



Protocolos

Manual do usuário

Revisão 103 (5d909bc) · 14/10/2026

shc-docs.web.app/protocolos/

Sumário

- Native Device: Introdução e Vantagens 3
- Native Device: Comunicação UDP e TCP 4
- Native Device: Query Strings 6
- Ações 8
 - Descrição das Ações 8
 - Descrição da ação 9
 - Descrição da ação 10
 - LOUD — Controle de Zonas via UDP 11
 - Descrição da ação 17
 - Descrição da ação 18
 - Descrição da ação 19
 - Descrição da ação TCP 20
 - Descrição da ação UDP 21
 - Descrição da ação WOL 22
 - Descrição da ação 23
 - Descrição da ação IF 27

Native Device: Introdução e Vantagens

O **Native Device** é um sistema de automação residencial robusto e versátil que permite a comunicação padronizada entre os dispositivos, mesmo com adaptações conforme as necessidades individuais de cada equipamento.

Vantagens do Native Device

Vantagem	Descrição
Protocolo Aberto	Facilita a integração e o acesso aos dispositivos.
Unicidade	Protege contra duplicação, garantindo que cada mensagem seja única.
Start and Stop	Caracteres delimitadores asseguram uma comunicação clara e sem ruídos.
Redundância	Permite a transmissão simultânea em diversos locais para evitar perda de dados.
Adaptabilidade	O dispositivo define suas necessidades, facilitando a inclusão de novos equipamentos.
Legibilidade	Utiliza o sistema ASCII, tornando o protocolo de fácil entendimento.

Essas características garantem que o Native Device ofereça uma base de comunicação confiável e eficiente em ambientes de automação residencial.

Native Device: Comunicação UDP e TCP

Todos os dispositivos **Native Device** aceitam múltiplas formas de comunicação simultâneas: **UDP Multicast**, **TCP Unicast** e, quando conectados fisicamente, também via **XNET** (rede cabeada).

Isso garante alta confiabilidade na entrega das mensagens e rapidez na comunicação, sem exigir nenhuma intervenção do usuário ou integrador.

O sistema usa todos os protocolos disponíveis simultaneamente, garantindo sempre a melhor performance e eficiência.

Protocolo UDP Multicast

O protocolo **UDP Multicast** permite enviar uma mensagem para **todos os dispositivos** conectados à mesma rede simultaneamente, sem precisar conhecer antecipadamente os endereços IP individuais.

- **IP Multicast:** 239.5.0.2
- **Porta:** 5202

Essa abordagem é essencial para comunicação rápida e simultânea em redes com muitos dispositivos conectados, facilitando comandos em massa ou descoberta de equipamentos.

Como Funciona:

- Uma única mensagem é enviada para o endereço multicast.
 - Todos os dispositivos na rede recebem a mensagem simultaneamente.
 - O dispositivo irá responder pelo multicast via UDP, ao IP solicitante via TCP, e pela XNET se estiver cabeado.
-

Protocolo TCP Unicast

O protocolo **TCP Unicast** permite comunicação direta entre o sistema e cada equipamento de forma individual e confiável, garantindo a entrega das mensagens.

- **Porta TCP utilizada pelos equipamentos:** 5203

Como Funciona:

- O sistema estabelece uma conexão direta com cada equipamento individualmente.
 - Ao enviar uma mensagem diretamente (unicast), a conexão TCP garante que a mensagem chegue ao destino de maneira confiável.
 - O dispositivo responde pela mesma conexão TCP, também via UDP Multicast, e via XNET (se disponível).
-

Comunicação Simultânea e Automática

O sistema **Native Device** utiliza **simultaneamente** os protocolos **UDP**, **TCP** e, quando disponível, **XNET (cabeado)**, garantindo máxima confiabilidade e performance.

Todo o gerenciamento de envio e resposta é feito automaticamente — o usuário **não precisa escolher ou configurar nada**.

Como funciona:

- Ao enviar comandos ou consultas, o sistema transmite a mesma mensagem via **UDP Multicast** e **TCP Unicast**.
- O equipamento **sempre responde por todos os canais disponíveis**, conforme as seguintes regras:

Se a mensagem contiver o campo `_to`:

- O equipamento abrirá uma nova conexão TCP com o IP definido em `_to`, caso não exista uma conexão TCP aberta com esse IP.
- A resposta será enviada **exclusivamente** para o IP do campo `_to`, via TCP.
- A resposta também será enviada via **UDP Multicast** e via **XNET** se estiver cabeado.

Se não houver campo `_to`:

- O equipamento responderá via **UDP Multicast**.
- Se houver **conexões TCP já abertas**, responderá em **todas elas**.
- Também responderá via **XNET**, se aplicável.

Exemplo prático de funcionamento

- 1 O sistema envia uma solicitação automaticamente por **UDP Multicast** e **TCP Unicast**.
- 2 O dispositivo recebe a mensagem e verifica a presença do campo `_to`.
- 3 Se `_to=192.168.0.10` estiver presente:
 - Ele abre uma nova conexão TCP com `192.168.0.10` e envia a resposta por ela.
- 4 Simultaneamente, a resposta será enviada por **UDP** e **XNET** (se disponível).

Dica

Esse processo é totalmente transparente, automático e não requer nenhuma ação por parte do usuário.

Native Device: Query Strings

Nesta seção, apresentamos exemplos de mensagens UDP utilizando Query Strings para o protocolo Native Device.

Query String – Versão 1

```
$6A6F>/ac/rs@0?r0=off&r1=off&_to=192.168.0.10<99999#
```

Parte	Descrição
<code>\$6A6F</code>	Identifica o dispositivo que receberá a ação, utilizando os 4 últimos dígitos do endereço MAC. Para o dispositivo mestre, utilize o valor 0000.
<code>>/ac/rs</code>	Especifica a ação a ser executada no dispositivo. Veja lista de ações disponíveis.
<code>@0</code>	Refere-se ao Protocolo da mensagem, podendo ser 0, 1, 8 ou 9. Veja tabela abaixo
<code>?r0=off&r1=off</code>	São os valores da mensagem, que podem representar informações ou alterações, conforme o tipo de protocolo. Este campo não é obrigatório e, caso nenhum valor seja informado, o dispositivo retornará o estado de todas as saídas relacionadas à ação especificada.
<code>&_to=192.168.0.10</code>	IP de origem da solicitação (opcional): define o endereço IP que será utilizado pelo equipamento ao enviar a resposta. Caso informado, o dispositivo responderá exclusivamente para esse endereço.
<code><99999</code>	Número único e aleatório utilizado para identificar a solicitação e evitar interpretações duplicadas do mesmo protocolo. Caso não seja necessário, utilize o valor 0 para ignorar (não recomendado).
<code>#</code>	Indica o fim do protocolo.

Importante: os caracteres `$`, `>`, `@`, `?`, `<` e `#` são **obrigatórios** em todas as mensagens. Eles funcionam como **separadores** dos campos e devem ser incluídos mesmo que nenhum valor seja fornecido após eles. Os parâmetros dentro da mensagem são separados por `&`, e os que iniciam com `_` representam protocolos internos. Por exemplo, a ausência de parâmetros após `?` não dispensa o uso do próprio caractere, pois ele define o início do bloco de dados da mensagem. O mesmo vale para os demais separadores, garantindo que a estrutura seja corretamente interpretada pelo dispositivo.

Tabela de Protocolos

Protocolo	Nome	Descrição
@0	Set Data	Protocolo utilizado para definir ou solicitar valores no dispositivo, como consultar estados ou aplicar modificações.
@1	Delete data	Protocolo utilizado para remover dados do dispositivo (ex: cenas)
@8	Send All	Protocolo utilizado para enviar a mesma mensagem para múltiplos dispositivos simultaneamente.
@9	Description	Protocolo que solicita ao dispositivo o envio das descrições de suas ações e parâmetros disponíveis.

Query String – Versão 2

```
$6A6F>/ac/rs@0?r[0,1]=off&_to=192.168.0.1<99999#
```

A Query String da Versão 2, compatível com dispositivos a partir da série 2.x.x, apresenta uma otimização significativa na transmissão dos dados. Nesta versão, valores idênticos em arrays não são repetidos; por exemplo, os parâmetros que antes eram enviados como `r0=off` e `r1=off` são condensados para `r[0,1]=off`. Essa condensação reduz o tamanho do pacote, diminui o tráfego de rede e melhora a eficiência da transmissão. Devido às diferenças estruturais, a Versão 2 não é compatível com dispositivos que operam na Versão 1.x.x, sendo necessária atualização para aproveitar as melhorias.

Ações

Descrição das Ações

Todos os dispositivos **Native Device** geram uma lista de ações para que o programa de configurações possa identificar o que está disponível em cada equipamento. Essas ações são acessadas por meio do protocolo **UDP**.

Cada uma delas tem um propósito diferente, explicado na seção **Lista de Ações disponíveis** abaixo.

Lista de Ações disponíveis:

As ações possuem versões abreviadas para melhorar o desempenho da comunicação. Isso reduz a quantidade de dados trafegados, otimizando a velocidade e eficiência do sistema.

Caminho	Nome	Descrição
<code>/ac/rs</code>	Rele State	Utilizado para controlar e receber estados de saídas e relés.
<code>/ac/pers</code>	Persiana State	Utilizado para controlar e receber estados das persianas.
<code>/ac/zonaN</code>	LOUD	Utilizado para consultar e controlar zonas de amplificadores LOUD.
<code>control</code>	Control CS	Utilizado para controlar via interface XNET (rede cabeada).
<code>/ac/serial</code>	Serial	Envia evento serial para os botões.
<code>/ac/xc</code>	Cena do Control	Controla intensidades específicas por cena (0 a 240).
<code>/ac/xClick</code>	xClick	Define que um botão deve executar um clique remoto.
<code>/ac/xHold</code>	xHold	Define que um botão deve executar um comando de segurar remoto.
<code>/aa/tcp</code>	Ação TCP	Envia mensagem via protocolo TCP para IP e porta configurados.
<code>/aa/udp</code>	Ação UDP	Envia mensagem via protocolo UDP para IP e porta configurados.
<code>/aa/wol</code>	Wake-on-LAN (WOL)	Envia pacote WOL com MacAddress.
<code>if</code>	Info do Dispositivo (if)	Retorna informações do dispositivo pela rede.

Descrição da ação

As ações `ac/rs` são usadas para controlar as saídas de relés dos equipamentos. Abaixo, apresentamos um exemplo de mensagem via protocolo UDP para a ação:

Path: `/ac/rs` — Controle de Estados dos Relés

Esta ação permite consultar os estados das saídas de relé do equipamento.

Se a mensagem for enviada sem parâmetros de consulta (query) após o caractere `?`, o dispositivo retornará o estado de todas as saídas relacionadas à ação informada. Isso se aplica tanto ao path `/ac/rs` quanto a outras ações do tipo `/ac`, como `/ac/bt`, `/ac/dim`, etc.

Mensagem de solicitação dos estados:

```
$6A6F>/ac/rs@0?_to=192.168.0.10<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/ac/rs@0?r0=off&r1=off&_to=192.168.0.10<99999#
```

Parâmetros de resposta

- `r0` — Relé 0 (representa a saída 1)
- `r1` — Relé 1 (representa a saída 2)
- `r2` — Relé 2 (representa a saída 3)

Dica

Outros relés seguem a mesma lógica incremental: `r3`, `r4`, etc., conforme o número de saídas do dispositivo.

Descrição da ação

As ações `ac/pers` são utilizadas para controlar o estado dos manipuladores de persianas (On, Up e Down). Abaixo, apresentamos um exemplo de mensagem via protocolo UDP para a ação:

Path: `/ac/pers` — Consulta de Estados das Persianas

Esta ação permite consultar o estado atual dos manipuladores das persianas. A resposta retornará os estados dos comandos **On**, **Up** e **Down** para cada persiana disponível no equipamento.

Se a mensagem for enviada sem parâmetros de consulta (query) após o caractere `?`, o dispositivo retornará o estado de todas as saídas relacionadas à ação informada. Isso se aplica tanto ao path `/ac/pers` quanto a outras ações do tipo `/ac`, como `/ac/bt`, `/ac/rs`, etc.

Mensagem de solicitação dos estados:

```
$6A6F>/ac/pers@0?_to=192.168.0.10<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/ac/pers@0?on0=off&on1=on&...&up0=off&up1=on&...&down0=on&down1=off&...&_to=192.168.0.10<99999#
```

Parâmetros de resposta

Os parâmetros seguem o seguinte padrão:

- `on0`, `up0`, `down0` — manipuladores da persiana 1
- `on1`, `up1`, `down1` — manipuladores da persiana 2
- `on2`, `up2`, `down2` — manipuladores da persiana 3
- ...

Dica

A numeração pode variar de acordo com a quantidade de persianas suportadas pelo equipamento. Cada persiana possui três estados: `on`, `up`, `down`.

LOUD — Controle de Zonas via UDP

Referência técnica para integradores. Mostra como descobrir e controlar as zonas de um amplificador LOUD na rede local usando o protocolo ASCII dos dispositivos Xcene/LOUD.

ⓘ Zonas e entradas variam por aparelho

O protocolo é o mesmo para toda a linha LOUD, mas o número de zonas e de entradas depende do aparelho. Os exemplos abaixo mostram até 6 zonas (zona0 a zona5) e as entradas AUX1 a AUX5 , BT e OPT . O APL 420S, por exemplo, tem 2 zonas. Para saber o que um aparelho tem, use a descoberta (@9 , seção 3): ele responde com as zonas e os parâmetros que expõe.

1. Formato do pacote ASCII

```
$<ID>><PATH>@<PROTOCOLO>?<QUERY>&_to=<IP><SEED>#
```

Exemplo:

```
$3A7F>/ac/zona0@0?vl=50&_to=192.168.0.10<123456#
```

Campo	Descrição
\$3A7F	4 últimos dígitos do MAC do device destino. 0000 = broadcast.
>/ac/zona0	Path da ação.
@0	Número do protocolo (tabela abaixo).
?vl=50	Parâmetros chave=valor separados por & . Vazio = leitura do estado atual.
&_to=192.168.0.10	(Opcional) IP para resposta exclusiva. Device abre TCP de volta neste IP.
<123456	Seed — número único por envio, evita processamento duplicado. Use 0 apenas para testes.
#	Fim do pacote.

Os separadores \$ > @ ? < # são obrigatórios, mesmo que o campo esteja vazio.

Tabela de protocolos

Código	Nome	Uso
0	Set Data	Ler ou definir valores no device
1	Delete Data	Remover dados (ex.: itens de cena)
8	Send All	Broadcast — mesma mensagem para vários devices
9	Description	Descobrir ações e parâmetros disponíveis

2. Device LOUD

Propriedade	Valor
ID	3A7F
Modelo	Multiroom
Zonas de controle	/ac/zona0, /ac/zona1, /ac/zona2, /ac/zona3, /ac/zona4, /ac/zona5
Config avançada de zonas	/lb/Zona (indexada por _n=0..5)

Outras ações (Bluetooth, Inputs, IR, WiFi, modo Dualzone/Singlezone/4.1) também estão disponíveis via path `/lb/*`, mas não são cobertas aqui.

3. Descoberta — Protocolo 9 (DESCRIPTION)

Pergunta ao device quais zonas e parâmetros ele expõe. Suporta wildcard ou path específico.

Listar todas as ações do device:

```
$3A7F>/*@9?<12345#
```

Listar apenas a zona 0:

```
$3A7F>/ac/zona0@9?<12345#
```

Modelo de resposta (zona 0):

```
$3A7F>/ac/zona0@9?{"name":"zona0","path":"/ac/zona0","params": [
  {"name":"pw","friendlyName":"power","params":["toggle","on","off"],"type":"Boolean"},
  {"name":"mt","friendlyName":"mute","params":["toggle","on","off"],"type":"Boolean"},
  {"name":"in","friendlyName":"input","params":["AUX1","AUX2","AUX3","AUX4","AUX5","BT","OPT"],"type":
  {"name":"vl","friendlyName":"volume","params":["$inc=[1,20]","$dec=[1,20]","$max=99","$min=0"],"ty
  {"name":"bs","friendlyName":"bass","params":["$inc=[1,20]","$dec=[1,20]","$max=0","$min=-15"],"typ
  {"name":"tr","friendlyName":"treble","params":["$inc=[1,20]","$dec=[1,20]","$max=0","$min=-15"],"t
  {"name":"md","friendlyName":"middle","params":["$inc=[1,20]","$dec=[1,20]","$max=0","$min=-15"],"t
]}<12345#
```

4. Parâmetros de uma zona (/ac/zonaN)

Param	Nome amigável	Tipo	Valores / Faixa	Exemplo
<code>pw</code>	power	Boolean	<code>on</code> , <code>off</code> , <code>toggle</code>	<code>pw=on</code>
<code>mt</code>	mute	Boolean	<code>on</code> , <code>off</code> , <code>toggle</code>	<code>mt=off</code>
<code>in</code>	input	Chosen	<code>AUX1</code> , <code>AUX2</code> , <code>AUX3</code> , <code>AUX4</code> , <code>AUX5</code> , <code>BT</code> , <code>OPT</code>	<code>in=BT</code>
<code>vl</code>	volume	Integer	<code>0</code> a <code>99</code>	<code>vl=50</code>
<code>bs</code>	bass	Integer	<code>-15</code> a <code>0</code> (dB)	<code>bs=-5</code>
<code>tr</code>	treble	Integer	<code>-15</code> a <code>0</code> (dB)	<code>tr=-3</code>
<code>md</code>	middle	Integer	<code>-15</code> a <code>0</code> (dB)	<code>md=-2</code>

5. Comandos práticos

Todos os exemplos usam a zona 0 (/ac/zona0) do device `3A7F`. Para outras zonas, troque o path (/ac/zona1, /ac/zona2, ...). A seed (<123#) deve ser trocada a cada envio.

5.1 Consultar estado completo

Envio:

```
$3A7F>/ac/zona0@0?<111#
```

Resposta:

```
$3A7F>/ac/zona0@0?code=1&pw=on&mt=off&in=AUX1&vl=50&bs=0&tr=0&md=0<111#
```

`code=1` = sucesso.

5.2 Ligar / desligar / toggle

```
$3A7F>/ac/zona0@0?pw=on<112#
$3A7F>/ac/zona0@0?pw=off<113#
$3A7F>/ac/zona0@0?pw=toggle<114#
```

Resposta (eco de confirmação):

```
$3A7F>/ac/zona0@0?code=1&pw=on<112#
```

5.3 Mutar / desmutar

```
$3A7F>/ac/zona0@0?mt=on<115#
$3A7F>/ac/zona0@0?mt=off<116#
```

Resposta:

```
$3A7F>/ac/zona0@0?code=1&mt=on<115#
```

5.4 Trocar fonte de áudio

```
$3A7F>/ac/zona0@0?in=AUX1<117#
```

```
$3A7F>/ac/zona0@0?in=BT<118#
```

```
$3A7F>/ac/zona0@0?in=OPT<119#
```

Resposta:

```
$3A7F>/ac/zona0@0?code=1&in=BT<118#
```

5.5 Definir volume

```
$3A7F>/ac/zona0@0?vl=75<120#
```

Resposta:

```
$3A7F>/ac/zona0@0?code=1&vl=75<120#
```

5.6 Ajustar equalização (graves, médios, agudos)

```
$3A7F>/ac/zona0@0?bs=-5<121#
```

```
$3A7F>/ac/zona0@0?md=-3<122#
```

```
$3A7F>/ac/zona0@0?tr=-2<123#
```

Resposta:

```
$3A7F>/ac/zona0@0?code=1&bs=-5<121#
```

6.7 Comando combinado (múltiplos parâmetros)

```
$3A7F>/ac/zona0@0?pw=on&in=BT&vl=60&mt=off<124#
```

Resposta:

```
$3A7F>/ac/zona0@0?code=1&pw=on&in=BT&vl=60&mt=off<124#
```

6. Configuração avançada da zona (/lb/Zona)

/lb/Zona guarda a configuração persistente de cada zona (nome, limites, defaults). Como o device tem 6 zonas, as operações usam o parâmetro `_n` (0 a 5) para indicar qual zona está sendo lida/gravada.

Parâmetros

Param	Tipo	Faixa / Valores	Descrição
<code>nmc</code>	String	até 15 caracteres	Nome amigável da zona
<code>vli</code>	Integer	0 a 30	Volume inicial (ao ligar)
<code>vll</code>	Integer	40 a 99	Volume máximo permitido
<code>ii</code>	Chosen	AUX1..AUX5, BT, OPT	Fonte padrão ao ligar
<code>mdda</code>	Chosen	Normal, Power Automático	Modo de ligar automático
<code>md</code>	Integer	-15 a 0	Médio default
<code>tb</code>	Integer	-15 a 0	Agudo default
<code>bs</code>	Integer	-15 a 0	Grave default

6.1 Ler a configuração da zona 0

Envio:

```
$3A7F>/lb/Zona@0?_n=0<130#
```

Resposta:

```
$3A7F>/lb/Zona@0?code=1&nmc=Sala&vli=15&vll=99&ii=AUX1&mdda=Normal&md=-5&tb=-3&bs=-2&_n=0<130#
```

6.2 Renomear a zona 0 para “Sala”

```
$3A7F>/lb/Zona@0?nmc=Sala&_n=0<131#
```

O valor em `nmc` pode ser URI-encoded se tiver espaços/acentos (ex.: `nmc=Sala%20de%20Estar`).

6.3 Definir volume inicial e máximo da zona 1

```
$3A7F>/lb/Zona@0?vli=20&vll=80&_n=1<132#
```

7. Notas rápidas

- **Seed única por envio:** o device descarta pacotes com seed repetida recente (deduplicação anti-eco). Gere um número novo a cada pacote.
- **`_to=<IP>`** : opcional mas recomendado quando você quer resposta unicast direta para a sua máquina (ex.: atrás de firewall multicast restritivo). O device também pode abrir TCP de volta neste IP na porta `5203`.
- **Separadores:** `$ > @ ? < #` são obrigatórios mesmo se o campo estiver vazio. Ex.: leitura sem parâmetros é `$3A7F>/ac/zona0@0?<SEED#`.
- **Envios sequenciais:** equipamentos são low-energy; evite enviar múltiplos pacotes UDP em paralelo. Use intervalo de ~50 ms entre envios consecutivos.
- **code** na resposta: `code=1` (ou que começa com `1`) indica sucesso. Respostas sem `code` em protocolo `@9` (DESCRIPTION) são normais.
- **Broadcast `0000`** : substitua `$3A7F` por `$0000` para enviar a todos os devices da rede. Útil apenas para descoberta (`/if/device@0`), não para controle.

Descrição da ação

As ações `ac/xc` são usadas para controlar a intensidade das saídas dos equipamentos. Abaixo, apresentamos um exemplo de mensagem via protocolo UDP para a ação:

Path: `/ac/xc` — Cena do Control

Esta ação permite consultar os estados das intensidades das cargas/saídas do equipamento.

Se a mensagem for enviada sem parâmetros de consulta (query) após o caractere `?`, o dispositivo retornará o estado de todas as saídas relacionadas à ação informada. Isso se aplica tanto ao path `/ac/xc` quanto a outras ações do tipo `/ac`, como `/ac/bt`, `/ac/dim`, etc.

Mensagem de solicitação dos estados:

```
$6A6F>/ac/xc0@0?_to=192.168.0.10<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/ac/xc0@0?l0=0&l1=80&l2=0...&_to=192.168.0.10<99999#
```

Parâmetros de resposta

- `l0` — Carga/Saída 0 (representa a saída 1)
- `l1` — Carga/Saída 1 (representa a saída 2)
- `l2` — Carga/Saída 2 (representa a saída 3)

Dica

Outras cargas/saídas seguem a mesma lógica incremental: `l3`, `l4`, etc., conforme o número de saídas do dispositivo.

Atenção

Equipamentos com **versão 2.4.17 ou superior** usam valores de **0 a 100**, onde `0` representa **apagado** e `100` representa **intensidade máxima (100%)**. Equipamentos com **versões anteriores** usam valores de **0 a 240**, onde `0` também representa **apagado** e `240` representa **100%**. Para calcular a porcentagem real nas versões antigas, use a fórmula: $(\text{valor} / 240) * 100$. Exemplo: $(185 / 240) * 100 = 77\%$

Descrição da ação

As ações `ac/xClick` são usadas para executar um clique em um botão de determinado ID. Abaixo, apresentamos um exemplo de mensagem via protocolo UDP para a ação:

Path: `/ac/xClick` — Cena do Control

Esta ação permite consultar o Id e Botão configurados.

Se a mensagem for enviada sem parâmetros de consulta (query) após o caractere `?`, o dispositivo retornará o estado de todas as saídas relacionadas à ação informada. Isso se aplica tanto ao path `/ac/xClick` quanto a outras ações do tipo `/ac`, como `/ac/bt`, `/ac/dim`, etc.

Mensagem de solicitação dos estados:

```
$6A6F>/ac/xClick@0?_to=192.168.0.10<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/ac/xClick@0?id=3&bt=5&_to=192.168.0.10<99999#
```

Parâmetros de resposta

- `id` — Id respectivo do Control
 - `bt` — Botão do respectivo Id
-

Descrição da ação

As ações `ac/xHold` são usadas para executar um pressionar em um botão de determinado ID. Abaixo, apresentamos um exemplo de mensagem via protocolo UDP para a ação:

Path: `/ac/xHold` — Cena do Control

Esta ação permite consultar o Id e Botão configurados.

Se a mensagem for enviada sem parâmetros de consulta (query) após o caractere `?`, o dispositivo retornará o estado de todas as saídas relacionadas à ação informada. Isso se aplica tanto ao path `/ac/xHold` quanto a outras ações do tipo `/ac`, como `/ac/bt`, `/ac/dim`, etc.

Mensagem de solicitação dos estados:

```
$6A6F>/ac/xHold@0?_to=192.168.0.10<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/ac/xHold@0?id=3&bt=5&_to=192.168.0.10<99999#
```

Parâmetros de resposta

- `id` — Id respectivo do Control
 - `bt` — Botão do respectivo Id
-

Descrição da ação TCP

A ação TCP permite enviar mensagens para um IP e porta definidos, usando o protocolo TCP.

Path: /aa/tcp

Essa ação é usada para acionar dispositivos externos via rede TCP, podendo ser personalizada com IP, porta e payload (mensagem).

Mensagem de solicitação dos estados:

```
$6A6F>/aa/tcp@0?_to=192.168.0.10<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/aa/tcp@0?ip=192.168.0.50&payload=ligar&port=5000&_to=192.168.0.10<99999#
```

Parâmetros de resposta

- **IP** — IP de destino
 - **Porta** — Porta usada
 - **Payload** — Conteúdo da mensagem
-

Dica

Essa ação requer que o equipamento esteja com Ações Avançadas habilitadas.

Descrição da ação UDP

A ação **UDP** permite enviar mensagens para um IP e porta definidos, usando o protocolo UDP.

Path: `/aa/udp`

Essa ação é usada para acionar dispositivos externos via rede UDP, podendo ser personalizada com IP, porta e payload (mensagem).

Mensagem de solicitação dos estados:

```
$6A6F>/aa/udp@0?_to=192.168.0.10<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/aa/udp@0?ip=192.168.0.50&payload=ligar&port=5000&_to=192.168.0.10<99999#
```

Parâmetros de resposta

- **IP** — IP de destino
 - **Porta** — Porta usada
 - **Payload** — Conteúdo da mensagem
-

Dica

Essa ação requer que o equipamento esteja com Ações Avançadas habilitadas.

Descrição da ação WOL

A ação WOL (Wake on Lan) permite ligar um dispositivo através da rede.

Path: `/aa/wol`

Essa ação é usada para ligar o dispositivos externos via rede, podendo ser personalizada com MacAddress do dispositivo.

Mensagem de solicitação dos estados:

```
$6A6F>/aa/wol@0?_to=192.168.0.10<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/aa/wol@0?mac=&_to=192.168.0.10<99999#
```

Parâmetros de resposta

- `Mac` — MacAddress do destino

Dica

Essa ação requer que o equipamento esteja com Ações Avançadas habilitadas.

Descrição da ação

A interface **XNET** do Control CS permite a comunicação com o sistema de automação **XCENE** por meio de mensagens estruturadas. Essa comunicação é exclusiva para redes **cabeadas via XNET**.

⚠️ Atenção

Nunca interligue diretamente a rede XNET com uma rede serial RS232 sem a interface apropriada.

Configurações de Comunicação

As configurações da interface RS232 são fixas e devem ser respeitadas para garantir o correto funcionamento:

Configuração	Valor
Velocidade	19.200 baud
Paridade	Nenhuma
Bits de dados	8
Stop bit	1

Disparo de Cena

Esta mensagem envia ao equipamento um comando para ativar uma **cena** com tempos de fade e intensidades específicas para cada carga.

Estrutura da mensagem

ID | Função | Qbyte(-1) | Tempo HI | Tempo LO | Carga 1 ... Carga 8 | CRC

Exemplo — Disparar cena para equipamento no endereço 0x01 :

0x01	0x03	0x0D	0x00	0x05	0x50	0x50	0x50	0x50	0x50	0x50	0x50	0x50	0x50	0x0D
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

Parâmetros

- **ID:** Endereço do equipamento Control CS (0x00 a 0x0F)
- **Função:** 0x03 — disparar cena
- **Qbyte(-1):** quantidade de bytes a serem transmitidos menos 1
- **Tempo HI / LO:** tempo de fade da cena, em milissegundos (HI = parte alta, LO = parte baixa)
- **Carga 1 a 8:**
 - 0x00 a 0xF0 (0 a 240): intensidade
 - 0xFF (255): a carga não participa da cena
- **CRC:** valor fixo 0x0D (CHECK desativado)

Atualização de Status das Cargas

Permite que o sistema informe o status atual das cargas **após o envio de uma cena**. O equipamento **não envia** essa informação automaticamente, portanto, o software deve atualizá-la por conta própria.

Estrutura da mensagem

ID | Função | Qbyte(-1) | ID Control | Carga 1 ... Carga 8 | CRC

Exemplo — Status de cargas do equipamento 0x01 via broadcast:

0xFF | 0x02 | 0x0A | 0x01 | 0x50 | 0x50 | 0x50 | 0x50 | 0x50 | 0x50 | 0x50 | 0x50 | 0x0D

Dica

0xFF é usado como endereço **broadcast** para atualizar todos os dispositivos da rede com o status informado.

Comando de Botão

Simula a ativação física de um botão de cena no equipamento Control CS.

Estrutura da mensagem

ID | Função | Qbyte(-1) | Botão/Comando | CRC

Para calcular o valor de **Botão/Comando**, use:

- Botões de 1 a 12 → base hexadecimal **0xC0 + número do botão**
- Acrescente:
 - +1 para **Release**
 - +2 para **Press**
 - +3 para **Hold**

Atenção

Os valores resultantes devem ser enviados como **um único byte hexadecimal**.

Exemplos

Pressionar botão 12 (**Press**)

0x01 | 0x01 | 0x04 | 0xE3 | 0x0D

Soltar botão 12 (**ReLease**)

0x01 | 0x01 | 0x04 | 0xE2 | 0x0D

Segurar botão 4 (**HoLd**)

- Botão 4 = $0xC0 + 4 + 3 = 0xCC$

Comando completo:

0x01 | 0x01 | 0x04 | 0xCC | 0x0D

Tabela de referência rápida

Botão	Release	Press	Hold
1	0xC1	0xC2	0xC3
2	0xC4	0xC5	0xC6
3	0xC7	0xC8	0xC9
4	0xCA	0xCB	0xCC
5	0xCD	0xCE	0xCF
6	0xD0	0xD1	0xD2
7	0xD3	0xD4	0xD5
8	0xD6	0xD7	0xD8
9	0xD9	0xDA	0xDB
10	0xDC	0xDD	0xDE
11	0xDF	0xE0	0xE1
12	0xE2	0xE3	0xE4

Dica

Use o valor da coluna correspondente no campo **Botão/Comando** da mensagem:

0x01 | 0x01 | 0x04 | <valor acima> | 0x0D

Status do LED de Cena

Essa ação permite consultar o estado atual dos LEDs de cena no equipamento.

O comando retorna quais LEDs estão acesos, ou seja, quais cenas estão visivelmente ativas no painel frontal.

Atenção

Trata-se de uma leitura — não é possível acender ou apagar LEDs com esse comando.

Estrutura da mensagem

ID | Função | Qbyte(-1) | Valor do LED | CRC

Essa estrutura solicita ao equipamento o estado visual atual (ligado/desligado) dos LEDs de cena.

Exemplo — Verificar LED da cena 5 no equipamento 0x01 :

0x01 | 0x0A | 0x04 | 0x10 | 0x0D

Explicação dos campos

- **ID** : Endereço do equipamento que receberá o comando (ex: `0x01`)
- **Função** : Código da função — sempre `0x0A` para consulta de LEDs de cena
- **Qbyte(-1)** : Quantidade de bytes a serem transmitidos, menos 1 (ex: `0x04`)
- **Valor do LED** : Byte com bits ligados representando os LEDs a serem consultados
- **CRC** : Código fixo de fim da mensagem — sempre `0x0D`

Tabela de referência de LEDs de cena

Cena	Valor em Hex	Descrição
1	<code>0x01</code>	Consultar status do LED da cena 1
2	<code>0x02</code>	Consultar status do LED da cena 2
3	<code>0x04</code>	Consultar status do LED da cena 3
4	<code>0x08</code>	Consultar status do LED da cena 4
5	<code>0x10</code>	Consultar status do LED da cena 5
6	<code>0x20</code>	Consultar status do LED da cena 6
1 + 2	<code>0x03</code>	LEDs das cenas 1 e 2
1–5	<code>0x1F</code>	LEDs das cenas 1 a 5
1–6 (todos)	<code>0x3F</code>	Todos os LEDs de cena

Como montar a mensagem

- 1 Escolha os LEDs que deseja consultar.
- 2 Some os valores hexadecimais correspondentes.
- 3 Monte a mensagem com base na estrutura.

Exemplo:

Para consultar os LEDs das cenas 1, 3, 5 e 6:

`0x01 + 0x04 + 0x10 + 0x20 = 0x35`

Mensagem final: `0x01 | 0x0A | 0x04 | 0x35 | 0x0D`

Resumo

- A comunicação ocorre via rede cabeada **XNET**
- A mensagem **consulta o status visual** dos LEDs de cena
- **Não altera o estado dos LEDs**
- O comando é binário, enviado em hexadecimal
- O CRC de término é sempre `0x0D`

Descrição da ação IF

Os paths que iniciam com o prefixo `/if` são responsáveis por **solicitar informações ao equipamento**.

A resposta trará, em sua query string, os seguintes dados:

- `model` : modelo do equipamento
- `deviceName` : nome definido para o dispositivo
- `version` : versão do firmware
- `serialNumber` : número de série do equipamento
- `type` : tipo de equipamento
- `id` : identificador único
- `date` : data de fabricação
- `nAC` : número de ações disponíveis no dispositivo

A seguir, apresentamos um exemplo de uso da ação via protocolo UDP:

Path: `/if/device` — Obtem informações sobre o dispositivo.

Esta ação permite consultar as informações detalhadas de um equipamento específico.

Caso a mensagem seja enviada com `0000` no lugar do identificador (por exemplo, em vez de `6A6F`), o comando será interpretado como uma solicitação para **todos os dispositivos conectados na rede**, que responderão individualmente.

Mensagem de solicitação das informações:

```
$6A6F>/if/device@0?<99999#
```

Mensagem de resposta do equipamento:

```
$6A6F>/if/device@0?model=MRQ10&deviceName=Xcene_MRQ10.....<99999#
```