode=277 [277-Invalid Script Number,]
[23:52:38][25,01,02][000266] ALARM Event=Unit On/Off Line [1],Inf=000,Node=0,OnOff=On [1],Unit=2,ADC=0
[23:52:39][25,01,02][000267] AUDIT Event=Error Event [1],Inf=000,Parm [1011,2,0],Ecode=277 [277-Invalid Script Number,]
[19:12:03][17,01,02][000311] ALARM Event=Power Fuse Event [6],Inf=001,Node=1,OnOff=On [1],Unit=1
[19:13:24][17,01,02][000313] DOOR Event Door=1,Event=Door Closed [9],Inf=001,Src=0
[19:06:03][17,01,02][000291] AUDIT Event=Error Event [1],Inf=000,Parm [1053,1,0],Ecode=277 [277-Invalid Script Number,]
[19:06:03][17,01,02][000292] DOOR Event Door=1,Event=Door Forced Alarm [10],Inf=000,Src=0
```

```
[23:52:38][25,01,02][000263] AUDIT Event=Error Event [1],Inf=000,Parm [1042,1,0],Ecode=319 [319-Script not found,]
```

[23:52:38]	- Time of the event
[25,01,02]	- Day,Month, Year
[00263]	- Event Number
AUDIT Event=Error Event [1]	- Error Number
Inf=000,Parm [1042,1,0],	- Information for Genesis staff
Ecode=319 [319-Script not found,]	- Error Code

In this example, the Parm[0,119,115] indicate the reason, why this Group reject has happened.

```
[20:16:29][25,01,02][000015] AUDIT Event=Error Event [1],Inf=000,Parm [0,119,115],Ecode=906 [906-Group reject,]
```

[17:35:24][30,01,02][000101] AUDIT Event=Dialler Report [4],Inf=000,Parm [**1**,1,0],Ecode=723 [723-All Dialler Messages been send,]

[**1**,1,0]	First Dialler List

There are 10 types of reports:

1/ Input events
- Input Reset
- Input Set to Access
- Input Set to Secure
- Input Tamper Alarm
- Input Unsealed
- Input Sealed
- Input Timer Expired
- Input Isolated
- Input De-Isolated
- Input Bypassed
- Input Bypass removed
- Input Alarm ON
- Input Alarm OFF

2/ Output events
- Output OFF
- Output ON
- Output Slow
- Output Fast
- Output Tone 1
- Output Tone 2
- Output Tone 3

3/ Door events
- Door Set to Normal

- Door Lock entry
- Door Lock Entry,Exit
- Door Unlock
- Door Release
- Door Switch Bypass ON
- Door Switch Bypass OFF
- Door Opened
- Door Closed
- Door Forced Alarm
- Door DOTL Alarm
- Door Switch Tamper
- Door Timer Expired

4/ Area events
- Area Reset
- Area Accessed
- Area Secured
- Area Alarm
- Area Tamper Alarm
- Area Unsealed
- Area Sealed
- Area Timer Expired
- Area Isolation ON
- Area Isolation OFF
- Area Bypass ON
- Area Bypass OFF

5/ Clock events
- Clock Enabled
- Clock Disable
- Clock Validated
- Clock Void

6/ Access events
- User Door Access
- User Menu Access
- User Access Denied
- Card Door Access
- Card Access Allowed
- Card Door Access Denied
- Card Presented
- User Duress
- Duress Cancelled
- User 100/101 Access

7/ Alarm events
- Unit On/Off Line
- Duress activated
- Cabinet Tamper
- Mains Event
- Battery Event
- Power Fuse Event
- Siren Fuse Event
- Strobe Fuse Event
- Dialler Event
- Securitel Event
- Modem Event
- Ras Lock/Unlock Event
- Unit Enable/disable Event
- Dialler Enable/Disabled
- Siu Enable/Disable

8/ Alert events
- System Reset
- Time Changed

- Service Request
- Battery Test
- Remote Access
- Event Log Reset
- Power Off Request

9/ Audit events

- Error Event
- Buffer Fault
- Script Runtime Error
- Dialler Report
- Securitel Report

10/ Script events

- Input Script
- Area Script
- Clock Script
- Door Script
- Alarm Script
- Alert Script
- Access Script
- Control Script

Please see 'Technical Manual` for description of ERROR CODES.

Fault finding scripts

There are a number of sources of errors from scripts. They include:

- The wrong script is assigned to the input, area, etc. This is best avoided by using self-explanatory names that clearly indicate their usage.
- The panel has not been reset after changes have been made.
- The code does not flow as expected. Check the IF-ENDIF pairs to ensure that the code is executed only when it is meant to. See the Flow Control section for more details.
- The wrong output is triggered. Check the code and use unique, self explanatory names for all outputs.

Most of the common mistakes are easily fixed by ensuring that a good naming system is used throughout the system.

Care should be taken when using multiple IF statements within each other ("nesting" is the technical term for this). When trying to find a problem with a script's logic, try placing extra OUTPUT statements within the script. The output should drive the RAS LED or briefly drive a siren or strobe. They are physical confirmations that the script has reached a certain point.

We have provided large number of LOG EVENTS, which will indicate majority of script or system programming errors in the system.

We urge you to view your log, as we have included over 200 error messages, providing you with the maximum help.

Happy Scripting.

Script summary

The following table groups events, commands and tests by their type. The second column provides a brief description and the final column displays the page number for the detailed explanation.

When viewing the PDF version of this document online, selecting the underlined commands will display the detailed information.

Fault finding

A logical and systematic approach to fault finding should be developed. The benefits of such an approach include:

- Rapid identification of the symptoms
- Saved time on not doing unnecessary tests
- The problem can be fixed sooner
- Less time on site saving the customer money and allowing the technician to service other customers
- A better reputation for the product and the company
-
- There are three fundamental steps to fault finding:
- Testing for symptoms
- Identifying the possible causes of the fault based upon the results
- Repeating the process to narrow in on the fault

Many technicians rely upon their experience with the product to quickly locate and fix the problem. Technicians often know about fundamental problems with a product or site. This approach provides quick solutions to many problems but relies upon the technician having seen the fault. When a new fault is encountered, some technicians do not know how to quickly test the system.

Identifying the symptoms

The purpose of identifying symptoms is to determine what is working and what is not working. Quite often identifying what is operational is just as critical as identifying what is not operational.

For example, a customer complains that no PINs are working in a RAS. In fact some PINs are working in the RAS (e.g. the technician PIN), but all of his department’s PINs are not working. In this case the problem is likely to be with the group programming, timer lists or holidays. The initial assumption that no PINs are working would lead the technician in the wrong direction of fault finding.

It is helpful to describe the fault in terms of “none”, “some”, “most”, and “all”. As shown in the above example, there is a big difference between “all” and “most” PINs not working.

The large number of diagnostic tools within the Genesis software package makes many tests quick and simple.

At this stage experience often helps in identifying known problem spots and the symptoms to look out for.

Identify the causes

Once some of the symptoms have been identified, start trying to work out what would cause the symptoms.

Causes do not necessarily have to be faults. Such issues as database programming, scripts and over rides should be considered. The most obvious example is that an isolated detector will not create an alarm.

Possible causes include (but are not limited to):

- Wiring
- Power
- Device failure

- User overrides
- Database programming
- Communication
- Database/RAM limitations
- Environment

Additional testing

Once a number of possible causes have been identified, additional tests should be done with the *aim of distinguishing between faults*. A test that could be due to any of the possible causes or none of them is useless.

Once one possible cause has been identified, additional testing may be necessary to confirm it. This can save wasted time, effort and money if the fix is expensive or time consuming.

If there are causes that have quick fixes, then doing the fix is often faster and more cost effective then confirming the fault.

If changing the system, only one change should be made at a time. The system should then be checked immediately after the change. If the change does not fix the problem, then the system should be restored back to its original condition if possible, and another change made and tested.

Making too many changes to the system in one go makes it difficult to pinpoint the actual fault. The fault may be either covered up but still exist, or may return. By clearly identifying the cause, steps can be done to avoid the problem in the future.

A “divide and conquer” approach should be taken. That is, tests should be conducted so as to rule out half the available options. This is usually faster then testing one option at a time. This is especially important for faults such as communication problems where the site is large and it takes time to break the communication run.

Resources

The following resources are intended to give an idea as to how to locate symptoms and test the system. It is not intended to be an exhaustive list. Many of the resources are not required on all sites.

Any test that does not interfere with the operation of the system should be considered before tests that do obstruct the system. For example, most of the software tests can be done without altering the operation of the system, but testing the loop resistance requires disconnecting wires and possibly turning the system off.

Software

Reset
Doing a reset before further fault finding could remove the problem immediately.
Do not use a RESET, unless essential.

Unit status
The first check that should be done is to ensure that all units are on-line. A split system, or a system with off-line devices, will not operate completely.

A device can be off-line in software if:
- It is not connected
- The communication line is broken
- The serial number is incorrect in the database
- The device has no power

- The device has been damaged and is not working properly

Other tests must be run to determine why the device is offline.

The first step in maintaining a site is to ensure that all devices are on-line.

Input status
The input states provide information on what the system thinks are the states of the input.

Particular attention should be paid to any inputs that are isolated or in tamper.

If the inputs are in the correct state it is often advisable to ensure that the correct script has been assigned to the areas and inputs.

Scripts
Scripts provide a huge amount of flexibility. The standard scripts also allow the technician to quickly setup a standard site.

If, however, a script has been created or modified, careful consideration of the consequences, and complete testing should be completed.

The ability to search the scripts to see which ones drive particular outputs provides a very powerful fault finding tool. For example, if a script accidentally drives the wrong output, the search function will be able to find it.

For building automation applications, the input logic can become more complex. This is particularly true when multiple scripts are used to control the same outputs. In this case, the state of the outputs may not depend upon the states of the inputs, but rather the order that the inputs were activated.

Hardware

LEDs provide instant and usually reliable information on the status of the hardware.

Links are limited with Genesis, but the few links within the system can cause faults if incorrectly set. When changing boards care should be given to ensuring the link settings are duplicated.

Connectors should be inspected as well as the cables being inserted. It is not uncommon to have a broken cable or a connector that is not quite on.

Physical damage such as water damage, scrape marks and smoke residue either on or around the equipment can identify possible causes for the fault.

Equipment

The **multimeter** is essential for many faults. Where possible, voltage tests should be done. Voltage testing is generally a non-intrusive test and the system can operate normally.
Good technician have always mutlimeter and good set of tools. For example, right size screwdriver will reduce damage to equipment.

Intrusive testing such as resistance and current testing should only be done where voltage testing is not sufficient. They require cables to be removed and so impact on the operation of the system.

A **portable oscilloscope** is an expensive but powerful device. It is very useful for tracking down communication problems. It is recommended that a company should invest in at least one once they employ a number of technicians.

A **screwdriver** or **side cutters** can also come in handy when connecting or disconnecting equipment.

People

Talk to the building manager, user, department or company manager and the secretary as well as the occupants. They can often provide information on who else has been on site, past problems and changes that have occurred to the system.

A logbook should also be kept so that past problems are recorded and the information is available to the technician.

Periodic meetings should be held with technicians to ensure that they are up to date with their knowledge of the product and their approach to fault finding.

Key questions

At times it is necessary to seek additional help. Technical Support typically asks the following questions. Answers as many of these questions as possible before calling Technical Support will increase the amount of assistance that they can provide.

Scope of the problem

What devices are affected?

How many inputs are affected?

How many doors are affected?

How many users are affected?

How many cards are affected?

Does the PC communicate with the system?

Are any other systems experiencing problems (e.g. computers, BMS, fire, lights etc)?

History

When was the system last working correctly?

What hardware has been replaced or modified recently?

What database changes have occurred recently?

What user actions have occurred recently?

Who else has been working on site (e.g. on the BMS, fire, or access control systems, electricians, plumbers etc)?

Has this problem occurred before?

What other problems have been experienced recently?

Reproducibility

How can the problem be demonstrated on the system?

Can the problem by demonstrated on another site or in the office?

Does the problem occur at a specific time, regular intervals or with another event?

System information

What versions of hardware, firmware and software are used on site?

What are the PC specifications?

How old is the system?

Specific examples

Device(s) off-line

A device can be off-line in software if:

- It is not connected
- The communication line is broken
- The serial number is incorrect in the database
- The device has no power
- The device had locked up

Typically if more than one device is offline the cause will be either a broken or disconnected comms lead, or power failure. Knowledge of the cabling routes is essential for a quick repair. For communication cabling faults, the problem will be between an on-line device and an off-line device.

For a power failure, the fault could be at any of the devices (i.e. the one with the power supply). Power failure should be easy to diagnose since no LEDs will be on.

If additional RS-485 devices have been added to the system, and they are not true RS-485, it is possible that they can cause null points on the bus. This fault is very difficult to trace and will often provide misleading or contradicting symptoms. For example, extending or shortening the bus could cause devices along the run to come on or offline.

Another common problem with communications is to leave the end of line resistors in for additional devices. If end of line termination is used, it must only be used on the first and last device.

User/card can not do anything

The typical causes are:

- The user or card has expired
- The user does not need to enter their user ID first (user IDs < 100) or the user must enter their user ID first (user IDs > 100)
- The card does not have the correct site code
- The group does not have any access
- The timers associated with the group are void (not between the start and stop times)

Many users/cards can not do anything

The most common causes are the group programming and holidays.

Other possible causes are:

- The associated clock list has been modified
- The time/date has been changed

User can not access an area

The user can only view an area on a RAS if:

- The RAS' area list contains the area
- The group assigned to the user has access to the group
- The clock list associated with the group's area access is valid (i.e. between the start and end time)

The first two can be checked within the database. The timer list can be tested using the diagnostic screens.

No one can open a door

The typical causes are:

- A holiday is active
- The door has been locked by the PC or script
- The door is physically jammed
- The reader is faulty
- The strike or lock is faulty
- Power has been lost to the controller or strike
- The strike has jammed. Try pushing the door before opening it
- The time or date are wrong

With this particular fault it is important to know if anyone (that is, at least one person) can access the door, or if literally no one can open the door. If no one can open the door, the problem is typically a hardware related problem. This, however, cannot be automatically assumed. That is why the above list includes many database related causes.

A script is not working as expected

The quick checks are:

- The correct script has been assigned to the inputs, areas etc
- The system has been reset after the changes have been downloaded
- The input is not isolated

Common scripting mistakes include:

- Closing an IF statement at the wrong place and so code is run at the wrong time
- Driving the wrong output
- Not considering the impact of isolating inputs

Scripts involving nested IF statements (IF statements within IF statements) can be checked by printing out the script and joining each IF statement with its ENDIF statement. Remember that each ENDIF is associated with the previous unassociated IF statement.

To test scripts it is possible to use quick checks. These include turning RAS LEDs on or off, turning buzzers on or even driving a siren or strobe briefly. This can confirm what is happening within the script.

Index