



# 2N<sup>®</sup> Helios IP Automation



## Konfigurační manuál

Verze 2.3.0  
Firmware 2.3

[www.2n.cz](http://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.



2N<sup>®</sup> je registrovaná ochranná známka společnosti 2N TELEKOMUNIKACE a.s. Jména výrobků a jakákoli jiná jména zde zmíněná jsou registrované ochranné známky a/nebo ochranné známky a/nebo značky chráněné příslušným zákonem.



Prohlášení o shodě

Společnost 2N TELEKOMUNIKACE a.s. tímto prohlašuje, že zařízení 2N<sup>®</sup> Helios IP je ve shodě se základními požadavky a dalšími příslušnými ustanoveními směrnice 1999/5/ES. Plné znění prohlášení o shodě naleznete na příloženém CD-ROM nebo na [www.2n.cz](http://www.2n.cz).



Společnost 2N TELEKOMUNIKACE a.s. je vlastníkem certifikátu ISO 9001:2000. 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

<b>1.</b>	<b>2N<sup>®</sup> Helios IP Automation.....</b>	<b>5</b>
1.1	<b>Termíny a piktogramy .....</b>	<b>6</b>
	Piktogramy.....	6
1.2	<b>Konfigurace 2N<sup>®</sup> Helios IP Automation .....</b>	<b>7</b>
	Nastavení parametrů bloků .....	8
1.3	<b>Události (Event) .....</b>	<b>10</b>
	Event.KeyPressed .....	11
	Event.CodeEntered .....	12
	Event.CardEntered .....	13
	Event.CallStateChanged .....	14
	Event.InputChanged.....	15
	Event.Delay .....	16
	Event.Timer .....	17
	Event.HttpTrigger .....	18
1.4	<b>Akce (Action) .....</b>	<b>19</b>
	Action.ActivateSwitch .....	20
	Action.SetOutput .....	21
	Action.BeginCall .....	22
	Action.AnswerCall .....	23
	Action.EndCall .....	24
	Action.SendHttpRequest .....	25
1.5	<b>Podmínky (Condition) .....</b>	<b>26</b>
	Condition.ProfileState.....	27
	Condition.CallState.....	28
	Condition.InputState.....	29
	Condition.LogicalAnd .....	30
	Condition.LogicalOr .....	30
	Condition.LogicalNot .....	31
	Condition.True .....	31
	Condition.False.....	31
	Condition.FlipFlopD .....	32
	Condition.FlipFlopRS .....	33
1.6	<b>Dostupné digitální vstupy a výstupy.....</b>	<b>34</b>
1.7	<b>Příklady použití .....</b>	<b>36</b>



# 1

# 2N<sup>®</sup> Helios IP Automation

V této kapitole je popsána konfigurace produktu.

Zde je přehled toho, co v kapitole naleznete:

- Termíny a piktogramy
- Konfigurace 2N<sup>®</sup> Helios IP Automation
- Popis konfigurovatelných bloků

# 1.1 Termíny a piktogramy

## Piktogramy

---



### Nebezpečí úrazu

- Věnujte **vždy** pozornost těmto informacím, abyste předešli úrazu!



### Varování

- Věnujte **vždy** pozornost těmto informacím, abyste předešli zničení výrobku.



### Upozornění

- Informace důležité pro správnou funkci výrobku.



### Tip

- Užitečné rady.



### Poznámka

- Doplnující informace.

## 1.2 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 2N® Helios IP Automation. Typické použití 2N® Helios IP Automation je v aplikacích, které vyžadují složitější propojení se systémy třetích stran.

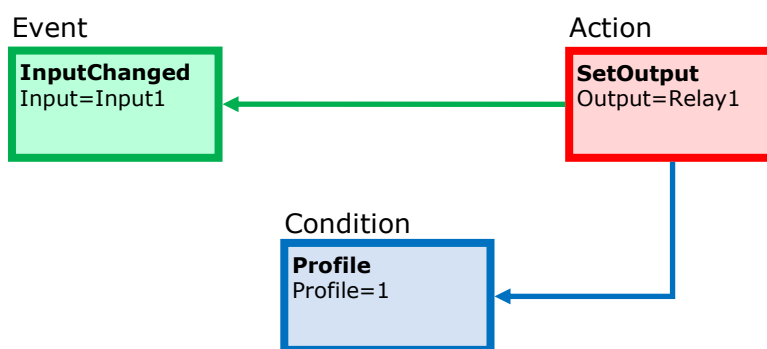


### Poznámka

- Funkcionalita 2N® Helios IP 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). 2N® Helios IP Automation umožňuje využít všechny tyto vstupy a výstupy a propojit je v různých kombinacích.

2N® Helios IP 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. 2N® Helios IP 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.



**Nastavení automatizace**

Stav automatizace: běží

Id	Typ bloku	Parametry	Stav
1	Event.InputChanged	Input=tamper; Edge=falling	✓
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	✓
6	Žádný		
7	Žádný		
8	Žádný		
9	Žádný		
10	Žádný		
11	Žádný		
12	Žádný		

2N® Helios IP umožňuje vytvořit a propojit až 12 bloků (nezáleží na tom, zda se jedná o události, akce nebo podmínky). Na událost nebo podmínku lze navázat více akcí. Můžete např. vytvořit 6 akcí a navázat je na 6 událostí nebo 11 akcí navázat na 1 událost.

## Nastavení parametrů bloků

Ve sloupci **Typ bloku** 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. 2N® Helios IP Automation pracuje pouze tehdy, pokud všechny vytvořené bloky jsou správně nakonfigurovány (tj. u všech je zelená značka). V opačném případě je funkce 2N® Helios IP 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.

## 1.3 Události (Event)

2N Helios IP Automation definuje následující typy událostí:

<b>KeyPressed</b>	Stisk tlačítka
<b>CodeEntered</b>	Zadaný numerický kód
<b>CardEntered</b>	Protažení RFID karty
<b>CallStateChanged</b>	Změna stavu hovoru
<b>InputChanged</b>	Změna digitálního vstupu
<b>HttpTrigger</b>	Přijat HTTP příkaz
<b>Delay</b>	Definované zpoždění
<b>Timer</b>	Časovač periodické události

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, .., %54** 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)

---

### Příklad

---

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

Action.KeyPressed: Key=#, %3, %4

## 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

#### 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)

---

### Příklad

---

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

#### 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)

---

### Příklad

---

Událost generovaná při protažení karty s ID 0\*0012456:

Event.CardEntered: Card=0\*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**.

---

### 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 vstup.

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 digitální 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.

---

### 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 zpozdít 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.

Příklad platných hodnot:

**10** – 10 sekund (jednotky není nutné uvádět)

**10s** – 10 sekund

**100ms** – 100 milisekund

---

### Příklad

---

Událost generovaná po 1s od vzniku události na řádce 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 zpozdřit 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 Event. Tento parametr je nepovinný.

#### Period

Definuje periodu časovače.

Příklad platných hodnot:

**10** – 10 sekund (jednotky není nutné uvádět)

**10s** – 10 sekund

**100ms** – 100 milisekund

#### 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).

---

### Příklad

---

Událost generovaná 3krát po 1s 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.

---

### Příklad

---

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

Event.HttpTrigger: Name=opendoor

## 1.4 Akce (Action)

2N Helios IP Automation definuje následující typy akcí:

<b>ActivateSwitch</b>	Aktivace spínače
<b>SetOutput</b>	Nastavení stavu digitálního výstupu
<b>BeginCall</b>	Vytvoření odchozího hovoru
<b>AnswerCall</b>	Vyzvednutí příchozího hovoru
<b>EndCall</b>	Ukončení hovoru
<b>SendHttpRequest</b>	Odeslání HTTP příkazu

## 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).

---

### 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.

Platné 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  
**cr\_relay1** – Relé 1 na čtečce karet  
**cr\_relay2** – Relé 2 na čtečce karet  
**cr\_output** – Výstup 1 na čtečce karet  
**redled** – Červená signalizační led  
**led1** – Signalizační LED 1  
**led2** – Signalizační LED 2  
**led3** – Signalizační LED 3

**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 výstupu Output1 v případě vzniku události definované na řádce 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

#### 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).

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.

---

### 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/message; Event=2
```

## 1.5 Podmínky (Condition)

2N Helios IP Automation definuje následující typy podmínek:

<b>ProfileState</b>	Stav časového profilu
<b>CallState</b>	Stav probíhajícího hovoru
<b>InputState</b>	Stav digitálního vstupu
<b>LogicalAnd</b>	Logický součin podmínek
<b>LogicalOr</b>	Logický součet podmínek
<b>LogicalNot</b>	Negace podmínky
<b>FlipFlopD</b>	Klopný obvod typu D
<b>FlipFlopRS</b>	Klopný obvod typu RS
<b>True</b>	Vždy pravdivá podmínka
<b>False</b>	Vždy nepravdivá podmínka

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: Id=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

**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=Inactive

## 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 současném 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 současném 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 splněna 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 2N Helios IP 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:

- 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 události. Výstup bloku FlipFlopRS lze použít jako podmínku řídící další akce ve složitějších případech použití 2N Helios IP 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:

**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

## 1.6 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

---

#### Výstupy

**relay1** – reléový výstup na základní jednotce

**relay2** – reléový výstup na přídatném spínači (pokud je instalován)

**cr\_relay1** – reléový výstup 1 na čtečce karet (pokud je instalována)

**cr\_relay2** – reléový výstup 2 na čtečce karet (pokud je instalována)

**redled** – 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** – digitální výstup 12V na základní jednotce (pouze verze desek 555v3 a vyšší, desky 555v2 mají digitální výstup 12V spojený s výstupem relay1)

**relay2** – reléový výstup na přídatném spínači (pokud je instalován)

**output2** – digitální výstup 12V na přídatném spínači (pokud je instalován)

**cr\_relay1** – reléový výstup na čtečce karet (pokud je instalována)

**cr\_output1** – digitální výstup 12V na čtečce karet (pokud je instalována)

**redled** – červená signalizační LED na čtečce karet (pokud je čtečka instalována)

#### 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)

---

## 2N Helios IP Uni

---

### Výstupy

**relay1** – reléový výstup na základní jednotce

### Vstupy

Nejsou k dispozici.

---

## 2N Helios IP Audio/Video Kit

---

### 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

## 1.7 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

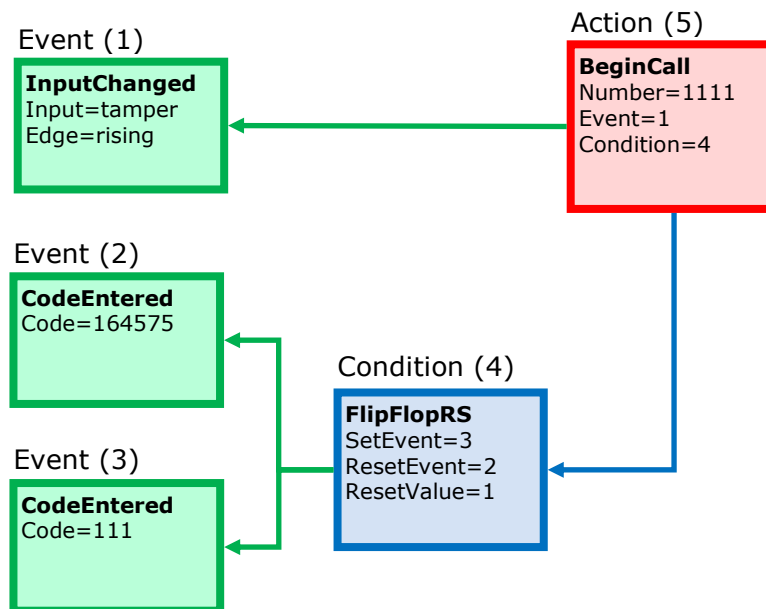
## Volání na dispečink při neoprávněném otevření zařízení s možností blokování servisním kódem

### 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

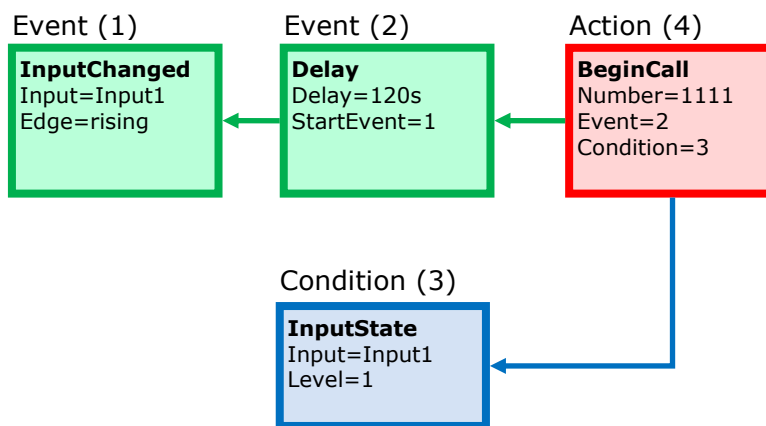
## Poplach(volání na dispečink) při otevření dveří delším než 2 minuty

### 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 120s (2: Event.Delay) se zavolá na zvolené telefonní číslo (4: Action.BeginCall). Volání se provede jen tehdy, pokud po uplynutí 120s 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

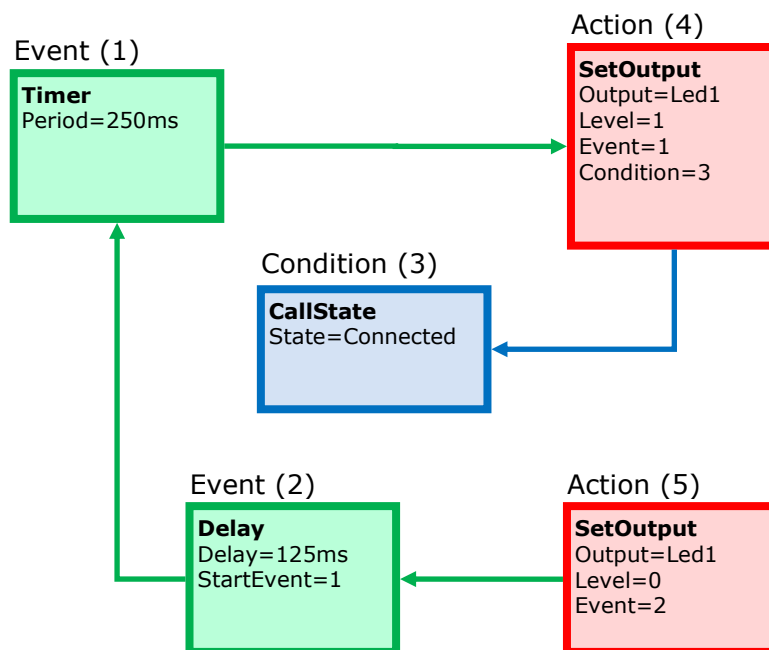
## Blikání s LED v průběhu hovoru / Blikání v průběhu odemčení elektrického zámku dveří

### 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 (250ms) a střidu signálu – resp. dobu rozsvícení LED (125ms). 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 Praha 4, Česká Republika  
Tel.: +420 261 301 111, Fax: +420 261 301 999  
E-mail: [obchod@2n.cz](mailto:obchod@2n.cz)  
Web: [www.2n.cz](http://www.2n.cz)