

4.15 Protocolo

Esta seção apresenta o protocolo do OpenDevice. As principais características do protocolo é que ele foi projetado para ser um protocolo aberto, leve, simples, de fácil

implementação, legível para humanos e voltado para dispositivos com restrições de memória e processamento, como por exemplo, microcontroladores (AVR 8-bits, 2Kb de RAM).

O protocolo é baseado em ASCII (assim como o HTTP), orientado a mensagens/comandos e assíncrono. Surgiu de algumas influências do protocolo MIDI (formato usado para comunicação com instrumentos musicais) e do protocolo REST na sua estrutura básica.

O protocolo foi projetado para permitir a abstração, controle de dispositivos (atuadores), realizar a leitura de sensores e ser de fácil extensão, mas não limitado a isto. Ele pode ser utilizado em conjunto com outros protocolos, como WebSocket e MQTT, e com outras tecnologias de comunicação, como: USB, Bluetooth, Ethernet, Wi-Fi, etc.

4.15.1 Formato da Mensagem

Os comandos do OpenDevice possuem um “cabeçalho” fixo, contendo o tipo do comando (*CommandType*) e o ID do comando, em seguida, o bloco “*Command Extension*”, que varia de acordo com o tipo de comando.

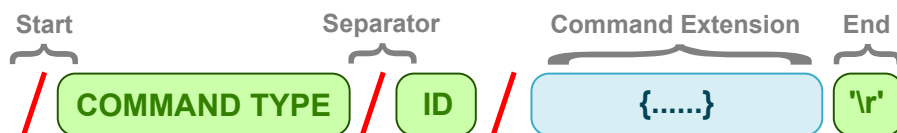


Figura 4.19 Formato do protocolo OpenDevice

- O **tipo do comando** define a estrutura do bloco “*Command Extension*”. Os tipos suportados estão definidos na tabela 4.5.
- O **ID** do comando é numérico, sequencial e gerenciado pela aplicação cliente. Ele serve para a aplicação identificar o retorno(resposta) de algum comando enviado. Quando atinge seu valor máximo a contagem reinicia.
- Os blocos do comando são separados através do caractere “/”.
- O comando finaliza com o terminador “\r” (Carriage return).

OBSERVAÇÃO: Na implementação atual do firmware, tanto o “tipo do comando” quanto o “id”, são armazenados em variáveis de 8bits (uint8_t / byte), limitando a valores na faixa

de 0 a 255. Porém, os mesmos são parametrizados podendo escolher outros tipos como: `uint16_t` ou `uint32_t`.

Convenções e pontos em aberto

Algumas convenções foram adotadas para determinar observações, e destacar pontos em aberto. Caso o texto apresente “[*Numero]”, significa que deve ser verificado as observações abaixo:

- Observação [*1]: Na implementação atual do firmware, o “DeviceID” é armazenado em uma variável de 8bits (byte), porém, o tipo é parametrizável.
- Observação [*2]: Significa que o parâmetro ou tipo é configurável.
- Observação [*3]: Significa que a versão atual do firmware não implementa o recurso.

4.15.2 Tipos de Comandos

A tabela [4.5](#), apresenta os tipos de comandos suportados.

Código	Tipo de Comando	Ref.	Formato
1	DIGITAL	4.15.3	[#/#]/{{DeviceID}}/{{value}}
2	ANALOG	4.15.4	[#/#]/{{DeviceID}}/{{value}}
3	NUMERIC	4.15.5	[#/#]/{{DeviceID}}/{{value}}
4...9	[Reservado]		–
10	COMMAND_RESPONSE	4.15.6	[#/#]/{{status}}/{{DeviceID?}}
11	SET_PROPERTY	4.15.7	[#/#]/{{DeviceID}}/{{property}}/{{value}}
12	GET_PROPERTIES	4.15.8	[#/#]/{{DeviceID}}
13	GET_PROPERTIES_RESPONSE	4.15.9	[#/#]/{{DeviceID}}/[[{property1:value},...,N]
14	ACTION	4.15.10	[#/#]/{{DeviceID}}/{{action}}/[[{value1?},...,{valueN}]]
15	GET_ACTIONS	4.15.11	[#/#]/{{DeviceID}}
16	GET_ACTIONS_RESPONSE	4.15.12	[#/#]/{{DeviceID}}/[[{action1},{actionN}]]
16...19	[Reservado]		–
20	PING_REQUEST	4.15.13	[#/#][somente cabeçalho]
21	PING_RESPONSE	4.15.14	[#/#][somente cabeçalho]
22	DISCOVERY_REQUEST	4.15.15	[#/#][somente cabeçalho]
23	DISCOVERY_RESPONSE	4.15.16	[#/#]/{{name}}/{{port}}/{{deviceLength}}
24...29	[Reservado]		–
30	GET_DEVICES	4.15.17	[#/#]/{{DeviceID?}}/{{value?}}
31	GET_DEVICES_RESPONSE	4.15.18	[#/#]/[{{DeviceInfo1}},{{DeviceInfoN?}}
32	DEVICE_ADD	4.15.19	[#/#]/{{DeviceInfo}}
33	DEVICE_DEL	4.15.20	[#/#]/{{DeviceID?}}
41..100	[Reservado]		

Tabela 4.5 Tipos de comandos do protocolo

4.15.3 Comando: DIGITAL

Este comando permite controlar os dispositivos digitais (atuadores) ou notificar alguma mudança no estado/valor de um sensor. O comando adiciona dois blocos adicionais: DeviceID e Value (Figura 4.20).

Ao enviar um comando para um dispositivo, por exemplo, ligar lâmpada, o dispositivo deve retornar uma resposta (geralmente assíncrona), com o status do comando.

O formato da resposta é definida em “[Reposta: COMMAND_RESPONSE](#)”, onde o ID da resposta será o mesmo ID do comando enviado.

**Figura 4.20** Comando: DIGITAL

Parâmetro	Tipo	Descrição
DeviceID	byte [*1]	Identificador único que representa o dispositivo
Value	bool	0 (Desligar / Desligado) ou 1 (Ligar / Ligado)

Tabela 4.6 Parâmetros - Comando: DIGITAL

4.15.4 Comando: ANALOG

Obedece o formato do “[Comando: DIGITAL](#)”, mudando apenas o tipo do comando, no caso 2, e o tipo de dado do bloco “*Value*”, que pode aceitar valores do tipo “*unsigned long*”, com tamanho de 32 bits (4 bytes).

Este comando pode ser utilizado tanto para sensores como para atuadores, no caso de sensores, ele é utilizado para notificar uma mudança de estado (já que sensores são apenas para “leitura”).

O formato da resposta é definida em “[Reposta: COMMAND_RESPONSE](#)”, onde o ID da resposta será o mesmo ID do comando enviado.

4.15.5 Comando: NUMERIC

Obedece o formato do “[Comando: ANALOG](#)”. Este comando é utilizado em conjunto com os dispositivos (principalmente sensores) do tipo *NUMERIC*, que permitem o envio de informações repetidas. Um exemplo, o sensor RFID, que gera um evento toda vez que uma etiqueta é lida, mesmo sendo a mesma etiqueta.

4.15.6 Reposta: COMMAND_RESPONSE

Representa a resposta de um determinado comando. As respostas são associadas ao comando que a originou (requisição), através do atributo “ID” do cabeçalho fixo. Adi-

cionalmente os comandos que referenciam um dispositivo, na resposta, será incluído o mesmo “DeviceID” do comando da “requisição”.

HEADER	Status	DeviceID
--------	--------	----------

Tabela 4.7 Reposta: COMMAND_RESPONSE

Parâmetro	Tipo	Descrição
ID	byte [*2]	Mesmo ID enviado na requisição
Status	byte	Obedece aos valores da tabela: 4.9
DeviceID	byte [*1] (Opcional)	Mesmo ID enviado na requisição

Tabela 4.8 Parâmetros - Comando: COMMAND_RESPONSE

Status	Código
SUCCESS	200
NOT_FOUND	404
BAD_REQUEST	400
UNAUTHORIZED [*3]	401
FORBIDDEN [*3]	403
PERMISSION_DENIED [*3]	550
INTERNAL_ERROR	500
NOT_IMPLEMENTED	501

Tabela 4.9 Status da resposta do comando

4.15.7 Comando: SET_PROPERTY

Comando utilizado para atualizar alguma propriedade do dispositivo. O formato da resposta é definido em “[Reposta: COMMAND_RESPONSE](#)”.

HEADER	DeviceID	property	value
--------	----------	----------	-------

Tabela 4.10 Comando: SET_PROPERTY

Parâmetro	Tipo	Descrição
DeviceID	byte [*1]	Identificador único que representa o dispositivo
property	String	Propriedade a ser alterada
value	String	Valor da propriedade

Tabela 4.11 Parâmetros: SET_PROPERTY

4.15.8 Comando: GET_PROPERTIES

Comando utilizado para recuperar a lista de propriedades do dispositivo e seus respectivos valores. O formato da resposta é definido em “[Comando: GET_PROPERTIES_RESPONSE](#)”

HEADER	DeviceID
--------	----------

Tabela 4.12 Comando: GET_PROPERTIES

Parâmetro	Tipo	Descrição
DeviceID	byte [*1]	Identificador único que representa o dispositivo

Tabela 4.13 Parâmetros: GET_PROPERTIES

4.15.9 Comando: GET_PROPERTIES_RESPONSE

Resposta do comando “*GET_PROPERTIES*”, com a lista de propriedades do dispositivo. A lista é retornada no formato de um Array, e os elementos no formato: “Propriedade:Valor”.

HEADER	DeviceID	[property:value,...,propertyN:value]
--------	----------	--------------------------------------

Tabela 4.14 Comando: GET_PROPERTIES_RESPONSE

Parâmetro	Tipo	Descrição
DeviceID	byte [*1]	Identificador único que representa o dispositivo
property	String	Propriedade
value	String	Valor da propriedade

Tabela 4.15 Parâmetros: GET_PROPERTIES_RESPONSE

4.15.10 Comando: ACTION

Comando usado para executar as ações definidas pelos dispositivos. As ações podem receber uma lista de valores, que são especificados em formato de Array. O formato da resposta é definido em “[Reposta: COMMAND_RESPONSE](#)”.

HEADER	DeviceID	action	[{ value1 },..., { valueN }]
--------	----------	--------	------------------------------

Tabela 4.16 Comando: ACTION

Parâmetro	Tipo	Descrição
DeviceID	byte [*1]	Identificador único que representa o dispositivo
action	String	Ação que deve ser executada
value	Array[String] (Opcional)	A lista de valores que a ação recebe

Tabela 4.17 Parâmetros: ACTION

4.15.11 Comando: GET_ACTIONS

Comando utilizado para recuperar a lista de ações definidas para o dispositivo. O formato da resposta é definido em “[Comando: GET_ACTIONS_RESPONSE](#)”

HEADER	DeviceID
--------	----------

Tabela 4.18 Comando: GET_ACTIONS

Parâmetro	Tipo	Descrição
DeviceID	byte [*1]	Identificador único que representa o dispositivo

Tabela 4.19 Parâmetros: GET_ACTIONS

4.15.12 Comando: GET_ACTIONS_RESPONSE

Resposta do comando “*GET_ACTIONS*”, com a lista de ações do dispositivo. A lista é retornada no formato de um Array.

HEADER	DeviceID	[[{ action1 },...,{ actionN }]]
--------	----------	---------------------------------

Tabela 4.20 Comando: GET_ACTIONS_RESPONSE

Parâmetro	Tipo	Descrição
DeviceID	byte [*1]	Identificador único que representa o dispositivo
action	Array[String]	Lista de ações do dispositivo

Tabela 4.21 Parâmetros: GET_ACTIONS_RESPONSE

4.15.13 Comando: PING_REQUEST

Comando enviado pelo firmware para notificar que está em funcionamento e ao mesmo tempo monitora o status do middleware/aplicação.

Este comando não possui parâmetros adicionais, apenas cabeçalho. O formato da resposta é definido em “[Resposta: PING_RESPONSE](#)”

4.15.14 Resposta: PING_RESPONSE

Resposta para o comando “PING_REQUEST”. Este comando não possui parâmetros adicionais, apenas cabeçalho.

4.15.15 Comando: DISCOVERY_REQUEST

Comando utilizado para realizar a descoberta de dispositivos (módulos) em uma rede. Geralmente é enviado via *broadcast*. Este comando não possui parâmetros adicionais, apenas cabeçalho.

4.15.16 Resposta: DISCOVERY_RESPONSE

Resposta com as informações de descoberta dos dispositivos (módulos) .

HEADER	name	port	deviceLength
--------	------	------	--------------

Tabela 4.22 Comando: GET_ACTIONS_RESPONSE

Parâmetro	Tipo	Descrição
name	String	Nome do dispositivo (módulo)
port	Array[String]	Lista de ações do dispositivo
deviceLength	int	Quantidades de dispositivos que o módulo possui

Tabela 4.23 Parâmetros: GET_ACTIONS_RESPONSE

4.15.17 Comando: GET_DEVICES

Comando usado para obter informações de um ou mais dispositivos. Os campos 'DeviceID' e 'Value' são opcionais e podem ser usados como filtro para o comando.

HEADER	DeviceID	Value
--------	----------	-------

Tabela 4.24 Comando: GET_DEVICES

Parâmetro	Tipo	Descrição
DeviceID (Opcional)	byte [*1]	Informar o ID do dispositivo ou 0 para todos
Value (Opcional)	<i>unsigned long</i>	Permite filtrar por valor ou 0 para todos

Tabela 4.25 Parâmetros: GET_DEVICES

4.15.18 Resposta: GET_DEVICES_RESPONSE

Resposta do comando “GET_DEVICES”, com a lista de dispositivos cadastrados. O bloco “*Command Extension*”, consiste em um Array de objetos do “**Tipo: DeviceInfo**”, separados por “,”. A quantidade de dispositivos deve ser inferida automaticamente.

HEADER	[DeviceInfo,...,DeviceInfoN]
--------	------------------------------

Tabela 4.26 Resposta: GET_DEVICES_RESPONSE

Parâmetro	Tipo	Descrição
ID	byte [*2]	Mesmo ID enviado da requisição
DeviceInfo	Array[DeviceInfo]	Lista dos atributos do Tipo: DeviceInfo

Tabela 4.27 Parâmetros - Resposta: GET_DEVICES_RESPONSE

4.15.19 Comando: DEVICE_ADD

Comando utilizado para configurar dinamicamente os dispositivos de um módulo. O formato da resposta é definido em “[Reposta: COMMAND_RESPONSE](#)”.

HEADER	DeviceInfo
--------	------------

Tabela 4.28 Comando: DEVICE_ADD

Parâmetro	Tipo	Descrição
DeviceInfo	DeviceInfo	Lista dos atributos do Tipo: DeviceInfo

Tabela 4.29 Parâmetros: DEVICE_ADD

4.15.20 Comando: DEVICE_DEL

Comando utilizado deletar o(s) dispositiv(o) de um módulo. O formato da resposta é definido em “[Comando: GET_ACTIONS_RESPONSE](#)”

HEADER	DeviceID
--------	----------

Tabela 4.30 Comando: DEVICE_DEL

Parâmetro	Tipo	Descrição
DeviceID	byte [*1] (Opcioanl)	Informar o ID do dispositivo ou 0 para todos

Tabela 4.31 Parâmetros: DEVICE_DEL

4.15.21 Tipo: DeviceInfo

Representa as informações e um dispositivo. É representado como um Array no seguinte formato: [ID, PIN, VALUE, TARGET, SENSOR, TYPE].

Atributo	Tipo	Descrição
ID	byte [*2]	ID do dispositivo
PIN	byte	Pino que o dispositivo está vinculado
VALUE	unsigned long	Valor atual do dispositivo
TARGET	byte	Quando for sensor, representa outro dispositivo (Opcional)
SENSOR	bool	0 para atuador, 1 para sensor
TYPE	byte	Tipo de dispositivo (Tabela 4.33)

Tabela 4.32 Objeto: DeviceInfo

Código	Tipo
1	DIGITAL
2	ANALOG
3	NUMERIC

Tabela 4.33 Tipos de Dispositivos