

State Monitoring Protocol

Til Blechschmidt

til@blechschmidt.de

February 25, 2017

Abstract

Since there is no universal, open and secure implementation of a protocol to control any arbitrary device from another this document aims to provide an outline of the SM-Protocol that aims to provide such a protocol.

Contents

I	Structure	2
1	Network layer	2
1.1	RFC 1112	2
1.2	JSON	2
2	Devices	3
II	Protocol	3

List of source codes

1.1	Simplest JSON example	2
-----	---------------------------------	---

Part I

Structure

1 Network layer

1.1 Host Extensions for IP Multicasting

At the heart of this protocol is RFC 1112 or the Host Extensions for IP Multicasting. It enables communication between multiple devices without any additional overhead by utilizing features that are baked into the TCP/IP stack instead of every device keeping track of every connected devices. This is of great importance since most devices utilizing this protocol do not have the processing power to handle multiple TCP connections.

1.2 JavaScript Object Notation

On top of the network layer this protocol utilizes the JavaScript Object Notation since it offers high flexibility and readability. This enables easy debugging due to it's great human-friendly structure as well as easy extendability so that the protocol does not restrict devices. An example of such a message looks like this:

```
1 {  
2     "action": "write",  
3     "channel": [0, 0, 1],  
4     "payload": 255  
5 }
```

Listing 1.1: Simplest JSON example

Note though that the action property can take any arbitrary value but these ones are reserved and should be used when possible to ensure interoperability between devices.

read	Requesting the current value from a device
write	Sending a value to a device to write
state	Notify other devices about a change in state

For more details have a look at Part II of the document.

2 Devices

Due to the previously in section 1.1 mentioned network layer all devices that make use of this protocol are required to have support for the TCP/IP stack and the multicast specifications. In addition to that the devices need to support basic JSON parsing and composition. If the device is only using actions that require distribution or reception of JSON data the opposing requirement can be omitted.

Part II

Protocol