ALARM ABSOLON

genesis

Bezpečnostní systém
Kontrola vstupu
Domácí automatizace
Automatizace budov

Stručný popis skriptů

Verze 2.0 (Build 134)

genesis

Úvodní poznámky:

Úvod

Vyhrazujeme sio právo na změny bez předchozího upozornění. Veškeré zde uvedené informace jsou platné a správné ke dni vydání tohoto manuálu. Neneseme žádnou odpovědnost za následky vzniklé používáním toho manuálu.

Poznámky

Úvod

Základní znalosti

Tento manuál předpokládá Vaše základní znalosti o systémech Elektronické zabezpečovací signalizace (EZS) a systémech kontroly přístupu (ACCESS). Doporučujeme zúčastnit se odborného školení, aby jste plně porozuměli systému Genesis.

Po instalaci a naprogramování systému Genesis musíte vykonat zkoušku všech funkcí v systému. Jedině tak zaručíte správnou funkci systému za všech okolností.

Vyhrazujeme si právo na změny bez předchozího upozornění.

Skripty

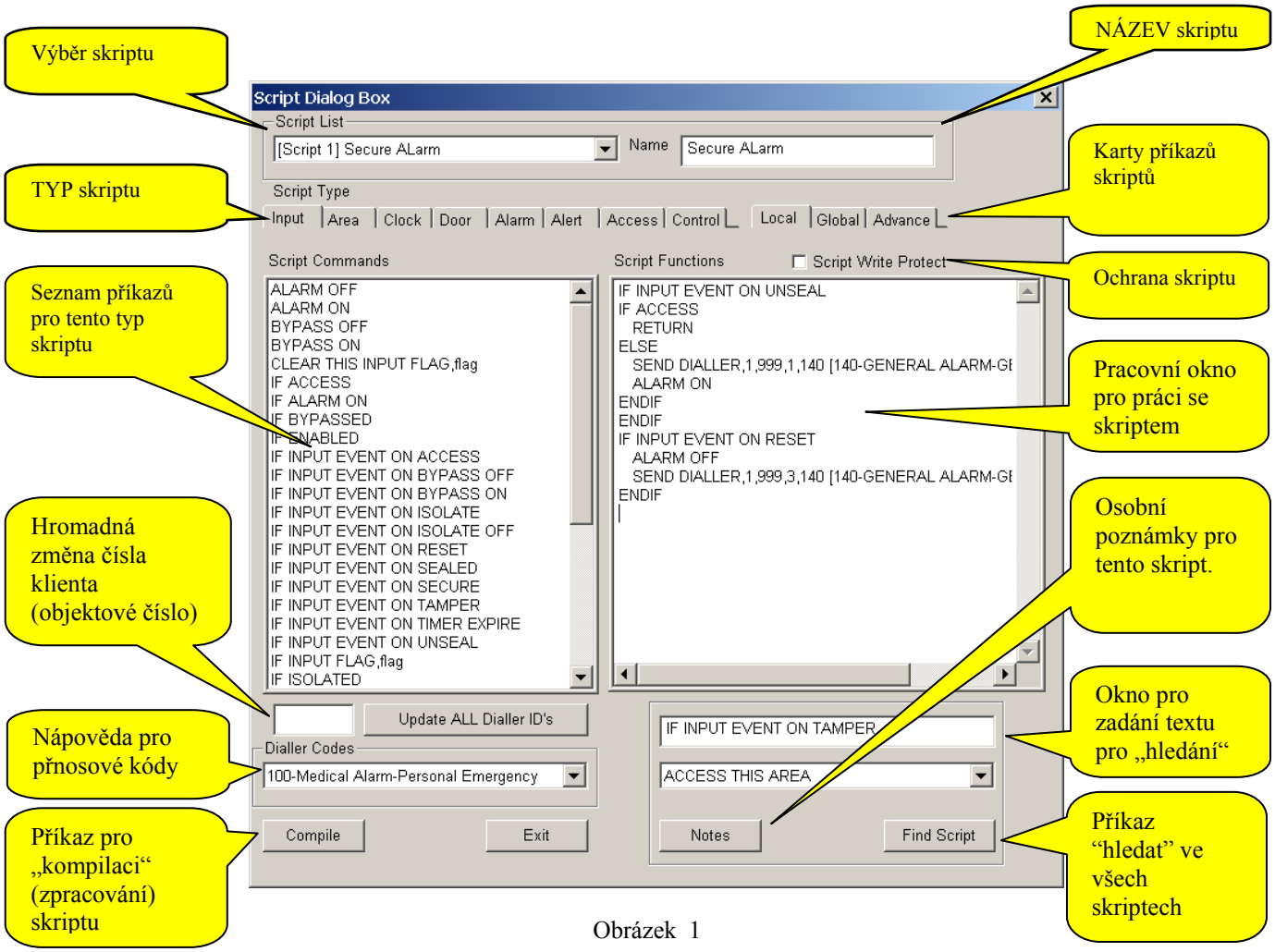
Pro běžné instalace může instalační technik použít standrdní skripty a nemusí detailně znát funkci skriptů. Pro rozsáhlejší a speciální aplikace je možno skripty modifikovat nebo psát nové. Doporučujeme každý skript před jeho použitím řádně vyzkoušet.

Skripty

Skripty jsou funkcí, která dělá ze systému Genesis naprosto unikátní systém který je schopen nabídnou téměř absolutní flexibilitu.
Skripty umožňují systému Genesis splnit jak běžné požadavky na funkci systému, tak i velmi specifické a náročné požadavky na funkce a vlastnosti.
Skripty jsou vlastně malé podprogramy napsané vlastním programovacím jazykem a tento způsob programování je velmi blízký programování osobních počítačů nebo programovatelných kalkulaček.

Programovací jazyk pro skripty je velmi jednoduchý. Je velmi podobný programovacím jazykům jako jsou C, Basic, Pascal, Delphi nebo Java. Věříme tomu, že velmi brzo porozumíte programovacímu jazyku skriptů. Nejsou ani nutné nějaké Vaše předchozí zkušenosti s těmito programovacími jazyky.

Okno programování skriptů



Obrázek 1

Výběr skriptu

Seznam všech skriptů použitelných v systému (mezi 100 – 300, v závislosti na konfiguraci systému). Skripty jsou zobrazeny včetně jejich názvu.

Název skriptu

Okno pro zadání libovolného názvu pro každý skript.

TYP skriptu

Typ skriptu určuje o jaký druh skriptu se jedná. Typ skriptu definuje, kde může být konkrétní skript použit a jaké příkazy mohou být použity. Nicméně většina příkazů může být použita ve všech typech skriptů.

Karty příkazů

Jakmile je vybrána karta příkazů “local”, jsou v seznamu příkazů zobrazeny pouze související příkazy skriptu, pokud je vybrána karta “global”, jsou zobrazeny všechny dostupné příkazy pro skripty.

Seznam příkazů

V seznamu příkazů jsou zobrazeny všechny dostupné příkazy. Dvojitým kliknutím levým tlačítkem myši. Pokud je vyžadován doplňující údaj (například číslo oblasti), zobrazí před kopírováním okno pro zadávání požadovaných údajů.

Pracovní okno

V této pravé části pracujete se skriptem tak, jak bude dále uvedeno. Jedná se vlastně o pracovní plochu podobně jako v textovém editoru.

Hromadná změna čísla klienta

Tato funkce slouží k velmi snadnému a splehlivému nastavení čísla klienta (resp. objektové číslo) pro komunikaci na PCO. Funkce najde všechny příkazy pro komunikaci ve skriptech a provede požadovanou změnu čísla klienta u všech položek.

Zpracování skriptu

Funkce zpracování skriptu provede synaktickou (formální) kontrolu skriptu a v případě, že je vše v pořádku, provede jeho začlenění do systému.

Hledání skriptu

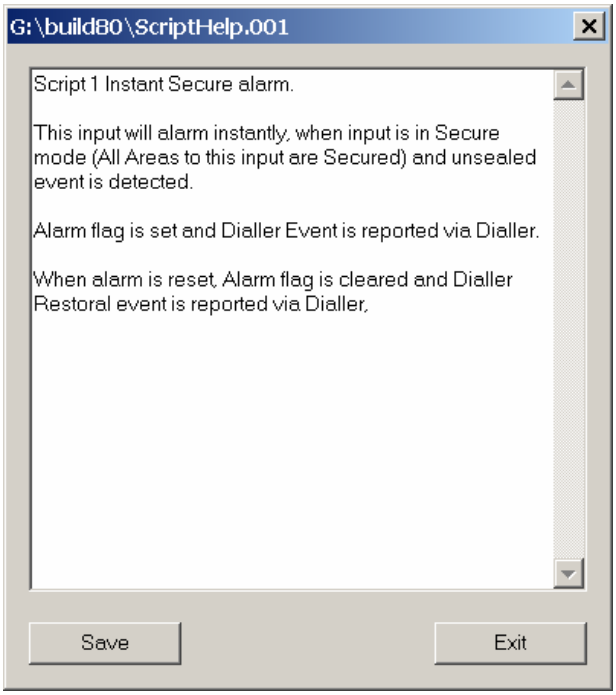
Funkce hledání skriptu umožňuje nalézt všechny skrity náležející k oblastem, výstupu, dveřím nebo vstupu. Toto je velmi užitečný nástroj pro určení co ovládá určitý bod.

Všiměte si, že funkce hledání neumožňuje nalézt co je ovládáno. Jednotlivé vstupy nebo zařízení musí být zobrazeny pro určení jaký skript je používá.

Poznámky

Funkce poznámky umožní připojit ke každému skriptu Vaši poznámky. Toto je velmi užitečné jak pro budoucnost (aby jste si připomněli, s jakým úmyslem byl skript vytvořen), ale také pro Vaše kolegy, kteří snáze pochopí smysl a funkci skriptu. Každá poznámka (nápověda ke skriptu) je uložena do souboru v aktuálním adresáři společně s databází – pod názvem ‘ScriptHelp.xxx`, kde xxx je číslo skriptu.

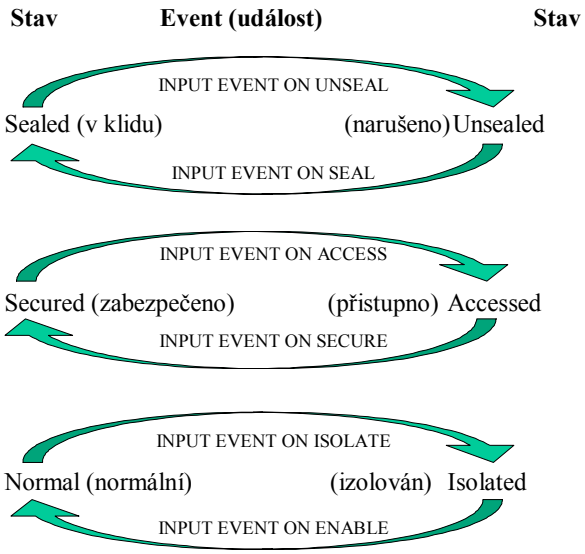
Příklad:



Výuka

Následující příklady představují základní myšlenky při programování se skripty. Příklady musíte prostudovat postupně, protože každý následující příklad předpokládá znalost předchozího.

Tato výuka Vám vysvětlí postup při vytváření nových skriptů a jak skripty modifikovat. Předpokládáme, že již znáte způsob programování ostatních částí systému Genesis (např. jednotky, hodiny, oblasti, uživatele). Viz "Technický programovací manuál".



Jakmile dojde ke změně stavu vstupu (například z "zabezpečeno" na "přístupno") nastane příslušná **událost** (EVENT) (INPUT EVENT ON ACCESS v tomto případě).

Tato metoda umožňuje programovat vlastnosti systému přesně tak, jak jsou požadovány. Tato flexibilita bude vysvětlena na následujících příkladech.

Událost "INPUT EVENT ON UNSEAL" může být použita například pro spouštění klimatizace. V tomto případě vstup nebude spouštět poplach.

Na dalším vstupu může být událost "INPUT EVENT ON UNSEAL" použita pro standardní bezpečnostní funkci, to znamená, pokud je vstup aktivován (SECURED) bude spouštět poplach.

Nějaký další vstup při události "INPUT EVENT ON UNSEAL" může aktivovat/deaktivovat (SECURE/ACCESSED) vstup nebo podsystém – funkce standardního klíčového ovladače (keyswitch).

V těchto třech příkladech je ukázáno, že skript může využívat stejnou **událost** (EVENT) pro různé funkce.

Síla systému Genesis vychází z možnosti spojovat skripty s téměř každou možnou událostí. Sytém je tak zcela programovatelný a pod kontrolou.

Ve skriptu je možno kontrlovat stav ostatních vstupů, oblastí, hodin apod. Toto umožňuje psát příkazy odpovídající těmto změnám, ale také s ohledem na stav ostatních částí systému. Pro spuštění chodu skriptu je zapotřebí vznik nějaké události, ale jeho chod může být závislý na dalších událostech, nebo stavech systému.

Každý SKRIPT, pokud nastane "událost" (EVENT) a pokud jsou jsou požadovány nějaké společné funkce může obsahovat "skryté" (*HIDDEN)* vstupy, oblasti, časovače, uživatele, zařízení, skupiny nebo číslo dveří.

TYPY SKRIPTU

Pro poskytnutí maximální flexibility systému je většina funkcí řízena prostřednictvím SKRIPTŮ. To, jak bude systém reagovat na jednotlivé události je závislé na tom, jak jsou skripty napsány. Je nezbytně nutné porozumět všem příkazůmpoužívaným ve skriptech.
Doporučujeme provádět testování systému, aby byla zaručena nejen správná činnost jednotlivých příkazů ve skriptech, ale i činnost celých skriptů a systému jako celku.
Každý SKRIPT obsahuje množství požadovaných UDÁLOSTÍ, OVLÁDACÍ a TESTOVACÍ funkce ale nikdy nezapomeňte na to, že prostřednictvím skriptů můžete ovlivňovat prakticky veškeré vlastnosti systému.

Řízení průběhu

Řízení průběhu ukazuje, jak bude řadič postupovat ve skriptu. Zejména bude rozhodovat, které řádky budou zařazeny a které přeskočeny.

Dosud byly používány instrukce IF a ENDIF. Dále bude vysvětlena třetí instrukce a to ELSE.

Příklad: SKRIPT TYPU AREA (OBLAST)

```
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
```

Nepřehlédněte:

1) Každý příkaz ve skriptu začíná s instrukcí **‘EVENT`** (v příkladu nahoře zvýrazněné červenou barvou). V každé EVENT mohou být další dodatečné instrukce pro ovládání nebo testování stavů systému.
2) Každá ‘EVENT` je zpracována jako samostatný příkaz.
3) Nepoužívejte příkaz ‘ELSE’ v instrukcích ‘EVENT’, dokud plně nezvládnete programovací jazyk skriptů.
4) Vždy používejte příkazy ‘TEST` a ‘CONTROL’ uzavřené v instrukci ‘EVENT`.
5) Každý příkaz ‘IF’ **MUSÍ** končit příkazem ‘ENDIF’.

Jestliže je oblast deaktivována (access), zapne se výstup (output) ovládající klimatizaci. Navíc, jestliže Hodiny 2 (CLOCK 2) jsou neplatné (např. mimo pracovní dobu), sepne se navíc výstup ovládající osvětlení.
Jestliže je oblast aktivována (secure) a pokud jsou Hodiny 1 neplatné (když se světla zapnula automaticky) výstup (OUTPUT) ovládající světla se vypne. Navíc také pokud jsou Hodiny 3 neplatné (když se klimatizace zapnula automaticky) tak se výstup ovládající klimatizaci vypne.

Všimněte si, že poslední dvě Hodiny (CLOCK) sledují dvě podmínky: jestli jsou platné a jestli je oblast aktivována. Oboje Hodiny jsou na sobě nezávislé, ale obě současně sledují stejnou podmínku AREA EVENT ON SECURE jestli je pravdivá.

Doporučujeme zachovávat výše uvedené seskupování ve skriptech při psaní nových skriptů i při dešifrování skriptů, které napsal někdo jiný.

Instrukce ‘IF` umožňuje skriptu buď pokračovat na následujících řádcích, nebo přeskočení na odpovídající instrukci 'ENDIF'. Jestliže je příslušná podmínka splněna, skript bude pokračovat na dalších řádcích. Jestliže podmínka není splněna, následující řádky budou přeskočeny. Instrukce ENDIF definuje kolik řádek bude pokračovat nebo kolik řádek bude přeskočeno.

Instrukce ELSE poskytuje alternativy. Například pokud se ráno někdo rozhoduje: “Pokud svítí slunce, tak si vezmu klobouk, jinak si vezmu deštník”. Instrukce IF pak má ve svém řešení dvě různé činnosti (akce).

Jestliže jsou více než dvě možné podmínky, je nutno zadat instrukci IF vícenásobně. Například skript pro oblast má čtyři možné stavy. Pro určení, která událost nastala jsou možné dva způsoby napsání skriptu:

Příklad ‘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
```

nebo

Příklad ‘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
```

Oba skripty dosáhnou zcela shodného výsledku. Oba jsou správné a každý z nich má své příznivce. Nejdůležitější je, aby kódům porozumněli i další lidé. Někdy je lepší při psaní skriptů používat více řádek méně komplikovaných a snadněji zapamatovatelných. Doporučujeme psát skripty spíše podle Příkladu “a” než podle Příkladu “b”.

Logika

V souvislosti s řízením průběhu je nutno provádět rozhodování založené na rozličných faktorech.

Cílem Genesis je zachovávat skripty jednoduché a snadno pochopitelné. Proto byla instrukce IF omezena na jednoduchý test. Nicméně je možné dosáhnout pomocí této struktury komplexních výsledků.

Použití instrukce **AND** je snadné. Každá instrukce AND znamená, že výsledek je pravdivý pouze tehdy, jsou-li pravdivé oba testy.

Například “jestliže prší AND (a) jdu ven, pak potřebuju deštník”. Deštník není zapotřebí, jestliže neprší a jdu ven, jestliže prší a nejdu ven, jestliže neprší a nejdu ven. Deštník potřebuji pouze když prší a současně jdu ven.

Instrukci AND dosáhneme řadíme-li instrukce IF za sebou. Například:

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

Kód <<a>> bude spuštěn pouze tehdy, pokud jsou obě oblasti (area) 1 a 2 deaktivovány (access). Jakmile je některá z nich aktivována (secure) pak kód <<a>> nebude vůbec dosažen.

Použití instrukce **OR** je také velmi snadné, vyžaduje však zadávání některých kódů (příkazů) vícenásobně. Instrukce OR znamená, že výsledek je pravdivý je-li pravdivý alespoň jeden z testů.

Například: “Jestliže prší nebo je zima tak potřebuji kabát”. Kabát není potřeba pouze tehdy, jestli neprší a není zima. Jakmile prší, nebo je zima, nebo obojí, tak pottřebuji kabát.

Pro získání instukce OR, je zapotřebí napsat kód v oddělených instrukcích IF. Například:

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

Kód <<a>> je napsán dvakrát – v každé instrukci IF. V editoru skriptů toho dosáhnete snadno požitím funkcí “kopírovat” (Ctrl-C) a vložit (Ctrl-V).
Dále je možné dosáhnout více logických funkcí nebo jejich vzájemných kombinací, jako například funkce “exklusivní součet” (XOR) kdy je je nutná pravda v jedné nebo druhé části, ale nikoliv v obou současně.

Bezpečnostní funkce

Dosud byla vysvětlena základní logika na příkladech pro automatizaci budovy. Tato kapitola bude soustředěna na Genesis jako bezpečnostní systém.

Genesis není bezpečnostním systémem okamžitě. Standardní skripty jsou sice vyhovující pro většinu bezpečnostních požadavků, ale tyto skripty musí být přiděleny příslušným vstupům. Pro běžné aplikace je přidělení skriptů rychlé a jednoduché..

Hlavní přednost a síla Genesis je v tom, že je možné při instalaci Genesis programovat jako zabezpečovací systém, systém pro kontrolu přístupu, systém pro řízení a automatizaci budovy nebo jejich vzájemnou kombinaci.

Je možné rozšiřovat standardní bezpečnostní funkce, nebo je naopak odstranit. Například nějaký vstup může být použit pro otevírání dveří. Změna stavu vstupu nezpůsobí poplach, ale tamper (sabotáž) bude pracovat normálně. Takové a další funkce umožňují skripty.

Typy stavů vstupu pro skripty

Vstup může mít řadu stavů. Nejvíce důležité jsou:
- Klidový (uzavřený) SEALED nebo narušený (otevřený) UNSEALED
- Přístupný (deaktivovaný) ACCESS nebo zabzpečený (aktivovaný) SECURE
- V poplachu (ALARM) nebo normální (NORMAL)
- Tamper nebo v klidu (SEAL) nebo narušený (UNSEAL)
- Izolace ZAP (ISOLATE ON) nebo Izolace VYP (ISOLATE OFF)
- Přemostění ZAP (BYPASS ON) nebo přemostění VYP (BYPASS OFF)

Změna vstupu byla z klidu na narušeno. Nastala událost "IF INPUT EVENT ON UNSEAL". Nyní nastane co? Poplach, nebo je třeba něco ovládat?

Běžný poplachový (bezpečnostní) skript pro vstup pracuje takto:

```
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
```

Jestliže byl vstup narušen (událost "UNSEAL"), pak bude systém pokračovat příkazy dále: Jestliže vstup není zabezpečen (stav "SECURE"), nestane se nic. To znamená, že všechny oblasti související se vstupem musí být zabezpečeny (SECURE), aby skript pokračooval dalšímy příkazy. Jestliže je vstup ve stavu poplach (ALARM), pak se skript vrátí zpět (return) a nestane se nic. Jestliže vstup není ve stavu poplach, skript sepne výstup pro sirénu a spustí komunikátor. Vstup se nastaví do stavu poplach (ALARM).

Tento skript demonstruje, jak skript rozhoduje o další činnosti. Vyjmutí příkazu pro sepnutí výstupu znamená tichý poplach. Přidáním podmínky "Hodiny" umožní spouštět sirénu pouze v určitých časech. Skriptů může být celá řada. Některé skripty mohou odpovídat jednou cestou, některé mohou být komplikované.

Příkaz "SEND DIALLER" je běžným příkazem. Formát přenosu (POINT ID), je popsán v části "reference". V této části předpokládejme, že tento příkaz vykoná sspojení s monitorovací službou.

Kód pro TAMPER je velmi podobný:

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

Jestliže se navstupu vyskytla událost TAMPER (sabotáž), pak se kontroluje, zda již není vstup ve stavu tamper. Jestliže ano, dále se nestane nic. Jestliže ne, vstup je nastaven do stavu tamper, vyšle se zpráva na komunikátor a sepne se siréna. Postup je obdobný, jako reakce na poplachový podnět z detektoru. Způsob odezvy je možno nastavit zcela libovolně.

Vstup má být resetován. Reakce je různá, podle typu poplachu:

```
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
```

Jestliže vystup byl ve stavu poplach nebo tamper, na monitorovací službu bude komunikátorem vyslána informace o resetu. Současně na vstupu bude zrušen poplachový stav (ALARM OFF, TAMPER OFF).

Oblasti

Oblasti jsou jednodušší než vstupy a mají méně stavů:

- Přístupna (ACCESS) nebo Zabezpečena (SECURE)
- Poplach (ALARM) nebo Zabezpečeno (SECURE)
- Narušena (UNSEALED) nebo V klidu (SEALED)
- Tamper nebo Zabezpečeno (SECURE)
- Izolace ZAP (ISOLATE ON) nebo Izolace VYP (ISOLATE OFF)
- Přemostění ZAP (BYPASS ON) nebo přemostění VYP (BYPASS OFF)

Stav **přístupno/zabezpečeno** je ovládán uživatelem nebo skripty. Může být ovládán z klávesnic (RAS), čteček, počítače PC nebo prostřednictvím skriptů.

Poplach / normální stav je nicméně určován systémem. Oblast je v poplachu, jesliže je alespoň jeden ze vstupů přidružených k této oblasti v poplachu (ALARM ON) nebo ve stavu tamper (TAMPER ON). Povšimněte si rozdílu mezi vstupem který je narušen a v poplachu. Pokud je vstup narušen, tak teprve příslušným skriptem je nastaven do stavu poplach (ALARM).
Stav "Narušeno" je aktivován s prvním narušeným vstupem přidruženým k této oblasti, stav "v klidu" je nastaven, jakmile se poslední vstup uvede do stavu "v klidu". Používá se pro události Izolace a Přemostění.

Průběh událostí

Vztahy mezi vstupy a oblastmi jsou provázány. Činnost jednoho bude mít automaticky vliv na události druhého.

Pokud je oblast zabezpečena (například uživatelem z klávesnice), průběh událostí je:
- Jestliže je oblast narušena, stop a zobrazení narušených vstupů, jinak
- pro každý vstup v této oblasti spustit událost ‘IF INPUT EVENT ON SECURE` jestliže všechny oblasti přidružené k tomuto vstupu jsou nyní zabezpečeny.
- Spustit událost ‘IF AREA EVENT ON SECURE` pro tuto oblast.

Bod číslo 2 možná potřebuje vysvětlení. Například oblast 1 zahrnuje dva vstupy. Vstup 1 je přiřazen pouze oblasti 1 a ta je nyní zabezpečena. Událost "IF INPUT EVENT ON SECURE"je pro tento vstup spuštna. Vstup 2 je přiřazen oblastem 2, 3, a 4. Pro tento vstup je spuštěna událost "IF INPUT EVENT ON SECURE" pouze tehdy, budou-li všechny tyto oblasti současně zabezpečeny.

Vstup v oblasti:

Jestliže je vstup narušen (unseal):
- Událost “IF INPUT EVENT ON UNSEAL” je spuštěna.
- Jestli jeaktivní funkce “ALARM ON”, a toto je první poplach v oblasti, potom

- Událost “IF AREA EVENT ON ALARM” bude spuštěna.

Poznámka: Událost "IF AREA EVENT ON ALARM" je spuštěna prvním vstupem v oblasti, který vydá příkaz "ALARM ON". Událost "IF AREA EVENT ON TAMPER" je spuštěna prvním vstupem v oblasti který vydá příkaz "TAMPER ON".

Jestliže je oblast deaktivována (access) pak jsou spuštěny události vstupu:
- Jestli byl nějaký vstup v poplachu je spuštěno “IF INPUT EVENT ON RESET”,
- Je spuštěna událost “IF INPUT EVENT ON ACCESS”
- Jestli byla oblast v poplachu, je spuštěna událost “IF AREA EVENT ON”,
- Je spuštěna událost “IF AREA EVENT ON ACCESS”.

Tato otevřená struktura znamená, že systém je flexibilní. To znamená, že je možno přizpůsobit se většině aplikací.

Standardní skripty zajistí normální funkci bezpečnostního systému. To znamená, že systém bude automaticky fungovat tak, jak bylo výše popsáno, bez nutnosti vytvářet či měnit skripty.

Absolutní a relativní adresování

Mnoho příkazů umožňuje buď absolutní nebo relativní adresování.

Při absolutním adresování jsou adresa, pozice, číslo oblasti, vstup přesně specifikovány. Toto číslo je jediné platné pro příkaz - stejná akce bez ohledu z jakého bodu byl skript spuštěn.

Například příkaz “ACCESS AREA, 3” bude vždy deaktivovat (accessed) oblast 3. Tento příkaz může být použitý v jekémkoliv typu skriptu.

Relativní adresování umožňuje spustit příkaz na na bodu, který spustil událost. Povšimněte si, že příkaz musí být stejného typu jako událost.

Například příkaz “ACCESS THIS AREA” bude deaktivovat oblast, která tento skript spustila. Pokud je tomuto skriptu asociováno více oblastí, jakákoliv z nich může být deaktivována. Záleží na tom, která oblast spustila skript.

Je možné kombinovat absolutní a relativní adresování uvnitř jednoho skriptu, ale relativní adresování musí být použito pouze pro příkazy stejného typu jako je typ skriptu.

Relativní adresování je velmi účinná metoda pro opakované používání skriptu. Například standardni skript pro vstup obsahuje následující kód:
IF INPUT EVENT ON UNSEAL
IF SECURED
IF ALARM ON
V tomto příkladu se první a třetí řádek odvolávají na vstup, který spustil událost. Tento skript pak může být použitý pro různé vstupy.

Input Script LOCAL

ALARM OFF

ALARM ON

BYPASS OFF

BYPASS ON

CLEAR THIS INPUT FLAG,1

ELSE

ENDIF

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 SEAL

IF INPUT EVENT ON SECURE

IF INPUT EVENT ON TAMPER

IF INPUT EVENT ON TIMER EXPIRE

IF INPUT EVENT ON UNSEAL

IF ISOLATED

IF SEALED

IF SECURED

IF TAMPER

IF THIS INPUT FLAG,1

IF THIS INPUT TIMER EXPIRED,1 [TIMER]

IF THIS INPUT TIMER ON,1 [TIMER]

IF UNSEALED

ISOLATE OFF

ISOLATE ON

RETURN

SET THIS INPUT FLAG,1

START THIS INPUT TIMER,1,10

STOP THIS INPUT TIMER,1 [TIMER]

TAMPER OFF

TAMPER ON

TRACE OFF

TRACE ON

AREA LOCAL

ACCESS THIS AREA

CLEAR THIS AREA FLAG,1

ELSE

ENDIF

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

IF AREA IN ACCESS CONDITION

IF AREA IN ALARM CONDITION

IF AREA IN BYPASSED CONDITION

IF AREA IN ISOLATED CONDITION

IF AREA IN SEALED CONDITION

IF AREA IN SECURE CONDITION

IF AREA IN TAMPER CONDITION

IF THIS AREA FLAG,1

IF THIS AREA TIMER EXPIRED,1 [TIMER]

IF THIS AREA TIMER ON,1 [TIMER]

RESET THIS AREA

RETURN

SECURE THIS AREA

SET THIS AREA FLAG,1

START THIS AREA TIMER,1,10

STOP THIS AREA TIMER,1 [TIMER]

TRACE OFF

TRACE ON

CLOCK LOCAL

ELSE

ENDIF

IF CLOCK EVENT TO VALID

IF CLOCK EVENT TO VOID

RETURN

TRACE OFF

TRACE ON

DOOR LOCAL

CLEAR THIS DOOR FLAG,1

ELSE

ENDIF

IF DOOR EQUAL,1 [DOOR=1]

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

IF DOOR RANGE,1,3 [DOOR=1][]

IF THIS DOOR FLAG,1

IF THIS DOOR TIMER EXPIRED,1 [TIMER]

IF THIS DOOR TIMER ON,1 [TIMER]

LOCK ALL ON THIS DOOR

LOCK THIS DOOR ENTRY

RELEASE THIS DOOR

RESET THIS DOOR

RETURN

SECURE THIS DOOR

SET THIS DOOR FLAG,1

START THIS DOOR TIMER,1,10

STOP THIS DOOR TIMER,1 [TIMER]

TRACE OFF

TRACE ON

UNLOCK THIS DOOR

ALARM LOCAL

ELSE

ENDIF

RETURN

TRACE OFF

TRACE ON

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

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

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

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

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

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

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

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

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

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

IF ALARM EVENT OFF,11 [[011] MODEM ALARM]
IF ALARM EVENT OFF,12 [[012] RAS LOCK]
IF ALARM EVENT OFF,13 [[013] UNIT ENABLE/DISABLE]
IF ALARM EVENT OFF,14 [[014] DIALLER ENABLE/DISABLE]
IF ALARM EVENT OFF,15 [[015] SIU ENABLE/DISABLE]
IF ALARM EVENT OFF,16 [[016] MODEM ENABLE/DISABLE]
IF ALARM EVENT OFF,17 [[017] CARD/PIN READER 1 CONTROL CHANGED]
IF ALARM EVENT OFF,18 [[018] CARD/PIN READER 2 CONTROL CHANGED]
IF ALARM EVENT OFF,19 [[019] BATTERY HIGH/LOW ALARM]
IF ALARM EVENT ON,1 [[001] UNIT ONLINE/ONLINE]
IF ALARM EVENT ON,2 [[002] DURESS ALARM]
IF ALARM EVENT ON,3 [[003] TAMPER ALARM]
IF ALARM EVENT ON,4 [[004] MAINS ALARM]
IF ALARM EVENT ON,5 [[005] BATTERY LOW ALARM]
IF ALARM EVENT ON,6 [[006] POWER FUSE ALARM]
IF ALARM EVENT ON,7 [[007] SIREN FUSE ALARM]
IF ALARM EVENT ON,8 [[008] STROBE FUSE ALARM]
IF ALARM EVENT ON,9 [[009] DIALLER ON-LINE]
IF ALARM EVENT ON,10 [[010] SIU ON-LINE]
IF ALARM EVENT ON,11 [[011] MODEM ALARM]
IF ALARM EVENT ON,12 [[012] RAS LOCK]
IF ALARM EVENT ON,13 [[013] UNIT ENABLE/DISABLE]
IF ALARM EVENT ON,14 [[014] DIALLER ENABLE/DISABLE]
IF ALARM EVENT ON,15 [[015] SIU ENABLE/DISABLE]
IF ALARM EVENT ON,16 [[016] MODEM ENABLE/DISABLE]
IF ALARM EVENT ON,17 [[017] CARD/PIN READER 1 CONTROL CHANGED]
IF ALARM EVENT ON,18 [[018] CARD/PIN READER 2 CONTROL CHANGED]
IF ALARM EVENT ON,19 [[019] BATTERY HIGH/LOW ALARM]

ALERT LOCAL
ELSE
ENDIF
IF ALERT EVENT ON,1 [[001] SYSTEM RESET]
IF ALERT EVENT ON,2 [[002] TIME CHANGED]
IF ALERT EVENT ON,3 [[003] SERVICE REQUESTED]
IF ALERT EVENT ON,4 [[004] BATTERY TEST STARTED]
IF ALERT EVENT ON,5 [[005] REMOTE ACCESS]
IF ALERT EVENT ON,6 [[006] EVENT LOG RESET]
IF ALERT EVENT ON,7 [[007] REQUEST TO POWER DOWN]
IF ALERT EVENT ON,8 [[008] CHANGE TO HOLIDAYS]
IF ALERT EVENT ON,9 [[009] DIALLER FAULT]
RETURN
TRACE OFF
TRACE ON

ACCESS LOCAL

ELSE

ENDIF

IF ACCESS DOOR EQUAL,1 [DOOR=1]

IF ACCESS DOOR RANGE,1,3 [DOOR=1][]

IF ACCESS EVENT,1 [USER ACCESS TO DOOR]

IF ACCESS EVENT,2 [USER ACCESS TO MENU]

IF ACCESS EVENT,3 [USER ACCESS DENIED]

IF ACCESS EVENT,4 [USER ACCESS TO DOOR]

IF ACCESS EVENT,5 [CARD DOOR ACCESS ALLOWED]

IF ACCESS EVENT,6 [CARD ACCESS DENIED]

IF ACCESS EVENT,7 [CARD USED AS BUNDY]

IF ACCESS EVENT,8 [USER DURESS ACTIVATED]

IF ACCESS EVENT,9 [USER DURESS CANCELLED]

IF ACCESS EVENT,10 [USER 100/101 ACCESS]

IF ACCESS EVENT,11 [USER DOOR ACCESS ALLOWED]

IF REJECT EVENT, 101 [USER NOT ALLOWED]

IF REJECT EVENT, 102 [USER SUSPENDED]

IF REJECT EVENT, 103 [USER EXPIRED]

IF REJECT EVENT, 104 [USER 100/101 ERROR]

IF REJECT EVENT, 105 [USER DURESS ACTIVATED]

IF REJECT EVENT, 106 [USER NO VALID ACCESS]

IF REJECT EVENT, 107 [USER PIN CHANGE DUE]

IF REJECT EVENT, 108 [USER PIN EXPIRED]

IF REJECT EVENT, 110 [CARD PASSBACK ERROR]

IF REJECT EVENT, 111 [CARD VALID NO DOOR]

IF REJECT EVENT, 113 [GROUP SUSPENDED]

IF REJECT EVENT, 114 [UNIT LOCKED ERROR]

IF REJECT EVENT, 115 [MENU ACCESS NOT ALLOWED]

IF REJECT EVENT, 116 [MENU ACCESS NOT VALID]

IF REJECT EVENT, 117 [AREA ACESS NOT ALLOWED]

IF REJECT EVENT, 118 [AREA ACCESS NOT VALID]

IF REJECT EVENT, 119 [DOOR ACCESS NOT ALLOWED]

IF REJECT EVENT, 120 [DOOR ACCESS NOT VALID]

IF REJECT EVENT, 121 [DOOR LOCKED]

IF REJECT EVENT, 122 [AZONE FULL]

IF GROUP EQUAL,1 [MASTER GROUP 1]

IF GROUP RANGE,1,2 [MASTER GROUP 1][]

IF THIS GROUP FLAG,1

IF THIS USER FLAG,1

IF USER EQUAL,1 [TEST]

IF USER RANGE,1,2 [TEST][]

RESET THIS GROUP FLAG,1

RESET THIS USER FLAG,1

RETURN

SET THIS GROUP FLAG,1

SET THIS USER FLAG,1

TRACE OFF

TRACE ON

CONTROL LOCAL

ELSE

ENDIF

RETURN

TRACE OFF

TRACE ON

IF CONTROL VALUE EQUAL,1

GLOBALS

ACCESS AREA,1 [AREA 1]

BATTERY TEST,1,1,10 [MASTER 1] [START]

BUZZER CONTROL,1,1,00 [MASTER 1]

BYPASS INPUT,1 [INP=1]

CALL SCRIPT,17,1 [CONTROL 17]

CHAIN,17 [CONTROL 17]

CLEAR AREA FLAG,1,1 [AREA 1] []

CLEAR CLOCK FLAG,1,1 [CLOCK 1]

CLEAR DOOR FLAG,1,1 [DOOR=1][]

CLEAR GLOBAL FLAG,1 []

CLEAR GROUP FLAG,1,1

CLEAR INPUT FLAG,1,1 [INP=1] []

CLEAR USER FLAG,1,1 [TEST] []

CLEAR VARIABLE,1

DECREMENT VARIABLE,1

DISABLE CLOCK,1 [CLOCK 1]

ENABLE CLOCK,1 [CLOCK 1]

IF ANY DEVICE OFFLINE

IF AREA DELAY TIMER ON,1,1 [AREA 1] [TIMER]

IF AREA FLAG ON,1,1 [AREA 1] []

IF AREA IN ACCESS,1 [AREA 1]

IF AREA IN ALARM,1 [AREA 1]

IF AREA IN SECURE,1 [AREA 1]

IF AREA IN TAMPER,1 [AREA 1]

IF AREA IS BYPASSED,1 [AREA 1]

IF AREA IS ISOLATED,1 [AREA 1]

IF AREA IS SEALED,1 [AREA 1]

IF AREA IS UNSEALED,1 [AREA 1]
IF AZONE IS,1,1,1 [LESS][AZONE 1,VALUE]
IF AZONE COMPARE,1,1,2[LESS][AZONE1,AZONE2]
IF CLOCK VALID,1 [CLOCK 1]
IF CLOCK VOID,1 [CLOCK 1]
IF DATE IS,1,14,01,04 [LESS,DD,MM,YY]
IF DOOR DELAY TIMER ON,1,1 [DOOR=1] [TIMER]
IF DOOR FLAG ON,1,1 [DOOR=1][]
IF GLOBAL FLAG ON,1 []
IF GROUP FLAG ON,1,1
IF INPUT DELAY TIMER ON,1,1 [INP=1] [TIMER]
IF INPUT FLAG ON,1,1 [INP=1] []
IF INPUT IN ACCESS,1 [INP=1]
IF INPUT IN ALARM,1 [INP=1]
IF INPUT IN SECURE,1 [INP=1]
IF INPUT IN TAMPER,1 [INP=1]
IF INPUT IS BYPASSED,1 [INP=1]
IF INPUT IS ENABLED,1 [INP=1]
IF INPUT IS ISOLATED,1 [INP=1]
IF INPUT IS SEALED,1 [INP=1]
IF INPUT IS UNSEALED,1 [INP=1]
IF OUTPUT ON,1
IF SYSTEM ALARM OFF,1 [[001] UNIT ONLINE/OFFLINE]
IF SYSTEM ALARM ON,1 [[001] UNIT ONLINE/OFFLINE]
IF TIME IS,1,19,44 [LESS,HH,MM]
IF USER FLAG ON,1,1 [TEST] []
IF VARIABLE COMPARE,1,1 [] []
IF VARIABLE,1,1,1 [LESS]
INCREMENT VARIABLE,1
INPUT ALARM OFF,1 [INP=1]
INPUT ALARM ON,1 [INP=1]
ISOLATE INPUT,1 [INP=1]
LOCK ALL ON DOOR,1 [DOOR=1]
LOCK DOOR ENTRY,1 [DOOR=1]
LOCK RAS,2,0 [RAS 1][UNLOCK]
LOG EVENT,1
OLIST CONTROL,1,1,00 [] [OUTPUT OFF]
OUTPUT CONTROL,1,1,00 [OP=1] [OUTPUT OFF]
RAS CARDPIN,2,1,0 [RAS 1][READER 1][OFF]
RELEASE DOOR,1 [DOOR=1]
REMOVE BYPASS,1 [INP=1]
REMOVE ISOLATION,1 [INP=1]
RESET AREA,1 [AREA 1]

RESET DOOR ALARM,1 [DOOR=1]

RESET LOG EVENTS

SECURE AREA,1 [AREA 1]

SECURE DOOR,1 [DOOR=1]

SEND COMM MESSAGE,2,1,1

SEND DIALLER,1,999,1,132 [132-BURGLAR ALARM-INTERIOR]

SEND RAS MESSAGE,2,1 [RAS 1][]

SET AREA FLAG ON,1,1 [AREA 1] []

SET AZONE,1,1 [AZONE 1]

SET CLOCK FLAG,1,1 [CLOCK 1]

SET DOOR FLAG,1,1 [DOOR=1][]

SET GLOBAL FLAG,1 []

SET GROUP FLAG ON,1,1

SET INPUT FLAG,1,1 [INP=1] []

SET USER FLAG ON,1,1 [TEST] []

SET VARIABLE,1,0

START AREA DELAY TIMER,1,1,10

START DOOR DELAY TIMER,1,1,10

START INPUT DELAY TIMER,1,1,10

STOP AREA DELAY TIMER,1,1

STOP DOOR DELAY TIMER,1,1

STOP INPUT DELAY TIMER,1,1

UNLOCK DOOR,1 [DOOR=1]

Výrobce:
Genesis Electronics Australia Pty Ltd
www.genesiselectronics.com.au

VÝHRADNÍ AUTORIZOVANÝ DISTRIBUTOR
PRO ČESKOU REPUBLIKU:

ALARM ABSOLON spol. s r.o.
Březinova 9, Praha-8
tel.: 224816766, 224816026, 224819539,
224819541, 604 233133,
fax: 224814028, 224819541
absolon@absolon.cz,
www.absolon.cz www.genesiselectronics.cz

Výrobce si ponechává právo na změny produktů bez předchozího upozornění.
Tento manuál předpokládá, že instalační technik tohoto produktu je vyškolen a znalý všech předpisů a norem na bezpečnostní systémy, systémy kontroly vstupu a na systémy automatizace.
Z tohoto důvodu Genesis Electronics Ltd. a Alarm Absolon spol. s r.o. nenesou žádnou odpovědnost za nějaké poškození, finanční ztráty nebo škody způsobené na jakémkoliv majetku nebo osobě vyplývající ze správného nebo nesprávného používání jakéhokoliv komponentu Genesis Electronics.