2N Helios IP Automation

IP Interkom

Konfigurační manuál

Firmware:
Verze: 2.17

www.2n.cz

Společnost 2N TELEKOMUNIKACE a.s. je českým výrobcem a dodavatelem telekomunikační techniky.

K produktovým řadám, které společnost vyvíjí, patří GSM brány, pobočkové ústředny, dveřní a výtahové komunikátory. 2N TELEKOMUNIKACE a.s. se již několik let řadí mezi 100 nejlepších firem České republiky a již dvě desítky let symbolizuje stabilitu a prosperitu na trhu telekomunikačních technologií. V dnešní době společnost vyváží do více než 120 zemí světa a má exkluzivní distributory na všech kontinentech.

Pro rychlé nalezení informací a zodpovězení dotazů týkajících se 2N produktů a služeb 2N TELEKOMUNIKACE spravuje databázi FAQ nejčastějších dotazů. Na www.faq.2n.cz naleznete informace týkající se nastavení produktů, návody na optimální použití a postupy „Co dělat, když...".

Společnost 2N TELEKOMUNIKACE a.s. tímto prohlašuje, že zařízení 2N® je ve shodě se základními požadavky a dalšími příslušnými ustanoveními směrnice 1999/5/ES. Plné znění prohlášení o shodě naleznete CD-ROM (pokud je přiloženo) nebo na www.2n.cz.

Společnost 2N TELEKOMUNIKACE a.s. je vlastníkem certifikátu ISO 9001:2009. Všechny vývojové, výrobní a distribuční procesy společnosti jsou řízeny v souladu s touto normou a zaručují vysokou kvalitu, technickou úroveň a profesionalitu všech našich výrobků.

Obsah:

1. Termíny a piktogramy

V manuálu jsou použity následující symboly a piktogramy:

Nebezpečí úrazu

- **Vždy dodržujte** tyto pokyny, abyste se vyhnuli nebezpečí úrazu.

Varování

- **Vždy dodržujte** tyto pokyny, abyste se vyvarovali poškození zařízení.

Upozornění

- **Důležité upozornění.** Nedodržení pokynů může vést k nesprávné funkci zařízení.

Tip

- **Užitečné informace** pro snazší a rychlejší používání nebo nastavení.

Poznámka

- Postupy a rady pro efektivní využití vlastností zařízení.

2. Konfigurace 2N® Helios IP Automation

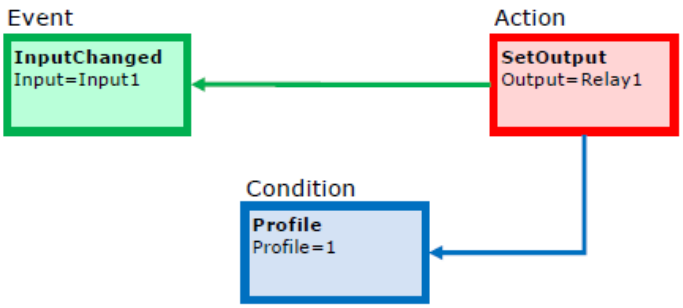
2N Helios IP poskytuje velmi flexibilní možnosti nastavení dle různorodých požadavků uživatele. Existují situace, kdy běžný rozsah nastavení (např. nastavení chování spínačů nebo volání) nedostačuje a pro tyto případy poskytuje 2N Helios IP speciální programovatelné rozhraní Automation. Typické použití Automation je v aplikacích, které vyžadují složitější propojení se systémy třetích stran.

Poznámka

- Funkcionalita Automation je dostupná pouze po vložení platného licenčního klíče pro licenci Enhanced Integration nebo Gold.

Některé modely 2N Helios IP jsou vybaveny řadou digitálních vstupů a výstupů, z nichž většinu lze konfigurovat jako běžné spínače 2N Helios IP (viz kap. Spínače). Automation umožňuje využít všechny tyto vstupy a výstupy a propojit je v různých kombinacích.

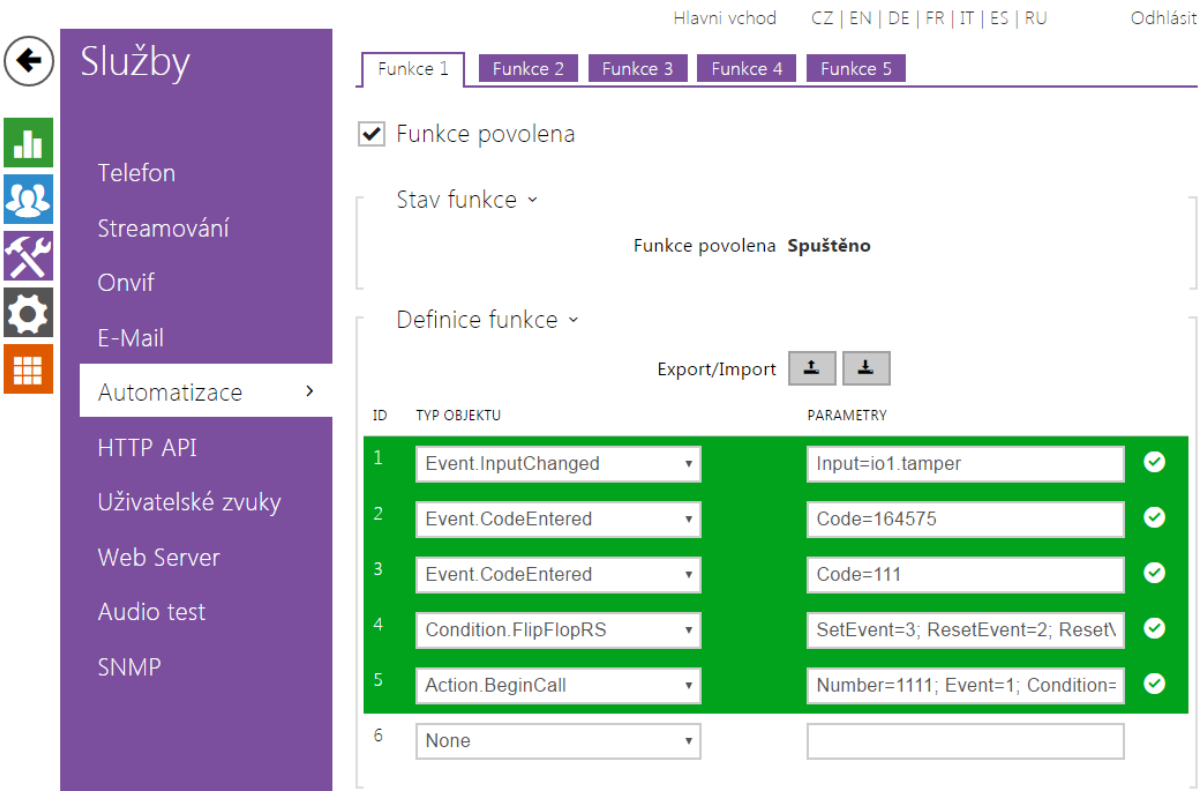
Automation umožňuje podle potřeby propojit události – Event, které v zařízení vznikají (např. stisk tlačítka, protažení RFID karty, změna stavu digitálního vstupu apod.) se specifickými akcemi – Action (např. sepnutí digitálního výstupu, přehrání uživatelského zvuku nebo volání apod.). Zároveň lze provádění akcí podmínit různými podmínkami – Condition (např. stavem časového profilu, stavem logického vstupu apod.).



Na obrázku výše je znázorněn příklad vzájemného propojení jednotlivých typů bloků – události, akce a podmínky. Obecně platí, že akce je vždy navázána na jednu konkrétní volitelnou událost a je vykonána, pokud je splněna konkrétní volitelná podmínka. Podmínka není povinná, a pokud není uvedena, akce je provedena vždy, když nastane jí přiřazená událost. **Automation** definuje množství událostí, akcí a podmínek, které lze podle potřeby parametrizovat. Jejich úplný seznam je uveden v následujících kapitolách.

Konkrétní případ propojení uvedený na obrázku lze interpretovat takto: Akce **SetOutput** (nastavení digitálního výstupu) se provede v případě vzniku události **InputChanged** (změna logického vstupu input1 z log 0 na log 1) za předpokladu podmínky **Profile** (aktivní profil č. 1).

Webové rozhraní **2N Helios IP** umožňuje jednoduchým způsobem konfigurovat propojení bloků na stránce Automatizace. Konfigurace na obrázku odpovídá výše uvedenému příkladu.



Ovládání Automation

- Záložky Funkce – 2N Helios IP umožňuje vytvořit a propojit až 30 bloků na 5 nezávislých stránkách (nezáleží na tom, zda se jedná o události, akce nebo podmínky). Na událost nebo podmínku lze navázat více akcí. Můžete např. vytvořit 15 akcí a navázat je na 15 událostí nebo 29 akcí navázat na 1 událost.
- Funkce povolena – povoluje definovanou funkci.
- Definice funkce – export nadefinované funkce je možné provést stiskem tlačítka  a nahrát dříve uloženou konfiguraci stiskem  .

Nastavení parametrů bloků

Ve sloupci **Typ objektu** vyberte požadovanou událost (Event.xxx), akci (Action.xxx) nebo podmínku (Condition.xxx). U většiny bloků je potřeba uvést jeden nebo více parametrů. Všechny podporované parametry naleznete dále v textu, v kapitolách popisujících vlastnosti jednotlivých bloků. Parametry bloku vyplňte do příslušného řádku ve sloupci **Parametry**. Pokud potřebujete definici bloku uvést více než jeden parametr, oddělte je středníkem.

Změny se uplatní, až po stisku tlačítka Uložit v pravém dolním rohu stránky.

Pokud jste zadali všechny parametry správně, pak se na konci příslušného řádku definice bloku zobrazí zelená značka. Pokud zadáte některý z parametrů nesprávně (neplatný název nebo hodnota parametru, případně není uveden povinný parametr bloku), pak se na konci příslušného řádku definice bloku zobrazí červená značka. Pokud najedete myší nad tuto značku, objeví se nápověda s popisem chyby. **Automation** pracuje pouze tehdy, pokud všechny vytvořené bloky jsou správně nakonfigurovány (tj. u všech je zelená značka). V opačném případě je funkce **Automation** vypnuta.

Většina bloků obsahuje parametry (např. Event, Condition, StartEvent apod.), které se odkazují na jiné bloky. Nastavením těchto parametrů se provádí vzájemné propojení definovaných bloků. Hodnota těchto parametrů musí odpovídat číslu řádku v tabulce, ve kterém je definován blok, na který se odkazuje. Pokud uvedete nesprávnou hodnotu (neodpovídá správnému typu bloku nebo odpovídá bloku, který není nadefinován), pak se u příslušného řádku po stisku tlačítka Uložit objeví červená značka.

Tip

- V názvech parametrů a hodnotách se nerozlišují velká a malá písmena.
- Některé parametry bloků jsou nepovinné. Pokud nepovinný parametr v definici bloku neuvedete, bude nastaven na implicitní hodnotu.

Použití proměnných

V některých případech je užitečné mezi jednotlivými bloky předávat doplňující informace – např. odeslat ID detekované karty pomocí HTTP příkazu jinému zařízení, parametry přijaté v HTTP příkazu použít pro nastavení parametrů navázané akce. K tomuto účelu slouží tzv. proměnné (parametry) bloků událostí. Hodnoty proměnných jsou aktualizovány vždy v okamžiku vyvolání události. Na hodnotu proměnné se lze odkázat v konfiguračních parametrech jiného bloku pomocí následující syntaxe:

$(číslo_bloku.název_proměnné) – číslo bloku je oddělené od názvu proměnné tečkou.

- Příklad:
  - 1: Event.KeyPressed: Key=Any
  - 2: Action.SendHttpRequest: Event=1; Uri=**http://192.168.1.1/enu/trigger/ABCD?Key=$(1.Key)**

V případě stisku libovolné klávesy (blok 1 Event.KeyPressed) se na IP adresu 192.168.1.1 odešle HTTP požadavek (blok 2 Action.SendHttpRequest). Konkrétně v případě stisku klávesy * bude Uri HTTP požadavku: **http://192.168.1.1/enu/trigger/ABCD?Key=***

Každá událost definuje proměnné **TimeStamp** a **Count**.

Proměnná **TimeStamp** obsahuje zakódované datum a čas posledního vyvolání události ve formátu Unix Time (počet sekund od 00:00:00 1.1.1970).

Proměnná **Count** obsahuje počet vyvolání události od spuštění zařízení nebo od poslední změny konfigurace bloků. Po každém vyvolání události se proměnná zvyšuje o 1.

Další proměnné jsou specifické pro konkrétní události a jsou popsány v následujících kapitolách.

Tip

- V názvech proměnných se nerozlišují velká a malá písmena.

Upozornění

- Proměnné nelze použít v parametrech definujících vazby mezi bloky, tj. v parametrech Event, Condition apod.

3. Události (Event)

Automation definuje následující typy událostí:

- KeyPressed – stisk tlačítka
- KeyReleased – uvolnění tlačítka
- DtmfPressed – přijat DTMF kód v hovoru
- DtmfEntered – detekován numerický kód v hovoru
- CodeEntered – zadaný numerický kód
- CardEntered – protažení RFID karty
- CallStateChanged – změna stavu hovoru
- InputChanged – změna digitálního vstupu
- Delay – definované zpoždění
- Timer – časovač periodické události
- HttpTrigger – přijat HTTP příkaz
- MulticastTrigger – přijat příkaz pro více zařízení současně
- AudioLoopTest – proveden audio test
- Time – definovaný čas
- MotionDetected – detekován pohyb před kamerou
- NoiseDetected – hluk detekován mikrofonem

Detailní popis událostí, jejich parametry a použití je popsáno v následujícím textu.

Event.KeyPressed

Blok KeyPressed definuje událost generovanou při stisku definovaného tlačítka nebo libovolného tlačítka z definované skupiny.

Parametry

- Key – definuje tlačítko (příp. skupinu tlačítek). V případě, že tento parametr není uveden, událost je vygenerovaná při stisku libovolné klávesy (implicitní hodnota Any).
  - Platné hodnoty:
    - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, # – pro tlačítka na numerické klávesnici
    - %1, %2, .., %999 – pro tlačítka rychlé volby
    - any – pro libovolné tlačítko (implicitní hodnota).
  - Pokud chcete specifikovat více tlačítek, oddělte hodnoty čárkou.
- SuppressTones – umožňuje potlačit zvukovou signalizaci spojenou se stiskem nenaprogramovaného tlačítka rychlé volby. Parametr je nepovinný.
  - Platné hodnoty:
    - 0 – Zvuky nejsou potlačeny
    - 1 – Zvuky jsou potlačeny (implicitní hodnota)

Proměnné

- Key – zaznamenaný kód stisknuté klávesy, která naposledy vyvolala tuto událost. Kód klávesy je uložen stejném formátu jako parametr Key.

Příklad

Událost vygenerovaná při stisku klávesy # a tlačítka rychlé volby 3 nebo 4:

- Event.KeyPressed: Key=#, %3, %4

Event.KeyReleased

Blok KeyReleased definuje událost generovanou při uvolnění definovaného stisknutého tlačítka nebo libovolného tlačítka z definované skupiny.

Poznámka

- Model Vario: událost je generována při stisku klávesy, odpovídá tedy události KeyReleased

Parametry

- Key – definuje tlačítko (příp. skupinu tlačítek). V případě, že tento parametr není uveden, událost je vygenerovaná při uvolnění libovolné klávesy (implicitní hodnota Any).
  - Platné hodnoty:
    - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, # – pro tlačítka na numerické klávesnici
    - %1, %2, .., %999 – pro tlačítka rychlé volby
    - any – pro libovolné tlačítko (implicitní hodnota).
  - Pokud chcete specifikovat více tlačítek, oddělte hodnoty čárkou.

Proměnné

- Key – zaznamenaný kód uvolněné klávesy, která naposledy vyvolala tuto událost. Kód klávesy je uložen stejném formátu jako parametr Key.

Příklad

Událost vygenerovaná při uvolnění klávesy 1 a tlačítka rychlé volby 2:

- Event.KeyReleased: Key=1, %2

Event.DtmfPressed

Blok DtmfPressed definuje událost generovanou při přijetí definovaného DTMF kódu nebo libovolného DTMF kódu z definované skupiny. DTMF kódu jsou detekovány jak v příchozích, tak odchozích hovorech.

Parametry

- Key – definuje DTMF kód (příp. skupinu DTMF kódů). V případě, že tento parametr není uveden, událost je vygenerovaná při detekci libovolného DTMF kódu (implicitní hodnota Any).
  - Platné hodnoty
    - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, A, B, C, D
    - any pro libovolný DTMF kód (implicitní hodnota).
  - Pokud chcete specifikovat skupinu kódů, oddělte hodnoty čárkou.
- Direction – definuje směr hovoru.
  - Platné hodnoty:
    - incoming – příchozí hovory
    - outgoing – odchozí hovory
    - any – oba směry
  - Parametr je nepovinný, implicitní hodnota je any.

Proměnné

- Key – zaznamenaný přijatý DTMF kód, který naposledy vyvolal tuto událost. DTMF kód je uložen stejném formátu jako parametr Key.

Příklad

Událost vygenerovaná při detekci DTMF kódu #:

- Event.DtmfPressed: Key=#

Event.DtmfEntered

Blok DtmfEntered definuje událost generovanou při zadání numerického kódu potvrzeného klávesou * pomocí DTMF v příchozím nebo odchozím hovoru.

Parametry

- Code – definuje numerický kód.
  - Platné hodnoty:
    - numerický kód – např. 12345

Proměnné

- Code – zaznamenaný přijatý numerický kód, který naposledy vyvolal tuto událost.

Příklad

Událost vygenerovaná při detekci DTMF kódu 12345*

- Event.DtmfEntered: Code=12345

Event.CodeEntered

Blok CodeEntered definuje událost generovanou při zadání numerického kódu potvrzeného klávesou * (pouze pro modely s numerickou klávesnicí).

Parametry

- Code – definuje numerický kód.
  - Platné hodnoty:
    - numerický kód - např. 12345
    - valid – libovolný platný kód
    - invalid – libovolný neplatný kód
    - any – libovolný platný i neplatný kód
- SuppressTones – umožňuje potlačit zvukovou signalizaci spojenou s přijetím neplatného numerického kódu. Parametr je nepovinný.
  - Platné hodnoty:
    - 0 – Zvuky nejsou potlačeny
    - 1 – Zvuky jsou potlačeny (implicitní hodnota)

Proměnné

- Code – zaznamenaný numerický kód, který naposledy vyvolal tuto událost.

Příklad

Událost vygenerovaná po zadání kódu 12345* na klávesnici

- Event.CodeEntered: Code=12345

Event.CardEntered

Blok **CardEntered** definuje událost generovanou při detekci (protažení) RFID karty se zadaným ID (pouze pro modely vybavených čtečkou RFID karet).

Parametry

- **Card** – definuje ID RFID karty, viz kapitola Čtečka Karet v konfiguračním manuálu **2N Helios IP**.
  - Platné hodnoty:
    - **valid** – libovolná platná karta (uvedena v seznamu karet v interkomu)
    - **invalid** – libovolná neplatná karta
    - **any** – libovolná platná i neplatná karta
    - **card ID** – zadané hodnota karty, např. 3F00F34F78
- **Reader** – definuje použitou čtečku karet
  - Platné hodnoty
    - **internal_cardreader** – vnitřní čtečka karet (modely **2N® Helios IP Vario, Force**)
    - **external_cardreader** – externí čtečka karet (modely **2N® Helios IP Vario, Force**)
    - **<jméno_modulu>** – jméno modulu nastavené v menu Hardware / Rozšiřující moduly / Použitý modul, parametr Jméno modulu (model **2N® Helios IP Verso**)
    - **any** – libovolná čtečka
  - Parametr je nepovinný, implicitní hodnota je **any**
- **Direction** – definuje směr průchodu
  - Platné hodnoty
    - **in** – vstupní čtečka
    - **out** – výstupní čtečka
    - **any** – nespecifikovaný směr
  - Parametr je nepovinný, implicitní hodnota je **any**
- **SuppressTones** – umožňuje potlačit zvukovou signalizaci spojenou s detekcí neplatné karty. Parametr je nepovinný.
  - Platné hodnoty:
    - **0** – zvuky nejsou potlačeny
    - **1** – zvuky jsou potlačeny (implicitní hodnota)

Proměnné

- Card – zaznamenané ID detekované karty, která naposledy vyvolala tuto událost.
- Reader – jméno naposledy použité čtečky karet (internal_cardreader, external_cardreader, <jméno_modulu>).
- Direction – směr průchodu, hodnota je nastavena u čtečky karet (In, Out, Unspecified).

Příklad

Událost generovaná při protažení karty s ID 0012456:

- Event.CardEntered: Card=0012456

Event.CallStateChanged

Blok CallStateChanged definuje událost generovanou při změně stavu hovoru (vyzvánění, spojení navázáno, hovor ukončen apod.)

Parametry

- State – definuje změnu stavu hovoru.
  - Platné hodnoty:
    - ringing – okamžik začátku vyzvánění
    - connected – okamžik úspěšného spojení hovoru
    - terminated – okamžik ukončení hovoru
- Direction – definuje směr hovoru.
  - Platné hodnoty:
    - incoming – příchozí hovory
    - outgoing – odchozí hovory
    - any – oba směry
  - Parametr je nepovinný, implicitní hodnota je any.

Proměnné

- State – zaznamenaný stav hovoru, který vyvolal tuto událost. Možné hodnoty odpovídají parametru State.
- Direction – zaznamenaný směr hovoru, který vyvolal tuto událost. Možné hodnoty jsou incoming nebo outgoing.

Příklad

Událost generovaná při ukončení libovolného odchozího hovoru:

- Event.CallStateChanged: State=terminated; Direction=outgoing

Event.InputChanged

Blok InputChanged definuje událost generovanou při změně logické úrovně na definovaném digitálním vstupu.

Parametry

- Input – definuje logický vstupu.
  - Platné hodnoty:
    - tamper – vstup tamper spínače
    - input1 – digitální vstup 1
    - input2 – digitální vstup 2
    - cr_input1 – digitální vstup 1 na čtečce karet
    - cr_input2 – digitální vstup 2 na čtečce karet
  - Seznam platných hodnot se může lišit pro různé modely interkomů 2N Helios IP, viz kap. Dostupné digitální vstupy a výstupy.
- Edge – definuje detekovanou změnu na digitálním vstupu.
  - Platné hodnoty:
    - falling – sestupná hrana, změna z log 1 na log 0
    - rising – vzestupná hrana, změna z log 0 na log 1
  - Parametr je nepovinný, implicitní hodnota je rising.

Proměnné

- Input – zaznamenané ID vstupu, jehož změna naposledy vyvolala tuto událost. Možné hodnoty odpovídají hodnotám parametru Input.
- Edge – zaznamenaná změna digitálního, která naposledy vyvolala tuto událost. Možné hodnoty jsou falling nebo rising.

Příklad

Událost generovaná při rozpojení tamper spínače (otevření zařízení):

- Event.InputChanged: Input=tamper

Event.Delay

Blok **Delay** definuje událost generovanou na základě jiné definované události s definovaným zpožděním. Pomocí této události lze zpozdit reakci na událost o definovaný časový interval (Delay).

Parametry

- **StartEvent** – definuje událost, která odstartuje zpoždění.
- **StopEvent** – definuje událost, která zastaví zpoždění. Tento parametr je nepovinný.
- **Delay** – definuje délku zpoždění. Zadat lze pouze číselnou hodnotu, nikoliv hodnotu získanou z proměnné jiného eventu.
- Příklad platných hodnot parametrů výše:
  - **10** – 10 sekund (jednotky není nutné uvádět)
  - **10 s** – 10 sekund
  - **100 ms** – 100 milisekund

Proměnné

Tento blok nedefinuje žádné specifické proměnné.

Příklad

Událost generovaná po 1s od vzniku události na řádku 1:

- Event.Delay: StartEvent=1; Delay=1s

Event.Timer

Blok Timer definuje událost generovanou s definovaným zpožděním po jiné specifikované události s definovaným počtem opakování události. Pomocí této speciální události lze zpozdit reakci na jinou událost o definovaný časový interval, nebo provést reakci několikrát po sobě.

Parametry

- StartEvent – definuje událost, která odstartuje časovač (jedná se o číslo řádku v záložce Automation, na kterém je událost definovaná). Tento parametr je nepovinný. Pokud není uveden, časovač se spustí automaticky.
- StopEvent – definuje událost, která zastaví časovač (jedná se o číslo řádku v záložce Automation, na kterém je událost definovaná). Pokud nastane událost StopEvent, pak se časovač zastaví a znovu spuštěn bude pouze událostí StartEvent. Tento parametr je nepovinný.
- Period – definuje periodu časovače.
- Příklad platných hodnot parametrů výše:
  - 10 – 10 sekund (jednotky není nutné uvádět)
  - 10 s – 10 sekund
  - 100 ms – 100 milisekund
- Minimální perioda je 100ms.
- Count – definuje počet opakování. Parametr je nepovinný a implicitní hodnota parametru je 0. V tomto případě není počet vygenerovaných událostí časovače omezen. V případě, že nastavíte hodnotu 1, časovač se chová jako zpoždění (Delay).

Proměnné

Tento blok nedefinuje žádné specifické proměnné.

Příklad

Událost generovaná 3x po 1 s od vzniku události na řádku 1:

- Event.Timer: StartEvent=1; Period=1s; Count=3

Event.HttpTrigger

Blok HttpTrigger definuje událost generovanou při přijetí HTTP příkazu odeslaného na HTTP server interkomu. Po přijetí HTTP příkazu ve tvaru tvaru http://ip_addr/enu/trigger/id , dojde k vygenerování události, která má parameter id shodný s uvedeným za slovem 'trigger/' v http příkazu. Na tento požadavek intercom odpoví jednoduchou odpovědí (200 OK).

Parametry

- Name – definuje jednoznačný identifikátor pro HTTP příkaz. Může obsahovat znaky abecedy a číslice.

Proměnné

Událost HttpTrigger je vždy vyvolaná přijetím HTTP příkazu, který umožňuje nést seznam uživatelských parametrů, který je součástí URI příkazu.

http://ip_adresa/enu/trigger/id?param1=value1&param2=value2

Seznam parametrů následuje za znakem ?. Každý parametr musí mít uveden název a hodnotu. Hodnota a název parametru jsou odděleny znakem =. Pokud seznam obsahuje více než jeden parametr, pak jsou jednotlivé parametry odděleny znakem &.

Jednotlivé parametry jsou při přijetí http příkazu k dispozici jako proměnné bloku HttpTrigger. Název proměnné je shodný s názvem předaného parametru, tedy $(line.param1) a $(line.param2)

Příklad

Událost generovaná při přijetí HTTP příkazu ve tvaru http://ip_addr/enu/trigger/opendoor :

- Event.HttpTrigger: Name=opendoor

MulticastTrigger

Blok MulticastTrigger definuje událost generovanou při přijetí příkazu odeslaného pomocí akce SendMulticastRequest. Příkaz tvoří zpráva odeslaná pomocí UDP protokolu na multicastovou adresu (235.255.255.250:4433) a tudíž může být přijata více zařízeními současně. Zpráva obsahuje identifikaci příkazu (parametr Command) a případné další parametry. Zpráva může být zabezpečena pomocí hesla (parametr Password).

Parametry

- Command – definuje identifikátor příkazu. Pomocí tohoto parametru lze odlišit různé typy odesílaných příkazů. Blok MulticastTrigger reaguje na akci SendMulticastRequest pouze tehdy, pokud má uveden stejný identifikátor příkazu. Identifikátor může být libovolný text obsahující znaky A-Z, a-z a 0-9. V názvu příkazu se rozlišují velká a malá písmena.
- Password – definuje heslo, pomocí kterého je příkaz zabezpečen proti neautorizovanému přístupu. Uvedené heslo se musí shodovat s heslem definovaným v akci SendMulticastRequest, na kterou má MulticastTrigger reagovat.
- CheckTime – umožňuje zapnout režim kontroly času přijetí příkazu s časem uvedeným ve zprávě příkazu a tím zamezit možnosti útoku pomocí opakovaní již zpracované zprávy. Tato funkce vyžaduje synchronizovaný čas (pomocí NTP serveru) na všech zařízení vysílajících a přijímajících příkazy.
    - Platné hodnoty:

0 – čas zprávy není kontrolován

1 – čas zprávy je kontrolován (vyšší bezpečnost)

Parametr je nepovinný, implicitní hodnota je 0.

Proměnné

Událost MulticastTrigger je vždy vyvolaná přijetím hromadného příkazu, který umožňuje nést seznam uživatelských parametrů (parameter Params akce MulticastRequest). Každý z těchto parametrů je identifikován vlastním jménem, který si zvolí uživatel a tyto odeslané parametry jsou k dispozici jako proměnné (s identickým jménem) v bloku MulticastTrigger.

Příklad: Byl přijat hromadný příkaz vyvolaný akcí MulticastRequest, u které byl uveden parametr Params="AAA=123". Událost MulticastTrigger, která tento příkaz zpracuje, bude mít automaticky nastavenou proměnnou AAA na hodnotu 123. Na tuto proměnnou je možné se odkázat v dalších navazujících blocích.

Příklad

Událost generovaná při přijetí hromadného příkazu opendoor:

- Event.MulticastTrigger: Command=opendoor

Event.AudioLoopTest

Blok AudioLoopTest definuje událost generovanou po provedení testu funkčnosti mikrofonu a reproduktoru zařízení (Audio Test). Na základě výsledku testu lze provést další akce.

Parametry

- Result – umožňuje specifikovat požadovaný výsledek audio testu.
  - Platné hodnoty:
    - any – událost je generována po každém provedení testu bez ohledu na výsledek
    - passed – událost je generovaná po provedení testu, pokud je výsledek pozitivní
    - failed – událost je generovaná po provedení testu, pokud je výsledek negativní
  - Parametr je nepovinný, implicitní hodnota je failed.

Proměnné

Tento blok nedefinuje žádné specifické proměnné.

Příklad

Událost generovaná po provedení audio testu, pokud je výsledek testu negativní (tj. mikrofon nebo reproduktor není funkční):

- Event.AudioLoopTest: Result=failed

Event.Time

Blok Time definuje událost generovanou každý den v zadaném čase. Pro omezení platnosti pouze na určité dny použijte u spouštěné akce podmínku Condition. ProfileState a do použitého časového profilu nastavte požadované dny.

Parametry

- Time – definuje čas, který spouští událost. Čas se zadává ve formátu hh:mm.

Proměnné

Tento blok nedefinuje žádné specifické proměnné.

Příklad

Událost generovaná každý den v 17:30.

- Event.Time: Time=17:30

Event.MotionDetected

Blok MotionDetected definuje událost generovanou v případě detekování pohybu. Pohyb může být detekován pouze interní kamerou. Parametry detekce pohybu jsou nastaveny v menu Hardware / Kamera / Interní kamera, sekce Nastavení detekce pohybu.

Parametry

- State – umožňuje nastavit, jestli bude událost generovat v okamžiku začátku nebo konce pohybu.
  - Platné hodnoty:
    - 1 – začátek detekce pohybu
    - 0 – konec detekce pohybu
  - Parametr je nepovinný, implicitní hodnota je 1.

Proměnné

Tento blok nedefinuje žádné specifické proměnné.

Příklad

Událost generovaná v případě, kdy začal být detekován pohyb.

- Event.MotionDetected: State=1

Event.NoiseDetected

Blok NoiseDetected definuje událost generovanou v případě detekování hluku. Hluk může být detekován pouze interním mikrofonem. Parametry detekce hluku jsou nastaveny v menu Hardware / Audio, sekce Nastavení detekce hluku.

Parametry

- State – umožňuje nastavit, jestli bude událost generovat v okamžiku začátku nebo konce hluku.
  - Platné hodnoty:
    - 1 – začátek detekce hluku
    - 0 – konec detekce hluku
  - Parametr je nepovinný, implicitní hodnota je 1.

Proměnné

Tento blok nedefinuje žádné specifické proměnné.

Příklad

Událost generovaná v případě, kdy začal být detekován hluk.

- Event.NoiseDetected: State=1

4. Akce (Action)

Automation definuje následující typy akcí:

- ActivateSwitch – aktivace spínače
- SetOutput – nastavení stavu digitálního výstupu
- BeginCall – vytvoření odchozího hovoru
- AnswerCall – vyzvednutí příchozího hovoru
- EndCall – ukončení hovoru
- SendHttpRequest – odeslání HTTP příkazu
- SendMulticastRequest – odeslání příkazu více zařízením současně
- PlayUserSound – přehrání uživatelského zvuku
- StartMulticastSend – spuštění vysílání audio streamu
- StopMulticastSend – zastavení vysílání audio streamu
- StartMulticastRecv – spuštění příjmu audio streamu
- StopMulticastRecv – zastavení vysílání audio streamu
- SetCameraInput – nastavuje použitou kameru
- ControlRtpStream – řídí přehrávání RTP streamu
- LogEvent – pošle zachycený event na syslog server
- SendEmail – odesílá email
- SetOnvifVirtualInput – virtuální vstup pro ONVIF
- SendWiegandCode – odešle kód na sběrnici Wiegand
- UploadSnapshotToFtp – nahraje snímek na FTP server
- StartAutoUpdate – stáhne aktuální firmware a konfigurace

Action.ActivateSwitch

Blok ActivateSwitch definuje akci pro sepnutí spínače interkomu nakonfigurovaného v záložkách Spínač 1–4. Činnost, která se provede při aktivaci spínače je zcela závislá na nastavení konkrétního spínače (může dojít k sepnutí digitálního výstup, odeslání HTTP požadavku apod.) Deaktivace (vypnutí) spínače je řízeno nastavením parametrů spínače.

Parametry

- Event – Definuje událost, která spustí tuto akci.
- Condition – Definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Switch – Definuje aktivovaný spínač (1 až 4).
- State – Definuje stav bistabilního spínače (parametr se neuplatní pro monostabilní režim spínače).
    - Platné hodnoty:
        - on – spínač je aktivován
        - off – spínač je deaktivován
        - toggle – spínač je přepnut
    - Parametr je nepovinný, implicitní hodnota je on.

Příklad

Aktivuje spínač 1 v případě vzniku události definované na řádku 2 a platnosti podmínky definované na řádku 3:

- Action.ActivateSwitch: Switch=1; Event=2; Condition=3

Action.SetOutput

Blok SetOutput definuje akci pro nastavení výstupu interkomu do požadované úrovně.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Output – definuje nastavovaný výstup.
  - Vzorové hodnoty:
    - relay1 – relé 1 na základní jednotce
    - relay2 – relé 2 na základní jednotce
    - output1 – výstup 1 na základní jednotce
    - output2 – výstup 2 na základní jednotce
  - Seznam platných hodnot se může lišit pro různé modely interkomů 2N Helios IP, viz kap. Dostupné digitální vstupy a výstupy.
- Level – požadovaný úroveň výstupu. Tento parametr je nepovinný.
  - Platné hodnoty:
    - 0 – vypnutí výstupu
    - 1 – zapnutí výstupu (implicitní hodnota)

Příklad

Aktivuje výstup Output1 v případě vzniku události definované na řádku 2:

- Action.SetOutput: Output=output1; Event=2

Action.BeginCall

Blok **BeginCall** definuje akci pro vytvoření odchozího hovoru na definované telefonní číslo, SIP URI nebo uživatele v telefonním seznamu interkomu.

Parametry

- **Event** – definuje událost, která spustí tuto akci.
- **Condition** – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- **Number** – volané telefonní číslo (pokud je **2N Helios IP** registrován k ústředně)
- **Uri** – volané SIP URI ve tvaru **sip:user@domain**
- **User** – volané číslo uživatele v telefonním seznamu. Platné hodnoty jsou 1 až 999 (podle modelu interkomu).
- **Device** – volaná aplikace **2N® Helios IP Mobile** ve tvaru **device:název_zařízení**
- Lze uvést pouze jeden z parametrů **Number**, **Uri** nebo **User**.

Příklad

Uskuteční odchozí hovor v případě vzniku události č. 2:

- Action.BeginCall: Number=1001; Event=2

Action.AnswerCall

Blok AnswerCall definuje akci pro vyzvednutí příchozího hovoru. V případě, kdy na interkom nepřichází žádný hovor nebo příchozí hovor není ve stavu vyzvánění, akce neprovede žádnou činnost.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.

Příklad

Provede vyzvednutí hovoru v případě vzniku události č. 2:

- Action.AnswerCall: Event=2

Action.EndCall

Blok EndCall definuje akci pro ukončení probíhajícího hovoru. V případě, kdy na interkom neprobíhá žádný hovor, akce neprovede žádnou činnost.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.

Příklad

Provede ukončení hovoru v případě vzniku události č. 2:

- Action.EndCall: Event=2

Action.SendHttpRequest

Blok **SendHttpRequest** definuje akci pro odeslání HTTP příkazu jinému zařízení v síti. Pomocí HTTP příkazu lze ovládat jiné zařízení v síti (IP relé, nahrávací systém, jiný interkom apod.)

Parametry

- **Event** – definuje událost, která spustí tuto akci.
- **Condition** – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- **Uri** – standardní HTTP URI obsahující adresu cílového zařízení a volitelně cestu a příp. další parametry.
- **Username** – definuje uživatelské jméno pro případ, že HTTP server vyžaduje autentizaci. Parametr je nepovinný – implicitní hodnota je "helios".
- **Password** – definuje heslo v případě, že HTTP server vyžaduje autentizaci. Parametr je nepovinný.

Příklad

Odešle HTTP příkaz na zařízení s IP adresou 192.168.1.1, v případě vzniku události č. 2:

- Action.SendHttpRequest: uri=**http://192.168.1.1/enu/trigger/message**; Event=2

> ⚠ **Upozornění**
>
> HTTP příkazy používají URL kódování. Při následujícím nastavení Automation:
>
> 1. Event.KeyPressed: Key=Any
>
> 2. Action.SendHttpRequest: Uri=**http://192.168.1.1/message=$(1.Key)**; Event=1 se po stisku tlačítka rychlé volby č. 1 pošle **http://192.168.1.1/message=%251** ("%" je zakódováno jako "%25")

Příkaz	Stisknuté tlačítko	Poslaný příkaz
http://192.168.1.1 /message=$(1.Key)	Tlačítko rychlé volby 1	**http://192.168.1.1/message=%251** ("%" je zakódováno jako "%25")
http://192.168.1.1/mess? age=$(1.Key)	Tlačítko rychlé volby 1	**http://192.168.1.1/mess?age=%1** (obsahuje speciální znak ?)

Příkaz	Stisknuté tlačítko	Poslaný příkaz
http://192.168.1.1 /message=$(1.Key)	Tlačítko na klávesnici 1	http://192.168.1.1/message=1
http://192.168.1.1/mess? age=$(1.Key)	Tlačítko na klávesnici 1	http://192.168.1.1/mess?age=1

Action.SendMulticastRequest

Blok SendMulticastRequest definuje akci pro odeslání uživatelského příkazu více zařízením současně. Příjem takto odeslaného příkazu lze zpracovat pomocí bloku MulticastTrigger. Příkaz tvoří zpráva odeslaná pomocí UDP protokolu na multicastovou adresu (235.255.255.250:4433) a tudíž může být přijata více zařízeními současně. Zpráva obsahuje identifikaci příkazu (parametr Command) a případné parametr (parametr Params). Zpráva může být zabezpečena pomocí hesla (parametr Password).

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Command – definuje identifikátor příkazu. Pomocí tohoto parametru lze odlišit různé typy odesílaných příkazů. Blok MulticastTrigger reaguje na akci SendMulticastRequest pouze tehdy, pokud má uveden stejný identifikátor příkazu. Identifikátor může být libovolný text obsahující znaky A–Z, a–z a 0–9.
- Params – definuje parametry příkazu, které budou odeslány v UDP zprávě. Lze uvést jeden nebo více parametrů oddělených čárkou. Jednotlivé parametry musí být ve formátu "název_parametru=hodnota_parametru".
  - Příklad:
    - Params="Address=192.168.1.1", "Port=10000"
  - Takto odeslané parametry budou dostupné v události HttpTrigger reagující na tento příkaz jako proměnné Address a Port. Lze je následně využít např. v akcích navázaných na HttpTrigger.
- Password – definuje heslo, pomocí kterého je příkaz zabezpečen proti neautorizovanému přístupu. Parametr je nepovinný. Pokud není heslo uvedeno, příkaz není zabezpečen. Heslo může být libovolný text obsahující znaky A–Z, a–z a 0–9.

Příklad

Odešle příkaz na opendoor na všechna zařízení v síti, která mají odpovídajícím způsoben nastaven blok Event.MulticastTrigger v případě vzniku události č. 2:

- Action.SendMulticastRequest: Command=opendoor; Event=2

Action.PlayUserSound

Blok PlayUserSound definuje akci pro přehrání definovaného uživatelského zvuku.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Sound – definuje přehrávaný zvuk
  - Platné hodnoty
    - 1 – 10 – číslo uživatelského zvuku
    - *1 – moderní vyzvánění
    - *2 – velký gong
    - *3 – štěkání psa
    - *4 – lodní siréna
    - *5 – decentní gong
- Destination – definuje, kde se přehrává uživatelský zvuk.
  - Platné hodnoty:
    - speaker – zvuk se přehrává na interkomu
    - call – zvuk se přehraje do hovoru
  - Parametr je nepovinný, implicitní hodnota je speaker.

Příklad

Přehraje uživatelský zvuk č 1 v případě vzniku události č. 2:

- Action.PlayUserSound: Sound=1; Event=2

Action.StartMulticastSend

Blok StartMulticastSend definuje akci pro spuštění vysílání audio streamu na multicastovou IP adresu. Akcí lze ovládat až čtyři nezávislé vysílací kanály. Pro vysílání audio streamu je použit protokol RTP/UDP.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Channel – definuje číslo ovládaného kanálu (0–3).
- Address – definuje multicastovou IP adresu audio streamu.
- Port – definuje UDP port, na který bude audio stream odesílán.
- Source – definuje zdroj zvuku
  - Platné hodnoty:
    - mic – zdrojem zvuku je mikrofon
    - call – zdrojem zvuku hovor
  - Parametr je nepovinný, implicitní hodnota je mic.
- Codec – Definuje použitý audio kodek
  - Platné hodnoty
    - pcmu – kodek G.711 u-law
    - pcma – kodek G.711 A-law
    - g729 – kodek G.729
    - g722 – kodek G.722
    - l16 – kodek L16, 16 kHz
  - Parametr je nepovinný, implicitní hodnota je pcmu.

Příklad

Spuštění odesílání audio streamu pomocí kanálu 1 na adresu 239.0.0.1:10000 v případě vzniku události č. 2:

- Action.StartMulticastSend: Channel=1; Address=239.0.0.1; Port=10000; Event=2

Action.StopMulticastSend

Blok StopMulticastSend definuje akci pro zastavení vysílání audio streamu na multicastovou IP adresu.

Parametry

- Event – Definuje událost, která spustí tuto akci.
- Condition – Definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Channel – Definuje číslo ovládaného kanálu (0–3).

Příklad

Zastaví odesílání audio streamu v kanálu 1 v případě vzniku události č. 2:

- Action.StopMulticastSend: Channel=1; Event=2

Action.StartMulticastRecv

Blok StartMulticastRecv definuje akci spuštění příjmu audio streamu a jeho přehrávání. Akcí lze ovládat až čtyři nezávislé přijímací kanály. Pro příjem audio streamu je použit protokol RTP/UDP.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Channel – definuje číslo ovládaného kanálu (0–3).
- Address – definuje multicastovou IP adresu audio streamu.
- Port – definuje UDP port, na který bude audio stream přijímán.
- Volume – definuje relativní hlasitost přehrávaného audio streamu v rozsahu –6 dB až +6 dB.
  - Platné hodnoty:
    - -6 – minimální hlasitost
    - 0 – střední hlasitost (implicitní hodnota)
    - 6 – maximální hlasitost
  - Parametr je nepovinný, implicitní hodnota je 0.
- Codec – definuje použitý audio kodek
  - Platné hodnoty
    - pcmu – kodek G.711 u-law
    - pcma – kodek G.711 A-law
    - g729 – kodek G.729
    - g722 – kodek G.722
    - l16 – kodek L16, 16 kHz
  - Parametr je nepovinný, implicitní hodnota je pcmu.

Příklad

Spustí příjem audio streamu na multicastové IP adrese 239.0.0.1:10000 v kanálu 1 v případě vzniku události č. 2:

- Action.StartMulticastRecv: Channel=1; Address=239.0.0.1; Port=10000; Event=2

Action.StopMulticastRecv

Blok StopMulticastRecv definuje akci pro zastavení příjmu audio streamu na multicastovou IP adresu.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Channel – definuje číslo ovládaného kanálu (0–3).

Příklad

Zastaví příjem audio streamu v kanálu 1 v případě vzniku události č. 2:

- Action.StopMulticastRecv: Channel=1; Event=2

Action.SetCameraInput

Blok SetCameraInput definuje akci umožňující přepínat mezi různými zdroji video signálu pro aktivní hovor. Tato akce umožňuje přepínat mezi zabudovanou kamerou a externí IP kamerou, případně mezi dvěma vstupy pro připojení analogové kamery k modelu 2N Helios IP Video Kit. Touto akcí nelze přepínat zdroj videa pro RTSP stream.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Type – definuje typ zdroje video signálu. Změna během hovoru se projeví pouze pro tento hovor. Ostatní přijímače videa dostávají stále stejný kanál.
  - Platné hodnoty:
    - internal – interní kamera (resp. externí analogová připojená přímo k zařízení)
    - external – externí IP kamera
  - Parametr je nepovinný, implicitní hodnota je internal.

  Id – definuje kanál video signálu. Parametr je dostupný pouze na modelu 2N Helios IP Video Kit. Uplatňuje pouze v případě, kdy parametr Type je nastaven na hodnotu internal.
  - Platné hodnoty:
    - 1 – analogová kamera připojená na vstup 1
    - 2 – analogová kamera připojená na vstup 2
  - Parametr je nepovinný, implicitní hodnota je 1.

Příklad

Přepne zdroj video signálu na první vstup pro externí analogovou kameru v případě vzniku události č. 2:

- Action.SetCameraInput: Type=internal; Id=1; Event=2

Action.ControlRtpStream

Blok ControlRtpStream definuje akci pro nastavení přehrávání RTP streamu. Řízeny jsou pouze hovorové streamy, streamy multicastu ovlivněny nejsou.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Direction – definuje směr přehrávání RTP hovorového streamu.
  - Platné hodnoty:
    - in – příchozí stream na komunikátor
    - out – stream posílaný z komunikátoru
    - both – příchozí i odchozí stream
  - Parametr je nepovinný, implicitní hodnota je both.

  Operation – definuje operaci provedenou se streamem
  - Platné hodnoty:
    - mute – přehrávání streamu je zastaveno
    - unmute – přehrávání streamu je obnoveno.

Příklad

Vypne přehrávání streamů v obou směrech v případě vzniku události č. 2:

- Action.ControlRtpStream: Direction=both; Operation=mute; Event=2

Action.LogEvent

Blok LogEvent definuje akci, která pomocí syslogu pošle event, který akci vyvolal. Blok lze použít k ověření nastavení automation.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.

Příklad

Pomocí syslogu pošle zachycený event č.2 (Event.CardEntered) v případě vzniku události č. 2.:

- Action.LogEvent: Event=2

Poslaná syslog zpráva:

- LOCAL0.DEBUG: Jan 10 12:49:14.305 ACT: aut_action_logevent_callback(): Logged event 'CardEntered'

Action.SendEmail

Blok SendEmail definuje akci pro odeslání emailu.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Sender – definuje adresu odesílatele pro odchozí e-maily ze zařízení.
- Subject – definuje předmět odesílané e-mailové zprávy.
  - Parametr je nepovinný, implicitní hodnota je nastavena pomocí parametru Předmět v menu Služby/E-Mail/E-Mail – hovor.
- Body – definuje obsah zprávy, kterou lze upravit obsah odesílané zprávy. V textu lze používat formátovací značky jazyka HTML. Do textu lze vkládat speciální zástupné symboly pro jméno uživatele, datum a čas, identifikaci zařízení příp. volané číslo. Tyto zástupné symboly budou před odesláním zprávy nahrazeny aktuální hodnotou. Viz následující tabulka zástupných symbolů:
  1. $User$ – jméno volaného uživatele
  2. $DateTime$ – aktuální datum a čas
  3. $DialNumber$ – volané číslo
  4. $HeliosId$ – identifikace interkomu
  - Parametr je nepovinný, implicitní hodnota je nastavena pomocí parametru Obsah zprávy v menu Služby/E-Mail/E-Mail – hovor.
- Snapshots – definuje počet snímků, které budou k e-mailu přiloženy (0–1).
  - Parametr je nepovinný, implicitní hodnota je 0.
- Width – definuje parametr nastavující šířku rozlišení přiloženého obrázku z kamery. Šířka snímku musí odpovídat jedné z podporovaných variant rozlišení interkomu.
  - Parametr je nepovinný, implicitní hodnota je 640.
- Height – definuje parametr nastavující výšku rozlišení přiloženého obrázku z kamery. Výška snímku musí odpovídat jedné z podporovaných variant rozlišení interkomu.
  - Parametr je nepovinný, implicitní hodnota je 480.
- User – definuje uživatele, kterému se pošle email.
  - Platné hodnoty:
    - pozice_uživatele – Numerické vyjádření pozice uživatele

- **Email** – Definuje emailovou adresu, na kterou se pošle email. V případě potřeby lze zadat více e-mailových adres oddělených čárkou v uvozovkách.
  - Platné hodnoty:
    - uživatel@doménové_jméno
    - uživatel@ip_adresa
    - "uživatel@doménové_jméno, uživatel@ip_adresa"

Tip

- Parametr **User** má přednost před parametrem **Email**.

Příklad

Pošle email na emailovou adresu nastavenou na uživatel2@doménové_jméno v případě vzniku události č. 1:

- Action.SendEmail: Event=1; Sender=uživatel1@doménové_jméno; Email=uživatel2@doménové_jméno; Subject=Subject; Body=Body; Snaphots=1; Width=640; Height=480

Action.SetOnvifVirtualInput

Blok SetOnvifVirtualInput definuje akci pro poslání změny virtuální vstupu pomocí protokolu ONVIF. Pro odzkoušení je možné použít například program ONVIF Device Manager (verze 2.2.250).

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Port – definuje číslo virtuálního portu,
  - Platné hodnoty:
    - 0–10 – číslo portu

  Level – definuje vstupní úroveň
  - Platné hodnoty:
    - 1 – nastavuje logickou hodnotu true
    - 0 – nastavuje logickou hodnotu false.
  - Parametr je nepovinný, implicitní hodnota je 1 (true).

Příklad

Pošle pomocí eventu v ONVIF informaci o tom, že port číslo 8 změnil svou hodnotu na úroveň logická 1 v případě vzniku události č. 2:

- Action.SetOnvifVirtualInput: Port=8; Level=1; Event=2

Na ONVIF je posláno:

- InputToken: onvif_port_08
- LogicalState:true

Action.SendWiegandCode

Blok SendWiegandCode definuje akci pro poslání zadaného kódu na sběrnici Wiegand.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Code – definuje kód, který se posílá přes sběrnici Wiegand. Pokud je zadaný kód delší, jež odpovídá zvolenému formátu, tak jsou oříznuty nejvyšší bity kódu.
  - Platné hodnoty
    - Dekadické číslo
  - Parametr je povinný
- Format – definuje formát zprávy na sběrnici Wiegand
  - Platné hodnoty
    - wiegand26 – 26 bitů
    - wiegand32 – 32 bitů
    - wiegand37 – 37 bitů
  - Parametr je nepovinný, implicitní hodnota je wiegand26
- Facility – facility kód zařízení. Nastavení se uplatní pouze pro formát „wiegand26".
  - Platné hodnoty
    - Dekadické číslo v rozmezí 0–255
  - Parametr je nepovinný, pokud není parametr nastaven, tak není použit.
- Module – definuje modul, přes který je kód odeslán.
  - Platné hodnoty
    - Jméno modulu nastavené v menu Hardware / Rozšiřující moduly / Modul Wiegand, parametr Jméno modulu
  - Parametr je povinný pro model Verso a Access Unit, ostatních modelů se netýká

Příklad

Pošle na sběrnici Wiegand kód zadaný v události 1. pomocí Event.CodeEntered.

- 1. Event.CodeEntered: Code=Any
- 2. Action.SendWiegandCode: Code=$(1.code); Event=1

Pro Verso a Access Unit pak druhý řádek vypadá následovně:

- 2. Action.SendWiegandCode: Code=$(1.code); Event=1; Module=1

Action.UploadSnapshotToFtp

Blok UploadSnapshotToFtp definuje akci pro nahrání snímku z kamery na FTP server. Parametry FTP serveru a snímku jsou nastaveny v menu Služby / Streamování / FTP.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.
- Source – definuje zdroj videa, ze kterého se má nahrát snímek na FTP server.
  - Platné hodnoty:
    - Auto – zdroj videa je vybrán podle nastavení Hardware / Kamera / Společné nastavení / Výchozí zdroj videa
    - Internal – interní kamera
    - External – externí kamery.
  - Parametr je nepovinný, implicitní hodnota je Auto

Příklad

Uloží fotku z kamery na FTP server v případě vzniku události č. 2:.

- Action.UploadSnapshotToFtp: Event=2

Action.StartAutoUpdate

Blok StartAutoUpdate definuje akci pro spuštění automatické aktualizace firmware a konfigurace. Parametry aktualizace jsou nastaveny v menu Systém / Aktualizace.

Parametry

- Event – definuje událost, která spustí tuto akci.
- Condition – definuje podmínku, která musí být splněna, aby akce byla provedena. Tento parametr je nepovinný.

5. Podmínky (Condition)

Automation definuje následující typy podmínek:

- ProfileState – stav časového profilu
- CallState – stav probíhajícího hovoru
- AccountState – stav registrace k SIP účtu
- InputState – stav digitálního vstupu
- LogicalAnd – logický součin podmínek
- LogicalOr – logický součet podmínek
- LogicalNot – negace podmínky
- True – vždy pravdivá podmínka
- False – vždy nepravdivá podmínka
- FlipFlopD – klopný obvod typu D
- FlipFlopRS – klopný obvod typu RS

Detailní popis podmínek, jejich parametry a použití jsou popsány v následujícím textu.

Condition.ProfileState

Blok **ProfileState** definuje podmínku, která je splněna v případě aktivního (příp. neaktivního) časového profilu.

Parametry

- **Profile** – číslo časového profilu (1–20 podle modelu interkomu).
- **State** – požadovaný stav profilu. Tento parametr je nepovinný.
  - Platné hodnoty:
    - **active** – profil je aktivní (implicitní hodnota)
    - **inactive** – profil je neaktivní

Příklad

Podmínka splněna v případě neaktivního časového profilu č. 1:

- Condition.ProfileState: Profile=1; State=Inactive

Condition.CallState

Blok CallState je splněn v případě definovaného stavu probíhajícího hovoru.

Parametry

- State – definuje stav hovoru.
  - Platné hodnoty:
    - idle – hovor neprobíhá
    - connecting – hovor se spojuje (pouze odchozí hovory)
    - ringing – probíhá vyzvánění
    - connected – hovor je spojen
- Direction – definuje směr hovoru.
  - Platné hodnoty:
    - incoming – příchozí hovory
    - outgoing – odchozí hovory
    - any – oba směry
  - Parametr je nepovinný, implicitní hodnota je any.

Příklad

Podmínka splněna v případě neaktivního hovoru:

- Condition.CallState: State=Idle

Condition.AccountState

Blok **AccountState** je splněn v případě úspěšné registrace k SIP účtu.

Parametry

- **Account** – definuje použitý SIP účet.
  - Platné hodnoty:
    - **1** – účet 1
    - **2** – účet 2
  - Parametr je nepovinný, implicitní hodnota je 1.
- **State** – Definuje stav registrace.
  - Platné hodnoty:
    - **registered** – účet je zaregistrován
    - **unregistered** – účet není zaregistrován
  - Parametr je nepovinný, implicitní hodnota je **registered**.

Příklad

Podmínka splněna v případě nezaregistrovaného 1. účtu:

- Condition.AccountState: State=unregistered

Condition.InputState

Blok InputState definuje podmínku, která je splněna v případě připojení definované logické úrovně na definovaný digitální vstup.

Parametry

- Input – definuje digitální vstup.
  - Platné hodnoty:
    - tamper – tamper spínač
    - input1 – digitální vstup 1
    - input2 – digitální vstup 2
    - cr_input1 – digitální vstup 1 na čtečce karet
    - cr_input2 – digitální vstup 2 na čtečce karet
  - Seznam platných hodnot se může lišit pro různé modely interkomů 2N Helios IP, viz kap. Dostupné digitální vstupy a výstupy.
- Level – Definuje požadovanou úroveň na digitálním vstupu. Parametr je nepovinný.
  - Platné hodnoty:
    - 0 – logická 0
    - 1 – logická 1 (implicitní hodnota)

Příklad

Podmínka splněna v případě sepnutého spínače tamperu (zařízení není otevřeno):

- Condition.InputState: Input1=tamper; Level=0

Condition.LogicalAnd

Blok LogicalAnd umožňuje vytvářet složitější zřetězení různých podmínek. Blok je splněn v případě splnění všech podmínek z definované skupiny.

Parametry

- Condition – definuje seznam podmínek, které mají být splněny. Jednotlivé podmínky se oddělují čárkou.

Příklad

Podmínka splněna v případě současného splnění podmínek 1, 2 a 3:

- Condition.LogicalAnd: Condition=1, 2, 3

Condition.LogicalOr

Blok LogicalOr umožňuje vytvářet složitější zřetězení různých podmínek. Tento blok je splněn v případě splnění alespoň jedné podmínky z definované skupiny.

Parametry

- Condition – Definuje seznam podmínek, které mají být splněny. Jednotlivé podmínky se oddělují čárkou.

Příklad

Podmínka splněna v případě současného splnění alespoň jedné z podmínek 1, 2 nebo 3:

- Condition.LogicalOr: Condition=1, 2, 3

Condition.LogicalNot

Blok LogicalNot definuje podmínku, která je splněna v případě nesplnění jiné definované podmínky.

Parametry

- Condition – definuje podmínku, které nemá být splněna

Příklad

Podmínka splněna v případě nesplnění podmínky 1:

- Condition.LogicalNot: Condition=1

Condition.True

Blok **True** definuje podmínku, která je vždy splněna.

Parametry

- Blok True nemá žádné parametry.

Příklad

Podmínka je vždy splněna:

- Condition.True

Condition.False

Blok False definuje podmínku, která není nikdy splněna.

Parametry

- Blok False nemá žádné parametry.

Příklad

Podmínka je vždy nesplněna:

- Condition.False

Condition.FlipFlopD

Blok FlipFlopD je jednobitová paměťová buňka (proměnná), která si zaznamená stav jiné podmínky v okamžiku vzniku definované události pro pozdější použití. Výstup bloku FlipFlopD lze použít jako podmínku řídící další akce ve složitějších případech použití Automation. Jedná se o simulaci klopného obvodu typu D.

Parametry

- ClockEvent – definuje událost, při které je zaznamenán aktuální stav podmínky Condition.
- Condition – definuje podmínku, která je zaznamenána při vzniku události ClockEvent.
- ResetValue – výchozí hodnota podmínky po restartu zařízení. Parametr není povinný.
- Platné hodnoty parametrů:
  - 0 – podmínka je nesplněná (implicitní hodnota)
  - 1 – podmínka je splněná

Příklad

Stav podmínky bude shodný se stavem podmínky 2 v okamžiku vzniku události 1:

- Condition.FlipFlopD: ClockEvent=1; Condition=2

Condition.FlipFlopRS

Blok FlipFlopRS je jednobitová paměťová buňka (proměnná), která mění svůj stav na 1 příp. 0 při vzniku definovaných událostí. Výstup bloku FlipFlopRS lze použít jako podmínku řídící další akce ve složitějších případech použití Automation. Jedná se o simulaci klopného obvodu typu R-S.

Parametry

- SetEvent – definuje událost, která nastaví podmínku do stavu splněno (1).
- ResetEvent – definuje událost, která nastaví podmínku do stavu nesplněno (0).
- ResetValue – výchozí hodnota podmínky po restartu zařízení. Parametr není povinný.
- Platné hodnoty parametrů:
  - 0 – podmínka je nesplněná (implicitní hodnota)
  - 1 – podmínka je splněná

Příklad

Podmínka bude splněna při vzniku události 1 a nesplněna při vzniku události 2:

- Condition.FlipFlopRS: SetEvent=1; ResetEvent=2

6. Dostupné digitální vstupy a výstupy

V této kapitole jsou popsány dostupné digitální vstup a výstup na jednotlivých modelech interkomů 2N Helios IP.

- 2N® Helios IP Vario
- 2N® Helios IP Force/Safety
- 2N® Helios IP Audio/Video Kit
- 2N® Helios IP Verso
- 2N® SIP Speaker
- 2N® Access Unit

2N ® Helios IP Vario

Výstupy

- relay1 – reléový výstup na základní jednotce
- relay2 – reléový výstup na přídavném spínači (pokud je instalován)
- cr_relay1 – reléový výstup 1 na čtečce karet (pokud je instalována)
- cr_relay2 – reléový výstup 2 na čtecce karet (pokud je instalována)
- led_secured – červená signalizační led pod jmenovkami (pouze modely 9137xxxU bez displeje)

Vstupy

- cr_input1 – digitální vstup 1 na čtečce karet (pokud je instalována)
- cr_input2 – digitální vstup 2 na čtečce karet (pokud je instalována)

2N ® Helios IP Force/Safety

Výstupy

- relay1 – reléový výstup na základní jednotce
- output1 – aktivní digitální výstup na základní jednotce (pouze verze desek 555v3 a vyšší, desky 555v2 mají aktivní digitální výstup spojený s výstupem relay1)
- relay2 – reléový výstup na přídavném spínači (pokud je instalován)
- output2 – aktivní digitální výstup na přídavném spínači (pokud je instalován)
- cr_relay1 – reléový výstup na čtečce karet (pokud je instalována)
- cr_output1 – aktivní digitální výstup na čtečce karet (pokud je instalována)
- led_secured – červená signalizační led na čtečce karet (pokud je čtečka instalována)
- led_ringing – oranžová led signalizující vyzvánění (pouze modely s piktogramy)
- led_connected – modrá led signalizující spojený hovor (pouze modely s piktogramy)
- led_door – zelená led signalizující otevření dveří (pouze modely s piktogramy)
- led_key1 – podsvětlení prvního tlačítka na Safety
- led_key2 – podsvětlení druhého tlačítka na Safety
- led_key3 – podsvětlení třetího tlačítka na Safety

Vstupy

- tamper – tamper spínač (pokud je instalován)
- cr_input1 – digitální vstup 1 na čtečce karet (pokud je instalována)
- cr_input2 – digitální vstup 2 na čtečce karet (pokud je instalována)

Výstupy

- relay1 – reléový výstup
- output1 – digitální výstup 1
- output2 – digitální výstup 2
- led1 – výstup pro řízení LED 1
- led2 – výstup pro řízení LED 2
- led3 – výstup pro řízení LED 3

Vstupy

- input1 – digitální vstup 1
- input2 – digitální vstup 2

2N ® Helios IP Verso

Základní jednotka

Výstupy

- output1 – digitální výstup 12 V
- relay1 – reléový výstup
- led_secured – červená signalizační led na panelu

Vstupy

- input1 – digitální vstup na základní jednotce

Modul I/O

Vstupy / výstupy jsou adresovány <jméno_modulu>.<jméno_vstupu/výstupu>, např. modul5.relay1.

Jméno modulu se nastavuje v menu Hardware / Rozšiřující moduly, parametr jméno modulu.

Výstupy

- relay1 – reléový výstup 1
- relay2 – reléový výstup 2

Vstupy

- input1 – digitální vstup 1
- input2 – digitální vstup 2
- tamper – tamper spínač (pokud je instalován)

Modul Wiegand

Vstup je adresován <jméno_modulu>.<jméno_vstupu>, např. modul2.tamper

Jméno modulu se nastavuje v menu Hardware / Rozšiřující moduly, parametr jméno modulu.

Výstupy

- output1 – výstup LED OUT

Vstupy

- input1 – vstup LED IN
- tamper – tamper spínač (pokud je instalován)

2N ® SIP Speaker

Výstupy

- relay1 – reléový výstup

Vstupy

- Nejsou k dispozici.
- Lze použít Event.KeyPressed: Key=%1 pro události vygenerované na vstupu LOGIC IN

2N ® Access Unit

Základní jednotka

Výstupy

- output1 – digitální výstup 12 V
- relay1 – reléový výstup
- led_secured – červená signalizační led na panelu

Vstupy

- input1 – digitální vstup č. 1 na základní jednotce
- input2 – digitální vstup č. 2 na základní jednotce
- input3 – digitální vstup č. 3 na základní jednotce
- tamper – tamper spínač

Modul I/O

Vstupy / výstupy jsou adresovány <jméno_modulu>.<jméno_vstupu/výstupu>, např. modul5.relay1.

Jméno modulu se nastavuje v menu Hardware / Rozšiřující moduly, parametr jméno modulu.

Výstupy

- relay1 – relélový výstup 1
- relay2 – reléový výstup 2

Vstupy

- input1 – digitální vstup 1
- input2 – digitální vstup 2
- tamper – tamper spínač (pokud je instalován)

Modul Wiegand

Vstup je adresován <jméno_modulu>.<jméno_vstupu>, např. modul2.tamper

Jméno modulu se nastavuje v menu Hardware / Rozšiřující moduly, parametr jméno modulu.

Výstupy

- output1 – výstup LED OUT

Vstupy

- input1 – vstup LED IN
- tamper – tamper spínač (pokud je instalován)

7. Příklady použití

Volání na dispečink při neoprávněném otevření zařízení

Zadání

Po rozepnutí tamper spínače (otevření zařízení) volat na zvolené telefonní číslo.

Blokové schéma

V okamžiku náběžné hrany na vstupu tamper (1: Event.InputChanged) se vyvolá akce volání na zvolené telefonní číslo (2: Action.BeginCall).

Nastavení interkomu

1: Event.InputChanged: Input=tamper

2: Action.BeginCall: Number=1111; Event=1

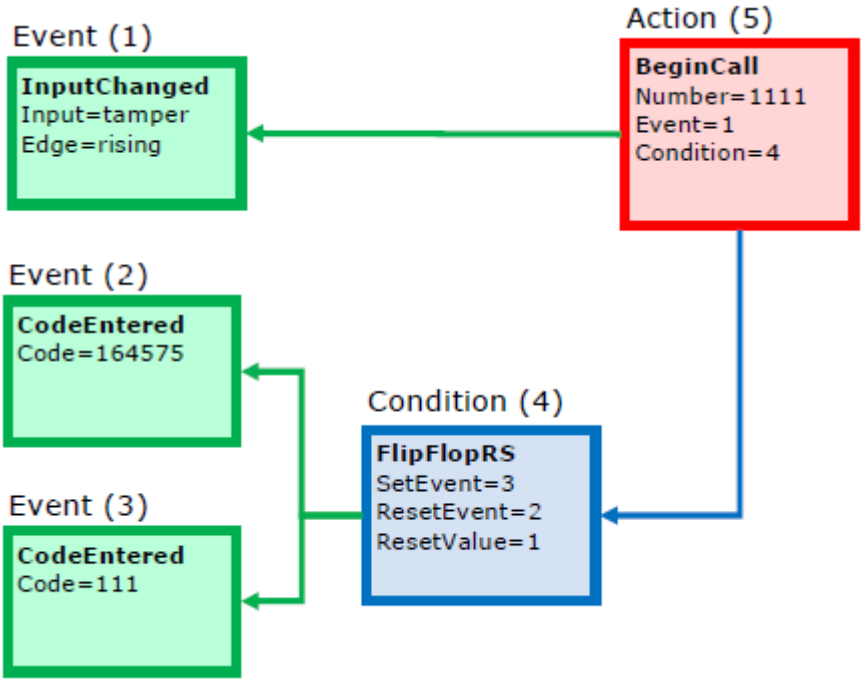
Volání na dispečink při neoprávněném otevření zařízení s možností blokování servisním kódem

Zadání

Po rozepnutí tamper spínače (otevření zařízení) volat na telefonní zvolené telefonní číslo. Umožnit blokování a opětovné povolení poplachu numerickým kódem zadaným na klávesnici interkomu.

Blokové schéma

V okamžiku náběžné hrany na vstupu tamper (1: Event.InputChanged) se volá na zvolené telefonní číslo (5: Action.BeginCall) v případě splnění podmínky. Podmínka (4: Condition.FlipFlopRS) je platná po restartu interkomu nebo po zadání zvoleného kódu (2: Condition.CodeEntered) na numerické klávesnici. Podmínka bude neplatná po zadání jiného zvoleného kódu (3: Condition.CodeEntered).



Nastavení interkomu

1: Event.InputChanged: Input=tamper; Edge=rising

2: Event.CodeEntered: Code=164575

3: Event.CodeEntered: Code=111

4: Condition.FlipFlopRS: SetEvent=3; ResetEvent=2; ResetValue=1

5: Action.BeginCall: Number=1111; Event=1; Condition=4

Otevření dveří po protažení RFID karty

Zadání

Po přiložení konkrétní RFID karty aktivovat spínač dveřního kontaktu.

Blokové schéma

V okamžiku přiložení RFID karty se zadaným ID (1: Event.CardEntered) dojde k aktivaci spínače 1 (2: Action.ActivateSwitch).

Nastavení interkomu

1: Event.CardEntered: Card=0*0000

2: Action.ActivateSwitch: Switch=1; Event=1

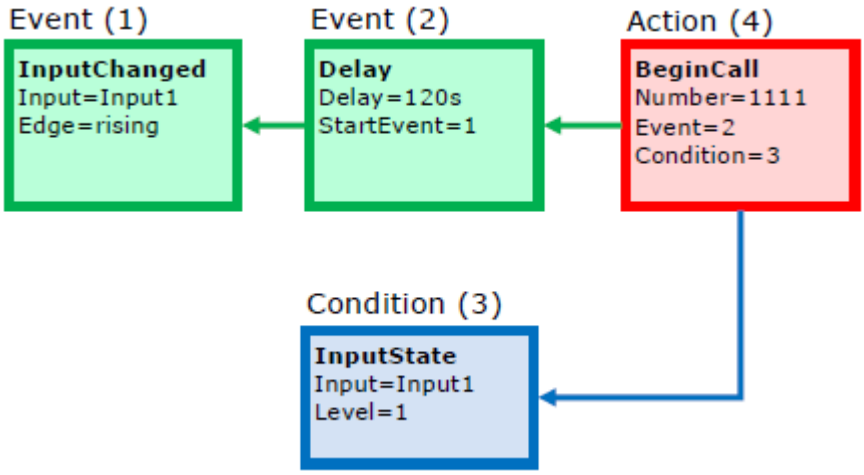
Poplach (volání na dispečink) při otevření dveří delším než 2 minuty

Zadání

V případě, že jsou dveře otevřeny po dobu delší než 2 minuty volat na dispečink. Příklad předpokládá, že na vstup Input1 je připojen kontakt, který signalizuje otevření dveří.

Blokové schéma

Po otevření dveří – náběžné hrana na signálu Input1 (1: Event.InputChanged) se se zpožděním 120 s (2: Event.Delay) se zavolá na zvolené telefonní číslo (4: Action. BeginCall). Volání se provede jen tehdy, pokud po uplynutí 120 s jsou dveře stále otevřeny (3: Condition.InputState).



Nastavení interkomu

1: Event.InputChanged: Input=input1; Edge=rising

2: Event.Delay: Delay=120s; StartEvent=1

3: Condition.InputState: Input=input1; Level=1

4: Action.BeginCall: Number=1111; Event=2; Condition=3

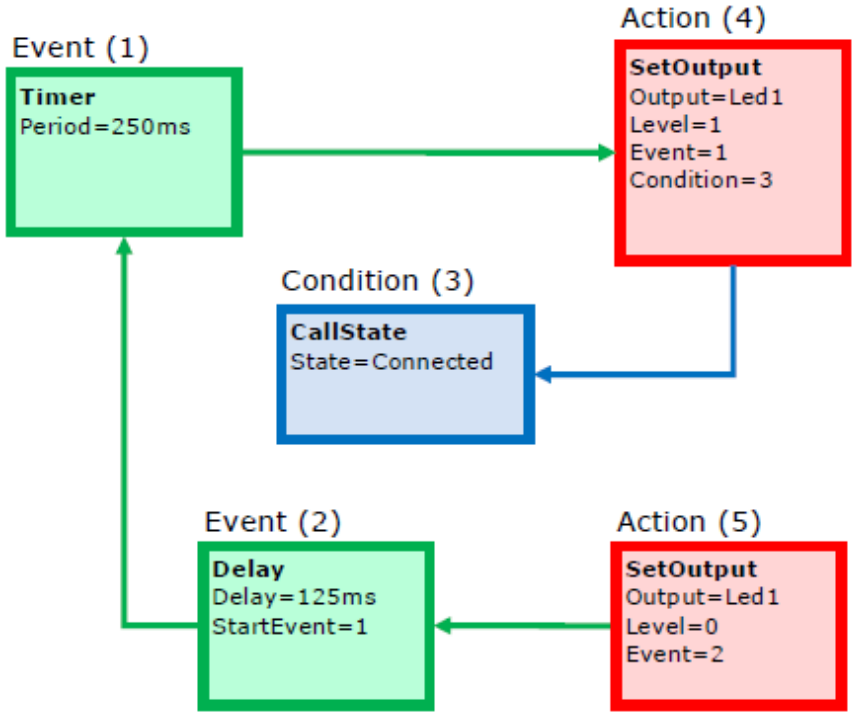
Blikání s LED v průběhu hovoru / Blikání v průběhu odemčení elektrického zámku dveří

Zadání

V průběhu aktivního hovoru blikat s LED.

Blokové schéma

Funkce blikání je řešena pomocí periodického časovače (1: Event.Timer) a zpoždění (2: Event.Delay). Tyto dva bloky definují periodu (250 ms) a střídu signálu – resp. dobu rozsvícení LED (125 ms). Na tyto dvě události jsou navázány akce pro rozsvícení (4: Action.SetOutput) a zhasnutí led (5: Action.SetOutput). Akce pro rozsvícení led je podmíněna probíhajícím hovorem (3: Condition.CallState).



Nastavení interkomu

1: Event.Timer: Period=250ms

2: Event.Delay: Delay=125ms; StartEvent=1

3: Condition.CallState: State=Connected

4: Action.SetOutput: Output=led1; Level=1; Event=1; Condition=3

5: Action.SetOutput: Output=led2; Level=0; Event=2

2N TELEKOMUNIKACE a.s.

Modřanská 621, 143 01 Prague 4, Czech Republic

Phone: +420 261 301 500, Fax: +420 261 301 599

E-mail: sales@2n.cz

Web: www.2n.cz

v2.17